Molecule Generation for Target Receptor Binding via Continuous Normalizing Flows

Anonymous authors
Paper under double-blind review

Abstract

We propose an algorithm for learning a conditional generative model of a molecule given a target. Specifically, given a receptor molecule that one wishes to bind to, the conditional model generates candidate ligand molecules that may bind to it. The distribution should be invariant to rigid body transformations that act *jointly* on the ligand and the receptor; it should also be invariant to permutations of either the ligand or receptor atoms. Our learning algorithm is based on a continuous normalizing flow. We establish semi-equivariance conditions on the flow which guarantee the aforementioned invariance conditions on the conditional distribution. We propose a graph neural network architecture which implements this flow, and which is designed to learn effectively despite the vast differences in size between the ligand and receptor. We evaluate our method on the CrossDocked2020 dataset, attaining a 52.7% relative improvement over the current state of the art.

1 Introduction

The design of new molecules is an important topic, with applications in medicine, biochemistry, and materials science. Recently there have been quite a number of promising directions for applying machine learning techniques to this problem; most relevant to the current work are those which produce a generative model of molecules. However, to date most of the approaches have focused on the *unconditional* setting, in which the goal is simply to produce molecules without regard for a more specific purpose. Such techniques can, for example, effectively produce “drug-like” molecules. However, given two different targets that one wishes to bind to, the models will produce precisely the same distribution of candidates in both cases.

In this paper, we focus on learning generative models for molecules in the *conditional* setting. Specifically, we assume that we are given a target receptor molecule; the aim is then to generate ligand molecules that may successfully bind to this receptor. This kind of conditional generative model is very useful in the context of drug design, in which one often has a target (receptor) in mind, and the goal is then to find drugs (ligands) which will bind to the target.

It is important that our generative model be probabilistic, so that we can generate multiple potential candidates; diversity is useful in this context, as not all candidates will be equally suitable experimentally, due to considerations such as toxicity. We choose to describe both the receptor and ligand as 3D graphs, so our goal is to learn a conditional probabilistic model of one 3D graph given another.

We implement the central part of this conditional probabilistic model as a continuous normalizing flow. The work of Satorras et al. (2021a) pioneered this approach in the unconditional setting. In the conditional setting, a number of important changes must be made, and the conditioning variable (the receptor) must enter the flow in a very particular way. Specifically, we know that the probabilistic model must capture natural invariances, to rigid motions and permutations of the atoms. Note that the invariance to rigid motions is a kind of “conditional invariance”, which is expressed *jointly* in terms of the ligand and the receptor. Due to the fact that the receptor is a conditioning variable, this leads to a novel form of *semi-equivariance* of the flow, which we prove. An additional issue which arises is the vastly different sizes of the two molecules: the receptor is typically 1-2 orders of magnitude larger than the ligand. We introduce an architecture for implementing the flow which takes this into account, and enables effective learning despite the size disparity. We train our method on the CrossDocked2020 dataset (Francoeur et al., 2020) and attain high quality performance, with a relative improvement of 52.7% over the current state of the art in the key ΔBinding metric.

Our principal contributions are as follows:

- The derivation of conditions on a continuous normalizing flow which allows for joint invariance of the ligand and receptor to both rigid motions and permutations.
- The design of an architecture which implements the conditional distribution of ligand given receptor, enabling effective learning despite the size disparity between the molecules.
- A demonstration of the effectiveness of the method on the CrossDocked2020 dataset, attaining a 52.7% relative improvement over the state of the art in terms of ΔBinding.

2 Related Work

Continuous normalizing flows Normalizing flows transform an initial density distribution into a desired density distribution by applying a sequence of invertible transformations. They are easy to sample from and can be trained by maximum likelihood using the change of variables formula (Rippel & Adams, 2013; Dinh et al., 2014; Rezende & Mohamed, 2015; Dinh et al., 2016; Kingma & Dhariwal, 2018). Chen et al. (2018a) introduced a continuous flow based on ODEs, which was then extended in (Grathwohl et al., 2018). Further works aimed to improve the accuracy of the model (Zhuang et al., 2021; Zhang & Zhao, 2022); improve stability and deal with ODE stiffness issues through regularization (Finlay et al., 2020; Kelly et al., 2020; Ghosh et al., 2020); and deal with discrete data (Ho et al., 2019; Hoogeboom et al., 2021).

GNNs for drug design and discovery Graph Neural Networks (GNNs) are well-suited to problems in molecular modelling, with atoms and bonds represented as vertices and edges, respectively. GNN methods were shown to be useful for detecting receptor's binding site (pocket) and predicting the receptor pocket-ligand or protein-protein interactions (Gainza et al., 2020; Sverrisson et al., 2021; 2022); as well as the drug binding structure or the rigid body protein-protein docking structure (Ganea et al., 2021; Stärk et al., 2022). Several generative methods to produce molecules were previously demonstrated, for the unconditional setting (Gebauer et al., 2019; Satorras et al., 2021a; Hoogeboom et al., 2022; Trippe et al., 2022). Satorras et al. (2021a) constructed an E(n) equivariant normalizing flows, by incorporating equivariant graph neural networks (Satorras et al., 2021b) into an ODE framework to obtain an invertible equivariant function. The model to jointly generates molecular features and 3D positions.

3D molecular design conditioned on a receptor binding site This line of research is newer than the general (unconditioned) GNN-based methods described above, with fewer results. LiGAN (Ragoza et al., 2022) is a pioneering work in this area, which uses a condition VAE on an image-like discretized 3D atomic density grid; a post-sampling step converts this grid structure into molecules using an atom fitting algorithm. A more recent class of works (Luo et al., 2021; Liu et al., 2022) use an auto-regressive approach. Luo et al. (2021) derive a model which captures the probability that a point 3D space is occupied by an atom of a particular chemical element. Liu et al. (2022) propose the GraphBP framework, which eliminates the need to discretize space; to place a new atom, they generate its atom type and relative location while preserving the equivariance property. Other works include fragment-based ligand generation, in which new molecular fragments are sequentially attached to the growing molecule (Powers et al., 2022); abstraction of the geometric interaction features of the receptor–ligand complex to a latent space, for generative models such as Bayesian sampling (Wang et al., 2022b) and RNNs (Zhang & Chen, 2022); and use of experimental electron densities as training data for the conditional generative model (Wang et al., 2022a).

Data Datasets with a large number of receptor-ligand complexes are critical to our endeavour. Many models have relied on the high quality PDBbind dataset which curates the Protein Data Bank (PDB) (Liu et al., 2017); however, for the training of generative models, this dataset is relatively small. CrossDocked2020 (Francoeur et al., 2020) is the first large-scale standardized dataset for training ML models with ligand poses cross-docked against non-cognate receptor structure, greatly expanding the number of poses available for training. The dataset is organized by clustering of similar binding pockets across the PDB; each cluster contains ligands cross-docked against all receptors in the pocket. Each receptor-ligand structure also contains information indicating the nature of the docked pair, such as root mean squared deviation (RMSD) to the reference crystal pose and Vina cross-docking score (Trott & Olson, 2010) as implemented in Smina (Koes et al., 2013). The dataset contains 22.5 million poses of ligands docked into multiple similar binding pockets across the PDB.

3 Invariant Conditional Model of Ligands Given a Receptor

Objective Our overall goal can be stated as follows: wish to learn a conditional distribution over ligand molecules given a particular receptor molecule. As both ligand and receptor are described by 3D graphs, this will be a distribution of the form $p(\mathcal{G}|\hat{\mathcal{G}})$. To be physically plausible, this distribution must be invariant to certain groups of transformations: rigid motions applied jointly the entire ligand-receptor complex, as well as permutations applied to each of the ligand and receptor separately.

Organization This section proposes a method for learning such a conditional distribution $p(\mathcal{G}|\hat{\mathcal{G}})$. Section 3.1 introduces notation. Section 3.2 decomposes the distribution into 4 subdistributions using a Markov decomposition, and shows how the invariance properties apply to each subdistribution. Section 3.3 is an intermezzo, describing two types of Equivariant Graph Neural Networks (EGNNs); these EGNNs are then used in Sections 3.4 - 3.6 to propose forms for each of the subdistributions with the appropriate invariance properties. The main result appears in Section 3.5, which shows that the vertex subdistribution can be implemented as a particular type of continuous normalizing flow.

3.1 Notation and Goal

Notation We use the following notation. A molecular graph is given by

$$\mathcal{G} = (N, V, E, A) \tag{1}$$

where N is the number of atoms in the molecule; V is the list of vertices, which are the atoms; E is the list of edges, which are the bonds; and A is the set of global molecular properties, i.e. properties which apply to the entire molecule. The vertex list[1] is $V = (\mathbf{v}_i)_{i=1}^N$ where each vertex is specified by a vector $\mathbf{v}_i = (\mathbf{x}_i, \mathbf{h}_i)$; in which $\mathbf{x}_i \in \mathbb{R}^3$ is the position of the atom, and $\mathbf{h}_i \in \mathbb{R}^{d_h}$ contains the properties of the atom, such as the atom type. More generally, this may include continuous properties, discrete ordinal properties, and discrete categorical properties (using the one-hot representation); $\mathbf{h}_i$ may be thought of a concatenation of all such properties. The edges in the graph $\mathcal{G}$ are undirected, and the edge list E is specified by a neighbourhood relationship. Specifically, if η_i is the set of vertex i's neighbours, then we write $E = (\mathbf{e}_{ij})_{i<j:j\in\eta_i}$. The vector $\mathbf{e}_{ij} \in \mathbb{R}^{d_e}$ contains the properties of the bond connecting atom i and atom j, such as the bond type; more generally, $\mathbf{e}_{ij}$ may contain a concatenation of various properties in a manner analogous to the atom properties $\mathbf{h}_i$ as described above. Finally, the graph properties are given by K individual properties, i.e. $A = (\mathbf{a}_1, \dots, \mathbf{a}_K)$. A given property $\mathbf{a}_k$ can be either continuous, categorical or ordinal.

Rigid Transformations The action[2] of a rigid transformation $T \in E(3)$ on a graph $\mathcal{G}$ is given by $T\mathcal{G} = (TN, TV, TE, TA)$ where

$$TV = (T\mathbf{v}_i)_{i=1}^N \quad \text{with} \quad T\mathbf{v}_i = (T\mathbf{x}_i, \mathbf{h}_i) \tag{2}$$

and the other variables are unaffected by T; that is $TN = N$, $TE = E$, and $TA = A$.

Permutations The action of a permutation $\pi \in \mathbb{S}_N$ on a graph $\mathcal{G}$ with $N(\mathcal{G}) = N$ is given by $\pi\mathcal{G} = (\pi N, \pi V, \pi E, \pi A)$ where

$$\pi V = (\mathbf{v}_{\pi_i})_{i=1}^N \qquad \text{and} \qquad \pi E = \left(\mathbf{e}_{\pi_i \pi_j}\right)_{i<j:j\in\eta_i} \tag{3}$$

and the other variables are unaffected by π; that is, $\pi N = N$ and $\pi A = A$.

Goal We assume that we have both a receptor and a ligand, each of which is specified by a molecular graph. We denote

$$\mathcal{G} = (N, V, E, A) = \text{ligand graph} \qquad \text{and} \qquad \hat{\mathcal{G}} = (\hat{N}, \hat{V}, \hat{E}, \hat{A}) = \text{receptor graph} \tag{4}$$

(Note: if there molecular properties of the entire ligand-receptor complex, these are subsumed into the ligand molecular properties, A.) Our goal is to learn a conditional generative model: given the receptor, we would like to generate possible ligands. Formally, we want to learn

$$p(\mathcal{G}|\hat{\mathcal{G}}) \tag{5}$$

[1] We use lists, rather than sets, so as to make the action of permutations clear.

