GEOMETRY-COMPLETE DIFFUSION FOR 3D MOLECULE GENERATION

Alex Morehead & Jianlin Cheng
Department of Electrical Engineering & Computer Science
University of Missouri
Columbia, MO 65211, USA
{acmwhb, chengji}@missouri.edu

ABSTRACT

Denoising diffusion probabilistic models (DDPMs) (Ho et al. (2020)) have recently taken the field of generative modeling by storm, pioneering new state-of-the-art results in disciplines such as computer vision and computational biology for diverse tasks ranging from text-guided image generation (Ramesh et al. (2022); Saharia et al. (2022); Rombach et al. (2022)) to structure-guided protein design (Ingraham et al. (2022); Watson et al. (2022)). Along this latter line of research, methods such as those of Hoogeboom et al. (2022) have been proposed for generating 3D molecules using equivariant graph neural networks (GNNs) within a DDPM framework. Toward this end, we propose GCDM, a geometry-complete diffusion model that achieves new state-of-the-art results for 3D molecule diffusion generation by leveraging the representation learning strengths offered by GNNs that perform geometry-complete message-passing. Our results with GCDM also offer preliminary insights into how physical inductive biases impact the generative dynamics of molecular DDPMs. The source code, data, and instructions to train new models or reproduce our results are freely available at https://github.com/BioinfoMachineLearning/Bio-Diffusion.

1 INTRODUCTION

Generative modeling has recently been experiencing a renaissance in modeling efforts driven largely by denoising diffusion probabilistic models (DDPMs). At a high level, DDPMs are trained by learning how to denoise a noisy version of an input example. For example, in the context of computer vision, Gaussian noise may be successively added to an input image with the goals of a DDPM in mind. We would then desire for a generative model of images to learn how to successfully distinguish between the original input image's feature signal and the noise signal added to the image thereafter. If a model can achieve such outcomes, we can use the model to generate novel images by first sampling multivariate Gaussian noise and then iteratively removing, from the current state of the image, the noise predicted by our model. This classic formulation of DDPMs has achieved significant results in the space of image generation (Rombach et al. (2022)), audio synthesis (Kong et al. (2020)), and even meta-learning by learning how to conditionally generate neural network checkpoints (Peebles et al. (2022)). Furthermore, such an approach to generative modeling has expanded its reach to encompass scientific disciplines such as computational biology (Anand & Achim (2022)), computational chemistry (Xu et al. (2022)), and even computational physics (Mudur & Finkbeiner (2022)).

Concurrently, the field of geometric deep learning (GDL) (Bronstein et al. (2021)) has seen a sizeable increase in research interest lately, driven largely by theoretical advances within the discipline (Joshi et al. (2023)) as well as by novel applications of such methodology (Stärk et al. (2022)). Notably, such applications even include what is considered by many researchers to be a solution to the problem of predicting 3D protein structures from their corresponding amino acid sequences (Jumper et al. (2021)). Such an outcome arose, in part, from recent advances in sequence-based language modeling efforts (Vaswani et al. (2017)) as well as from innovations in equivariant neural network modeling (Thomas et al. (2018)).

With such diverse, successful use cases of DDPMs and GDL in mind, in this work, we explore the intersection of geometric graph representation learning and DDPMs to answer the following questions.

- What is the impact of geometric representation learning on DDPMs designed to generate 3D molecular data?
- What are the limitations of current equivariant graph neural networks empowering contemporary molecular DDPMs?
- What role do physical inductive biases play within the generative denoising of molecular DDPMs?

2 Related Work

Generative Modeling. The field of deep generative modeling (Ruthotto & Haber (2021)) has pioneered a variety of techniques by which to train deep neural networks to create new content similar to that of an existing data repository (e.g., a text dataset of English sentences). Language models such as GPT-3 and ChatGPT (Brown et al. (2020); Schulman et al. (2022)) have become known as hallmark examples of successful generative modeling of text data. In the domains of computer vision and computational biology, techniques such as latent diffusion (Rombach et al. (2022)) and equivariant graph diffusion (Luo et al. (2022)) have established some of the latest state-of-the-art results in generative modeling of images (Tang et al. (2022)) and biomolecules (Hoogeboom et al. (2022)) such as proteins (Anand & Achim (2022); Yim et al. (2023)), respectively.

Geometric Deep Learning. Data residing in a geometric or physical space (e.g., $\mathbb{R}^3$) can be processed by machine learning algorithms in a plethora of ways. Subsequently, in recent years, the field of geometric deep learning has become known for its proficiency in introducing powerful new deep learning methods designed specifically to process geometric data (Cao et al. (2020)). Examples of popular GDL algorithms include convolutional neural networks designed for working with image data (LeCun et al. (1995)), recurrent neural networks for processing sequence-based data (Medsker & Jain (1999)), and graph neural networks for handling graph-structured model inputs (Zhou et al. (2020)).

Equivariant Neural Networks. To process geometric data efficiently, however, recent GDL research (Cohen & Welling (2016); Bronstein et al. (2021); Bulusu et al. (2021)) has specifically shown that designing one's machine learning algorithm to be equivariant to the symmetry groups the input data points naturally respect (e.g., 3D rotation symmetries) often helps such an algorithm generalize to datasets beyond those used for its cross-validation (e.g., training and testing datasets). As a particularly relevant example of a neural network that is equivariant to several important and common symmetry groups of geometric data, equivariant graph neural networks (Fuchs et al. (2020); Satorras et al. (2021b); Kofinas et al. (2021); Morehead & Cheng (2022)) that are translation and rotation equivariant to inputs residing in $\mathbb{R}^3$ have become known as hallmark examples of geometric deep learning algorithms that generalize remarkably well to new inputs and require notably fewer training iterations to converge.

Representation Learning of Scientific Data. Scientific data, in particular, requires careful consideration in the context of representation learning. As much scientific data contains within it a notion of geometry or latent structure, equivariance has become a key algorithmic component for processing such inputs as well (Han et al. (2022)). Moreover, equivariant graph representation learning algorithms have recently become a de facto methodology for processing scientific data of many shapes and origins Musaelian et al. (2022); Batzner et al. (2022).

Contributions. In this work, we connect ideas at the forefront of GDL and generative modeling to advance the state-of-the-art (SOTA) for 3D molecule generation. In detail, we provide the following contributions.

- We introduce the Geometry-Complete Diffusion Model (GCDM) which establishes new SOTA results for unconditional and conditional 3D molecule generation on the QM9 dataset and for unconditional 3D molecule generation on the larger GEOM-Drugs dataset.
- We investigate the impact of geometric message-passing on the behavior and performance of DDPMs trained to generate 3D molecular data.

- Our experiments demonstrate the importance of incorporating physical inductive biases such as molecular chirality within DDPM denoising neural networks when training them on data from physical domains.

