

Structure-based Drug Design with Equivariant Diffusion Models

Arne Schneuing^{1*†}, Charles Harris^{2†}, Yuanqi Du^{3†}, Kieran Didi²,
 Arian Jamasb^{2,9}, Ilia Igashov¹, Weitao Du⁴, Carla Gomes³,
 Tom L. Blundell^{2,10}, Pietro Lio^{2,5}, Max Welling^{6,11}, Michael Bronstein^{7,8},
 Bruno Correia^{1*}

¹École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland.

²University of Cambridge, Cambridge, UK.

³Cornell University, Ithaca, USA.

⁴Chinese Academy of Mathematics and System Science, Beijing, China.

⁵University of Rome “La Sapienza”, Rome, Italy.

⁶Microsoft Research AI4Science, Amsterdam, Netherlands.

⁷University of Oxford, Oxford, UK.

⁸AITHYRA Institute, Vienna, Austria.

⁹Current affiliation: Prescient Design, Genentech, Basel, Switzerland.

¹⁰Current affiliation: Heart and Lung Research Institute, University of Cambridge, Cambridge, UK.

¹¹Current affiliation: University of Amsterdam, Amsterdam, Netherlands.

*Corresponding author(s). E-mail(s): arne.schneuing@epfl.ch; bruno.correia@epfl.ch;

†These authors contributed equally to this work.

Abstract

Structure-based drug design (SBDD) aims to design small-molecule ligands that bind with high affinity and specificity to pre-determined protein targets. Generative SBDD methods leverage structural data of drugs in complex with their protein targets to propose new drug candidates. These approaches typically place one atom at a time in an autoregressive fashion using the binding pocket as well as previously added ligand atoms as context in each step. Recently a surge of diffusion generative models has entered this domain which hold promise to capture the statistical properties of natural ligands more faithfully. However, most existing methods focus exclusively on bottom-up *de novo* design of compounds or tackle other drug development challenges with task-specific models. The latter requires curation of suitable datasets, careful engineering of the models and retraining from scratch for each task. Here we show how a single pre-trained diffusion model can be applied to a broader range of problems, such as off-the-shelf property optimization, explicit negative design, and partial molecular design with inpainting. We formulate SBDD as a 3D-conditional generation problem and present DiffSBDD, an SE(3)-equivariant diffusion model that generates novel ligands conditioned on protein pockets. Our *in silico* experiments demonstrate that DiffSBDD captures the statistics of the ground truth data effectively. Furthermore, we show how additional constraints can be used to improve the generated drug candidates according to a variety of computational metrics. These results support the assumption that diffusion models represent the complex distribution of structural data more accurately than previous methods, and are able to incorporate additional design objectives and constraints changing nothing but the sampling strategy. We anticipate that our findings may contribute to accelerate progress on several computational drug design frontiers as more powerful distribution learners emerge, that can be inserted into our flexible framework.

Keywords: structure-based drug design, conditioned diffusion models, equivariance

1 Introduction

The rational design of small-molecules with drug-like properties remains an outstanding challenge in both fundamental and biopharmaceutical research. Structure-based drug design (SBDD) aims to find small-molecule ligands that bind to specific three-dimensional sites in proteins with high affinity and specificity [1]. Traditionally, SBDD campaigns are usually initiated either by high-throughput experimental or virtual screening [2, 3] of large chemical databases. Generally, these approaches are expensive and time-consuming, but they also restrict the exploration of the chemical space to previously studied molecules, with a further emphasis usually placed on commercial availability [4]. Moreover, the optimization of initial lead molecules is often a biased process, with significant reliance on human intuition [5]. Recent advances in geometric deep learning, especially in modelling geometric structures of biomolecules [6–8], provide a promising direction for SBDD [9]. Despite remarkable progress in the use of deep learning as surrogate docking models [10–12], deep learning-based design of ligands that bind to target proteins remains an overarching problem in molecular modeling. Early attempts have been made to represent molecules as atomic density maps, with variational auto-encoders generating new atomic density maps corresponding to novel molecules [13]. However, it is nontrivial to map atomic density maps back to molecular space, requiring an additional atom-fitting stage. An alternative is to represent molecules as 3D graphs with atomic coordinates and types which naturally circumvents the post-processing steps. Li et al. [14] proposed an autoregressive generative model to sample ligands given the protein pocket as a conditioning constraint. Peng et al. [15] improved this method by using an $E(3)$ -equivariant graph neural network which respects rotation and translation symmetries in 3D space. Similarly, Drotár et al. [16] and Liu et al. [17] used autoregressive models to generate atoms sequentially and incorporate angles during the generation process. However, the main premise of sequential generation methods may not hold in real scenarios, since it imposes an artificial ordering scheme in the generation process and, as a result, the global context of the generated ligands may be lost. Very recently, a number of diffusion models have been put forward for target-specific molecule design [18–22]. These models place all atoms simultaneously, allowing them to reason about the whole molecule at once and typically enabling faster sampling. While this class of models has already shown great promise in *de novo* ligand generation, their potential in other parts of the drug design pipeline has not been thoroughly explored.