[2] Note to the reader: other papers such as Satorras et al. (2021a) use the notation $E(n)$ rather than $E(3)$, as the transformation is applied to multiple atoms. Here, we opt to use the notation $E(3)$, and $O(3)$ in the case of rotations, as there is a single transformation being applied to all of the atoms. Our notation is made sensible and precise given the definitions in Equation (2) and the surrounding text.

We want our generative model to observe two types of symmetries. First, if we transform both the ligand and the receptor with the same rigid transformation, the probability should not change:

$$p(T\mathcal{G}|T\hat{\mathcal{G}}) = p(\mathcal{G}|\hat{\mathcal{G}}) \quad \text{for all } T \in E(3) \tag{6}$$

Second, permuting the order of either the ligand or the receptor should not affect the probability:

$$p(\pi\mathcal{G}|\hat{\pi}\hat{\mathcal{G}}) = p(\mathcal{G}|\hat{\mathcal{G}}) \quad \text{for all } \pi \in \mathbb{S}_N,\ \hat{\pi} \in \mathbb{S}_{\hat{N}} \tag{7}$$

3.2 Decomposition of the Conditional Distribution and Invariance Properties

Decomposition We may breakdown the conditional generative model as follows:

$$p(\mathcal{G}|\hat{\mathcal{G}}) = p(N, V, E, A|\hat{\mathcal{G}}) = p(N|\hat{\mathcal{G}}) \cdot p(V|N, \hat{\mathcal{G}}) \cdot p(E|N, V, \hat{\mathcal{G}}) \cdot p(A|N, V, E, \hat{\mathcal{G}}) \tag{8}$$

We refer the four terms on the right-hand side of the equation as the Number Distribution, the Vertex Distribution, the Edge Distribution, and the Property Distribution, respectively. We will specify a model for each of these distributions in turn. First, however, we examine how the invariance properties affect the distributions.

Invariance Properties Using the relations for rigid body transformations in (2), we have that

$$p(T\mathcal{G}|T\hat{\mathcal{G}}) = p(N|T\hat{\mathcal{G}}) \cdot p(TV|N, T\hat{\mathcal{G}}) \cdot p(E|N, TV, T\hat{\mathcal{G}}) \cdot p(A|N, TV, E, T\hat{\mathcal{G}}) \tag{9}$$

Similarly, using the relations for permutations in (3), we have that

$$p(\pi\mathcal{G}|\hat{\pi}\hat{\mathcal{G}}) = p(N|\hat{\pi}\hat{\mathcal{G}}) \cdot p(\pi V|N, \hat{\pi}\hat{\mathcal{G}}) \cdot p(\pi E|N, \pi V, \hat{\pi}\hat{\mathcal{G}}) \cdot p(A|N, \pi V, \pi E, \hat{\pi}\hat{\mathcal{G}}) \tag{10}$$

Comparing Equation (8) with (9) and (10), the following conditions are sufficient for conditional rigid body invariance (6) and conditional permutation invariance (7):

$$p(N|T\hat{\mathcal{G}}) = p(N|\hat{\mathcal{G}}) \qquad p(N|\hat{\pi}\hat{\mathcal{G}}) = p(N|\hat{\mathcal{G}}) \tag{11}$$

$$p(TV|N, T\hat{\mathcal{G}}) = p(V|N, \hat{\mathcal{G}}) \qquad p(\pi V|N, \hat{\pi}\hat{\mathcal{G}}) = p(V|N, \hat{\mathcal{G}}) \tag{12}$$

$$p(E|N, TV, T\hat{\mathcal{G}}) = p(E|N, V, \hat{\mathcal{G}}) \qquad p(\pi E|N, \pi V, \hat{\pi}\hat{\mathcal{G}}) = p(E|N, V, \hat{\mathcal{G}}) \tag{13}$$

$$p(A|N, TV, E, T\hat{\mathcal{G}}) = p(A|N, V, E, \hat{\mathcal{G}}) \qquad p(A|N, \pi V, \pi E, \hat{\pi}\hat{\mathcal{G}}) = p(A|N, V, E, \hat{\mathcal{G}}) \tag{14}$$

3.3 Intermezzo: Two Flavours of EGNNs

In order to incorporate the relevant invariance properties, it will be helpful to use Equivariant Graph Neural Networks, also known as EGNNs (Satorras et al., 2021b). We now introduce two separate flavours of EGNNs, one which applies to the receptor alone, and a second which applies to the combination of the receptor and the ligand.

Receptor EGNN This is the standard EGNN which is described in (Satorras et al., 2021b), applied to the receptor. As we are dealing with the receptor we use hatted variables:

$$\hat{\mathbf{m}}_{ij}^{\ell} = \hat{\phi}_e(\hat{\mathbf{h}}_i^{\ell}, \hat{\mathbf{h}}_j^{\ell}, \|\hat{\mathbf{x}}_i^{\ell} - \hat{\mathbf{x}}_j^{\ell}\|^2, \|\hat{\mathbf{x}}_i^{0} - \hat{\mathbf{x}}_j^{0}\|^2, \hat{\mathbf{e}}_{ij}, \{\hat{\mathbf{a}}_k\}) \qquad \hat{b}_{ij}^{\ell} = \sigma(\hat{\phi}_b(\hat{\mathbf{m}}_{ij}^{\ell})) \qquad \hat{\mathbf{m}}_i^{\ell} = \sum_{j \in \hat{\eta}_i} \hat{b}_{ij}^{\ell} \hat{\mathbf{m}}_{ij}^{\ell}$$

$$\hat{\mathbf{x}}_i^{\ell+1} = \hat{\mathbf{x}}_i^{\ell} + \left(\sum_{j \neq i} \frac{(\hat{\mathbf{x}}_i^{\ell} - \hat{\mathbf{x}}_j^{\ell})}{\|\hat{\mathbf{x}}_i^{\ell} - \hat{\mathbf{x}}_j^{\ell}\| + 1} \right) \hat{\phi}_x(\hat{\mathbf{m}}_{ij}^{\ell}) \qquad \hat{\mathbf{h}}_i^{\ell+1} = \hat{\mathbf{h}}_i^{\ell} + \hat{\phi}_h(\hat{\mathbf{h}}_i^{\ell}, \hat{\mathbf{m}}_i^{\ell}) \tag{15}$$

The particular Receptor EGNN is thus specified by the functions $\hat{\phi}_e, \hat{\phi}_b, \hat{\phi}_x, \hat{\phi}_h$. In practice, prior to applying the EGNN one may apply: (i) an ActNorm layer to the atom positions and features, as well as the edge features and graph features; (ii) a linear transformation to the atom properties.

Receptor-Conditional Ligand EGNN It is possible to design a joint EGNN on the receptor and the ligand, by constructing a single graph to capture both. The main problem with this approach is that the receptor is much larger (1-2 orders of magnitude) than the ligand. As a result, this naive approach will lead to a situation in which the ligand is "drowned out" by the receptor, making it difficult to learn about the ligand.

We therefore take a different approach: we compute summary signatures of the receptor based on the Receptor EGNN, and use these as input to a ligand EGNN. Our signatures will be based on the feature variables $\hat{\mathbf{h}}_j^\ell$ from each layer $\ell = 1, \ldots, \hat{L}$. These variables are invariant to rigid body transformations by construction; furthermore, we can introduce permutation invariance by averaging, that is $\hat{\mathbf{h}}_{av}^\ell = \frac{1}{\hat{N}} \sum_{j=1}^{\hat{N}} \hat{\mathbf{h}}_j^\ell$. In Section 3.5, we will see that the vertex distribution is described by a continuous normalizing flow. In anticipation of this, we wish to introduce a dependence (crucial in practice) on the time variable t of the ODE corresponding to this flow. Thus, the receptor at layer ℓ of the ligand's EGNN is summarized by the signature $\hat{\mathbf{g}}^\ell$ which depends on both $\{\hat{\mathbf{h}}_{av}^\ell\}$ and t:

$$\hat{\mathbf{g}}^0 = \phi_g^0\left(\hat{\mathbf{h}}_{av}^1, \ldots, \hat{\mathbf{h}}_{av}^{\hat{L}}, t\right) \qquad \text{and} \qquad \hat{\mathbf{g}}^\ell = \phi_g^\ell\left(\hat{\mathbf{g}}^{\ell-1}\right) \quad \ell = 1, \ldots, L \tag{16}$$

These invariant receptor signatures $\{\hat{\mathbf{g}}^\ell\}_{\ell=1}^L$ are then naturally incorporated into the Receptor-Conditional Ligand EGNN as follows:

$$\mathbf{m}_{ij}^\ell = \phi_e\left(\mathbf{h}_i^\ell, \mathbf{h}_j^\ell, \|\mathbf{x}_i^\ell - \mathbf{x}_j^\ell\|^2, \|\mathbf{x}_i^0 - \mathbf{x}_j^0\|^2, \hat{\mathbf{g}}^\ell\right) \qquad b_{ij}^\ell = \sigma(\phi_a(\mathbf{m}_{ij}^\ell, \hat{\mathbf{g}}^\ell)) \qquad \mathbf{m}_i^\ell = \sum_{j=1}^N b_{ij}^\ell \mathbf{m}_{ij}^\ell$$

$$\mathbf{x}_i^{\ell+1} = \mathbf{x}_i^\ell + \left(\sum_{j \neq i} \frac{(\mathbf{x}_i^\ell - \mathbf{x}_j^\ell)}{\|\mathbf{x}_i^\ell - \mathbf{x}_j^\ell\| + 1}\right) \phi_x(\mathbf{m}_{ij}^\ell, \hat{\mathbf{g}}^\ell) \qquad \mathbf{h}_i^{\ell+1} = \mathbf{h}_i^\ell + \phi_h(\mathbf{h}_i^\ell, \mathbf{m}_i^\ell, \hat{\mathbf{g}}^\ell) \tag{17}$$

The particular Receptor-Conditional Ligand EGNN is thus specified by the functions $\{\phi_g^\ell\}_{\ell=0}^L, \phi_e, \phi_b, \phi_x, \phi_h$.

3.4 The Number Distribution: $p(N|\hat{\mathcal{G}})$

Construction Given the invariance conditions for the number distribution $p(N|\hat{\mathcal{G}})$ described in Equation (11), we propose the following distribution. Let $\boldsymbol{\zeta}_N$ indicate a one-hot vector, where the index corresponding to N is filled in with a 1. Based on the output of the receptor EGNN, compute

$$p(N|\hat{\mathcal{G}}) = \boldsymbol{\zeta}_N^T \texttt{MLP}' \left(\frac{1}{\hat{N}} \sum_{i=1}^{\hat{N}} \texttt{MLP}\left(\hat{\mathbf{h}}_i^L\right) \right) \tag{18}$$

where the outer `MLP`'s last layer is a softmax of size equal to the maximum number of atoms allowed.

Due to the fact that we use the $\hat{\mathbf{h}}_i^L$ vectors (and not the $\hat{\mathbf{x}}_i^L$ vectors), we have the first invariance condition, as $T\hat{\mathbf{h}}_i^L = \hat{\mathbf{h}}_i^L$. Due to the fact that we use an average, we have the second invariance condition. Note that we can choose to make the inner `MLP` the identity, if we so desire.

