Streamlining Generative Models for Structure-Based Drug Design

Anonymous authors
Paper under double-blind review

Abstract

Generative models for structure-based drug design (SBDD) aim to generate novel 3D molecules for specified protein targets *in silico*. The prevailing paradigm focuses on model expressivity - typically with powerful Graph Neural Network (GNN) models - but is agnostic to binding affinity during training, potentially overlooking better molecules. We address this issue with a two-pronged approach: learn an economical surrogate for affinity to infer an unlabeled molecular graph, and optimize for labels conditioned on this graph and desired molecular properties (e.g., QED, SA). The resulting model FastSBDD achieves state-of-the-art results as well as streamlined computation and model size (up to 1000x faster and with 100x fewer trainable parameters compared to existing methods), paving way for improved docking software. We also establish rigorous theoretical results to expose the representation limits of GNNs in SBDD contexts and the generalizability of our affinity scoring model, advocating more emphasis on generalization going forward.

1 Introduction

Identifying *ligands* that bind well to a protein target and have desired properties is a key challenge of Structure-based drug design (SBDD), while experimental determination of binding is costly. Recently deep generative models have proposed to accelerate SBDD by suggesting good candidate molecules from learned conditional ligand distribution given a target protein from experimental or *in silico* binding observations (Vamathevan et al., 2019; Bilodeau et al., 2022). These approaches include autoregressive models (Peng et al., 2022; Luo et al., 2021a), variational autoencoders (Ragoza et al., 2022), reinforcement learning (Li et al., 2021), and diffusion models (Schneuing et al., 2022).

However, existing methods are oblivious to binding affinity as they seek to match the observed ligand distribution, which often includes molecules with varying affinities. Furthermore, they optimize jointly atom types and 3D coordinates with powerful GNN encoders with typically expensive training and sampling procedures. We address these issues with a novel two-phase generative method *FastSBDD* that draws inspiration from the success of performance-aware methods in other domains (Joachims, 2005), and directly optimizes for the metrics of interest, i.e. properties and binding affinity.

Specifically, we first learn a cheap affinity surrogate to infer an unlabeled molecular graph, and then optimize its atom labels and coordinates. We demonstrate the versatility of our approach with three versions of FastSBDD tailored to the problems of (i) drug repurposing, which generates samples from a known ligand set; (ii) drug generation, which generates novel unseen ligands; and (iii) property optimisation, which seeks to control both molecular properties and binding. FastSBDD achieves SBDD state of the art with up to 1000x speed up and 100x fewer parameters.

A broader objective, and contribution, of this work is to foster theoretical underpinnings for SBDD. We take two important steps in this pursuit. First, we expose the limits on the expressivity of GNNs to distinguish distinct molecules conditioned on protein targets. Next we prove generalization bounds for our binding surrogate to ground its strong empirical performance. While such results on representational limits and generalization ability are known for GNN models in unconditional settings (Garg et al., 2020; Joshi et al., 2023; Ju et al., 2023), to our knowledge, these are the first results in conditional drug discovery contexts.

1.1 Our contributions

We summarize our main contributions below. Specifically, we

1. **(Conceptual)** introduce metric-aware optimization for generative SBDD settings;
2. **(Methodological)** propose FastSBDD as a streamlined generative framework that optimizes for docking scores and physicochemical properties;
3. **(Empirical)** show that FastSBDD significantly outperforms existing methods with orders of magnitude fewer parameters and faster runtime; and
4. **(Theoretical)** analyze representational challenges of GNNs in SBDD contexts, and establish that FastSBDD generalizes well with respect to predicting scores for new target-ligand pairs.

2 Preliminaries and related work

Deep (Geometric) Learning for Drug Design. Deep learning has enabled progress on various facets of drug design across molecular generation (Jin et al., 2018; Zang & Wang, 2020; Verma et al., 2022; Shi et al., 2020; Luo et al., 2021b; Hoogeboom et al., 2022; Igashov et al., 2022), molecular optimization (Jin et al., 2019) and drug repurposing (Pushpakom et al., 2019). It has also accelerated advancements in protein folding and design (Jumper et al., 2021; Wu et al., 2022b; Ahdritz et al., 2022; Verkuil et al., 2022; Hie et al., 2022; Shi et al., 2023; Watson et al., 2022; Ingraham et al., 2022; Wu et al., 2022a; Verma et al., 2023) and docking (Stärk et al., 2022b; Ganea et al., 2022; Corso et al., 2023). We refer the reader to the surveys by Chen et al. (2018) and Pandey et al. (2022).

Graph Neural Networks (GNNs) (Kipf & Welling, 2017; Hamilton et al., 2017; Veličković et al., 2018) have recently emerged as a workhorse for modeling data, such as molecules, that can be represented as graphs. They can be extended to encode important symmetries occurring in geometric graphs (Satorras et al., 2021), so have found success across diverse tasks in the drug design pipeline including docking (Corso et al., 2023) and molecular property prediction (Stärk et al., 2022a).

Molecular properties The goal of SBDD is to generate drug candidates with high binding affinity to a target protein, while controlling for other properties. We list below common molecular properties, calculated in practice using the RDKit software (Landrum et al., 2020), that are relevant to drug design (Bilodeau et al., 2022; Peng et al., 2022):

- Binding affinity – the strength of the natural connection between the ligand and the protein. A common surrogate for affinity is the Vina score (Alhossary et al., 2015).
- LogP – logarithm of the partition coefficient, which is a solubility indicator.
- QED – quantitative estimate of drug-likeness, a mixture of properties correlated with drugs.
- SA – Synthetic accessibility, an estimate of how easy it would be to synthesize the molecule.
- Lipinski's Rule of Five – A heuristic about whether a molecule would be an active oral drug.