In this study, we propose DiffSBDD, an $SE(3)$ -equivariant 3D-conditional diffusion model for SBDD that respects translation, rotation, and permutation symmetries. With a free *de novo* generation benchmark, we first establish that the diffusion model captures molecular properties of real molecules more accurately than previously popular autoregressive models. Secondly, we show how the design space can be effectively constrained based on prior knowledge to improve the sample quality. Our inpainting-inspired approach furthermore allows us to tackle a diverse set of molecular design problems, such as scaffold hopping/elaboration and fragment growing/merging, all without the need to retrain new models. Lastly, we demonstrate how pre-trained DiffSBDD models can be used out-of-the-box to optimize arbitrary molecular properties via a noise/denoise scheme in combination with an evolutionary algorithm.

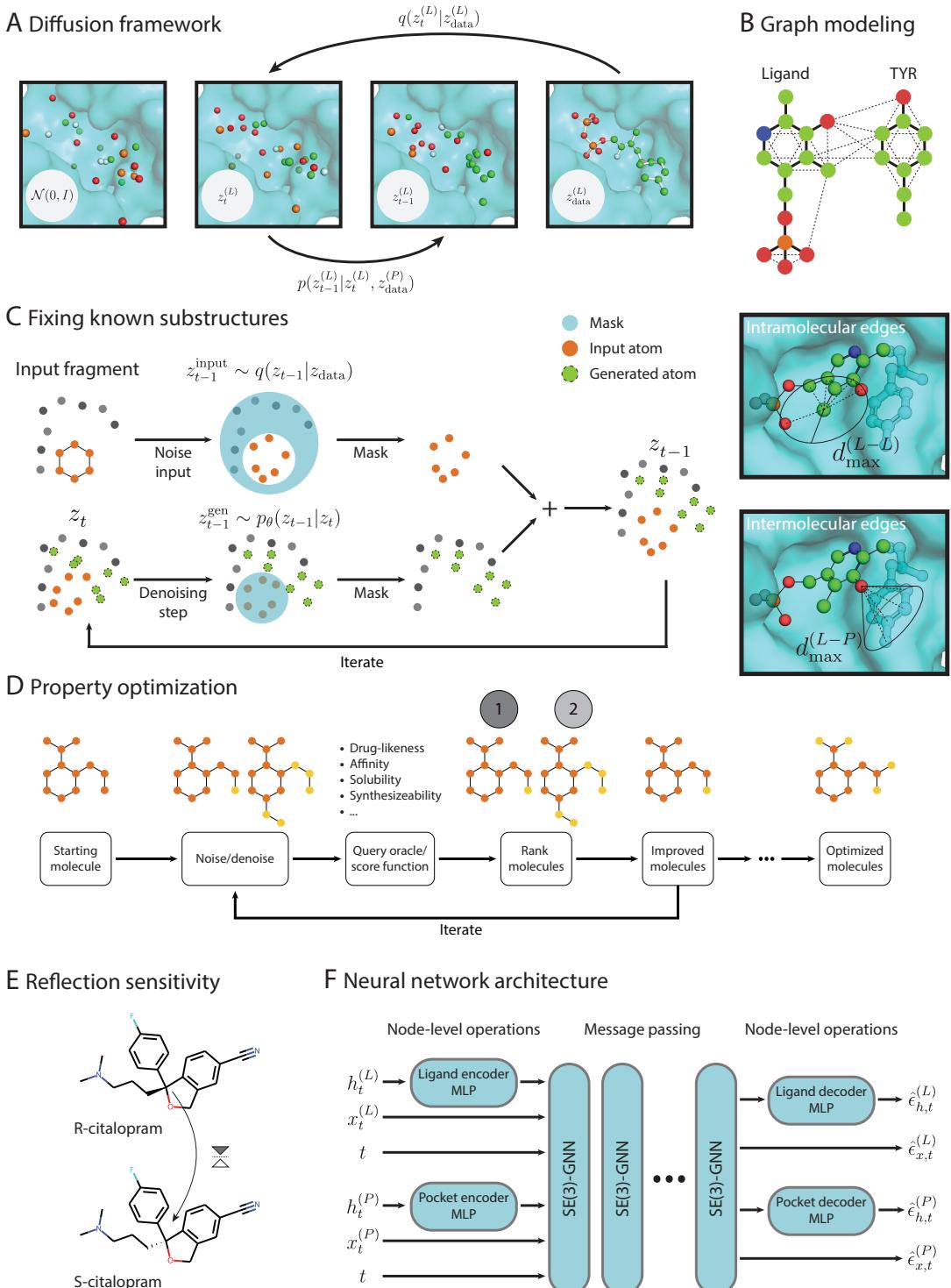


Fig. 1: (A) Overview of the 3D diffusion setup. The diffusion process q yields a noised version of the original atomic point cloud for a time step $t \leq T$. The neural network model is trained to approximate the reverse process conditioned on the target protein structure $\mathbf{z}^{(P)}$. Once trained, an initial noisy point cloud is sampled from a Gaussian distribution $\mathbf{z}_T^{(L)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and progressively denoised using the learned model. Covalent bonds are added to the resultant point cloud at the end of generation. (B) Each state is processed as a graph where edges are introduced according to distance thresholds within the ligand d_{\max}^{L-L} , within the pocket d_{\max}^{P-P} and between ligand and pocket nodes d_{\max}^{L-P} . (C) Replacement method for fixing molecular substructures. To complete the known part of the molecule (orange) with newly generated chemical matter (green atoms), we apply the learned denoising process to the entire molecule (orange & green), but at every step we replace the prediction for the known part (orange) with the ground-truth noised version computed with q . The protein context (gray) remains unchanged in every step. (D) Iterative procedure to tune molecular features. We find variations of a starting molecule by applying small amounts of noise and running an appropriate number of denoising steps. The new set of molecules is ranked by an arbitrary oracle and the procedure is repeated for the strongest candidates. (E) Antidepressant Citalopram as an example in which stereochemistry is essential for its therapeutic effect. (F) The neural network backbone is composed of MLPs that map scalar features of ligand and pockets nodes into a joint embedding space, and SE(3)-equivariant message passing layers that operate on these features and each node's coordinates. It outputs the predicted noise values for every vertex.