Loss Function The loss function is straightforward here – it is simply the negative log-likelihood of the number distribution, i.e. $L(\theta) = \mathbb{E}_{\mathcal{G}, \hat{\mathcal{G}}}\left[-\log p(N|\hat{\mathcal{G}}; \theta)\right]$.

3.5 The Vertex Distribution: $p(V|N, \hat{\mathcal{G}})$ via Continuous Normalizing Flows

General Notation Given the invariance conditions for the vertex distribution $p(V|N, \hat{\mathcal{G}})$ described in Equation (12), we now outline a procedure for constructing such a distribution. We begin with some notation. Let us define a vectorization operation on the vertex list V, which produces a vector $\mathbf{v}$; we refer to this as a *vertex vector*. Recall that $V = (\mathbf{v}_i)_{i=1}^N$ where $\mathbf{v}_i = (\mathbf{x}_i, \mathbf{h}_i)$. Let

$$\mathbf{x} = \texttt{concat}(\mathbf{x}_1, \ldots, \mathbf{x}_N) \qquad \mathbf{h} = \texttt{concat}(\mathbf{h}_1, \ldots, \mathbf{h}_N) \qquad \mathbf{v} = \texttt{concat}(\mathbf{x}, \mathbf{h}) \tag{19}$$

The vertex vector $\mathbf{v} \in \mathbb{R}^{d_v^N}$ where $d_v^N = (d_h + 3)N$. We denote the mapping from the vertex list V to the vertex vector $\mathbf{v}$ as the vectorization operation $\texttt{vec}(\cdot)$:

$$\mathbf{v} = \texttt{vec}(V) \qquad \text{and} \qquad V = \texttt{vec}^{-1}(\mathbf{v}) \tag{20}$$

We have already described the action of rigid body transformation T and permutations π on the vertex list V in Equations (2) and (3). It is easy to extend this to vertex vectors $\mathbf{v}$ using the `vec` operation; we have $T\mathbf{v} = \texttt{vec}(T\texttt{vec}^{-1}(\mathbf{v}))$ and $\pi\mathbf{v} = \texttt{vec}(\pi\texttt{vec}^{-1}(\mathbf{v}))$.

Given the above, it is sufficient for us to describe the distribution $p_{vec}(\mathbf{v}|\hat{\mathcal{G}})$ from which the vertex distribution $p(V|N,\hat{\mathcal{G}})$ follows directly, $p(V|N,\hat{\mathcal{G}}) = p_{vec}(\texttt{vec}(V)|\hat{\mathcal{G}})$. Note that we have suppressed N in the condition in $p_{vec}(\cdot)$, as $\mathbf{v}$ is a vector of dimension d_v^N, so the N dependence is already implicitly encoded.

Complex-to-Ligand Mapping and Semi-Equivariance Let γ be a function which takes as input the ligand graph $\mathcal{G}$ and receptor graph $\hat{\mathcal{G}}$, and outputs a new vertex list V' for the ligand graph $\mathcal{G}$:

$$\gamma : \mathcal{G} \times \hat{\mathcal{G}} \to \mathcal{V} \qquad V' = \gamma(\mathcal{G}, \hat{\mathcal{G}}) \tag{21}$$

We refer to γ as a *Complex-to-Ligand Mapping*. A rigid body transformation $T \in E(3)$ consists of a rotation and translation; let the rotation be denoted as T_{rot}. Then we say that γ is *rotation semi-equivariant* if

$$\gamma(T_{rot}\mathcal{G}, T\hat{\mathcal{G}}) = T_{rot}\gamma(\mathcal{G}, \hat{\mathcal{G}}) \quad \text{for all } T \in E(3) \tag{22}$$

where, as before, the action of T on a graph is given by Equation (2). γ is said to be *permutation semi-equivariant* if

$$\gamma(\pi\mathcal{G}, \hat{\pi}\hat{\mathcal{G}}) = \pi\gamma(\mathcal{G}, \hat{\mathcal{G}}) \quad \text{for all } \pi \in \mathbb{S}_N \text{ and } \hat{\pi} \in \mathbb{S}_{\hat{N}} \tag{23}$$

where, as before, the action of the permutation on a graph is given by Equation (3). Note in the definitions of both types of semi-equivariance, the differing roles played by the ligand and receptor; as the equivariant behaviour only applies to the ligand, we have used the term semi-equivariance.

Receptor-Conditioned Ligand Flow Let $\gamma : \mathcal{G} \times \hat{\mathcal{G}} \to \mathcal{V}$ be a Complex-to-Ligand Mapping. If $\mathbf{v}$ is a vertex vector, define $\mathcal{G}_{\mathbf{v}}$ to be the graph $\mathcal{G}$ with the vertex set replaced by $\texttt{vec}^{-1}(\mathbf{v})$. Then the following ordinary differential equation is referred to as a *Receptor-Conditioned Ligand Flow*:

$$\frac{d\mathbf{u}}{dt} = \texttt{vec}\left(\gamma(\mathcal{G}_{\mathbf{u}}, \hat{\mathcal{G}})\right), \quad \text{with } \mathbf{u}(0) = \mathbf{z} \tag{24}$$

where the initial condition $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ is a Gaussian random vector of dimension d_v^N, and the ODE is run until $t = 1$. $\mathbf{u}(1)$ is thus the output of the Receptor-Conditioned Ligand Flow.

Vertex Distributions with Appropriate Invariance We now have all of the necessary ingredients in order to construct a distribution $p_{vec}(\mathbf{v}|\hat{\mathcal{G}})$ which yields a vertex distribution $p(V|N,\hat{\mathcal{G}})$ that satisfies the invariance conditions that we require. The following is our main result:

Theorem. *Let $\mathbf{u}(1)$ be the output of a Receptor-Conditioned Ligand Flow specified by the Complex-to-Ligand Mapping γ. Let the mean position of the receptor be given by $\hat{\mathbf{x}}_{av} = \frac{1}{\hat{N}}\sum_{i=1}^{\hat{N}} \hat{\mathbf{x}}_i$, and define the following quantities*

$$\alpha = \frac{N}{N+\hat{N}} \qquad \Omega_{\hat{\mathcal{G}}} = \begin{bmatrix} \mathbf{I}_{3N} - \frac{\alpha}{N}\mathbf{1}_{N\times N} \otimes \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{d_h N} \end{bmatrix} \qquad \omega_{\hat{\mathcal{G}}} = \begin{bmatrix} -(1-\alpha)\mathbf{1}_{N\times 1} \otimes \hat{\mathbf{x}}_{av} \\ \mathbf{0} \end{bmatrix} \tag{25}$$

where $\otimes$ indicates the Kronecker product. Finally, let

$$\mathbf{v} = \Omega_{\hat{\mathcal{G}}}^{-1}\left(\mathbf{u}(1) - \omega_{\hat{\mathcal{G}}}\right) \tag{26}$$

Suppose that γ is both rotation semi-equivariant and permutation semi-equivariant. Then the resulting distribution on $\mathbf{v}$, that is $p_{vec}(\mathbf{v}|\hat{\mathcal{G}})$, yields a vertex distribution $p(V|N,\hat{\mathcal{G}}) = p_{vec}(vec(V)|\hat{\mathcal{G}})$ that satisfies the invariance conditions in Equation (12).

Proof: See Appendix A.1.

Designing the Complex-to-Ligand Mapping Examining the theorem, we see that the one degree of freedom that we have is the Complex-to-Ligand Mapping γ. For this, we choose to use the Receptor-Conditional Ligand EGNN, where the output of the Complex-to-Ligand Mapping $V' = \gamma(\mathcal{G}, \hat{\mathcal{G}})$ is simply the final layer, i.e. the i^{th} vertex of V' is given by $v_i' = (\mathbf{x}_i^L, \mathbf{h}_i^L)$.

The rotation semi-equivariance of γ follows straightforwardly from the rotation semi-equivariance of EGNNs and the rotation invariance of the receptor signatures $\{\hat{\mathbf{g}}^\ell\}_{\ell=1}^L$. Similarly, the permutation semi-equivariance follows from the permutation semi-equivariance of EGNNs and the permutation invariance of the receptor signatures.

Loss Function The vertex distribution is described by a continuous normalizing flow. The loss function and its optimization are implemented using standard techniques from this field (Chen et al., 2018b; Grathwohl et al., 2018; Chen et al., 2018a). As the feature vectors can contain discrete variables such as the atom type, then techniques based on variational dequantization (Ho et al., 2019) and argmax flows Hoogeboom et al. (2021) can be used, for ordinal and categorical features respectively. This is parallel to the treatment in (Satorras et al., 2021a).

3.6 Extensions: The Edge and Properties Distributions

We now outline methods for computing the Edge and Properties Distributions. In practice, we do not implement these methods, but rather use a standard simple technique based on inferring edge properties directly from vertices, see Section 4. Nevertheless, we describe these methods for completeness.

The Edge Distribution Given the invariance conditions for the edge distribution $p(E|N, V, \hat{\mathcal{G}})$ described in Equation (13), we propose a distribution which displays *conditional independence*: $p\left(E = (\mathbf{e}_{ij})_{i<j:j\in\eta_i} |N, V, \hat{\mathcal{G}}\right) = \prod_{i<j:j\in\eta_i} p(\mathbf{e}_{ij}|N, V, \hat{\mathcal{G}})$. We opt for conditional independence for two reasons: (1) The usual Markov decomposition of the probability distribution with terms of the form $p(\mathbf{e}_{ij}|\mathbf{e}_{<ij}, N, V, \hat{\mathcal{G}})$ implies a particular ordering of the edges, and is therefore not permutation-invariant. (2) V is a deterministic and invertible function of the flow's noise vector $\mathbf{z}$; thus, conditioning on V is the same as conditioning on $\mathbf{z}$. If E is a deterministic (but not necessarily invertible) function of $\mathbf{z}$, then conditional independence is correct.

To compute $p(\mathbf{e}_{ij}|N, V, \hat{\mathcal{G}})$, we use a second Receptor-Conditional Ligand EGNN. The key distinction between between this network and the Receptor-Conditional Ligand EGNN used in computing the vertex distribution is the initial conditions. In the case of the vertex distribution, the initial conditions are $\mathbf{x}_i^1 = \mathbf{0}$ and $\mathbf{h}_i^1 = \mathbf{0}$. In the current case of the edge distribution, we are given V (we are conditioning on it); thus, we take the initial conditions to be $\tilde{\mathbf{x}}_i^1 = \mathbf{x}_i(V)$ and $\tilde{\mathbf{h}}_i^1 = \mathbf{h}_i(V)$. In other words, the initial values are given the vertex list V itself.

Given this second Receptor-Conditional Ligand EGNN, we can compute the edge distribution as

$$p\left(\mathbf{e}_{ij}|N, V, \hat{\mathcal{G}}\right) = \mathbf{e}_{ij}^T \, \texttt{MLP}\left(\tilde{\mathbf{m}}_{ij}^L\right) \tag{27}$$

in the case of categorical properties (where MLP's output is a softmax with d_e entries); analogous expressions exist for ordinal or continuous properties. It is straightforward to see that this distribution satisfies the invariance properties in Equation (13). The corresponding loss function is a simple cross-entropy loss (or regression loss for non-categorical properties).