3 METHODS

3.1 PROBLEM SETTING

In the context of this work, our goal is to generate new 3D molecules either *ab initio* or to capture a specific molecular property. We represent a molecular point cloud as a fully-connected 3D graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V}$ and $\mathcal{E}$ representing the graph's set of nodes and set of edges, respectively, and $N = |\mathcal{V}|$ and $E = |\mathcal{E}|$ representing the number of nodes and the number of edges in the graph, respectively. In addition, $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N) \in \mathbb{R}^{N\times 3}$ represents the respective Cartesian coordinates for each node (i.e., atom). Each node in $\mathcal{G}$ is described by scalar features $\mathbf{H} \in \mathbb{R}^{N\times h}$ and m vector-valued features $\boldsymbol{\chi} \in \mathbb{R}^{N\times(m\times 3)}$. Likewise, each edge in $\mathcal{G}$ is described by scalar features $\mathbf{E} \in \mathbb{R}^{E\times e}$ and x vector-valued features $\boldsymbol{\xi} \in \mathbb{R}^{E\times(x\times 3)}$. Then, let $\mathcal{M} = [\mathbf{X}, \mathbf{H}]$ represent the molecules our method is to generate, where $[\cdot,\cdot]$ denotes the concatenation of two variables. Important to note is that the input features $\mathbf{H}$ and $\mathbf{E}$ are invariant to 3D rotations, reflections, and translations, whereas the input features $\mathbf{X}$, $\boldsymbol{\chi}$, and $\boldsymbol{\xi}$ are equivariant to 3D rotations and reflections. In particular, we describe a denoising neural network $\boldsymbol{\Phi}$ as *SE*(3)-equivariant (i.e., 3D rotation and translation-equivariant) if it satisfies the following constraint on its outputs (denoted by $\square'$):

Definition 3.1. (*SE(3) Equivariance*)**.**

Given $(\mathbf{H}', \mathbf{E}', \mathbf{X}', \boldsymbol{\chi}', \boldsymbol{\xi}') = \boldsymbol{\Phi}(\mathbf{H}, \mathbf{E}, \mathbf{X}, \boldsymbol{\chi}, \boldsymbol{\xi})$,
we have $(\mathbf{H}', \mathbf{E}', \mathbf{Q}\mathbf{X}'^{\mathrm{T}} + \mathbf{g}, \mathbf{Q}\boldsymbol{\chi}'^{\mathrm{T}}, \mathbf{Q}\boldsymbol{\xi}'^{\mathrm{T}}) = \boldsymbol{\Phi}(\mathbf{H}, \mathbf{E}, \mathbf{Q}\mathbf{X}^{\mathrm{T}} + \mathbf{g}, \mathbf{Q}\boldsymbol{\chi}^{\mathrm{T}}, \mathbf{Q}\boldsymbol{\xi}^{\mathrm{T}})$,
$\forall \mathbf{Q} \in SO(3), \forall \mathbf{g} \in \mathbb{R}^{3\times 1}$.

3.2 OVERVIEW OF GCDM

We will now introduce GCDM, a new Geometry-Complete SE(3)-Equivariant Diffusion Model. In particular, we will describe how GCDM defines a joint noising process on equivariant atom coordinates $\mathbf{x}$ and invariant atom types $\mathbf{h}$ to produce a noisy representation $\mathbf{z} = [\mathbf{z}^{(\mathbf{x})}, \mathbf{z}^{(\mathbf{h})}]$ and then learns a generative *denoising* process using GCPNET (Morehead & Cheng (2022)). As we will show in subsequent sections, GCPNET is a desirable architecture for the task of denoising 3D graph inputs in that it contains two distinct feature channels for scalar and vector features, respectively, and supports geometry-complete and chirality-aware message-passing by embedding geometry information-complete local frames for each node (Barron (1986)). Moreover, in our subsequent experiments, we demonstrate that this enables GCPNET to learn more useful SE(3)-equivariant graph representations for generative modeling of 3D molecules.

As an extension of the DDPM framework (Ho et al. (2020)) outlined in Appendix A.1, GCDM is designed to generate molecules in 3D while maintaining SE(3) equivariance, in contrast to previous methods that generate molecules solely in 2D (Jin et al. (2018)) or other dimensionalities (Segler et al. (2018)). GCDM generates molecules by directly placing atoms in continuous 3D space and assigning them discrete types, which is accomplished by modeling forward and reverse diffusion processes, respectively:

$$q(\mathbf{z}_{1:T}|\mathbf{z}_0) = \prod_{t=1}^{T} q(\mathbf{z}_t|\mathbf{z}_{t-1}) \qquad (1) \qquad\qquad p_{\boldsymbol{\Phi}}(\mathbf{z}_{0:T-1}|\mathbf{z}_T) = \prod_{t=1}^{T} p_{\boldsymbol{\Phi}}(\mathbf{z}_{t-1}|\mathbf{z}_t) \qquad (2)$$

Overall, these processes describe a latent variable model $p_{\boldsymbol{\Phi}}(\mathbf{z}_0) = \int p_{\boldsymbol{\Phi}}(\mathbf{z}_{0:T}) d\mathbf{z}_{1:T}$ given a sequence of latent variables $\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_T$ matching the dimensionality of the data $\mathcal{M} \sim p(\mathbf{z}_0)$. As illustrated in Figure 1, the forward process (directed from right to left) iteratively adds noise to an input, and the learned reverse process (directed from left to right) iteratively denoises a noisy input to generate new examples from the original data distribution. We will now proceed to formulate GCDM's joint diffusion process and its remaining practical details.

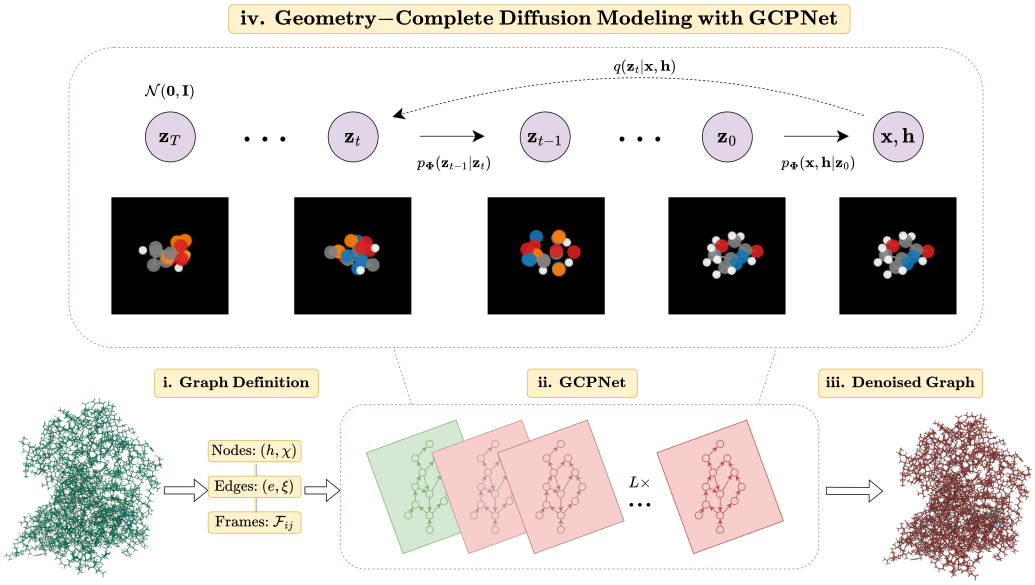


Figure 1: A framework overview for our proposed *Geometry-Complete Diffusion Model* (GCDM). Our framework consists of (**i.**) a graph (topology) definition process, (**ii.**) a GCPNET-based graph neural network for 3D graph representation learning, (**iii.**) denoising of 3D input graphs using GCPNET, and (**iv.**) application of a trained GCPNET denoising network for 3D molecule generation. Zoom in for the best viewing experience.