Setting. We view proteins P as labelled graphs: each pocket $G = (V, E)$ consists of nodes $V = \{v_1, \ldots, v_N\}$ with v_i representing atoms, and edges $E \subseteq V \times V$. Each node $v = (a, \mathbf{s})$ is associated with an atom type $a \in \mathcal{A} = \{\texttt{C}, \texttt{N}, \texttt{O}, \ldots\}$ and 3D coordinates $\mathbf{s} \in \mathbb{R}^3$, and each edge $e \in E$ with a bond type from $\{1, 2, 3\}$. We represent the molecule M analogously as another graph. We often include superscripts G^P and G^M to distinguish between protein and molecule graphs. We also distinguish between a labelled molecular graph G^M and unlabelled one U^M. The unlabelled graph contains information about the number of nodes and connectivity between them, but no information about atom types or 3D coordinates.

The problem of SBDD can be posed as modeling the conditional distribution $p(G^M|G^P)$. The main challenge of SBDD is to generate candidate ligands with high binding affinity to a target protein.

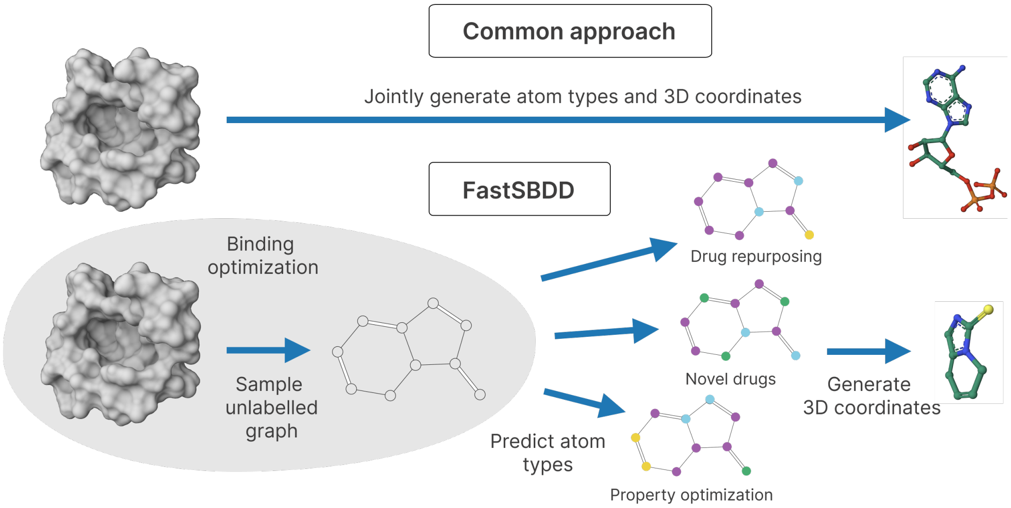


Figure 1: **Comparison of FastSBDD to common approaches**. Top: SBDD approaches commonly learn to approximate the data distribution of atom types and 3D coordinates conditioned on the protein pocket. Bottom: FastSBDD first generates the unlabelled graph explicitly optimized for binding affinity. Then it predicts atom types using different strategies designed for solving different tasks independently of the protein pocket. Finally, it generates a 3D configuration.

3 Fast structure-based drug design

3.1 Decoupling the unlabelled molecular graph from atom types

One of the main innovations underlying our framework is the separation of molecular representation into the unlabeled molecular graph and atom types. As a motivation, we first investigate how Vina binding scores behave when we modify the atom types and coordinates, while keeping the underlying graph intact. After making these adjustments, we measured the Vina scores of the modified molecules and compared them to the scores of their unaltered counterparts.

Remarkably, our analysis showed a strong coefficient of determination, with an R^2 value of 0.76, between the scores of the original molecules and their modified counterparts. This significant result highlights that a considerable portion of the variability in binding affinity, as quantified by the Vina score, can be attributed to the information contained within the unlabeled molecular graph itself, even before incorporating atom-specific details. Please see Appendix A for details.

It is important to recognize that these findings pertain to the Vina software and not the intrinsic biological process of binding. While Vina is the most prevalent approximation, it remains just that – an approximation. Nonetheless, upon conducting similar experiments with Gnina (McNutt et al., 2021), a reportedly significantly more accurate binding approximation than Vina, we found analogous results. Please see Appendix A.4 for details. This suggests that the observed behavior is not solely an artifact of Vina but may be more widespread.

Inspired by these findings, we propose a new disentangled model FastSBDD (see Figure 1). We first generate the high-binding unlabelled graph structures U^M, and generate atom types $\mathbf{a}^M$ independently of the protein. The model is defined by a disentangled conditional distribution

$$p(G^M|G^P) \overset{\text{def}}{=} p(\mathbf{a}^M, \mathbf{s}^M|U^M \cancel{, G^P})p(U^M|G^P), \tag{1}$$

where we assume atoms $\mathbf{a}^M$ are conditionally independent of the protein pocket G^P given the unlabelled graph structure U^M. The generative model thus reduces to two independent components: unlabelled graph sampler and atom sampler. We discuss them below.

3.2 Unlabelled graph sampler

Typically, one would train the unlabelled graph model $p(U^M|G^P)$ to match the empirical distribution. However, we hypothesize that observed ligand-protein complexes contain examples with suboptimal binding. We first learn a binding surrogate g_θ to predict binding affinity from unlabelled graph alone,

$$g_\theta(U^M, G^P) \approx \text{Vina}(G^M, G^P). \tag{2}$$

Next, we use g_θ to score unlabelled graphs from a repository of molecules, and choose those with the best predicted affinities. We define a generative distribution

$$p_\theta(U^M|G^P) = \text{UNIFORM}\Big(\big\{U^M \in \mathcal{U}^M \mid v_{\min} \leq g_\theta(G^P, U^M) \leq v_{\max}\big\}\Big), \tag{3}$$

where $\mathcal{U}^M$ is a database of unlabelled graph structures, and $v_{\min}, v_{\max}$ are threshold hyperparameters. We set $v_{\min}$ and $v_{\max}$ to be the 5th and 10th percentiles of all predictions for $\mathcal{U}^M$, and filter out the best 5% of the scores to avoid outliers. As $\mathcal{U}^M$, we used the ZINC250k dataset (Irwin et al., 2012) instead of the CrossDocked2020 (Francoeur et al., 2020), which only contains 8,000 unique ligands.