The Property Distribution Given the invariance conditions for the property distribution $p(A|N, V, E, \hat{\mathcal{G}})$ described in Equation (14), we propose the following distribution. We use a standard Markov decomposition: $p(A|N, V, E, \hat{\mathcal{G}}) = \prod_{k=1}^K p\left(\mathbf{a}_k|\mathbf{a}_{1:(k-1)}, N, V, E, \hat{\mathcal{G}}\right)$. Let

$$\boldsymbol{\xi}_h = \frac{1}{N}\sum_{i=1}^N \texttt{MLP}\left(\tilde{\mathbf{h}}_i^L\right) \qquad \boldsymbol{\xi}_e = \frac{1}{|E|}\sum_{i<j:j\in\eta_i} \texttt{MLP}\left(\mathbf{e}_{ij}\right) \qquad \boldsymbol{\xi}_{a,k} = \texttt{MLP}\left(\sum_{j=1}^{k-1} W_j \mathbf{a}_j\right) \tag{28}$$

where the matrices $W_1, \dots W_K$ all have the same number of rows. Then we set

$$p\left(\mathbf{a}_k \left| \mathbf{a}_{1:(k-1)}, N, V, E, \hat{\mathcal{G}}\right.\right) = \mathbf{a}_k^T \texttt{MLP}\left(\texttt{concat}\left(\boldsymbol{\xi}_h, \boldsymbol{\xi}_e, \boldsymbol{\xi}_{a,k}\right)\right) \tag{29}$$

in the case of categorical properties; analogous expressions exist for ordinal or continuous properties. Note that the only item which changes for the different properties k is the vector $\boldsymbol{\xi}_{a,k}$. It is easy to see that this distribution satisfies the invariance properties in Equation (14). The corresponding loss function is a simple cross-entropy loss (or regression loss for non-categorical properties).

4 Experiments

Data We use the CrossDocked2020 dataset (Francoeur et al., 2020) which contains poses of ligands docked into multiple similar binding pockets across the Protein Data Bank. We use the authors'

Validity	
Ours	GraphBP
99.86%	99.75%

Bond Length Distribution			
	Ref. Mols.	Ours	GraphBP
std	0.08	0.10	0.95
mean	1.42	1.45	1.65

(a) Ligand validity and bond length distribution

ΔBinding	
Ours	GraphBP
34.75%	22.76%

Predicted Affinity Distribution			
	Ref. Mols.	Ours	GraphBP
std	5.1	4.52	3.64
mean	1.16	1.01	1.09

(b) ΔBinding and predicted affinity distribution

Table 1: Comparison of molecule validity and ΔBinding between proposed method and GraphBP.

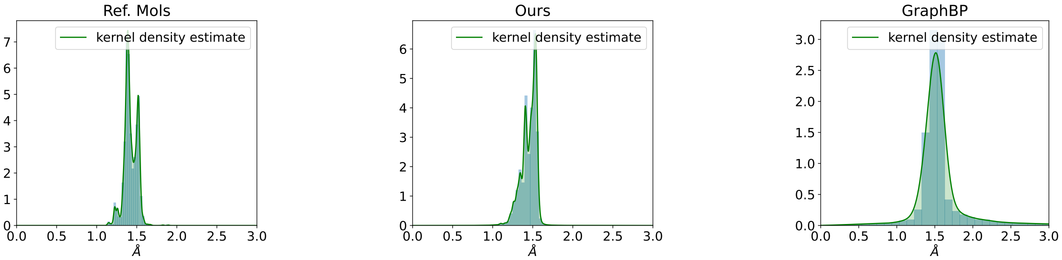

Figure 1: Normalized histogram of relative distances between atoms

suggested split into training and validation sets. The dataset contains docked receptor-ligand pairs whose binding pose RMSD is lower than 2 Å. We keep only those data points whose ligand has 30 atoms or fewer with atom types in {C, N, O, F}; which do not contain duplicate vertices; and whose predicted Vina scores (Trott & Olson, 2010) are within distribution. The refined datasets consist of 132,863 training data points and 63,929 validation data points. A full description of the data is summarized in Appendix A.2.

Features The ligand features that we wish to predict include the atom type $\in$ {C, N, O, F} (categorical); the stereo parity $\in$ {not stereo, odd, even} (categorical); and charge $\in \{-1, 0, +1\}$ (ordinal). The receptor features that are used are computed with the Graphein library (Jamasb et al., 2022). The vertex features include: the atom type $\in$ {C, N, O, S, "other"} where "other" is a catch-all for less common atom types[3] (categorical); the Meiler Embeddings (Meiler et al., 2001) (continuous $\in \mathbb{R}^7$). The bond (edge) properties include: the bond order $\in$ {Single, Double, Triple} (categorical); covalent bond length (continuous $\in \mathbb{R}$). The receptor overall graph properties ($\hat{A}$) contain the weight of all chains contained within a polypeptide structure, see (Jamasb et al., 2022).

Training Training the model takes approximately 14 days using a single NVIDIA A100 GPU for 30 epochs. We train with the Adam optimizer, weight decay of 10^{-12}, batch size of 128, and learning rate of 2×10^{-4}. We also train the baseline (state of the art) technique, GraphBP (Liu et al., 2022) on our filtered dataset. We train it for 100 epochs, using the hyperparameters given in the paper, with one exception: we set the atom number range of the autoregressive generative process according to the atom distribution of the filtered dataset.

Evaluation For both the proposed method as well as GraphBP, we perform inference using the method suggested in Liu et al. (2022). Given a receptor, we sample from the learned distribution, which generates the ligands' vertices; we then then apply OpenBabel (Hummell et al., 2021) to construct bonds. Evaluation follows the standard procedure (Francoeur et al., 2020; Liu et al., 2022). First, the receptor target is computed by taking all the atoms in the receptor that are less than 15 Å from the center of mass of the reference ligand; if the target has fewer than 200 atoms, the threshold of 15 Å until the 200 atom minimum is reached. We then generate 100 ligands for each reference binding site in the evaluation set, and compute statistics (i.e. validity and ΔBinding, see below) on this set of samples. As in (Francoeur et al., 2020; Liu et al., 2022), 10 target receptors for evaluation; each target receptor has multiple associated ligands, leading to 90 (receptor, reference-ligand) pairs.

[3] Specifically: Na, Mg, P, Cl, K, Ca, Co, Cu, Zn, Se, Cd, I, Hg.

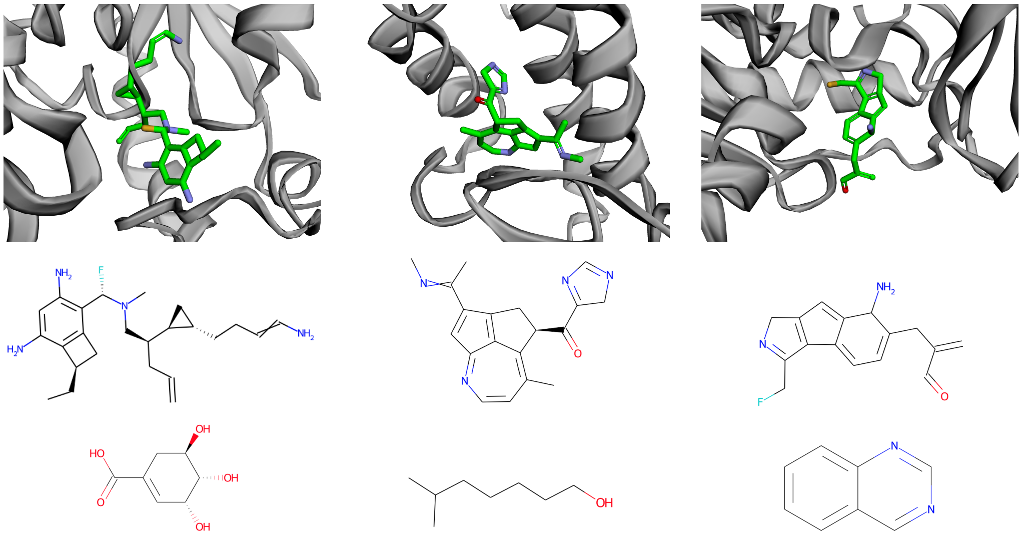

Figure 2: Comparison between generated 3D molecules for target binding-site and reference molecules. Receptor IDs, left to right: lzyu, 2qu9, 2hw1. Top: generated ligand (colour) + receptor. Middle: generated ligand chemical structure. Bottom: reference ligand chemical structure.

Validity The validity is defined as the percentage of molecules that are chemically valid among all generated molecules. A molecule is valid if it can be sanitized by RDKit; for an explanation of the sanitization procedure, see (Landrum, 2016). As shown in Table 1(a), our model produces ligands with a validity 0f 99.86%, surpassing the previous state of the art. We also compute the distribution of bond distances of the two methods, and compare this to distribution of the reference ligands; see Figure 1. Our method's distribution is considerably closer to the reference distribution than GraphBP; some non-trivial fraction of the time, GraphBP produces unusual, very high bond distances. (In fact, we have discarded values higher than 10 Å on the GraphBP plot so as to display the distributions on similar scales.) This impression is reinforced in Table 1(a) which compares the mean and standard deviation of these distributions.

Binding Affinity A more interesting measure than validity is ΔBinding, which measures the measures the percentage of generated molecules that have higher predicted binding affinity to the target binding site than the corresponding reference molecule. To compute binding affinities, we follow the procedure used by GraphBP. Briefly, we refine the generated 3D molecules by Universal Force Field minimization (Rappé et al., 1992); then, Vina minimization and CNN scoring are applied to both generated and reference molecules by using gnina, a molecular docking program (McNutt et al., 2021). As can be seen in Table 1(b), our result improves significantly on the state of the art. Raw GraphBP attains ΔBinding = 13.45%. By playing with the minimum and the maximum atom number of the baseline autoregressive model, we were able to improve this to 22.76%; however, note that this results in a reduction in validity from 99.76% to 99.54%. Our method attains ΔBinding = 34.75%, which is a relative improvement of 52.7% over the better of the two GraphBP scores.

Qualitative Results We show examples of generated ligands in Figure 2, along with their chemical structures. Note that the structures of the generated molecules differ substantially from the reference molecules, indicating that the model has indeed learn to generalize to interesting novel structures.

5 Conclusions

We have presented a method for learning a conditional distribution of ligands given a receptor. The method, which is based on a continuous normalizing flow, has provable invariance properties based on semi-equivariance conditions on the flow. Empirically, our method improves upon the state of the art by a considerable margin in ΔBinding, promising the potential to generate previously undiscovered molecules with high binding affinity.

6 Reproducibility Statement

In this research we establish semi-equivariance conditions on the continuous graph normalizing flow which guarantee the invariant to rigid body transformations conditions on the conditional distribution. The theory is described in Section 3.5, and the proof in Appendix A.1 shows formally that the semi-equivariance conditions yield the desired invariant distribution. For empirical testing of the method, we use the CrossDocked2020 dataset (Francoeur et al., 2020), with a data-split into a training set (see here) and validation set (see here); with additional filtering, as described in Section 4. The evaluation set is given here, and further changes are described in Appendix A.2. In order to add bonds to the resulting molecules, we rely on the LiGAN implementation, see here.

7 Ethics Statement

The work presents theory and experiments without discrimination/bias/fairness concerns, and without any legal compliance issues. No experiments on humans or animals were performed. The data used for experiments is publicly available and based on Protein Data Bank (PDB) (Berman et al., 2003; Zardecki et al., 2016). PDB is a data resource of 3D biomolecular structure information that exemplifies the FAIR (findability, accessibility, interoperability, and reusability) principles (Wilkinson et al., 2016; Burley et al., 2018). Within the PBD, the personal identifying information (PII) maintained on contributors is limited to the minimal contact information required to conduct the operations of the archive.

References

Helen Berman, Kim Henrick, and Haruki Nakamura. Announcing the worldwide protein data bank. Nature Structural & Molecular Biology, 10(12):980–980, 2003.