3.3 JOINT MOLECULAR DIFFUSION

Recall that our model's molecular graph inputs, $\mathcal{G}$, associate with each node a 3D position $\mathbf{x}_i \in \mathbb{R}^3$ and a feature vector $\mathbf{h}_i \in \mathbb{R}^h$. By way of adding random noise to these model inputs at each time step t and using a fixed, Markov chain variance schedule $\sigma_1^2, \sigma_2^2, \ldots, \sigma_T^2$, we can define a joint molecular diffusion process for equivariant atom coordinates $\mathbf{x}$ and invariant atom types $\mathbf{h}$ as the product of two distributions (Hoogeboom et al. (2022)):

$$q(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}_x(\mathbf{z}_t^{(x)}|\alpha_t \mathbf{z}_{t-1}^{(x)}, \sigma_t^2 \mathbf{I}) \cdot \mathcal{N}_h(\mathbf{z}_t^{(h)}|\alpha_t \mathbf{z}_{t-1}^{(h)}, \sigma_t^2 \mathbf{I}). \tag{3}$$

where the first distribution, $\mathcal{N}_x$, represents the noised node coordinates, the second distribution, $\mathcal{N}_h$, represents the noised node features, and $\alpha_t = \sqrt{1-\sigma_t^2}$ following the variance preserving process of Ho et al. (2020). Using $\mathcal{N}_{xh}$ as concise notation to denote the product of two normal distributions, we can further simplify Eq. 3 as:

$$q(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}_{xh}(\mathbf{z}_t|\alpha_t \mathbf{z}_{t-1}, \sigma_t^2 \mathbf{I}). \tag{4}$$

With $\alpha_{t|s} = \alpha_t/\alpha_s$ and $\sigma_{t|s}^2 = \sigma_t^2 - \alpha_{t|s}\sigma_s^2$ for any $t > s$, we can directly obtain the noisy data distribution $q(\mathbf{z}_t|\mathbf{z}_0)$ at any time step t:

$$q(\mathbf{z}_t|\mathbf{z}_0) = \mathcal{N}_{xh}(\mathbf{z}_t|\alpha_{t|0}\mathbf{z}_0, \sigma_{t|0}^2 \mathbf{I}). \tag{5}$$

Bayes Theorem then tells us that if we then define $\boldsymbol{\mu}_{t\to s}(\mathbf{z}_t, \mathbf{z}_0)$ and $\sigma_{t\to s}$ as

$$\boldsymbol{\mu}_{t\to s}(\mathbf{z}_t, \mathbf{z}_0) = \frac{\alpha_s \sigma_{t|s}^2}{\sigma_t^2}\mathbf{z}_0 + \frac{\alpha_{t|s}\sigma_s^2}{\sigma_t^2}\mathbf{z}_t \text{ and } \sigma_{t\to s} = \frac{\sigma_{t|s}\sigma_s}{\sigma_t},$$

we have that the inverse of the noising process, the *true denoising process*, is given by the posterior of the transitions conditioned on $\mathcal{M} \sim \mathbf{z}_0$, a process that is also Gaussian (Hoogeboom et al. (2022)):

$$q(\mathbf{z}_s|\mathbf{z}_t, \mathbf{z}_0) = \mathcal{N}(\mathbf{z}_s|\boldsymbol{\mu}_{t\to s}(\mathbf{z}_t, \mathbf{z}_0), \sigma_{t\to s}^2 \mathbf{I}). \tag{6}$$

3.4 GEOMETRY-COMPLETE PARAMETRIZATION OF THE EQUIVARIANT REVERSE PROCESS

Noise parametrization. We now need to define our learned generative reverse process that *denoises* pure noise into realistic examples from the original data distribution. Towards this end, we can directly use the noise posteriors $q(\mathbf{z}_s|\mathbf{z}_t, \mathbf{z}_0)$ of Eq. 16 with $\mathbf{z}_0 \sim (\mathcal{M} = [\mathbf{x}, \mathbf{h}])$. However, to do so, we must replace the input variables $\mathbf{x}$ and $\mathbf{h}$ with the approximations $\hat{\mathbf{x}}$ and $\hat{\mathbf{h}}$ predicted by our denoising neural network $\mathbf{\Phi}$:

$$p_{\mathbf{\Phi}}(\mathbf{z}_s|\mathbf{z}_t) = \mathcal{N}_{xh}(\mathbf{z}_s|\boldsymbol{\mu}_{\mathbf{\Phi}_{t\to s}}(\mathbf{z}_t, \tilde{\mathbf{z}}_0), \sigma^2_{t\to s}\mathbf{I}), \tag{7}$$

where the values for $\tilde{\mathbf{z}}_0 = [\hat{\mathbf{x}}, \hat{\mathbf{h}}]$ depend on $\mathbf{z}_t$, t, and our denoising neural network $\mathbf{\Phi}$.

In the context of diffusion models, many different parametrizations of $\boldsymbol{\mu}_{\mathbf{\Phi}_{t\to s}}(\mathbf{z}_t, \tilde{\mathbf{z}}_0)$ are possible. Prior works have found that it is often easier to optimize a diffusion model using a noise parametrization to predict the noise $\hat{\boldsymbol{\epsilon}}$. In this work, we use such a parametrization to predict $\hat{\boldsymbol{\epsilon}} = [\hat{\boldsymbol{\epsilon}}^{(x)}, \hat{\boldsymbol{\epsilon}}^{(h)}]$, which represents the noise individually added to $\hat{\mathbf{x}}$ and $\hat{\mathbf{h}}$. We can then use the predicted $\hat{\boldsymbol{\epsilon}}$ to derive:

$$\tilde{\mathbf{z}}_0 = [\hat{\mathbf{x}}, \hat{\mathbf{h}}] = \mathbf{z}_t/\alpha_t - \hat{\boldsymbol{\epsilon}}_t \cdot \sigma_t/\alpha_t. \tag{8}$$

Invariant likelihood. Ideally, we desire for a 3D molecular diffusion model to assign the same likelihood to a generated molecule even after arbitrarily rotating or translating it in 3D space. To ensure our model achieves this desirable property for $p_{\mathbf{\Phi}}(\mathbf{z}_0)$, we can leverage the insight that an invariant distribution composed of an equivariant transition function yields an invariant distribution (Satorras et al. (2021a); Xu et al. (2022); Hoogeboom et al. (2022)). Moreover, to address the translation invariance issue raised by Satorras et al. (2021a) in the context of handling a distribution over 3D coordinates, we adopt the zero center of gravity trick proposed by Xu et al. (2022) to define $\mathcal{N}_x$ as a normal distribution on the subspace defined by $\sum_i \mathbf{x}_i = \mathbf{0}$. In contrast, to handle node features $\mathbf{h}_i$ that are rotation and translation-invariant, we can instead use a conventional normal distribution $\mathcal{N}$. As such, if we parametrize our transition function $p_{\mathbf{\Phi}}$ using an SE(3)-equivariant neural network after using the zero center of gravity trick of Xu et al. (2022), our model will have achieved the desired likelihood invariance property.

Geometry-completeness. Furthermore, in this work, we postulate that certain types of geometric neural networks serve as more effective 3D graph denoising functions for molecular DDPMs. We formalize this notion as follows.

Proposition 3.2. *(Geometry-Complete Denoising).*

Geometric neural networks that achieve geometry-completeness are principally more capable of denoising noisy 3D molecular graph inputs, in that geometry-complete methods encode local reference frames under which the directions of arbitrary global forces can be mapped.

This proposition comes as an extension of the definition of geometry-completeness from Morehead & Cheng (2022). An intuition for its implications on molecular diffusion models is that geometry-complete networks should be able to more effectively learn data distributions in which a global force field is present, as is typically the case with 3D molecules (Du et al. (2022)). As a complement to understanding the theoretical benefits offered to geometry-complete networks, we support this claim through specific ablation studies in Section 4.1.

GCPNETS. Inspired by their recent success in modeling 3D molecular structures with geometry-complete message-passing, as mentioned previously, we will parametrize $p_{\mathbf{\Phi}}$ using Geometry-Complete Perceptron Networks (GCPNETS) as introduced by Morehead & Cheng (2022). GCPNET is a geometry-complete graph neural network that is equivariant to SE(3) transformations of its graph inputs and, as such, satisfies our SE(3) equivariance constraint (3.1) and maps nicely to the context of Proposition 3.2.