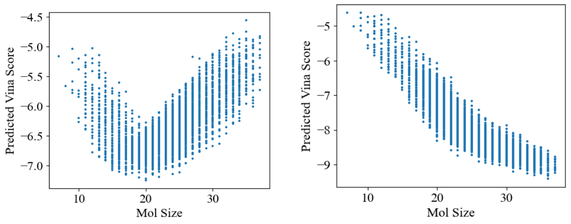
We note that instead of sampling from a database, one can define a deep generative model over unlabelled graphs. However, given that there is substantial variability in the molecular space even with predefined unlabelled graphs, we have not investigated this direction.

Architecture of the scoring model g_θ To parametrize the binding surrogate g_θ, we chose the E(3)-equivariant graph neural network (EGNN) (Satorras et al., 2021), which computes a learnable representation of the protein pocket. This protein pocket representation is concatenated with features describing the ligand's unlabelled graph. We chose as features (i) number of nodes, (ii) number of rings, (iii) number of rotatable bonds, and (iv) graph diameter. The features are passed through an MLP with ReLU activations. Remarkably, these four simple features already perform well, and more sophisticated methods like GNNs did not bring significant improvements. See Appendix B for details.

Training data for g_θ To train the binding affinity predictor g_θ, we need pairs of protein-ligand pairs with their affinity estimates. We chose not to use the CrossDocked2020 dataset, the gold standard for SBDD, for the following reasons: 1) even though it contains 100,000 protein-ligand pairs, there are only around 8,000 unique ligands, which can be insufficient for generalisation, 2) for a given protein pocket there may exist ligands with better affinity than what is present in the data and 3) for the scoring model to assign good values only to good structures, it needs to be trained also on examples with poor binding affinity that are not available.

To address the above, we construct a dataset ourselves. We sample 1000 diverse ligand-protein complexes from CrossDocked2020, and couple 50 additional random molecules from the ZINC250k. This results in 51,000 pairs, for which we compute the Vina ground-truth binding affinity. The scoring model g_θ is trained with stochastic gradient descent to minimize the mean squared error between prediction g and Vina scores. See Appendix B.1 for details.

Scoring model detects optimal molecule size To understand what the scoring learned, we visualized its predictions as a function of the size of the molecule. We notice that protein pockets fall under two cases: either there is an optimal size of the ligand (Figure 2a) or there is a monotonic relationship favouring larger ligands (Figure 2b). We hypothesize the model learnt the optimal ligand size per pocket, but for some pockets this size is larger than the largest available molecule (38 atoms) in the ZINC250k dataset.



(a) Protein pocket with optimal ligand size ≈ 20

(b) Protein pocket with optimal ligand size ≥ 38

Figure 2: **Optimal ligand size has two modes**. Representative examples showing that the scoring model learns the optimal ligand size which is different for different proteins.

3.3 Atom sampler

The atom sampler $p(\mathbf{a}^M, \mathbf{s}^M|U^M)$ generates atom types and coordinates of each atom. As mentioned in Section 3.1, the precise 3D conformation has marginal effect on the Vina score, and therefore we simply generate any valid 3D configuration with RDKit. We choose a factorisation

$$p(\mathbf{s}^M, \mathbf{a}^M|U^M) = p(\mathbf{s}^M|\mathbf{a}^M, U^M)p(\mathbf{a}^M|U^M), \tag{4}$$

where $p(\mathbf{s}^M|\mathbf{a}^M, U^M)$ is an off-the-shelf conformation generation model available in RDKit. In practice, the 3D configuration of the ligand must be in the vicinity of the protein pocket in order to correctly compute its binding affinity using Vina. We therefore train a separate model which predicts the center of mass of the ligand based on the protein pocket, and use it to center the conformers. See Appendix C for more details.

We propose three different approaches to modeling the atom type sampler $p(\mathbf{a}^M|U^M)$ that are tailored for solving different problems.

3.3.1 Drug repurposing

We begin with a model FastSBDD-$\mathcal{DR}$ for drug repurposing, i.e. sampling from known molecules that are synthesizable. We define the atom sampler to follow the empirical distribution of known ligands with a given unlabelled graph U^M:

$$p_{\text{DR}}(\mathbf{a}^M|U^M) = \text{UNIFORM}\left(\{\mathbf{a}^M|(\mathbf{a}^M, U^M) \in \mathcal{D}\}\right), \tag{5}$$

where $(\mathbf{a}^M, U^M)$ denotes a molecule with unlabelled graph U^M and atoms $\mathbf{a}^M$, and $\mathcal{D}$ is a molecular database. In our experiments, we used ZINC250 dataset (Irwin et al., 2012).

3.3.2 Novel drug generation

FastSBDD can also be applied for generation of novel drugs. We adapt an existing deep hierarchical generative model MoFlow (Zang & Wang, 2020), consisting of an unlabelled graph model f_U and a conditional atom model $f_{\mathbf{a}|U}$. The atom model induces a conditional distribution $p_{f_{\mathbf{a}|U}}(\mathbf{a}^M|U^M)$, which we use as our atom sampling distribution for novel drug generation:

$$p_{\mathcal{ND}}(\mathbf{a}^M|U^M) = p_{f_{\mathbf{a}|U}}(\mathbf{a}^M|U^M). \tag{6}$$

We use a pre-trained MoFlow with no further training or fine-tuning, so omit them from the trainable parameter count of the result (Table 1). We denote this model variant as FastSBDD-$\mathcal{ND}$.

3.3.3 Property optimization

Finally, we show that our approach allows for joint optimization of molecular properties and binding affinity. As an illustrative example, we follow Gómez-Bombarelli et al. (2018) and choose $5 \cdot \text{QED} + \text{SA}$ as the property mixture to optimize. We first sample 50 candidates for atom assignments using $\mathbf{a}^M \sim p_{\mathcal{ND}}(\mathbf{a}^M|U^M)$, and choose the one with the best value of the desired property. We emphasize that this optimization is performed with respect to atom types, while keeping the unlabelled graph structure U^M fixed. We denote this model FastSBDD-$\mathcal{PO}$.

We also consider optimization in the latent space of the atom type generative model. Since the atom type generative model of Section 3.3.2 is a normalizing flow, we can compute a latent representation of each atom type assignment. We use the pre-trained model for predicting molecular properties based on the latent representation of a molecule, and used its gradients to perform optimization in the latent space. This approach has the benefit of being faster as it does not require passing through the flow model at each iteration. In our experiments however this yielded worse results than the random search, suggesting suboptimal performance of the property predictor model.