2 Equivariant Diffusion Models for SBDD

We leverage equivariant denoising diffusion probabilistic models (DDPMs) [23, 24] to generate molecules and binding conformations jointly with respect to a specific protein target. Figure 1A schematically depicts the 3D diffusion procedure. During training, varying amounts of random noise are applied to 3D structures of real ligands and a neural network learns to predict the noise-less features of the molecules. For sampling, these predictions are used to parameterize denoising transition probabilities which allow us to gradually move a sample from a standard normal distribution onto the data manifold. Both the protein and the ligand are represented as 3D point clouds, where atom types are encoded as one-hot vectors, and all objects are processed as graphs. For improved computational efficiency, we define independently tunable distance cutoffs for intermolecular edges between nodes of the ligand and pocket and intramolecular edges between two nodes from the same molecule (Figure 1B). This means information is only propagated between spatially proximal atoms. Our neural network is designed to respect natural symmetries of the molecular system, which include rotations and translations but excludes nonsuperposable transformations. That is, we process rigid transformations in an equivariant way but not reflections. This design choice is motivated by well-studied examples of drugs whose stereochemistry affects their activity and toxicity. For instance, the antidepressant Citalopram (Figure 1E) has two enantiomers but only the S-enantiomer has the desired therapeutic effect. The difference between the S- and R-form of the molecule, however, is only detectable by a reflection-sensitive GNN (Appendix E). To process ligand and pocket nodes with a single graph neural network (GNN), atom types and residue types are first embedded in a joint node embedding space by separate learnable MLPs. We also experimented with coarse-grained C_α descriptions of the pockets to reduce processing time even further but found this representation to be inferior in most cases (Appendix F.5). Further technical details of the diffusion framework and equivariant neural network are described in Method Sections 7.1 and 7.3.