In this setting, with ($h_i \in \mathbf{H}$, $\chi_i \in \boldsymbol{\chi}$, $e_{ij} \in \mathbf{E}, \xi_{ij} \in \boldsymbol{\xi}$), GCPNET consists of a composition of Geometry-Complete Graph Convolution (**GCPConv**) layers $(h_i^l, \chi_i^l), x_i^l = \mathbf{GCPConv}[(h_i^{l-1}, \chi_i^{l-1}), (e_{ij}^{l-1}, \xi_{ij}^{l-1}), x_i^{l-1}, \mathcal{F}_{ij}]$ which are defined as:

$$n_i^l = \boldsymbol{\phi}^l(n_i^{l-1}, \mathcal{A}_{\forall j \in \mathcal{N}(i)} \mathbf{\Omega}_\omega^l(n_i^{l-1}, n_j^{l-1}, e_{ij}^{l-1}, \xi_{ij}^{l-1}, \mathcal{F}_{ij})), \tag{9}$$

where $n_i^l = (h_i^l, \chi_i^l)$; $\boldsymbol{\phi}^l$ is a trainable function; l signifies the representation depth of the network; $\mathcal{A}$ is a permutation-invariant aggregation function; $\mathbf{\Omega}_\omega$ represents a message-passing function cor-

responding to the ω-th $\mathbf{GCP}$ message-passing layer; and node i's geometry-complete local frames are $\mathcal{F}_{ij}^t = (a_{ij}^t, b_{ij}^t, c_{ij}^t)$, with $a_{ij}^t = \frac{x_i^t - x_j^t}{\|x_i^t - x_j^t\|}$, $b_{ij}^t = \frac{x_i^t \times x_j^t}{\|x_i^t \times x_j^t\|}$, and $c_{ij}^t = a_{ij}^t \times b_{ij}^t$, respectively.

Lastly, if one desires to update the coordinate representations of each node in $\mathcal{G}$, as we do in the context of 3D molecule generation, $\mathbf{GCPConv}$ provides a simple, SE(3)-equivariant method to do so using a dedicated $\mathbf{GCP}$ module as follows:

$$(h_{p_i}^l, \chi_{p_i}^l) = \mathbf{GCP}_p^l(n_i^l, \mathcal{F}_{ij}) \tag{10}$$

$$x_i^l = x_i^{l-1} + \chi_{p_i}^l, \text{where } \chi_{p_i}^l \in \mathbb{R}^{1\times 3}, \tag{11}$$

where $\mathbf{GCP}_\cdot^l(\cdot, \mathcal{F}_{ij})$ is defined as in (Morehead & Cheng (2022)) to provide chirality-aware rotation and translation-invariant updates to h_i and rotation equivariant updates to χ_i following centralization of the input point cloud's coordinates $\mathbf{X}$ (Du et al. (2022)). The effect of using feature updates to χ_i to update x_i is, after decentralizing $\mathbf{X}$ following the final $\mathbf{GCPConv}$ layer, that updates to x_i then become SE(3)-equivariant. As such, all the transformations described above collectively satisfy the required equivariance constraint in Def. 3.1. Therefore, in adopting GCPNET as its 3D graph denoiser, GCDM achieves SE(3) equivariance, geometry-completeness, and likelihood invariance altogether.

3.5 Optimization Objective

Following previous works on diffusion models (Ho et al. (2020); Hoogeboom et al. (2022); Wu et al. (2022)), our noise parametrization chosen for GCDM yields the following model training objective:

$$\mathcal{L}_t = \mathbb{E}_{\epsilon_t \sim \mathcal{N}_{xh}(0,1)} \left[\frac{1}{2} w(t) \|\epsilon_t - \hat{\epsilon}_t\|^2 \right], \tag{12}$$

where $\hat{\epsilon}_t$ is our network's noise prediction as described above and where we empirically choose to set $w(t) = 1$ for the best possible generation results compared to $w(t) = (1 - \text{SNR}(t-1) \,/\, \text{SNR}(t))$ with $\text{SNR}(t) = \alpha_t^2 / \sigma_t^2$. Additionally, GCDM permits a negative log-likelihood computation using the same optimization terms as Hoogeboom et al. (2022), for which we refer interested readers to Appendices A.1 and A.2 for further details. Lastly, for remaining technical details regarding GCDM's training and sampling procedures, we refer readers to Appendix A.4.

4 Experiments

4.1 Unconditional 3D Molecule Generation - QM9

The QM9 dataset (Ramakrishnan et al. (2014)) contains molecular property descriptions and 3D atom coordinates for 130k small molecules. Each molecule in QM9 can contain up to 9 heavy atoms, that is, 29 atoms when including hydrogens. For the task of 3D molecule generation, we train GCDM to unconditionally generate molecules by producing atom types (H, C, N, O, and F), integer-valued atom charges, and 3D coordinates for each of the molecules' atoms. Following Anderson et al. (2019), we split QM9 into training, validation, and test partitions consisting of 100k, 18k, and 13k molecule examples, respectively.

Metrics. We adopt the scoring conventions of Satorras et al. (2021a) by using the distance between atom pairs and their respective atom types to predict bond types (single, double, triple, or none) for all but one baseline method (i.e., E-NF). Subsequently, we measure the proportion of generated atoms that have the right valency (atom stability) and the proportion of generated molecules for which all atoms are stable (molecule stability). To offer additional insights into each method's behavior for 3D molecule generation, we also report the validity of a generated molecule as determined by RDKit (Landrum et al. (2013)) and the uniqueness of the generated molecules overall.

Baselines. Besides including a reference point for molecule quality metrics using QM9 itself (i.e., Data), we compare GCDM to three existing E(3)-equivariant models: G-Schnet (Gebauer et al. (2019)), Equivariant Normalizing Flows (E-NF) (Satorras et al. (2021a)), and Equivariant Diffusion Models (EDM) (Hoogeboom et al. (2022)). For each of these three models, we report their results as reported in Hoogeboom et al. (2022). For comparison with models for this task that are not equivariant, we also report results from Hoogeboom et al. (2022) for Graph Diffusion Models (GDM)

Table 1: Comparison of GCPNET with baseline methods for 3D molecule generation. The results are reported in terms of the negative log-likelihood (NLL) - $\log p(\mathbf{x}, \mathbf{h}, N)$, atom stability, molecule stability, validity, and uniqueness of 10,000 samples drawn from each model, with standard deviations for each model across three runs on QM9. The top-1 (best) results for this task are in **bold**, and the second-best results are underlined.

Type	Method	NLL ↓	Atoms Stable (%) ↑	Mol Stable (%) ↑	Valid (%) ↑	Valid and Unique (%) ↑
Normalizing Flow	E-NF	-59.7	85.0	4.9	40.2	39.4
Graph Autoregression	G-Schnet	N/A	95.7	68.1	85.5	80.3
DDPM	GDM	-94.7	97.0	63.2	N/A	N/A
	GDM-aug	-92.5	97.6	71.6	90.4	89.5
	EDM	-110.7 $\pm$ 1.5	98.7 $\pm$ 0.1	82.0 $\pm$ 0.4	91.9 $\pm$ 0.5	90.7 $\pm$ 0.6
	Bridge	N/A	98.7 $\pm$ 0.1	81.8 $\pm$ 0.2	N/A	90.2
	Bridge + Force	N/A	**98.8** $\pm$ 0.1	84.6 $\pm$ 0.3	N/A	90.7
DDPM - Ours	GCDM w/o Frames	-162.3 $\pm$ 0.3	98.4 $\pm$ 0.0	81.7 $\pm$ 0.5	93.9 $\pm$ 0.1	92.7 $\pm$ 0.1
	GCDM w/o SMA	-131.3 $\pm$ 0.8	95.7 $\pm$ 0.1	51.7 $\pm$ 1.4	83.1 $\pm$ 1.7	82.8 $\pm$ 1.7
	GCDM	**-171.0** $\pm$ 0.2	98.7 $\pm$ 0.0	**85.7** $\pm$ 0.4	**94.8** $\pm$ 0.2	**93.3** $\pm$ 0.0
Data			99.0	95.2	97.7	97.7