4 Results

Data We follow related work of Peng et al. (2022); Schneuing et al. (2022) and use the CrossDocked2020 dataset (Francoeur et al., 2020) for evaluating the models in our experiments. We use the same train-test split, which seperates protein pockets based on their sequence similarity computed with MMseqs2 (Steinegger & Söding, 2017) and contains 100,000 train and 100 test protein-ligand pairs.

We follow the evaluation procedure described in Peng et al. (2022). For each of 100 test protein pockets, we

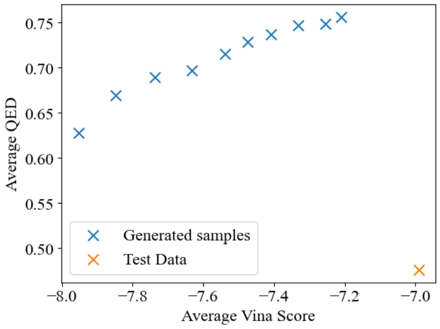


Figure 3: **Binding Affinity vs QED trade-off**. Our models allow for controlling the binding affinity of predictions.

Table 1: **FastSBDD finds high-quality drug candidates up to 1000x faster than competing methods**. Comparison of multiple properties of the CrossDocked2020 test data and other generation methods. Results for DiffSBDD were taken from Schneuing et al. (2022), where "High Affinity" was not reported. To report results for TargetDiff, we used samples provided by the authors. Reported runtime was taken from their paper, where standard deviation was not reported. † denotes method run exclusively on CPU. The #Params column refers to the number of trainable parameters.

	Vina Score (kcal/mol, ↓)	High Affinity (↑)	QED (↑)	SA (↑)	Diversity (↑)	#Params (↓)	Time (s, ↓)
Test set	-6.99 ± 2.16	-	0.48 ± 0.21	0.73 ± 0.14	-	-	-
DiffSBDD	-6.58 ± 2.06	-	0.50 ± 0.15	0.34 ± 0.09	**0.73** ± 0.11	3.5M	1634 ± 769
Pocket2Mol	-7.10 ± 2.56	0.55 ± 0.31	0.57 ± 0.16	0.74 ± 0.13	**0.72** ± 0.16	3.7M	2504 ± 220
TargetDiff	-6.91 ± 2.25	0.52 ± 0.32	0.48 ± 0.20	0.58 ± 0.13	**0.72** ± 0.09	2.5M	3428 ± NA
FastSBDD–$\mathcal{DR}$	**-7.82** ± 1.47	**0.72** ± 0.36	0.66 ± 0.15	**0.78** ± 0.08	0.68 ± 0.05	**23K**	**1.8**† ± 0.4
FastSBDD–$\mathcal{ND}$	**-7.78** ± 1.47	**0.71** ± 0.34	0.61 ± 0.18	0.69 ± 0.09	0.68 ± 0.06	**23K**	3.9† ± 0.9
FastSBDD–$\mathcal{PO}$	**-7.98**± 1.46	**0.75** ± 0.35	**0.80** ± 0.10	0.73 ± 0.08	0.66 ± 0.06	**23K**	115† ± 11

sample 100 molecules using each of our sampling strategies described in this section. As evaluation metrics we report 1) **Vina Score** estimating binding affinity (Alhossary et al., 2015), 2) **High Affinity**, which is the percentage of generated molecules with binding affinity at least as good as ground truth, 3) **QED**, a quantitative estimate of druglikeness, 4) **SA**, synthetic accesibility, 5) **Diversity** defined by average Tanimoto dissimilarity for the generated molecules in the pocket, 6) **#Params**, the number of trainable parameters and 7) **Time** to generate 100 molecules in seconds. We include more molecular metrics in Appendix D. As baselines, we use three recent methods for SBDD: Pocket2Mol (Peng et al., 2022), DiffSBDD (Schneuing et al., 2022) and TargetDiff (Guan et al., 2023).

We report the results in Table 1. Clearly, all our approaches significantly outperform the previous state of the art, Pocket2Mol while being 20-1000x faster despite being run solely on CPU (and not on GPU unlike the baselines). Additionally, our method has only 23k trainable parameters, which is two orders of magnitude less than the baselines. We also note a decrease in diversity compared to these baselines: this is expected given that our model uses known unlabelled graphs during sampling.

Trade-off between QED and Binding Affinity prediction As seen in Equation (3), our method allows for controlling the binding affinity of the generated ligands. We therefore investigated the performance of the model when we vary the $v_{\min}, v_{\max}$ thresholds. Specifically, we evaluated 10 different versions of the unlabelled graph sampler with $(v_{\min}, v_{\max})$ set to $(q_0, q_5), (q_5, q_{10}), \ldots, (q_{45}, q_{50})$, where q_k is the k^{th} percentile of the predictions for $\mathcal{U}^M$ (note that these percentiles depend on the protein pocket). We generated atom types using the repurposing model described in section 3.3.1. We note a trade-off between QED and Binding Affinity of these models, but all of them are still better on both of these criteria than the samples from the CrossDocked2020 dataset. We show this in Figure 3.

The model generalizes across molecular datasets In our experiments we have used the ZINC250k dataset (Irwin et al., 2012) to train the scoring model (2), to sample unlabelled graphs (3) and to sample atom types in the drug repurposing model (5). We now check whether the model generalizes across datasets. To do that, we evaluate the FastSBDD–$\mathcal{DR}$ model with, but with the ChEMBL dataset (Mendez et al., 2018; Davies et al., 2015) without retraining the scoring model. We found that the model performs very similarly in terms of binding affinity. See Table 2 for details. We discuss model's generalization ability from a theoretical perspective in Section 5.2.

We now move on to studying the representational challenges and generalization of models for SBDD.