To condition the 3D generative model on the structure of the protein pocket, we consider two distinct approaches. In the first approach, DiffSBDD-cond, we provide fixed three-dimensional context in each step of the denoising process. To this end, we supplement the ligand atomic point cloud $\mathbf{z}_t^{(L)}$, denoted by superscript L , with protein pocket nodes $\mathbf{z}_{\text{data}}^{(P)}$, denoted by superscript P , that remain unchanged throughout the reverse diffusion process (Figure 1A). For the second method, DiffSBDD-joint, we initially train a diffusion model to approximate the joint distribution $p(\mathbf{z}_{\text{data}}^{(L)}, \mathbf{z}_{\text{data}}^{(P)})$ of ligand-pocket pairs, and inject information about target pockets only at inference time. The methodology is analogous to the substructure inpainting approach described below (Section 7.4 and Figure 1C).

Table 1: Evaluation of generated molecules for targets from the CrossDocked and Binding MOAD test sets. To assess how well the models capture properties of real ligands, we compute the Wasserstein distance between the distributions of scores from generated molecules and the ground truth molecules from the test sets. The best performance (i.e. lowest Wasserstein distance) is highlighted in bold. * denotes that we re-evaluate the generated ligands provided by the authors. \dagger means inference times are taken from the original paper. \ddagger means inference time estimated based on five targets. *QED*: Quantitative Estimation of Drug-likeness [27]; *SA*: Synthetic Accessibility [28]; *LogP*: partition coefficient [29]; *CNN*: Convolutional Neural Network.

		Wasserstein distance to reference distribution						
		QED	SA	LogP	Lipinski	Vina score	CNN affinity	Time (s, \downarrow)
CrossDocked	Pocket2Mol [15]*	0.106	0.24	0.879	0.647	0.596	0.616	$2504 \pm 2207^\dagger$
	ResGen [30]	0.116	0.556	0.778	0.673	2.15	1.21	$\approx 936^\ddagger$
	DiffSBDD-cond	0.0205	1.29	0.589	0.261	0.848	0.145	135.866 ± 51.66
	DiffSBDD-joint	0.0187	1.51	1.67	0.405	0.694	0.281	160.314 ± 73.30
Binding MOAD	Pocket2Mol [15]	0.123	0.93	0.841	0.269	3.4	1.58	$\approx 613^\ddagger$
	ResGen [30]	0.0727	0.92	0.831	0.237	6.93	1.71	$\approx 697^\ddagger$
	DiffSBDD-cond	0.0924	1.15	0.833	0.143	4.26	0.903	336.061 ± 85.02
	DiffSBDD-joint	0.0716	0.76	1.23	0.183	9.56	0.628	369.873 ± 124.54

Both approaches are equally applicable to the small molecule design task and in practice differ only in whether the neural networks expects the original pocket or a noisy version as input.

To evaluate our approach we first show that diffusion models are a powerful framework for learning the distribution of three-dimensional molecular data by generating new target-specific ligands *de novo* without additional constraints or optimizing a particular property (Section 3). We then demonstrate how the flexibility of diffusion models enables partial molecular redesign to incorporate specific design constraints without needing to develop specialised models (Sections 4), and iterative improvement of molecular properties measured by arbitrary oracles (Section 5). While we provide empirical results only for our model, the methodology can be readily used in combination with other recently published diffusion models for molecule design [18–22]. Finally, we have curated an experimentally determined binding dataset derived from Binding MOAD [25], which supplements the commonly used synthetic dataset CrossDocked [26], to validate our model’s performance under realistic binding scenarios. While protein-ligand pairs of the former might contain non-native contacts, since ligands were cross-docked into structurally similar binding pockets, our new dataset consists exclusively of experimentally validated interactions.

3 DiffSBDD captures the data distribution faithfully

As a first test to our model, we probe its ability to accurately represent the properties of real binders, and compare the results with Pocket2Mol [15] and ResGen [30], two recently published autoregressive models, which represent the previous state-of-the-art class of machine learning models for structure-based drug design. We use publicly available code and weights of the models [31, 32].