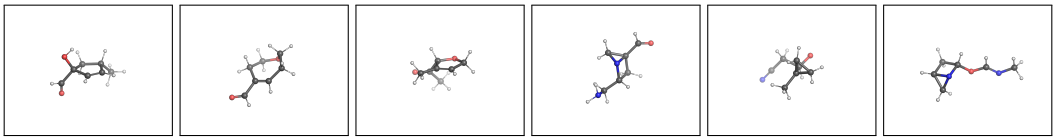


Figure 2: 3D molecules generated by GCDM for the QM9 dataset.

trained with random data rotations (GDM-aug) and without them (GDM). To the best of our knowledge, the force-guided molecule generation methods of Wu et al. (2022) (i.e., Bridge and Bridge + Force) are the most recent and performant state-of-the-art open-source methods for 3D molecule generation, so we include their results for this experiment as well.

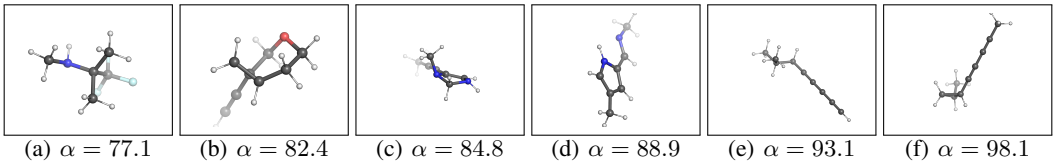
We further include two GCDM ablation models to more closely analyze the impact of certain key model components within GCDM. These two ablation models include GCDM without geometry-complete local frames $\mathcal{F}_{ij}$ (i.e., GCDM w/o Frames) and GCDM without scalar message attention (SMA) applied to each edge message (i.e., GCDM w/o SMA). For SMA, $\mathbf{m}_{ij} = e_{ij}\mathbf{m}_{ij}$, where $\mathbf{m}_{ij}$ represents the scalar messages learned by GCPNET during message-passing and e_{ij} represents a 1 if an edge exists between nodes i and j (and a 0 otherwise) via $e_{ij} \approx \phi_{inf}(\mathbf{m}_{ij})$. Here, $\phi_{inf} : \mathbb{R}^e \to [0, 1]^1$ resembles a linear layer followed by a sigmoid function Satorras et al. (2021b). All GCDM models train on QM9 for approximately 1,000 epochs using 9 **GCPConv** layers; SiLU activations (Elfwing et al. (2018)); 256 and 64 scalar node and edge hidden features, respectively; and 32 and 16 vector-valued node and edge features, respectively. All GCDM models are also trained using the AdamW optimizer (Loshchilov & Hutter (2017)) with a batch size of 64, a learning rate of 10^{-4}, and a weight decay rate of 10^{-12}.

Results. In Table 1, we see that GCDM matches or outperforms all previous methods (E-NF, G-Schnet, EDM, Bridge, and Bridge + Force) as well as their non-equivariant counterparts (GDM and GDM-aug) for all metrics, with generated samples shown in Figure 2. In particular, GCDM generates the highest percentage of valid and unique molecules compared to all other methods, improving upon previous SOTA results in such measures by 3%. GCDM also advances the SOTA results in terms of negative log-likelihood (NLL) and molecule stability by 54% and 1%, respectively. Moreover, our ablation of SMA within GCDM demonstrates that GCDM heavily relies on being able to perform a lightweight version of self-attention (Vaswani et al. (2017)) in the form of fully-connected attentive message-passing to generate stable 3D molecules. This finding suggests interesting avenues for future research into the impact of different kinds of attention-based geometric message-passing (e.g., type-2 tensor message-passing) on the performance of diffusion models for 3D molecular generation tasks.

Specifically, it is interesting to note how much lower the NLL of GCDM is compared to that of EDM, the previous NLL-based SOTA method for 3D molecule generation, indicating the generative

Table 2: Comparison of GCPNET with baseline methods for property-conditional 3D molecule generation. The results are reported in terms of the mean absolute error for molecular property prediction by an EGNN classifier ϕ_c on a QM9 subset, GCDM-generated samples, and two different baselines "Naive (Upper-bound)" and "# Atoms". The top-1 (best) results for this task are in **bold**, and the second-best results are underlined.

Task Units	α $Bohr^3$	$\Delta\epsilon$ meV	ϵ_{HOMO} meV	ϵ_{LUMO} meV	μ D	C_v $\frac{cal}{mol}K$
Naive (Upper-bound)	9.01	1470	645	1457	1.616	6.857
# Atoms	3.86	866	426	813	1.053	1.971
EDM	2.76	655	356	584	1.111	1.101
GCDM	**1.97**	**602**	**344**	**479**	**0.844**	**0.689**
QM9 (Lower-bound)	0.10	64	39	36	0.043	0.040



(a) $\alpha = 77.1$ (b) $\alpha = 82.4$ (c) $\alpha = 84.8$ (d) $\alpha = 88.9$ (e) $\alpha = 93.1$ (f) $\alpha = 98.1$

Figure 3: 3D molecules generated by GCDM using increasing values of α for the QM9 dataset.

distribution that GCDM learns from QM9 likely contains much sharper peaks compared to EDM even within the context of similar diffusion modeling frameworks. A possible explanation for why GCDM can achieve such results over other equivariant methods such as EDM and Bridge is that GCDM performs geometric (and geometry-complete) message-passing over each 3D input graph to remove the noise present therein, whereas other methods learn solely using scalar features (Joshi et al. (2023)). Our ablation of geometry-complete local frames within GCDM supports this claim in that, compared to EDM, message-passing with type-1 tensor (i.e., vectors) appears to improve GCDM's NLL over that of EDM by 47%, whereas with geometry-complete frames GCDM's NLL improves by another 7%. In fact, with geometry-complete and chirality-sensitive frame embeddings, all of GCDM's sample quality metrics improve to SOTA levels, providing support for Proposition 3.2.

4.2 CONDITIONAL 3D MOLECULE GENERATION - QM9

Baselines. Towards conditional generation of 3D molecules, we compare GCDM to an existing E(3)-equivariant model, EDM (Hoogeboom et al. (2022)), as well as to two naive baselines: "Naive (Upper-bound)" where a property classifier ϕ_c predicts molecular properties given a method's generated 3D molecules and shuffled (i.e., random) property labels; and "# Atoms" where one uses the numbers of atoms in a method's generated 3D molecules to predict their molecular properties. For each baseline method, we report its mean absolute error in terms of molecular property prediction by an EGNN classifier ϕ_c Satorras et al. (2021b) as reported in Hoogeboom et al. (2022). For GCDM, we train each conditional model by conditioning it on one of six distinct molecular properties - α, gap, homo, lumo, μ, and C_v - for approximately 1,500 epochs using the QM9 validation split of Hoogeboom et al. (2022) as the model's training dataset and the QM9 training split of Hoogeboom et al. (2022) as the corresponding EGNN classifier's training dataset. Consequently, one can expect the gap between a method's performance and that of "QM9 (Lower-bound)" to decrease as the method generates molecules that more accurately model a given molecular property.

Results. We see in Table 2 that GCDM outperforms all other methods in conditioning on a given molecular property, with conditionally-generated samples shown in Figure 3. In particular, GCDM improves upon the mean absolute error of the SOTA EDM method for all six molecular properties - α, gap, homo, lumo, μ, and C_v - by 29%, 8%, 3%, 18%, 24%, and 37%, respectively, demonstrating that, using geometry-complete message-passing, GCDM can more accurately model important molecular properties for 3D molecule generation.