5 Theoretical analysis

We showed empirically that very simple representations of the unlabelled graph perform remarkably well in terms of predicting binding affinity, and allow for significantly streamlined computation compared to much larger and more sophisticated models. Here we bring attention to the representational issues concerning large models. We show that despite adding considerable complexity, such models might have weaknesses in terms of what they can represent. While these limitations can possibly be circumvented with even more complex models, the expense in terms of their weaker

Table 2: **The model generalizes to other molecular datasets.** Comparison of the drug repurposing model with different sets of molecular structures. Both use the scoring model trained on the ZINC250k dataset (Irwin et al., 2012), but FastSBDD–$\mathcal{DR}$ (ZINC) uses ZINC250k also during sampling, while FastSBDD–$\mathcal{DR}$ (ChEMBL) uses ChEMBL (Mendez et al., 2018; Davies et al., 2015).

	Vina Score (kcal/mol, ↓)	High Affinity (↑)	QED (↑)	SA (↑)	Lipinski (↑)	LogP
Test set	-6.99 ± 2.16	-	0.48 ± 0.21	0.73 ± 0.14	4.34 ± 1.14	0.89 ± 2.73
FastSBDD–$\mathcal{DR}$ (ZINC)	-7.82 ± 1.47	0.72 ± 0.36	0.66 ± 0.15	0.78 ± 0.08	4.99 ± 0.05	2.97 ± 1.37
FastSBDD–$\mathcal{DR}$ (ChEMBL)	-8.03 ± 1.56	0.76 ± 0.33	0.56 ± 0.15	0.79 ± 0.08	4.99 ± 0.12	3.48 ± 0.95

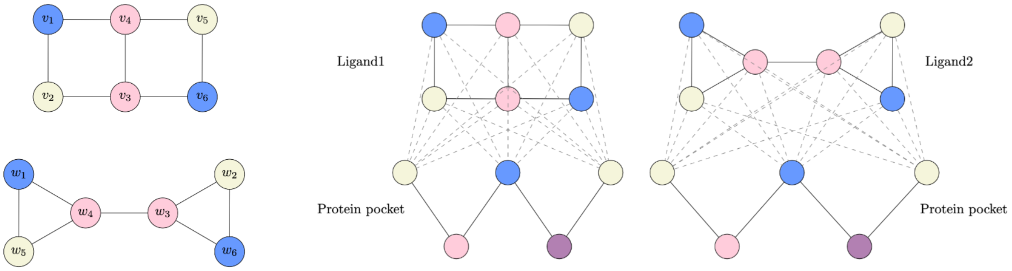


Figure 4: **Some ligand graphs cannot be distinguished by LU-GNNs even with additional protein context** Left: Construction for Lemma 1; Two non-isomorphic graphs differing in all properties stated in Lemma 1, but for which LU-GNNs produce identical embeddings. Right: Complex graphs constructed by joining the ligand graphs with the same protein graph remain identical whenever ligand graphs cannot be differentiated.

generalization ability (as known from learning theory results) could be excessive. Simpler, and likely less expressive, models like the ones considered in our work on the other hand might offer a better expressivity-generalization tradeoff and thus might be preferable from a theoretical perspective.

5.1 Representation limits of GNNs in SBDD

It is known that there exist graphs representing distinct molecular structures that cannot be distinguished using message passing graph neural networks (MPGNNs) (Sato, 2020; Garg et al., 2020). Following the notation in Garg et al. (2020), we consider *Locally Unordered* GNNs (LU-GNNs), which we summarize as follows. The updated embedding $h_v^{(l)}$ of node v at layer l is defined as

$$m_{u\to v}^{(l-1)} = \phi(h_u^{(l-1)}, e_{uv}); \tilde{h}_v^{(l-1)} = \text{AGG}\{m_{u\to v}^{(l-1)} \mid u \in N(v)\}; h_v^{(l)} = \text{COMBINE}\{h_v^{(l-1)}, \tilde{h}_v^{(l-1)}\},$$

where $N(v)$ denotes the set of neighbours of v, e_{uv} are edge features and ϕ is any function. It is known that there exist non-isomorphic graphs, for which LU-GNNs produce identical embeddings to every node at each layer (Garg et al., 2020). We now note that this result also holds for graphs embedded in 3D space. Let us introduce *Locally Unordered 3D* GNNs (LU3D-GNN) which aggregate embeddings for each node based on the embeddings of the neighbours, as well as their distance to these neighbours. Formally, in LU3D-GNNs, the messages are defined as:

$$m_{u\to v}^{(l-1)} = \phi\left(h_u^{(l-1)}, e_{uv}, \|x_u - x_v\|\right) \tag{7}$$

where x_v is the 3D position of v. We have the following result.

Lemma 1. *There exist connected non-isomorphic geometric graphs that differ in the number of conjoined cycles, girth, size of the largest cycle and cut-edges that LU3D-GNNs cannot distinguish.*

On the lefthand side of Figure 4, we show two graphs that are not isomorphic and differing in all mentioned properties, but which cannot be distinguished by LU3D-GNNs. For clarity of presentation, we presented the graphs in 2D, but any 3D configuration having all edges of equal length can be used. We prove that they cannot be distinguished by LU3D-GNNs in Appendix E.

In the context of SBDD, we are interested in modelling pairs of graphs corresponding to protein-ligand complexes. We now show that they also pose representational challenges for LU(3D)-GNNs.

Let $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ be any two graphs and $\mathcal{C}(G_1, G_2)$ denote a *complex* graph, i.e. $\mathcal{C}(G_1, G_2) = (V, E)$, where

$$V = V_1 \cup V_2 \text{ and } E = E_1 \cup E_2 \cup V_1 \times V_2 \tag{8}$$

and, importantly, features for all added edged $e_{uv} \in V_1 \times V_2$ only depend on the features of nodes at their endpoints. We have the following results.

Proposition 1.