Stephen K Burley, Helen M Berman, Cole Christie, Jose M Duarte, Zukang Feng, John Westbrook, Jasmine Young, and Christine Zardecki. Rcsb protein data bank: Sustaining a living digital data resource that enables breakthroughs in scientific research and biomedical education. Protein Science, 27(1):316–330, 2018.

Changyou Chen, Chunyuan Li, Liqun Chen, Wenlin Wang, Yunchen Pu, and Lawrence Carin Duke. Continuous-time flows for efficient inference and density estimation. In International Conference on Machine Learning, pp. 824–833. PMLR, 2018a.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. Advances in neural information processing systems, 31, 2018b.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. arXiv preprint arXiv:1410.8516, 2014.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. arXiv preprint arXiv:1605.08803, 2016.

Chris Finlay, Jörn-Henrik Jacobsen, Levon Nurbekyan, and Adam Oberman. How to train your neural ode: the world of jacobian and kinetic regularization. In International conference on machine learning, pp. 3154–3164. PMLR, 2020.

Paul G Francoeur, Tomohide Masuda, Jocelyn Sunseri, Andrew Jia, Richard B Iovanisci, Ian Snyder, and David R Koes. Three-dimensional convolutional neural networks and a cross-docked data set for structure-based drug design. Journal of chemical information and modeling, 60(9):4200–4215, 2020.

Pablo Gainza, Freyr Sverrisson, Frederico Monti, Emanuele Rodola, D Boscaini, MM Bronstein, and BE Correia. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. Nature Methods, 17(2):184–192, 2020.

Octavian-Eugen Ganea, Xinyuan Huang, Charlotte Bunne, Yatao Bian, Regina Barzilay, Tommi Jaakkola, and Andreas Krause. Independent se (3)-equivariant models for end-to-end rigid protein docking. arXiv preprint arXiv:2111.07786, 2021.

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.

Arnab Ghosh, Harkirat Behl, Emilien Dupont, Philip Torr, and Vinay Namboodiri. Steer: Simple temporal regularization for neural ode. Advances in Neural Information Processing Systems, 33: 14831–14843, 2020.

Will Grathwohl, Ricky TQ Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. arXiv preprint arXiv:1810.01367, 2018.

Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In International Conference on Machine Learning, pp. 2722–2730. PMLR, 2019.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. Advances in Neural Information Processing Systems, 34:12454–12465, 2021.

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

Nicholas A Hummell, Alexey V Revtovich, and Natalia V Kirienko. Novel immune modulators enhance caenorhabditis elegans resistance to multiple pathogens. Msphere, 6(1):e00950–20, 2021.

Arian Rokkum Jamasb, Ramon Viñas Torné, Eric J Ma, Yuanqi Du, Charles Harris, Kexin Huang, Dominic Hall, Pietro Lio, and Tom Leon Blundell. Graphein-a python library for geometric deep learning and network analysis on biomolecular structures and interaction networks. In ICML 2022 2nd AI for Science Workshop, 2022.

Jacob Kelly, Jesse Bettencourt, Matthew J Johnson, and David K Duvenaud. Learning differential equations that are easy to solve. Advances in Neural Information Processing Systems, 33:4370–4380, 2020.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. Advances in neural information processing systems, 31, 2018.

David Ryan Koes, Matthew P Baumgartner, and Carlos J Camacho. Lessons learned in empirical scoring with smina from the csar 2011 benchmarking exercise. Journal of chemical information and modeling, 53(8):1893–1904, 2013.

Greg Landrum. Rdkit: Open-source cheminformatics software. 2016. URL https://github.com/rdkit/rdkit/releases/tag/Release_2016_09_4.

Meng Liu, Youzhi Luo, Kanji Uchino, Koji Maruhashi, and Shuiwang Ji. Generating 3d molecules for target protein binding. arXiv preprint arXiv:2204.09410, 2022.

Zhihai Liu, Minyi Su, Li Han, Jie Liu, Qifan Yang, Yan Li, and Renxiao Wang. Forging the basis for developing protein–ligand interaction scoring functions. Accounts of chemical research, 50 (2):302–309, 2017.

Shitong Luo, Jiaqi Guan, Jianzhu Ma, and Jian Peng. A 3d generative model for structure-based drug design. Advances in Neural Information Processing Systems, 34:6229–6239, 2021.

Andrew T McNutt, Paul Francoeur, Rishal Aggarwal, Tomohide Masuda, Rocco Meli, Matthew Ragoza, Jocelyn Sunseri, and David Ryan Koes. Gnina 1.0: molecular docking with deep learning. Journal of cheminformatics, 13(1):1–20, 2021.

Jens Meiler, Michael Müller, Anita Zeidler, and Felix Schmäschke. Generation and evaluation of dimension-reduced amino acid parameter representations by artificial neural networks. Molecular modeling annual, 7(9):360–369, 2001.

Alexander Powers, Helen Yu, Patricia Suriana, and Ron Dror. Fragment-based ligand generation guided by geometric deep learning on protein-ligand structure. bioRxiv, 2022.

Matthew Ragoza, Tomohide Masuda, and David Ryan Koes. Generating 3d molecules conditional on receptor binding sites with deep generative models. Chemical science, 13(9):2701–2713, 2022.

Anthony K Rappé, Carla J Casewit, KS Colwell, William A Goddard III, and W Mason Skiff. Uff, a full periodic table force field for molecular mechanics and molecular dynamics simulations. Journal of the American chemical society, 114(25):10024–10035, 1992.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In International conference on machine learning, pp. 1530–1538. PMLR, 2015.

Oren Rippel and Ryan Prescott Adams. High-dimensional probability estimation with deep density models. arXiv preprint arXiv:1302.5125, 2013.

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.

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.

Freyr Sverrisson, Jean Feydy, Bruno E Correia, and Michael M Bronstein. Fast end-to-end learning on protein surfaces. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15272–15281, 2021.

Freyr Sverrisson, Jean Feydy, Joshua Southern, Michael M Bronstein, and Bruno Correia. Physics-informed deep neural network for rigid-body protein docking. In ICLR2022 Machine Learning for Drug Discovery, 2022.

Brian L Trippe, Jason Yim, Doug Tischer, Tamara Broderick, David Baker, Regina Barzilay, and Tommi Jaakkola. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. arXiv preprint arXiv:2206.04119, 2022.

Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. Journal of computational chemistry, 31(2):455–461, 2010.

Lvwei Wang, Rong Bai, Xiaoxuan Shi, Wei Zhang, Yinuo Cui, Xiaoman Wang, Cheng Wang, Haoyu Chang, Yingsheng Zhang, Jielong Zhou, et al. A pocket-based 3d molecule generative model fueled by experimental electron density. Scientific reports, 12(1):1–14, 2022a.

Mingyang Wang, Chang-Yu Hsieh, Jike Wang, Dong Wang, Gaoqi Weng, Chao Shen, Xiaojun Yao, Zhitong Bing, Honglin Li, Dongsheng Cao, et al. Relation: A deep generative model for structure-based de novo drug design. Journal of Medicinal Chemistry, 65(13):9478–9492, 2022b.

Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. The fair guiding principles for scientific data management and stewardship. Scientific data, 3(1):1–9, 2016.

Christine Zardecki, Shuchismita Dutta, David S Goodsell, Maria Voigt, and Stephen K Burley. Rcsb protein data bank: A resource for chemical, biochemical, and structural explorations of large and small biomolecules, 2016.

Hong Zhang and Wenjun Zhao. Pnode: A memory-efficient neural ode framework based on high-level adjoint differentiation. arXiv preprint arXiv:2206.01298, 2022.

Jie Zhang and Hongming Chen. De novo molecule design using molecular generative models constrained by ligand–protein interactions. Journal of Chemical Information and Modeling, 62(14): 3291–3306, 2022.

Juntang Zhuang, Nicha C Dvornek, Sekhar Tatikonda, and James S Duncan. Mali: A memory efficient and reverse accurate integrator for neural odes. arXiv preprint arXiv:2102.04668, 2021.

A Appendix

A.1 Proof of Theorem

Lemma 1. *Let the mean position of the receptor be given by $\hat{\mathbf{x}}_{av} = \frac{1}{\hat{N}} \sum_{i=1}^{\hat{N}} \hat{\mathbf{x}}_i$, and define the following quantities*

$$\alpha = \frac{N}{N + \hat{N}} \qquad \Omega_{\hat{\mathcal{G}}} = \begin{bmatrix} \mathbf{I}_{3N} - \frac{\alpha}{N} \mathbf{1}_{N \times N} \otimes \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{d_h N} \end{bmatrix} \qquad \omega_{\hat{\mathcal{G}}} = \begin{bmatrix} -(1-\alpha) \mathbf{1}_{N \times 1} \otimes \hat{\mathbf{x}}_{av} \\ \mathbf{0} \end{bmatrix}$$

where $\otimes$ indicates the Kronecker product. Given the following mapping:

$$\mathbf{v} = \Omega_{\hat{\mathcal{G}}}^{-1} \left(\mathbf{u} - \omega_{\hat{\mathcal{G}}} \right) \tag{30}$$

Let the inverse mapping be denoted by $\Gamma^1_{\hat{\mathcal{G}}}$, i.e. $\mathbf{u} = \Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v})$. For any rigid transformation $T \in E(3)$, which consists of both a rotation and a translation, denote the transformation consisting only of the rotation of T as $T_{rot} \in O(3)$. Then

$$\Gamma^1_{T\hat{\mathcal{G}}}(T\mathbf{v}) = T_{rot} \Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v}).$$

Furthermore, for any permutations $\pi \in \mathbb{S}_N$ and $\hat{\pi} \in \mathbb{S}_{\hat{N}}$, then

$$\Gamma^1_{\hat{\pi}\hat{\mathcal{G}}}(\pi \mathbf{v}) = \pi \Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v}).$$

Proof: The mapping $\mathbf{u} = \Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v})$ is given by

$$\mathbf{u} = \Omega_{\hat{\mathcal{G}}} \mathbf{v} + \omega_{\hat{\mathcal{G}}} \tag{31}$$

Let us denote the parts of $\mathbf{u}$ corresponding to the coordinates and the features as $\mathbf{x}^{\mathbf{u}}$ and $\mathbf{h}^{\mathbf{u}}$, respectively; and use similar notation for $\mathbf{v}$. Then we have that

$$\mathbf{h}^{\mathbf{u}} = \mathbf{h}^{\mathbf{v}} \tag{32}$$

and

$$\mathbf{x}^{\mathbf{u}} = \left(\mathbf{I}_{3N} - \frac{\alpha}{N} \mathbf{1}_{N \times N} \otimes \mathbf{I}_3 \right) \mathbf{x}^{\mathbf{u}} - (1-\alpha) \mathbf{1}_{N \times 1} \otimes \hat{\mathbf{x}}_{av} \tag{33}$$

Breaking down this last equation by atom gives

$$\begin{aligned} \mathbf{x}^{\mathbf{u}}_i &= \mathbf{x}^{\mathbf{v}}_i - \frac{\alpha}{N} \sum_{j=1}^{N} \mathbf{x}^{\mathbf{v}}_j - (1-\alpha) \hat{\mathbf{x}}_{av} \\ &= \mathbf{x}^{\mathbf{v}}_i - (\alpha \mathbf{x}^{\mathbf{v}}_{av} + (1-\alpha) \hat{\mathbf{x}}_{av}) \\ &= \mathbf{x}^{\mathbf{v}}_i - \bar{\mathbf{x}}^{\mathbf{v}} \end{aligned} \tag{34}$$

where $\mathbf{x}^{\mathbf{v}}_{av}$ is the average coordinate position of $\mathbf{x}^{\mathbf{v}}$, and $\bar{\mathbf{x}}^{\mathbf{v}}$ indicates the average of all atoms in the entire complex, i.e. taking both the ligand and the receptor together.