Table 3: Comparison of GCPNET with baseline methods for 3D molecule generation. The results are reported in terms of each method's atom stability and molecule stability with standard deviations across three runs on GEOM-Drugs, each drawing 10,000 samples from the model. The top-1 (best) results for this task are in **bold**, and the second-best results are underlined.

Type	Method	Atoms Stable (%) ↑	Mol Stable (%) ↑
Normalizing Flow	E-NF	75.0	0.0
DDPM	GDM	75.0	0.0
	GDM-aug	77.7	0.0
	EDM	81.3	0.0
	Bridge	81.0 ± 0.7	0.0
	Bridge + Force	82.4 ± 0.8	0.0
DDPM - Ours	GCDM	$\mathbf{86.4} \pm 0.2$	$\mathbf{3.7} \pm 0.3$
Data		86.5	2.8

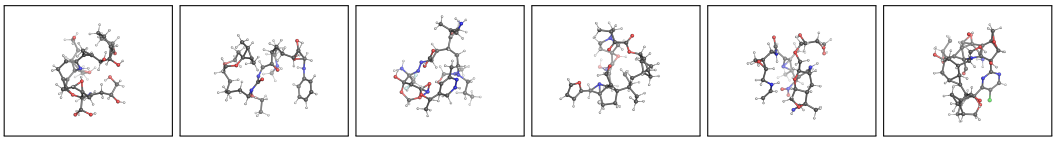


Figure 4: 3D molecules generated by GCDM for the GEOM-Drugs dataset.

4.3 UNCONDITIONAL 3D MOLECULE GENERATION - GEOM-DRUGS

The GEOM-Drugs dataset is a well-known source of large, 3D molecular conformers for downstream machine learning tasks. It contains 430k molecules, each with 44 atoms on average and with up to as many as 181 atoms. For this experiment, we collect the 30 lowest-energy conformers corresponding to a molecule and task each baseline method with generating new molecules with 3D positions and types for each constituent atom. Here, we also adopt the atom stability and molecule stability metrics as defined in Section 4.1 and train GCDM using the same hyperparameters as listed in Section 4.1 with the exception of training for approximately 50 epochs on GEOM-Drugs.

Baselines. In this experiment, we compare GCDM to several state-of-the-art baseline methods for 3D molecule generation on GEOM-Drugs. Similar to our experiments on QM9, in addition to including a reference point for molecule quality metrics using GEOM-Drugs itself (i.e., Data), here we also compare against E-NF, GDM, GDM-aug, EDM, and Bridge with its variant Bridge + Force.

Results. To start, Table 3 displays an interesting phenomenon: Due to the size of GEOM-Drugs' molecules and the subsequent errors accumulated when estimating bond types based on inter-atom distances, the baseline results for the molecule stability metrics measured here (i.e., Data) are much lower than those collected for the QM9 dataset. Nonetheless, for GEOM-Drugs, GCDM improves upon SOTA atom stability results by 5%, with generated samples shown in Figure 4. Remarkably, to our best knowledge, GCDM is also the first deep learning model that can generate any stable large molecules according to the definitions of atomic and molecular stability in Section 4.1, demonstrating that GCDM can not only effectively generate large molecules but can also closely model the true distribution of stable molecules within GEOM-Drugs.

5 CONCLUSION

In this work, we introduced GCDM, an SE(3)-equivariant geometry-complete denoising diffusion probabilistic model for 3D molecule generation. While previous equivariant methods for this task have had difficulty establishing sizeable performance gains over non-equivariant methods for this task, GCDM establishes a clear performance advantage over all other methods, generating more realistic, stable, valid, unique, and property-specific 3D molecules compared to existing approaches. Although GCDM's results here are promising, since the method falls into the traditional DDPM framework for generative modeling, using it to generate several thousands of large 3D molecules

takes a notable amount of time (e.g., 15 minutes to generate 100 new molecules). As such, future work in improving GCDM could involve introducing new time-efficient sampling algorithms for diffusion models (Song et al. (2020)) or even exploring other uses of GCDM in optimizing existing molecule's geometry or chemical properties.

ACKNOWLEDGMENTS

This work is partially supported by two NSF grants (DBI1759934 and IIS1763246), two NIH grants (R01GM093123 and R01GM146340), three DOE grants (DE-AR0001213, DE-SC0020400, and DE-SC0021303), and the computing allocation on the Summit compute cluster provided by the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05- 00OR22725, granted in part by the Advanced Scientific Computing Research (ASCR) Leadership Computing Challenge (ALCC) program.

REFERENCES

Namrata Anand and Tudor Achim. Protein structure and sequence generation with equivariant denoising diffusion probabilistic models. *arXiv preprint arXiv:2205.15019*, 2022.

Brandon Anderson, Truong Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. *Advances in neural information processing systems*, 32, 2019.

LD Barron. Symmetry and molecular chirality. *Chemical Society Reviews*, 15(2):189–223, 1986.

Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):2453, 2022.

Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Srinath Bulusu, Matteo Favoni, Andreas Ipp, David I Müller, and Daniel Schuh. Generalization capabilities of translationally equivariant neural networks. *Physical Review D*, 104(7):074504, 2021.

Wenming Cao, Zhiyue Yan, Zhiquan He, and Zhihai He. A comprehensive survey on geometric deep learning. *IEEE Access*, 8:35929–35949, 2020.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pp. 2990–2999. PMLR, 2016.

Weitao Du, He Zhang, Yuanqi Du, Qi Meng, Wei Chen, Nanning Zheng, Bin Shao, and Tie-Yan Liu. Se(3) equivariant graph neural networks with complete local frames. In *International Conference on Machine Learning*, pp. 5583–5608. PMLR, 2022.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018.

Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in Neural Information Processing Systems*, 33:1970–1981, 2020.

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

Jiaqi Han, Yu Rong, Tingyang Xu, and Wenbing Huang. Geometrically equivariant graph neural networks: A survey. *arXiv preprint arXiv:2202.07230*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, pp. 8867–8887. PMLR, 2022.

John Ingraham, Max Baranov, Zak Costello, Vincent Frappier, Ahmed Ismail, Shan Tie, Wujie Wang, Vincent Xue, Fritz Obermeyer, Andrew Beam, et al. Illuminating protein space with a programmable generative model. *bioRxiv*, pp. 2022–12, 2022.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2323–2332. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/jin18a.html`.

Chaitanya K Joshi, Cristian Bodnar, Simon V Mathis, Taco Cohen, and Pietro Liò. On the expressive power of geometric graph neural networks. *arXiv preprint arXiv:2301.09308*, 2023.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

Miltiadis Kofinas, Naveen Nagaraja, and Efstratios Gavves. Roto-translated local coordinate frames for interacting dynamical systems. *Advances in Neural Information Processing Systems*, 34:6417–6429, 2021.

Jonas Köhler, Leon Klein, and Frank Noé. Equivariant flows: exact likelihood generative learning for symmetric densities. In *International conference on machine learning*, pp. 5361–5370. PMLR, 2020.

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

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

Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

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

Shitong Luo, Yufeng Su, Xingang Peng, Sheng Wang, Jian Peng, and Jianzhu Ma. Antigen-specific antibody design and optimization with diffusion-based generative models. *bioRxiv*, pp. 2022–07, 2022.

Larry Medsker and Lakhmi C Jain. *Recurrent neural networks: design and applications*. CRC press, 1999.

Alex Morehead and Jianlin Cheng. Geometry-complete perceptron networks for 3d molecular graphs. *arXiv preprint arXiv:2211.02504*, 2022.