- *(i) If G_1 and G_2 are indistinguishable for LU-GNNs, then for any graph P, the complex graphs $\mathcal{C}(P, G_1)$, $\mathcal{C}(P, G_2)$ are also indistinguishable for LU-GNNs.*
- *(ii) There exist ligand graphs G_1, G_2 differing in graph properties listed in Lemma 1, such that for any protein P, the complexes $\mathcal{C}(P, G_1)$, $\mathcal{C}(P, G_2)$ are indistinguishable for models using LU3D-GNNs or LU-GNNs for intra-ligand and intra-protein message passing and LU-GNNs for inter ligand-protein message passing.*

The righthand side of Figure 4 visualizes the proposed statement. The proof can be found in Appendix E. Proposition 1 shows that a scoring model defined in Equation (2) parametrized as a LU-GNN would not differentiate between certain ligand graphs, no matter what the protein context would be. This stems from the fact that we model unlabelled graphs without 3D coordinates and therefore cannot use LU3D-GNNs for inter ligand-protein message passing. We provide a real-world example of a pair of molecules that have different binding affinities to a specific protein pocket, but which are identical from LU-GNN perspective.

5.2 Generalization of the scoring model

We now delve into the theoretical analysis of the generalization capabilities of our proposed scoring model within FastSBDD. Understanding generalization is paramount, as it provides insight into how well our model can perform on unseen data, which is a fundamental aspect of its practical utility in structure-based drug design (SBDD). Our analysis is focused on deriving generalization bounds, which give a theoretical measure of the model's performance across different molecular structures and protein targets. We begin with defining the generalization error.

Definition 1. (Generalization error). *Let $f : \mathcal{X} \to \mathcal{Y}$ and $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ be a loss function. Let $S = \{(x_1, y_1), \dots, (x_m, y_m)\} \subseteq \mathcal{X} \times \mathcal{Y}$ be a training sample. The generalization error of f is defined as the difference between expected loss and the sample loss:*

$$\mathcal{R}_S(f) = \mathbb{E}L(f(x), y) - \frac{1}{m}\sum_{i=1}^{m} L(f(x_i), y_i), \tag{9}$$

where the expectation is taken over (x, y) sampled from the underlying data distribution.

ε-soft normalized EGNN As described in Section 3.2, our scoring model first computes an embedding of the protein graph, then concatenates features of ligand's unlabelled graph and passes through an MLP. However, we use a slightly modified version of the EGNN model. As originally introduced, a layer of the EGNN network can be summarized as follows:

$$\mathbf{m}_{u\to v}^{(l-1)} = \phi_e\left(\mathbf{h}_u^{(l-1)}, \mathbf{h}_v^{(l-1)}, \|\mathbf{x}_u^{(l-1)} - \mathbf{x}_v^{(l-1)}\|, e_{uv}\right); \mathbf{m}_v^{(l-1)} = \sum_{u\in\mathcal{N}(v)} \mathbf{m}_{u\to v}^{(l-1)} \tag{10}$$

$$\mathbf{x}_v^{(l)} = \mathbf{x}_v^{(l-1)} + \frac{1}{|\mathcal{N}(v)|}\sum_{u\in\mathcal{N}(v)} \left(\mathbf{x}_v^{(l-1)} - \mathbf{x}_u^{(l-1)}\right)\phi_x(\mathbf{m}_{u\to v}^{(l-1)}) \tag{11}$$

$$\mathbf{h}_v^{(l)} = \phi_h(\mathbf{h}_v^{(l-1)}, \mathbf{m}_v^{(l-1)}), \tag{12}$$

where $\mathbf{h}_v^{(0)}$ and $\mathbf{x}_v^{(0)}$ are initialized with node's features and coordinates respectively and ϕ_x, ϕ_e, ϕ_h are learnable functions, typically MLPs. In our experiments, we replace (11) with

$$\mathbf{x}_v^{(l)} = \mathbf{x}_v^{(l-1)} + \frac{1}{|\mathcal{N}(v)|}\sum_{u\in\mathcal{N}(v)} \frac{\mathbf{x}_v^{(l-1)} - \mathbf{x}_u^{(l-1)}}{\|\mathbf{x}_v^{(l-1)} - \mathbf{x}_u^{(l-1)}\| + \varepsilon}\phi_x(\mathbf{m}_{u\to v}^{(l-1)}). \tag{13}$$

The authors include this version of the model in the official implementation[1], but do not use it in the experiments. With this formulation, we derive the following generalization bound:

Proposition 2 (Informal)**.** *Let f be the scoring model using ε-soft normalization in the EGNN network. Then for the sample of size m, the following holds*

$$\mathcal{R}_S(f) \leq \mathcal{O}\left(\frac{L^2 + k}{\sqrt{m}}\sqrt{\log\left(\frac{m(L+k)}{\varepsilon}\right)}\right), \tag{14}$$

where L is the number of layers of the EGNN and k is the number of layers in the MLP.

We note that we make no assumptions about ε other than $\varepsilon > 0$. If generalization is the primary concern, one can set $\varepsilon = 1$.

Proof sketch. The key step is establishing that the model is Lipschitz continuous w.r.t. parameters and determining the Lipschitz constant. We show it by first proving that Lipschitz continuity is preserved under ε-soft normalization. We then use an inductive argument to derive recursive relationship for the constants and determine their growth rate as a function of the depth of the network. Once the Lipschitz constants are established, we leverage results from Learning theory relating the covering number of a function class to the empirical Rademacher complexity, which can subsequently be used to bound the geneneralization error. Please see Appendix F for a detailed formulation and proof. □

Impact of ε-soft normalization Interestingly, when analyzing the generalization bounds of normalized EGNN model vis-à-vis the original formulation, we observe an important difference. Our derivations indicate that, in the absence of normalization, the original EGNN model would exhibit significantly less favorable generalization bounds. Specifically, using the same proof technique, the bound for the EGNN without normalization would be exponential in L as opposed to polynomial. Please see Appendix F.5 for details. This discovery underscores the importance of the normalization component, emphasizing its benefits in both numerical stability and theoretical generalizability.

6 Conclusion, Broader Impact and Limitations

Our research brings to the forefront efficient methodologies for Structure-Based Drug Design (SBDD), advocating for a shift in focus from mere model expressivity to robust generalization. The successes of FastSBDD serve as a testament to the significance of integrating metrics of interest into the optimization process. The unprecedented efficiency and success of FastSBDD not only highlights the potential to reshape the approach to SBDD but also paves the way for streamlining docking software, especially when assessed with widely adopted tools like Vina and Gnina.