Since distribution learning capabilities in the high-dimensional space of chemical compounds are difficult to quantify directly, we instead measure a range of molecular properties that are relevant for potential drug candidates. We then compare the distributions of these scores to the distributions we get from the real ligands in our test set using the Wasserstein distance. These results are summarized in Table 1 for computational scores of drug-likeness (QED), synthetic accessibility (SA), hydrophobicity (LogP) and two measures of target affinity, the empirical Vina scoring function and a neural network estimation of binding affinity (CNN affinity). Both were computed by GNINA [33] after local energy minimization to resolve minor clashes. The underlying distributions of scores are shown in supplementary Figures F2 and F3. We perform this analysis both for the test set targets from CrossDocked [26], a standard dataset extensively used in prior works [13, 15, 17, 30], and our newly

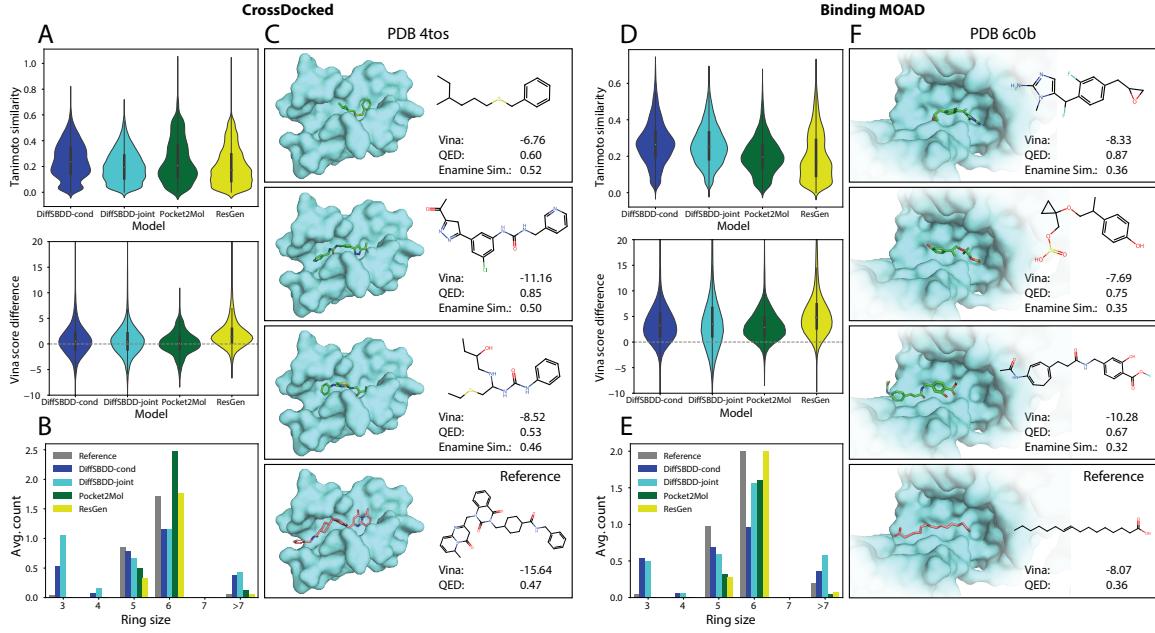


Fig. 2: Evaluation of distribution learning capabilities and generated examples. All targets are taken from the CrossDocked and Binding MOAD test sets. (A) Comparison of generated molecules with the reference molecule from the same pocket. We compare Tanimoto similarity of the molecular fingerprints and compute the difference $Vina_{gen} - Vina_{ref}$ between their Vina docking scores. (B) Average number of rings of different sizes per generated molecule. (C) Example molecules generated by DiffSBDD-cond for a pocket from the CrossDocked test set. We compared all generated molecules with the approximately 4.2M compounds from the Enamine Screening Collection, and selected the three closest hits with drug-likeness $QED > 0.5$. Vina docking score, QED drug-likeness score and fingerprint similarity to the most similar Enamine molecules are reported in each case. (D-F) Same analyses but for target pockets from the Binding MOAD test set.

curated dataset based on Binding MOAD [25]. Note that Pocket2Mol and ResGen were trained on the same CrossDocked training set but not on Binding MOAD.