Now, let us examine what happens when we apply the rigid transformation T to both $\mathbf{v}$ and the receptor graph $\hat{\mathcal{G}}$; that is, let us examine

$$\tilde{\mathbf{u}} = \Gamma^1_{T\hat{\mathcal{G}}}(T\mathbf{v}) \tag{35}$$

In the case of the features $\mathbf{h}$, they are invariant by design; thus

$$\begin{aligned} \mathbf{h}^{\tilde{\mathbf{u}}} &= \mathbf{h}^{\mathbf{T}\mathbf{v}} \\ &= \mathbf{h}^{\mathbf{v}} \\ &= \mathbf{h}^{\mathbf{u}} \end{aligned} \tag{36}$$

where the last line follows from Equation (32). In the case of the coordinates, the transformation is as follows:

$$\mathbf{x}_i^{T\mathbf{v}} = R\mathbf{x}_i^{\mathbf{v}} + t \tag{37}$$

where $R \in O(3)$ is the rotation matrix, and $t \in \mathbb{R}^3$ the translation vector, corresponding to rigid motion T. As we apply T to the receptor graph $\hat{\mathcal{G}}$, this has the effect of applying this transformation to each of the receptor atoms, and hence to their mean and the mean of the entire complex:

$$\hat{\mathbf{x}}_{av}^{T\hat{\mathcal{G}}} = R\hat{\mathbf{x}}_{av}^{\hat{\mathcal{G}}} + t \quad \Rightarrow \quad \bar{\mathbf{x}}^{\mathbf{Tv}} = R\bar{\mathbf{x}}^{\mathbf{v}} + t \tag{38}$$

Thus, following Equation (34), and substituting $T\mathbf{v}$ and $T\hat{\mathcal{G}}$ in place of $\mathbf{v}$ and $\hat{\mathcal{G}}$, we get

$$\begin{aligned} \mathbf{x}_i^{\tilde{\mathbf{u}}} &= \mathbf{x}_i^{T\mathbf{v}} - \bar{\mathbf{x}}^{T\mathbf{v}} \\ &= R\mathbf{x}_i^{\mathbf{v}} + t - (R\bar{\mathbf{x}}^{\mathbf{v}} + t) \\ &= R\left(\mathbf{x}_i^{\mathbf{v}} - \bar{\mathbf{x}}^{\mathbf{v}}\right) \\ &= R\mathbf{x}_i^{\mathbf{u}} \end{aligned} \tag{39}$$

Combining Equations (36) and (39), we have that

$$\tilde{\mathbf{u}} = T_{rot}\mathbf{u} \tag{40}$$

Since $\mathbf{u} = \Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v})$ and $\tilde{\mathbf{u}} = \Gamma^1_{T\hat{\mathcal{G}}}(T\mathbf{v})$, we have shown that $\Gamma^1_{T\hat{\mathcal{G}}}(T\mathbf{v}) = T_{rot}\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v})$, as desired.

In the case of the permutations, let us now set

$$\tilde{\mathbf{u}} = \Gamma^1_{\hat{\pi}\hat{\mathcal{G}}}(\pi\mathbf{v}) \tag{41}$$

It is easy to see that $\hat{\pi}$ has no effect; the only place the receptor enters is through the quantities $\hat{N}$ and $\hat{\mathbf{x}}_{av}$, both of which are permutation-invariant. For the features, we now have

$$\begin{aligned} \mathbf{h}^{\tilde{\mathbf{u}}} &= \mathbf{h}^{\pi\mathbf{v}} \\ &= \pi\mathbf{h}^{\mathbf{v}} \\ &= \pi\mathbf{h}^{\mathbf{u}} \end{aligned} \tag{42}$$

That is, the features are simply reordered according to π. With regard to the coordinates, we have that

$$\begin{aligned} \mathbf{x}_i^{\tilde{\mathbf{u}}} &= \mathbf{x}_i^{\pi\mathbf{v}} - \bar{\mathbf{x}}^{\pi\mathbf{v}} \\ &= \mathbf{x}_{\pi(i)}^{\mathbf{v}} - \bar{\mathbf{x}}^{\mathbf{v}} \\ &= \mathbf{x}_{\pi(i)}^{\mathbf{u}} \end{aligned} \tag{43}$$

The coordinates are also therefore simply reordered according to π. Summarizing, we have that

$$\tilde{\mathbf{u}} = \pi\mathbf{u} \tag{44}$$

This is exactly equal to

$$\Gamma^1_{\hat{\pi}\hat{\mathcal{G}}}(\pi\mathbf{v}) = \pi\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v})$$

which concludes the proof. □

Lemma 2. *Let $\mathbf{u}(1)$ be the output of a Receptor-Conditioned Ligand Flow specified by the Complex-to-Ligand Mapping γ which is rotation semi-equivariant and permutation semi-equivariant. This Receptor-Conditioned Ligand Flow maps the initial condition $\mathbf{z}$ to $\mathbf{u}(1)$; let the inverse mapping be denoted by $\Gamma^2_{\hat{\mathcal{G}}}$, i.e. $\mathbf{z} = \Gamma^2_{\hat{\mathcal{G}}}(\mathbf{u}(1))$. Then*

$$\Gamma^2_{T\hat{\mathcal{G}}}(T_{rot}\mathbf{u}) = T_{rot}\Gamma^2_{\hat{\mathcal{G}}}(\mathbf{u})$$

Furthermore, for any permutations $\pi \in \mathbb{S}_N$ and $\hat{\pi} \in \mathbb{S}_{\hat{N}}$, then

$$\Gamma^2_{\hat{\pi}\hat{\mathcal{G}}}(\pi\mathbf{u}) = \pi\Gamma^2_{\hat{\mathcal{G}}}(\mathbf{u})$$

Proof: Our first goal is to show that $\Gamma^2_{T\hat{\mathcal{G}}}(T_{rot}\mathbf{u}) = T_{rot}\Gamma^2_{\hat{\mathcal{G}}}(\mathbf{u})$. Let us define $F_{\hat{\mathcal{G}}}$ to be the inverse of $\Gamma^2_{\hat{\mathcal{G}}}$, and let $\mathbf{u} = F_{\hat{\mathcal{G}}}(\mathbf{z})$. Then

$$\begin{aligned}
\Gamma^2_{T\hat{\mathcal{G}}}(T_{rot}\mathbf{u}) = T_{rot}\Gamma^2_{\hat{\mathcal{G}}}(\mathbf{u}) \quad &\Leftrightarrow \quad F^{-1}_{T\hat{\mathcal{G}}}(T_{rot}F_{\hat{\mathcal{G}}}(\mathbf{z})) = T_{rot}F^{-1}_{\hat{\mathcal{G}}}(F_{\hat{\mathcal{G}}}(\mathbf{z})) \\
&\Leftrightarrow \quad F^{-1}_{T\hat{\mathcal{G}}}(T_{rot}F_{\hat{\mathcal{G}}}(\mathbf{z})) = T_{rot}\mathbf{z} \\
&\Leftrightarrow \quad F_{T\hat{\mathcal{G}}}(T_{rot}\mathbf{z}) = T_{rot}F_{\hat{\mathcal{G}}}(\mathbf{z}) \qquad (45)
\end{aligned}$$

Thus, it is sufficient to shows that $F_{T\hat{\mathcal{G}}}(T_{rot}\mathbf{z}) = T_{rot}F_{\hat{\mathcal{G}}}(\mathbf{z})$. For convenience, we shall set

$$\mathbf{u}(1) = F_{\hat{\mathcal{G}}}(\mathbf{z}) \quad \text{and} \quad \tilde{\mathbf{u}}(1) = F_{T\hat{\mathcal{G}}}(T_{rot}\mathbf{z}) \qquad (46)$$

In this case, $\mathbf{u}(1)$ is defined by the ODE

$$\frac{d\mathbf{u}}{dt} = \texttt{vec}\left(\gamma(\mathcal{G}_{\mathbf{u}}, \hat{\mathcal{G}})\right) \quad \text{with } \mathbf{u}(0) = \mathbf{z} \qquad (47)$$

whereas $\tilde{\mathbf{u}}(1)$ is defined by the ODE

$$\frac{d\tilde{\mathbf{u}}}{dt} = \texttt{vec}\left(\gamma(\mathcal{G}_{\tilde{\mathbf{u}}}, T\hat{\mathcal{G}})\right) \quad \text{with } \tilde{\mathbf{u}}(0) = T_{rot}\mathbf{z} \qquad (48)$$

Now, let us define $\breve{\mathbf{u}}(t) = T^{-1}_{rot}\tilde{\mathbf{u}}(t)$, so that $\tilde{\mathbf{u}}(t) = T_{rot}\breve{\mathbf{u}}(t)$. In this case, we have that:

1. $\breve{\mathbf{u}}(0) = T^{-1}_{rot}\tilde{\mathbf{u}}(0) = T^{-1}_{rot}T_{rot}\mathbf{z} = \mathbf{z}$.
2. $\frac{d\tilde{\mathbf{u}}}{dt} = T_{rot}\frac{d\breve{\mathbf{u}}}{dt}$.
3. $\texttt{vec}\left(\gamma(\mathcal{G}_{\tilde{\mathbf{u}}}, T\hat{\mathcal{G}})\right) = \texttt{vec}\left(\gamma(\mathcal{G}_{T_{rot}\breve{\mathbf{u}}}, T\hat{\mathcal{G}})\right) = T_{rot}\texttt{vec}\left(\gamma(\mathcal{G}_{\breve{\mathbf{u}}}, \hat{\mathcal{G}})\right)$, where the last equality is from the definition of rotation semi-equivariance of γ.

Plugging the above three results into the flow for $\tilde{\mathbf{u}}$ in Equation (48) yields

$$\begin{aligned}
T_{rot}\frac{d\breve{\mathbf{u}}}{dt} &= T_{rot}\texttt{vec}\left(\gamma(\mathcal{G}_{\breve{\mathbf{u}}}, \hat{\mathcal{G}})\right) \quad \text{with} \quad \breve{\mathbf{u}}(0) = \mathbf{z} \\
\Rightarrow \quad \frac{d\breve{\mathbf{u}}}{dt} &= \texttt{vec}\left(\gamma(\mathcal{G}_{\breve{\mathbf{u}}}, \hat{\mathcal{G}})\right) \quad \text{with} \quad \breve{\mathbf{u}}(0) = \mathbf{z} \qquad (49)
\end{aligned}$$

But this is precisely identical to the flow described in Equation (47); thus, we have that

$$\breve{\mathbf{u}}(t) = \mathbf{u}(t) \quad \text{for all } t \qquad (50)$$

But $\tilde{\mathbf{u}}(t) = T_{rot}\breve{\mathbf{u}}(t)$ so that $\tilde{\mathbf{u}}(t) = T_{rot}\mathbf{u}(t)$, and in particular $\tilde{\mathbf{u}}(1) = T_{rot}\mathbf{u}(1)$. Comparing with Equation (46) completes rigid motion part of the proof.