However, there are limitations to our findings. While we rely on Vina, the gold standard for evaluation in the SBDD realm, the ideal validation would entail wet lab experiments or more sophisticated molecular dynamics simulations. However, such methods are often prohibitively expensive. Additionally, the very nature of FastSBDD as a generative model, especially with its property optimization capabilities, brings with it risks. As SBDD models become increasingly adept at designing molecules with specific properties, we must remain vigilant of the potential unintended outcomes, since streamlining the design process could accelerate the design of harmful biochemicals.

Reproducibility statement All our experiments are reproducible. We will share our code as well as the trained models under the MIT License on GitHub upon the acceptance of the paper.

References

Gustaf Ahdritz, Nazim Bouatta, Sachin Kadyan, Qinghui Xia, William Gerecke, Timothy J O'Donnell, Daniel Berenberg, Ian Fisk, Niccolò Zanichelli, Bo Zhang, Arkadiusz Nowaczynski, Bei Wang, Marta M Stepniewska-Dziubinska, Shang Zhang, Adegoke Ojewole, Murat Efe Guney, Stella Biderman, Andrew M Watkins, Stephen Ra, Pablo Ribalta Lorenzo, Lucas Nivon, Brian Weitzner, Yih-En Andrew Ban, Peter K Sorger, Emad Mostaque, Zhao Zhang, Richard Bonneau, and

[1] https://github.com/vgsatorras/egnn

Mohammed AlQuraishi. Openfold: Retraining alphafold2 yields new insights into its learning mechanisms and capacity for generalization. *bioRxiv*, 2022.

Amr Alhossary, Stephanus Daniel Handoko, Yuguang Mu, and Chee-Keong Kwoh. Fast, accurate, and reliable molecular docking with QuickVina 2. *Bioinformatics*, 31(13):2214–2216, 2015.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/b22b257ad0519d4500539da3c8bcf4dd-Paper.pdf.

Camille Bilodeau, Wengong Jin, Tommi Jaakkola, Regina Barzilay, and Klavs F Jensen. Generative models for molecular discovery: Recent advances and challenges. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 12(5):e1608, 2022.

Hongming Chen, Ola Engkvist, Yinhai Wang, Marcus Olivecrona, and Thomas Blaschke. The rise of deep learning in drug discovery. *Drug discovery today*, 23(6):1241–1250, 2018.

Minshuo Chen, Xingguo Li, and Tuo Zhao. On generalization bounds of a family of recurrent neural networks. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1233–1243. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/chen20d.html.

Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi S. Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. In *International Conference on Learning Representations*, 2023.

Mark Davies, Michał Nowotka, George Papadatos, Nathan Dedman, Anna Gaulton, Francis Atkinson, Louisa Bellis, and John P. Overington. ChEMBL web services: streamlining access to drug discovery data and utilities. *Nucleic Acids Research*, 43(W1):W612–W620, 2015.

R. Fletcher. *Newton-Like Methods*, chapter 3, pp. 44–79. John Wiley & Sons, 2000.

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.

Octavian-Eugen Ganea, Xinyuan Huang, Charlotte Bunne, Yatao Bian, Regina Barzilay, Tommi S. Jaakkola, and Andreas Krause. Independent SE(3)-equivariant models for end-to-end rigid protein docking. In *International Conference on Learning Representations*, 2022.

Vikas K. Garg, Stefanie Jegelka, and Tommi Jaakkola. Generalization and representational limits of graph neural networks. In *International Conference on Machine Learning*, 2020.

Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

Jiaqi Guan, Wesley Wei Qian, Xingang Peng, Yufeng Su, Jian Peng, and Jianzhu Ma. 3d equivariant diffusion for target-aware molecule generation and affinity prediction. In *International Conference on Learning Representations*, 2023.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *International Conference on Neural Information Processing Systems*, 2017.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Brian Hie, Salvatore Candido, Zeming Lin, Ori Kabeli, Roshan Rao, Nikita Smetanin, Tom Sercu, and Alexander Rives. A high-level programming language for generative protein design. *bioRxiv*, 2022.

Emiel Hoogeboom, Víctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3D. In *International Conference on Machine Learning*, 2022.

Ilia Igashov, Hannes Stärk, Clément Vignac, Victor Garcia Satorras, Pascal Frossard, Max Welling, Michael M. Bronstein, and Bruno E. Correia. Equivariant 3d-conditional diffusion models for molecular linker design. *arXiv*, 2022.

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

John J. Irwin, Teague Sterling, Michael M. Mysinger, Erin S. Bolstad, and Ryan G. Coleman. Zinc: A free tool to discover chemistry for biology. *Journal of Chemical Information and Modeling*, 52 (7):1757–1768, 2012.

Stefanie Jegelka. Theory of graph neural networks: Representation and learning. *ArXiv*, abs/2204.07697, 2022.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International Conference on Machine Learning*, 2018.

Wengong Jin, Kevin Yang, Regina Barzilay, and T. Jaakkola. Learning multimodal graph-to-graph translation for molecular optimization. In *International Conference on Learning Representations*, 2019.

Thorsten Joachims. A support vector method for multivariate performance measures. In Luc De Raedt and Stefan Wrobel (eds.), *Proceedings of the 22nd International Conference on Machine Learning (ICML 2005), August 7-11, 2005, Bonn, Germany*, pp. 377–384. ACM Press, New York, NY, USA, 2005. ISBN 1-59593-180-5. URL http://doi.acm.org/10.1145/1102351.1102399.

Chaitanya K. Joshi, Cristian Bodnar, Simon V Mathis, Taco Cohen, and Pietro Lio. On the expressive power of geometric graph neural networks. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 15330–15355. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/joshi23a.html.

Haotian Ju, Dongyue Li, Aneesh Sharma, and Hongyang R. Zhang. Generalization in Graph Neural Networks: Improved PAC-Bayesian Bounds on Graph Diffusion, June 2023.