Nayantara Mudur and Douglas P Finkbeiner. Can denoising diffusion probabilistic models generate realistic astrophysical fields? *arXiv preprint arXiv:2211.12444*, 2022.

Albert Musaelian, Simon Batzner, Anders Johansson, Lixin Sun, Cameron J Owen, Mordechai Kornbluth, and Boris Kozinsky. Learning local equivariant representations for large-scale atomistic dynamics. *arXiv preprint arXiv:2204.05249*, 2022.

William Peebles, Ilija Radosavovic, Tim Brooks, Alexei A Efros, and Jitendra Malik. Learning to learn with generative models of neural network checkpoints. *arXiv preprint arXiv:2209.12892*, 2022.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Lars Ruthotto and Eldad Haber. An introduction to deep generative modeling. *GAMM-Mitteilungen*, 44(2):e202100008, 2021.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

Victor Garcia Satorras, Emiel Hoogeboom, Fabian B Fuchs, Ingmar Posner, and Max Welling. E (n) equivariant normalizing flows. *arXiv preprint arXiv:2105.09016*, 2021a.

Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pp. 9323–9332. PMLR, 2021b.

J Schulman, B Zoph, C Kim, J Hilton, J Menick, J Weng, JFC Uribe, L Fedus, L Metz, M Pokorny, et al. Chatgpt: Optimizing language models for dialogue, 2022.

Marwin HS Segler, Thierry Kogej, Christian Tyrchan, and Mark P Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS central science*, 4(1): 120–131, 2018.

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

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Hannes Stärk, Octavian Ganea, Lagnajit Pattanaik, Regina Barzilay, and Tommi Jaakkola. Equibind: Geometric deep learning for drug binding structure prediction. In *International Conference on Machine Learning*, pp. 20503–20521. PMLR, 2022.

Raphael Tang, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Jimmy Lin, and Ferhan Ture. What the daam: Interpreting stable diffusion using cross attention. *arXiv preprint arXiv:2210.04885*, 2022.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

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

Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models. *bioRxiv*, pp. 2022–12, 2022.

Lemeng Wu, Chengyue Gong, Xingchao Liu, Mao Ye, and Qiang Liu. Diffusion-based molecule generation with informative prior bridges. *arXiv preprint arXiv:2209.00865*, 2022.

Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.

Jason Yim, Brian L Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, Regina Barzilay, and Tommi Jaakkola. Se (3) diffusion model with application to protein backbone generation. *arXiv preprint arXiv:2302.02277*, 2023.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

A Appendix

A.1 Diffusion Models

Key to understanding our contributions in this work are denoising diffusion probabilistic models. As alluded to previously, once trained, DDPMs can generate new data of arbitrary shapes, sizes, formats, and geometries by learning to reverse a noising process acting on each model input. More precisely, for a given data point $\mathbf{x}$, a diffusion process adds noise to $\mathbf{x}$ for time step $t = 0, 1, ..., T$ to yield $\mathbf{z}_t$, a noisy representation of the input $\mathbf{x}$ at time step t. Such a process is defined by a multivariate Gaussian distribution:

$$q(\mathbf{z}_t|x) = \mathcal{N}(\mathbf{z}_t|\alpha_t \mathbf{x}_t, \sigma_t^2 \mathbf{I}), \tag{13}$$

where $\alpha_t \in \mathbb{R}^+$ regulates how much feature signal is retained and σ_t^2 modulates how much feature noise is added to input $\mathbf{x}$. Note that we typically model α as a function defined with smooth transitions from $\alpha_0 = 1$ to $\alpha_T = 0$, where a special case of such a noising process, the variance preserving process (Sohl-Dickstein et al. (2015); Ho et al. (2020)), is defined by $\alpha_t = \sqrt{1 - \sigma_t^2}$. To simplify notation, in this work, we define the feature signal-to-noise ratio as SNR(t) $= \alpha_t^2/\sigma_t^2$. Also interesting to note is that this diffusion process is Markovian in nature, indicating that we have transition distributions as follows:

$$q(\mathbf{z}_t|\mathbf{z}_s) = \mathcal{N}(\mathbf{z}_t|\alpha_{t|s}\mathbf{z}_s, \sigma_{t|s}^2 \mathbf{I}), \tag{14}$$

for all $t > s$ with $\alpha_{t|s} = \alpha_t/\alpha_s$ and $\sigma_{t|s}^2 = \sigma_t^2 - \alpha_{t|s}^2 \sigma_s^2$. In total, then, we can write the noising process as:

$$q(\mathbf{z}_0, \mathbf{z}_1, ..., \mathbf{z}_T|\mathbf{x}) = q(\mathbf{z}_0|x) \prod_{t=1}^{T} q(\mathbf{z}_t|\mathbf{z}_{t-1}). \tag{15}$$

If we then define $\boldsymbol{\mu}_{t\to s}(\mathbf{x}, \mathbf{z}_t)$ and $\sigma_{t\to s}$ as

$$\boldsymbol{\mu}_{t\to s}(\mathbf{x}, \mathbf{z}_t) = \frac{\alpha_{t|s}\sigma_s^2}{\sigma_t^2}\mathbf{z}_t + \frac{\alpha_s \sigma_{t|s}^2}{\sigma_t^2}\mathbf{x} \text{ and } \sigma_{t\to s} = \frac{\sigma_{t|s}\sigma_s}{\sigma_t},$$

we have that the inverse of the noising process, the *true denoising process*, is given by the posterior of the transitions conditioned on $\mathbf{x}$, a process that is also Gaussian:

$$q(\mathbf{z}_s|\mathbf{x}, \mathbf{z}_t) = \mathcal{N}(\mathbf{z}_s|\boldsymbol{\mu}_{t\to s}(\mathbf{x}, \mathbf{z}_t), \sigma_{t\to s}\mathbf{I}). \tag{16}$$

The Generative Denoising Process. In diffusion models, we define the generative process according to the *true denoising process*. However, for such a denoising process, we do not know the value of $\mathbf{x}$ *a priori*, so we typically approximate it as $\hat{\mathbf{x}} = \phi(\mathbf{z}_t, t)$ using a neural network ϕ. Doing so then lets us express the generative transition distribution $p(\mathbf{z}_s|\mathbf{z}_t)$ as $q(\mathbf{z}_s|\hat{\mathbf{x}}(\mathbf{z}_t, t), \mathbf{z}_t)$. As a practical alternative to Eq. 16, we can represent this expression using our approximation for $\hat{\mathbf{x}}$:

$$p(\mathbf{z}_s|\mathbf{z}_t) = \mathcal{N}(\mathbf{z}_s|\boldsymbol{\mu}_{t\to s}(\hat{\mathbf{x}}, \mathbf{z}_t), \sigma_{t\to s}^2 \mathbf{I}). \tag{17}$$

If we choose to define s as $s = t - 1$, then we can derive the variational lower bound on the log-likelihood of $\mathbf{x}$ given our generative model as:

$$\log p(\mathbf{x}) \geq \mathcal{L}_0 + \mathcal{L}_{base} + \sum_{t=1}^{T} \mathcal{L}_t, \tag{18}$$

where we note that $\mathcal{L}_0 = \log p(\mathbf{x}|\mathbf{z}_0)$ models the likelihood of the data given its noisy representation $\mathbf{z}_0$, $\mathcal{L}_{base} = -\text{KL}(q(\mathbf{z}_T|\mathbf{x})|p(\mathbf{z}_T))$ models the difference between a standard normal distribution and the final latent variable $q(\mathbf{z}_T|\mathbf{x})$, and

$$\mathcal{L}_t = -\text{KL}(q(\mathbf{z}_s|\mathbf{x}, \mathbf{z}_t)|p(\mathbf{z}_s|\mathbf{z}_t)) \text{ for } t = 1, 2, ..., T.$$