Let us now turn to permutations; the proof is similar, but we repeat it in full for completeness. Our goal is to show that $\Gamma^2_{\hat{\pi}\hat{\mathcal{G}}}(\pi\mathbf{u}) = \pi\Gamma^2_{\hat{\mathcal{G}}}(\mathbf{u})$. Let us define $F_{\hat{\mathcal{G}}}$ to be the inverse of $\Gamma^2_{\hat{\mathcal{G}}}$, and let $\mathbf{u} = F_{\hat{\mathcal{G}}}(\mathbf{z})$. Then

$$\begin{aligned}
\Gamma^2_{\hat{\pi}\hat{\mathcal{G}}}(\pi\mathbf{u}) = \pi\Gamma^2_{\hat{\mathcal{G}}}(\mathbf{u}) \quad &\Leftrightarrow \quad F^{-1}_{\hat{\pi}\hat{\mathcal{G}}}(\pi F_{\hat{\mathcal{G}}}(\mathbf{z})) = \pi F^{-1}_{\hat{\mathcal{G}}}(F_{\hat{\mathcal{G}}}(\mathbf{z})) \\
&\Leftrightarrow \quad F^{-1}_{\hat{\pi}\hat{\mathcal{G}}}(\pi F_{\hat{\mathcal{G}}}(\mathbf{z})) = \pi\mathbf{z} \\
&\Leftrightarrow \quad F_{\hat{\pi}\hat{\mathcal{G}}}(\pi\mathbf{z}) = \pi F_{\hat{\mathcal{G}}}(\mathbf{z}) \qquad (51)
\end{aligned}$$

Thus, it is sufficient to shows that $F_{\hat{\pi}\hat{\mathcal{G}}}(\pi\mathbf{z}) = \pi F_{\hat{\mathcal{G}}}(\mathbf{z})$. For convenience, we shall set

$$\mathbf{u}(1) = F_{\hat{\mathcal{G}}}(\mathbf{z}) \quad \text{and} \quad \tilde{\mathbf{u}}(1) = F_{\hat{\pi}\hat{\mathcal{G}}}(\pi\mathbf{z}) \qquad (52)$$

In this case, $\mathbf{u}(1)$ is defined by the ODE

$$\frac{d\mathbf{u}}{dt} = \texttt{vec}\left(\gamma(\mathcal{G}_{\mathbf{u}}, \hat{\mathcal{G}})\right) \quad \text{with } \mathbf{u}(0) = \mathbf{z} \qquad (53)$$

whereas $\tilde{\mathbf{u}}(1)$ is defined by the ODE

$$\frac{d\tilde{\mathbf{u}}}{dt} = \texttt{vec}\left(\gamma(\mathcal{G}_{\tilde{\mathbf{u}}}, \hat{\pi}\hat{\mathcal{G}})\right) \quad \text{with } \tilde{\mathbf{u}}(0) = \pi\mathbf{z} \qquad (54)$$

Now, let us define $\breve{\mathbf{u}}(t) = \pi^{-1}\tilde{\mathbf{u}}(t)$, so that $\tilde{\mathbf{u}}(t) = \pi\breve{\mathbf{u}}(t)$. In this case, we have that:

1. $\breve{\mathbf{u}}(0) = \pi^{-1}\tilde{\mathbf{u}}(0) = \pi^{-1}\pi\mathbf{z} = \mathbf{z}$.
2. $\frac{d\tilde{\mathbf{u}}}{dt} = \pi\frac{d\breve{\mathbf{u}}}{dt}$.
3. $\texttt{vec}\left(\gamma(\mathcal{G}_{\tilde{\mathbf{u}}}, \hat{\pi}\hat{\mathcal{G}})\right) = \texttt{vec}\left(\gamma(\mathcal{G}_{\pi\breve{\mathbf{u}}}, \hat{\pi}\hat{\mathcal{G}})\right) = \pi\texttt{vec}\left(\gamma(\mathcal{G}_{\breve{\mathbf{u}}}, \hat{\mathcal{G}})\right)$, where the last equality is from the definition of permutation semi-equivariance of γ.

Plugging the above three results into the flow for $\tilde{\mathbf{u}}$ in Equation (54) yields

$$\pi\frac{d\breve{\mathbf{u}}}{dt} = \pi\texttt{vec}\left(\gamma(\mathcal{G}_{\breve{\mathbf{u}}}, \hat{\mathcal{G}})\right) \quad \text{with} \quad \breve{\mathbf{u}}(0) = \mathbf{z}$$

$$\Rightarrow \quad \frac{d\breve{\mathbf{u}}}{dt} = \texttt{vec}\left(\gamma(\mathcal{G}_{\breve{\mathbf{u}}}, \hat{\mathcal{G}})\right) \quad \text{with} \quad \breve{\mathbf{u}}(0) = \mathbf{z} \tag{55}$$

But this is precisely identical to the flow described in Equation (53); thus, we have that

$$\breve{\mathbf{u}}(t) = \mathbf{u}(t) \quad \text{for all } t \tag{56}$$

But $\tilde{\mathbf{u}}(t) = \pi\breve{\mathbf{u}}(t)$ so that $\tilde{\mathbf{u}}(t) = \pi\mathbf{u}(t)$, and in particular $\tilde{\mathbf{u}}(1) = \pi\mathbf{u}(1)$. Comparing with Equation (52) completes the proof. □

Theorem. *Let $\mathbf{u}(1)$ be the output of a Receptor-Conditioned Ligand Flow specified by the Complex-to-Ligand Mapping γ. Let the mean position of the receptor be given by $\hat{\mathbf{x}}_{av} = \frac{1}{\hat{N}}\sum_{i=1}^{\hat{N}}\hat{\mathbf{x}}_i$, and define the following quantities*

$$\alpha = \frac{N}{N+\hat{N}} \qquad \Omega_{\hat{\mathcal{G}}} = \begin{bmatrix} \mathbf{I}_{3N} - \frac{\alpha}{N}\mathbf{1}_{N\times N}\otimes\mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{d_h N} \end{bmatrix} \qquad \omega_{\hat{\mathcal{G}}} = \begin{bmatrix} -(1-\alpha)\mathbf{1}_{N\times 1}\otimes\hat{\mathbf{x}}_{av} \\ \mathbf{0} \end{bmatrix} \tag{57}$$

where $\otimes$ indicates the Kronecker product. Finally, let

$$\mathbf{v} = \Omega_{\hat{\mathcal{G}}}^{-1}\left(\mathbf{u}(1) - \omega_{\hat{\mathcal{G}}}\right) \tag{58}$$

Suppose that γ is both rotation semi-equivariant and permutation semi-equivariant. Then the resulting distribution on $\mathbf{v}$, that is $p_{vec}(\mathbf{v}|\hat{\mathcal{G}})$, yields a vertex distribution $p(V|N,\hat{\mathcal{G}}) = p_{vec}(vec(V)|\hat{\mathcal{G}})$ that satisfies the invariance conditions in Equation (12).

Proof: The Receptor-Conditioned Ligand Flow maps from the Gaussian random variable $\mathbf{z}$ to the variable $\mathbf{u}(1)$. As this flow is a normalizing flow, it is invertible, so let us denote the inverse mapping by Γ^2:

$$\mathbf{z} = \Gamma^2_{\hat{\mathcal{G}}}(\mathbf{u}(1)) \tag{59}$$

Note that the dependence on the receptor graph $\hat{\mathcal{G}}$ is denoted using a subscript, as the invertibility does not apply to the receptor, but only to the ligand. Equation (58) maps from the variable $\mathbf{u}(1)$ to the variable $\mathbf{v}$; let us denote its inverse mapping by Γ^1:

$$\mathbf{u}(1) = \Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v}) \tag{60}$$

In this case, we have that

$$\mathbf{z} = \Gamma^2_{\hat{\mathcal{G}}}(\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v})) \equiv \Gamma_{\hat{\mathcal{G}}}(\mathbf{v}) \tag{61}$$

Now, our goal is to show that the following condition holds:

$$p(TV|N, T\hat{\mathcal{G}}) = p(V|N,\hat{\mathcal{G}}) \qquad \text{for } T \in E(3) \tag{62}$$

Using the p_{vec} notation, this translates to

$$p_{vec}(T\mathbf{v}|T\hat{\mathcal{G}}) = p_{vec}(\mathbf{v}|\hat{\mathcal{G}}) \tag{63}$$

Now, from Equation (61), the fact that Γ is invertible, and the change of variables formula, we have that

$$p_{vec}(\mathbf{v}|\hat{\mathcal{G}}) = p_{\mathbf{z}}(\Gamma_{\hat{\mathcal{G}}}(\mathbf{v}))|\det J_{\Gamma_{\hat{\mathcal{G}}}}(\mathbf{v})| \tag{64}$$

where $p_{\mathbf{z}}(\cdot)$ is the Gaussian distribution from $\mathbf{z}$ is sampled; and $J_{\Gamma_{\hat{\mathcal{G}}}}(\cdot)$ is the Jacobian of $\Gamma_{\hat{\mathcal{G}}}(\cdot)$. Since $\Gamma_{\hat{\mathcal{G}}} = \Gamma^2_{\hat{\mathcal{G}}} \circ \Gamma^1_{\hat{\mathcal{G}}}$, this can be expanded as

$$p_{vec}(\mathbf{v}|\hat{\mathcal{G}}) = p_{\mathbf{z}}(\Gamma^2_{\hat{\mathcal{G}}}(\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v})))|\det J_{\Gamma^2_{\hat{\mathcal{G}}}}(\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v}))||\det J_{\Gamma^1_{\hat{\mathcal{G}}}}(\mathbf{v})| \tag{65}$$

using the chain rule, and the fact that determinant of a product is the product of determinants. Plugging this into Equation (63), we must show that

$$\begin{aligned} & p_{\mathbf{z}}(\Gamma^2_{T\hat{\mathcal{G}}}(\Gamma^1_{T\hat{\mathcal{G}}}(T\mathbf{v})))|\det J_{\Gamma^2_{T\hat{\mathcal{G}}}}(\Gamma^1_{T\hat{\mathcal{G}}}(T\mathbf{v}))||\det J_{\Gamma^1_{T\hat{\mathcal{G}}}}(T\mathbf{v})| \\ & \quad = \quad p_{\mathbf{z}}(\Gamma^2_{\hat{\mathcal{G}}}(\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v})))|\det J_{\Gamma^2_{\hat{\mathcal{G}}}}(\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v}))||\det J_{\Gamma^1_{\hat{\mathcal{G}}}}(\mathbf{v})| \quad \text{for } T \in E(3) \end{aligned} \tag{66}$$

A rigid transformation $T \in E(3)$ consists of both a rotation and a translation. For brevity, denote the transformation consisting only of the rotation of T as $T_{rot} \in O(3)$. Now, from Lemma 1, we have that

$$\Gamma^1_{T\hat{\mathcal{G}}}(T\mathbf{v}) = T_{rot}\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v}) \tag{67}$$

From Lemma 2, we have that

$$\Gamma^2_{T\hat{\mathcal{G}}}(T_{rot}\mathbf{u}) = T_{rot}\Gamma^2_{\hat{\mathcal{G}}}(\mathbf{u}) \tag{68}$$

Combining Equations (67) and (68) gives that

$$\begin{aligned} p_{\mathbf{z}}(\Gamma^2_{T\hat{\mathcal{G}}}(\Gamma^1_{T\hat{\mathcal{G}}}(T\mathbf{v}))) &= p_{\mathbf{z}}(\Gamma^2_{T\hat{\mathcal{G}}}(T_{rot}\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v}))) \\ &= p_{\mathbf{z}}(T_{rot}\Gamma^2_{\hat{\mathcal{G}}}(\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v}))) \\ &= p_{\mathbf{z}}(\Gamma^2_{\hat{\mathcal{G}}}(\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v}))) \end{aligned} \tag{69}$$

where the last line follows from the rotation invariance of the Gaussian distribution.