John M. Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Zídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Andy Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David A. Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 596:583–589, 2021.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Greg Landrum, Paolo Tosco, Brian Kelley, sriniker, gedeck, NadineSchneider, Riccardo Vianello, Ric, Andrew Dalke, Brian Cole, AlexanderSavelyev, Matt Swain, Samo Turk, Dan N, Alain Vaucher, Eisuke Kawashima, Maciej Wójcikowski, Daniel Probst, guillaume godin, David Cosgrove, Axel Pahl, JP, Francois Berenger, strets123, JLVarjo, Noel O'Boyle, Patrick Fuller, Jan Holst Jensen, Gianluca Sforna, and Doliath Gavid. Rdkit, 2020.

Yibo Li, Jianfeng Pei, and Luhua Lai. Structure-based de novo drug design using 3d deep generative models. *Chemical Sciences*, 12:13664–13675, 2021.

Shitong Luo, Jiaqi Guan, Jianzhu Ma, and Jian Peng. A 3d generative model for structure-based drug design. In *Advances in Neural Information Processing Systems*, 2021a.

Youzhi Luo, Keqiang Yan, and Shuiwang Ji. GraphDF: A discrete flow model for molecular graph generation. In *International Conference on Machine Learning*, 2021b.

Andrew McNutt, Paul Francoeur, Rishal Aggarwal, Tomohide Masuda, Rocco Meli, Matthew Ragoza, Jocelyn Sunseri, and David Koes. Gnina 1.0: molecular docking with deep learning. *Journal of Cheminformatics*, 13, 06 2021. doi: 10.1186/s13321-021-00522-2.

David Mendez, Anna Gaulton, A Patrícia Bento, Jon Chambers, Marleen De Veij, Eloy Félix, María Paula Magariños, Juan F Mosquera, Prudence Mutowo, Michał Nowotka, María Gordillo-Marañón, Fiona Hunter, Laura Junco, Grace Mugumbate, Milagros Rodriguez-Lopez, Francis Atkinson, Nicolas Bosc, Chris J Radoux, Aldo Segura-Cabrera, Anne Hersey, and Andrew R Leach. ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Research*, 47(D1): D930–D940, 2018.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012. ISBN 9780262018258. URL http://www.jstor.org/stable/j.ctt5hhcw1.

Mohit Pandey, Michael Fernandez, Francesco Gentile, Olexandr Isayev, Alexander Tropsha, Abraham C Stern, and Artem Cherkasov. The transformational role of GPU computing and deep learning in drug discovery. *Nature Machine Intelligence*, 4(3):211–221, 2022.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Xingang Peng, Shitong Luo, Jiaqi Guan, Qi Xie, Jian Peng, and Jianzhu Ma. Pocket2Mol: Efficient molecular sampling based on 3D protein pockets. In *International Conference on Machine Learning*, 2022.

Sudeep Pushpakom, Francesco Iorio, Patrick A Eyers, K Jane Escott, Shirley Hopper, Andrew Wells, Andrew Doig, Tim Guilliams, Joanna Latimer, Christine McNamee, et al. Drug repurposing: progress, challenges and recommendations. *Nature reviews Drug discovery*, 18(1):41–58, 2019.

Matthew Ragoza, Tomohide Masuda, and David Koes. Generating 3d molecules conditional on receptor binding sites with deep generative models. *Chemical Science*, 13, 2022.

Ryoma Sato. A survey on the expressive power of graph neural networks. *arXiv*, 2020.

Víctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. In *International Conference on Machine Learning*, 2021.

Arne Schneuing, Yuanqi Du, Charles Harris, Arian Jamasb, Ilia Igashov, Weitao Du, Tom Blundell, Pietro Lió, Carla Gomes, Max Welling, Michael Bronstein, and Bruno Correia. Structure-based drug design with equivariant diffusion models. *arXiv*, 2022.

Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. Graphaf: a flow-based autoregressive model for molecular graph generation. In *International Conference on Learning Representations*, 2020.

Chence Shi, Chuanrui Wang, Jiarui Lu, Bozitao Zhong, and Jian Tang. Protein sequence and structure co-design with equivariant translation. In *International Conference on Learning Representations*, 2023.

Hannes Stärk, Dominique Beaini, Gabriele Corso, Prudencio Tossou, Christian Dallago, Stephan Günnemann, and Pietro Lió. 3D infomax improves GNNs for molecular property prediction. In *International Conference on Machine Learning*, 2022a.

Hannes Stärk, Octavian Ganea, Lagnajit Pattanaik, Dr.Regina Barzilay, and Tommi Jaakkola. EquiBind: Geometric deep learning for drug binding structure prediction. In *International Conference on Machine Learning*, 2022b.

Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35, 2017.

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.

Jessica Vamathevan, Dominic Clark, Paul Czodrowski, Ian Dunham, Edgardo Ferran, George Lee, Bin Li, Anant Madabhushi, Parantu Shah, Michaela Spitzer, et al. Applications of machine learning in drug discovery and development. *Nature reviews Drug discovery*, 18(6):463–477, 2019.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Robert Verkuil, Ori Kabeli, Yilun Du, Basile I. M. Wicky, Lukas F. Milles, Justas Dauparas, David Baker, Sergey Ovchinnikov, Tom Sercu, and Alexander Rives. Language models generalize beyond natural proteins. *bioRxiv*, 2022.

Yogesh Verma, Samuel Kaski, Markus Heinonen, and Vikas Garg. Modular flows: Differential molecular generation. In *Advances in Neural Information Processing Systems*, 2022.

Yogesh Verma, Markus Heinonen, and Vikas Garg. Abode: Ab initio antibody design using conjoined odes. In *International Conference on Machine Learning*, 2023.

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, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Regina Barzilay, Tommi S. Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models. *bioRxiv*, 2022.

Kevin E. Wu, Kevin K. Yang, Rianne van den Berg, James Y. Zou, Alex X. Lu, and Ava P. Amini. Protein structure generation via folding diffusion. *arXiv*, 2022a.

Ruidong Wu, Fan Ding, Rui Wang, Rui Shen, Xiwen Zhang, Shitong Luo, Chenpeng Su, Zuofan Wu, Qi Xie, Bonnie Berger, Jianzhu Ma, and Jian Peng. High-resolution de novo structure prediction from primary sequence. *bioRxiv*, 2022b.

Chengxi Zang and Fei Wang. Moflow: An invertible flow model for generating molecular graphs. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.