Note that, in this formation of diffusion models, our neural network ϕ directly predicts $\hat{\mathbf{x}}$. However, Ho et al. (2020) and others have found optimization of ϕ to be made much easier when instead predicting the Gaussian noise added to $\mathbf{x}$ to create $\hat{\mathbf{x}}$. An intuition for how this changes the neural network's learning dynamics is that, when predicting back the noise added to the model's input, the network is being trained to more directly differentiate which part of $\mathbf{z}_t$ corresponds to the input's feature signal (i.e., the underlying data point $\mathbf{x}$) and which part corresponds to added feature noise. In doing so, if we let $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}$, our neural network can then predict $\hat{\boldsymbol{\epsilon}} = \phi(\mathbf{z}_t, t)$ such that:

$$\hat{\mathbf{x}} = (1/\alpha_t)\, \mathbf{z}_t - (\sigma_t/\alpha_t)\, \hat{\boldsymbol{\epsilon}}. \tag{19}$$

Kingma et al. (2021) and others have since shown that, when parametrizing our denoising neural network in this way, the loss term $\mathcal{L}_t$ reduces to:

$$\mathcal{L}_t = \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[\frac{1}{2} (1 - \text{SNR}(t-1)/\text{SNR}(t)) \|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}\|^2 \right] \tag{20}$$

Note that, in practice, the loss term $\mathcal{L}_{base}$ should be close to zero when using a noising schedule defined such that $\alpha_T \approx 0$. Moreover, if and when $\alpha_0 \approx 1$ *and* $\mathbf{x}$ is a discrete value, we will find $\mathcal{L}_0$ to be close to zero as well.

A.2 Zeroth Likelihood Terms for GCDM Optimization Objective

For the zeroth likelihood terms corresponding to each type of input feature, we directly adopt the respective terms previously derived by Hoogeboom et al. (2022). Doing so enables a negative log-likelihood calculation for GCDM's predictions. In particular, for integer node features, we adopt the zeroth likelihood term:

$$p(\mathbf{h}|\mathbf{z}_0^{(h)}) = \int_{\mathbf{h}-\frac{1}{2}}^{\mathbf{h}+\frac{1}{2}} \mathcal{N}(\mathbf{u}|\mathbf{z}_0^{(h)}, \sigma_0) \mathrm{d}\mathbf{u}, \tag{21}$$

where we use the CDF of a standard normal distribution, Φ, to compute Eq. 21 as $\Phi((\mathbf{h} + \frac{1}{2} - \mathbf{z}_0^{(h)})/\sigma_0) - \Phi((\mathbf{h} - \frac{1}{2} - \mathbf{z}_0^{(h)})/\sigma_0) \approx 1$ for reasonable noise parameters α_0 and σ_0 (Hoogeboom et al. (2022)). For categorical node features, we instead use the zeroth likelihood term:

$$p(\mathbf{h}|\mathbf{z}_0^{(h)}) = C(\mathbf{h}|\mathbf{p}), \mathbf{p} \propto \int_{1-\frac{1}{2}}^{1+\frac{1}{2}} \mathcal{N}(\mathbf{u}|\mathbf{z}_0^{(h)}, \sigma_0) \mathrm{d}\mathbf{u}, \tag{22}$$

where we normalize $\mathbf{p}$ to sum to one and where C is a categorical distribution (Hoogeboom et al. (2022)). Lastly, for continuous node positions, we adopt the zeroth likelihood term:

$$p(\mathbf{x}|\mathbf{z}_0^{(x)}) = \mathcal{N}\left(\mathbf{x}|\mathbf{z}_0^{(x)}/\alpha_0 - \sigma_0/\alpha_0 \hat{\boldsymbol{\epsilon}}_0, \sigma_0^2/\alpha_0^2 \mathbf{I}\right) \tag{23}$$

which gives rise to the log-likelihood component $\mathcal{L}_0^{(x)}$ as:

$$\mathcal{L}_0^{(x)} = \mathbb{E}_{\boldsymbol{\epsilon}^{(x)} \sim \mathcal{N}_x(\mathbf{0}, \mathbf{I})} \left[\log Z^{-1} - \frac{1}{2} \|\boldsymbol{\epsilon}^x - \phi^{(x)}(\mathbf{z}_0, 0)\|^2 \right], \tag{24}$$

where $d = 3$ and the normalization constant $Z = (\sqrt{2\pi} \cdot \sigma_0/\alpha_0)^{(N-1)\cdot d}$ - in particular, its $(N-1) \cdot d$ term - arises from the zero center of gravity trick mentioned in Section 3.4 (Hoogeboom et al. (2022)).

A.3 Diffusion Models and Equivariant Distributions

In the context of diffusion generative models of 3D data, one often desires for the marginal distribution $p(\mathbf{x})$ of their denoising neural network to be an invariant distribution. Towards this end, we observe that a conditional distribution $p(y|x)$ is equivariant to the action of 3D rotations by meeting the criterion:

$$p(y|x) = p(\mathbf{R}y|\mathbf{R}x) \text{ for all orthogonal } \mathbf{R}. \tag{25}$$

Moreover, a distribution is invariant to rotation transformations $\mathbf{R}$ when

$$p(y) = p(\mathbf{R}y) \text{ for all orthogonal } \mathbf{R}. \tag{26}$$

As Köhler et al. (2020) and Xu et al. (2022) have collectively demonstrated, we know that if $p(\mathbf{z}_T)$ is invariant and the neural network we use to parametrize $p(\mathbf{z}_{t-1}|\mathbf{z}_t)$ is equivariant, we have, as desired, that the marginal distribution $p(\mathbf{x})$ of the denoising model is an invariant distribution.

A.4 Training and Sampling Procedures for GCDM

Equivariant Dynamics. In this work, we use our previous definition of GCPNet in Section 3.4 to learn an SE(3)-equivariant dynamics function $[\hat{\boldsymbol{\epsilon}}^{(x)}, \hat{\boldsymbol{\epsilon}}^{(h)}] = \phi(\mathbf{z}_t^{(x)}, \mathbf{z}_t^{(h)}, t)$ as:

$$\hat{\boldsymbol{\epsilon}}_t^{(x)}, \hat{\boldsymbol{\epsilon}}_t^{(h)} = \text{GCPNet}(\mathbf{z}_t^{(x)}, [\mathbf{z}_t^{(h)}, t/T]) - [\mathbf{z}_t^{(x)}, \mathbf{0}], \tag{27}$$

where we inform our denoising model of the current time step by concatenating t/T as an additional node feature and where we subtract the coordinate representation outputs of GCPNet from its coordinate representation inputs after subtracting from the coordinate representation outputs their collective center of gravity. With the parametrization in Eq. 8, GCDM subsequently achieves rotation equivariance on $\hat{\mathbf{x}}_i$, thereby achieving a 3D translation and rotation-invariant marginal distribution $p(\mathbf{x})$ as described in Appendix A.3.

Scaling Node Features. In line with Hoogeboom et al. (2022), to improve the log-likelihood of our model's generated samples, we find it useful to train and perform sampling with GCDM using scaled node feature inputs as $[\mathbf{x}, \frac{1}{4}\mathbf{h}^{(categorical)}, \frac{1}{10}\mathbf{h}^{(integer)}]$.

Deriving The Number of Atoms. Finally, to determine the number of atoms with which GCDM will generate a 3D molecule, we first sample $N \sim p(N)$, where $p(N)$ denotes the categorical distribution of molecule sizes over GCDM's training dataset. Then, we conclude by sampling $\mathbf{x}, \mathbf{h} \sim p(\mathbf{x}, \mathbf{h}|N)$.