Note that

$$\begin{aligned} J_{\Gamma^1_{T\hat{\mathcal{G}}}}(T\mathbf{v}) &= \frac{\partial}{\partial \mathbf{v}}\Big(\Gamma^1_{T\hat{\mathcal{G}}}(T\mathbf{v})\Big) \\ &= \frac{\partial}{\partial \mathbf{v}}\Big(T_{rot}\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v})\Big) \\ &= T_{rot}\frac{\partial}{\partial \mathbf{v}}\Big(\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v})\Big) \\ &= T_{rot}J_{\Gamma^1_{\hat{\mathcal{G}}}}(\mathbf{v}) \end{aligned} \tag{70}$$

Now, T_{rot} can be represented by the $d_v^N \times d_v^N$ block diagonal matrix given by

$$T_{rot} = \begin{bmatrix} \mathbf{1}_{N\times 1} \otimes R & 0 \\ 0 & \mathbf{I}_{d_h N} \end{bmatrix} \tag{71}$$

where $R \in O(3)$, the top-left block corresponds to the coordinates $\mathbf{x}$ and the bottom-right block corresponds to the feature $\mathbf{h}$. Thus,

$$\begin{aligned} \det\Big(J_{\Gamma^1_{T\hat{\mathcal{G}}}}(T\mathbf{v})\Big) &= \det\Big(T_{rot}J_{\Gamma^1_{\hat{\mathcal{G}}}}(\mathbf{v})\Big) \\ &= \det(T_{rot})\det\Big(J_{\Gamma^1_{\hat{\mathcal{G}}}}(\mathbf{v})\Big) \\ &= \det(R)^N \det(\mathbf{I}_{d_h N})\det\Big(J_{\Gamma^1_{\hat{\mathcal{G}}}}(\mathbf{v})\Big) \\ &= \pm\det\Big(J_{\Gamma^1_{\hat{\mathcal{G}}}}(\mathbf{v})\Big) \end{aligned} \tag{72}$$

where the second line follows from the fact that the determinant of a product is the product of determinants; the third line from the fact that the determinant of a block diagonal matrix is the product of the determinants of the blocks; and the fourth line from the fact that the determinant of a rotation matrix is ± 1.

To simplify $J_{\Gamma^2_{T\hat{\mathcal{G}}}}(\Gamma^1_{T\hat{\mathcal{G}}}(T\mathbf{v}))$, note that

$$\begin{aligned} \Gamma^2_{T\hat{\mathcal{G}}}(\mathbf{u}) &= \Gamma^2_{T\hat{\mathcal{G}}}(T_{rot}T_{rot}^{-1}\mathbf{u}) \\ &= T_{rot}\Gamma^2_{\hat{\mathcal{G}}}(T_{rot}^{-1}\mathbf{u}) \end{aligned} \tag{73}$$

where we have used Equation (68). Thus,

$$\begin{aligned} J_{\Gamma^2_{T\hat{\mathcal{G}}}}(\mathbf{u}) &= \frac{\partial}{\partial \mathbf{u}} \left(\Gamma^2_{T\hat{\mathcal{G}}}(\mathbf{u}) \right) \\ &= \frac{\partial}{\partial \mathbf{u}} \left(T_{rot} \Gamma^2_{\hat{\mathcal{G}}}(T_{rot}^{-1}\mathbf{u}) \right) \\ &= T_{rot} J_{\Gamma^2_{T\hat{\mathcal{G}}}}(T_{rot}^{-1}\mathbf{u}) T_{rot}^{-1} \end{aligned} \tag{74}$$

We wish to plug in $\mathbf{u} = \Gamma^1_{T\hat{\mathcal{G}}}(T\mathbf{v})$. Note that from Equation (67), $\Gamma^1_{T\hat{\mathcal{G}}}(T\mathbf{v}) = T_{rot}\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v})$. Thus,

$$\begin{aligned} J_{\Gamma^2_{T\hat{\mathcal{G}}}}(\Gamma^1_{T\hat{\mathcal{G}}}(T\mathbf{v})) &= J_{\Gamma^2_{T\hat{\mathcal{G}}}}(T_{rot}\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v})) \\ &= T_{rot} J_{\Gamma^2_{T\hat{\mathcal{G}}}}(T_{rot}^{-1} T_{rot} \Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v})) T_{rot}^{-1} \\ &= T_{rot} J_{\Gamma^2_{T\hat{\mathcal{G}}}}(\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v})) T_{rot}^{-1} \end{aligned} \tag{75}$$

where in the second line we substituted Equation (74). Taking determinants gives

$$\begin{aligned} \det\left(J_{\Gamma^2_{T\hat{\mathcal{G}}}}(\Gamma^1_{T\hat{\mathcal{G}}}(T\mathbf{v})) \right) &= \det\left(T_{rot} J_{\Gamma^2_{T\hat{\mathcal{G}}}}(\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v})) T_{rot}^{-1} \right) \\ &= \det\left(T_{rot}^{-1} T_{rot} J_{\Gamma^2_{T\hat{\mathcal{G}}}}(\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v})) \right) \\ &= \det\left(J_{\Gamma^2_{T\hat{\mathcal{G}}}}(\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v})) \right) \end{aligned} \tag{76}$$

where in the second line, we used the fact that permuting the order of a matrix multiplication does not affect the determinant.

Combining Equations (69), (72), and (76), we finally arrive at:

$$\begin{aligned} & p_{\mathbf{z}}(\Gamma^2_{T\hat{\mathcal{G}}}(\Gamma^1_{T\hat{\mathcal{G}}}(T\mathbf{v}))) |\det J_{\Gamma^2_{T\hat{\mathcal{G}}}}(\Gamma^1_{T\hat{\mathcal{G}}}(T\mathbf{v}))| |\det J_{\Gamma^1_{T\hat{\mathcal{G}}}}(T\mathbf{v})| \\ = \quad & p_{\mathbf{z}}(\Gamma^2_{\hat{\mathcal{G}}}(\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v}))) |\det J_{\Gamma^2_{\hat{\mathcal{G}}}}(\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v}))| |\det J_{\Gamma^1_{\hat{\mathcal{G}}}}(\mathbf{v})| \end{aligned} \tag{77}$$

which is exactly Equation (66). Thus, we have shown that $p(TV|N, T\hat{\mathcal{G}}) = p(V|N, \hat{\mathcal{G}})$, as desired.

Let us turn now to the permutation case, which is quite similar. Similar to Equation (66), we need to show

$$\begin{aligned} & p_{\mathbf{z}}(\Gamma^2_{\hat{\pi}\hat{\mathcal{G}}}(\Gamma^1_{\hat{\pi}\hat{\mathcal{G}}}(\pi\mathbf{v}))) |\det J_{\Gamma^2_{\hat{\pi}\hat{\mathcal{G}}}}(\Gamma^1_{\hat{\pi}\hat{\mathcal{G}}}(\pi\mathbf{v}))| |\det J_{\Gamma^1_{\hat{\pi}\hat{\mathcal{G}}}}(\pi\mathbf{v})| \\ = \quad & p_{\mathbf{z}}(\Gamma^2_{\hat{\mathcal{G}}}(\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v}))) |\det J_{\Gamma^2_{\hat{\mathcal{G}}}}(\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v}))| |\det J_{\Gamma^1_{\hat{\mathcal{G}}}}(\mathbf{v})| \quad \text{for } \pi \in \mathbb{S}_n \text{ and } \hat{\pi} \in \mathbb{S}_{\hat{N}} \end{aligned} \tag{78}$$

From Lemmata 1 and 2, we have that

$$\Gamma^1_{\hat{\pi}\hat{\mathcal{G}}}(\pi\mathbf{v}) = \pi\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v}) \quad \text{and} \quad \Gamma^2_{\hat{\pi}\hat{\mathcal{G}}}(\pi\mathbf{u}) = \pi\Gamma^2_{\hat{\mathcal{G}}}(\mathbf{u}) \tag{79}$$

Thus

$$\begin{aligned} p_{\mathbf{z}}(\Gamma^2_{\hat{\pi}\hat{\mathcal{G}}}(\Gamma^1_{\hat{\pi}\hat{\mathcal{G}}}(\pi\mathbf{v}))) &= p_{\mathbf{z}}(\Gamma^2_{\hat{\pi}\hat{\mathcal{G}}}(\pi\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v}))) \\ &= p_{\mathbf{z}}(\pi\Gamma^2_{\hat{\mathcal{G}}}(\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v}))) \\ &= p_{\mathbf{z}}(\Gamma^2_{\hat{\mathcal{G}}}(\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v}))) \end{aligned} \tag{80}$$

where the last line follows from the permutation-invariance of the Gaussian distribution $p_{\mathbf{z}}(\cdot)$.

In a manner parallel to the derivation of Equation (70), we can show that

$$J_{\Gamma^1_{\hat{\pi}\hat{\mathcal{G}}}}(\pi\mathbf{v}) = \boldsymbol{\pi} J_{\Gamma^1_{\hat{\mathcal{G}}}}(\mathbf{v}) \tag{81}$$

where $\boldsymbol{\pi}$ now indicates the permutation matrix associated with the permutation π. Thus, we have that

$$\det\left(J_{\Gamma^1_{\hat{\pi}\hat{\mathcal{G}}}}(\pi\mathbf{v}) \right) = \det(\boldsymbol{\pi}) \det\left(J_{\Gamma^1_{\hat{\mathcal{G}}}}(\mathbf{v}) \right) = \pm \det\left(J_{\Gamma^1_{\hat{\mathcal{G}}}}(\mathbf{v}) \right) \tag{82}$$

where we have used the fact that a permutation matrix has determinant of ± 1. Similarly, in a manner parallel to the derivation of Equation (75), we can show that

$$J_{\Gamma^2_{\hat{\pi}\hat{\mathcal{G}}}}(\Gamma^1_{\hat{\pi}\hat{\mathcal{G}}}(\pi\mathbf{v})) = \boldsymbol{\pi} J_{\Gamma^2_{\hat{\mathcal{G}}}}(\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v})) \boldsymbol{\pi}^{-1} \tag{83}$$

so that

$$\begin{aligned} \det\left(J_{\Gamma^2_{\hat{\pi}\hat{\mathcal{G}}}}(\Gamma^1_{\hat{\pi}\hat{\mathcal{G}}}(\pi\mathbf{v}))\right) &= \det\left(\boldsymbol{\pi} J_{\Gamma^2_{\hat{\mathcal{G}}}}(\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v}))\boldsymbol{\pi}^{-1}\right) \\ &= \det\left(\boldsymbol{\pi}^{-1}\boldsymbol{\pi} J_{\Gamma^2_{\hat{\mathcal{G}}}}(\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v}))\right) \\ &= \det\left(J_{\Gamma^2_{\hat{\mathcal{G}}}}(\Gamma^1_{\hat{\mathcal{G}}}(\mathbf{v}))\right) \end{aligned} \tag{84}$$

Combining Equations (80), (82), and (84) yields Equation (78); completing the proof. □

A.2 Data

We use the same approach to evaluate the generative model as done in previous works (Ragoza et al., 2022; Liu et al., 2022). The reference evaluation set consists of 10 target receptors with each having multiple associated ligands, leading to 90 receptor-ligand pairs. After the previously described filtering procedure (Sec. 4), 5 receptors remain corresponding to 27 receptor-ligand pairs. We complete the evaluation set, to yield 10 receptors corresponding to 90 pairs, by selecting data points from the validation set. These data points are chosen so that the ligands, receptors and pockets do not appear in the training set, and are not repeated in the evaluation set.