Generally, our diffusion models capture molecular properties of natural ligands more accurately than the autoregressive baselines despite significantly shorter sampling times. A notable exception is the Vina score, which Pocket2Mol matches particularly well on the CrossDocked dataset. Interestingly, this observation is not confirmed by GNINA’s CNN affinity which estimates the same quantity. Its distribution is better approximated by DiffSBDD. Figure 2A shows that both DiffSBDD and Pocket2Mol Vina scores are centered around the reference but the spread is larger in the case of the diffusion models, which means their samples contain larger fractions of low scoring molecules but also ligands that potentially bind more tightly than the native counterparts. The greater abundance of high scoring molecules is particularly important in anticipation of downstream design applications, where we often look for the most competitive binder rather than average candidates. A similar observation holds for the Binding MOAD dataset with experimentally determined binding complexes. However, unlike the CrossDocked case, docking scores are lower on average than the scores of corresponding reference ligands from this dataset. We believe the reason to be twofold: the Binding MOAD training set is much smaller and also contains more challenging ground-truth ligands (native binders) whereas CrossDocked complexes can have unrealistic protein-ligand interactions. This hypothesis is supported by less favorable Vina scores of reference molecules from the synthetic dataset on average (-7.68 vs. -9.17 kcal/mol). This result underscores the importance of high quality training sets for SBDD models that aim to design high affinity binders. Lastly, the DiffSBDD models also produce molecules that are slightly more similar to the reference on average (Figure 2A,D) and contain a comparable amount of 5- and 6-rings to natural ligands (Figure 2B,E). However, very small and very large ring systems are typically over-represented in DiffSBDD molecules.

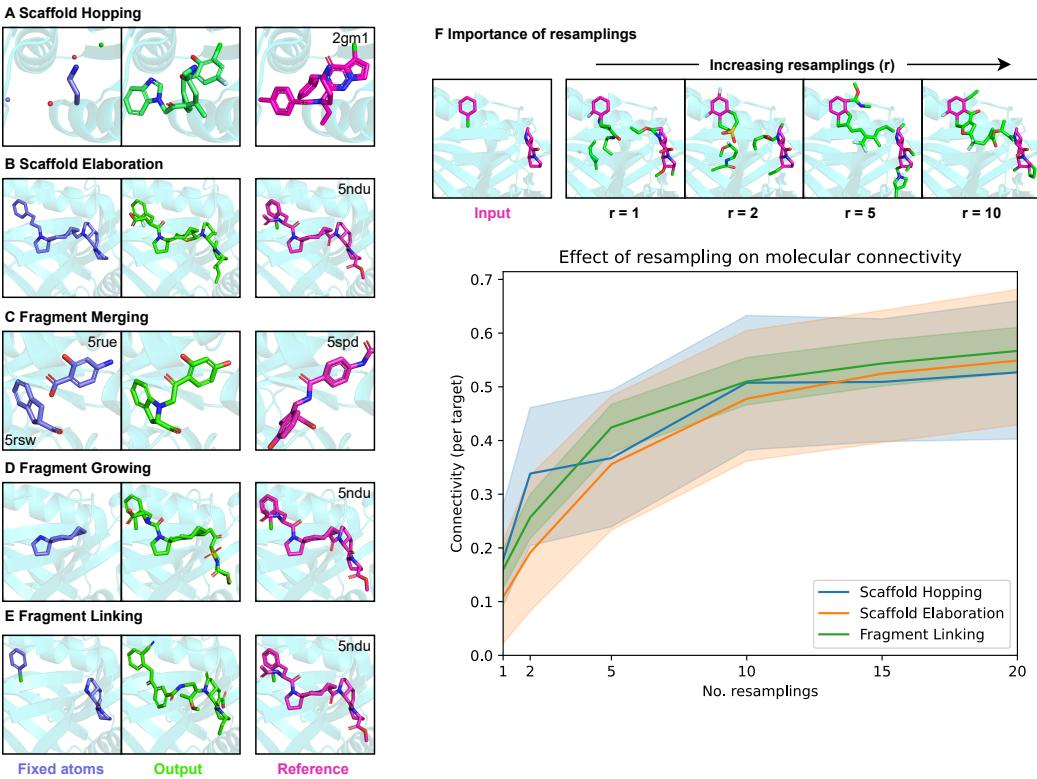


Fig. 3: Molecular inpainting design examples for scaffold hopping (A), scaffold elaboration (B), fragment merging (C), fragment growing (D) and fragment linking (E) respectively. The input to our model (the fixed atoms) is shown in blue, the outputs (designed molecules) are shown in green and the original molecule is shown in magenta for reference. PDB codes are shown for the ground truth structure. In the case of fragment merging, we compose fragments with two different crystal structures with PDB codes shown. (F) Importance of resampling for generating realistic and connected molecules. Top: Visual example; inpainted region (green) finally harmonizes with molecular context at high resamplings. Bottom: Effect of the number of resampling steps on molecular connectivity. Means and 95% confidence intervals are plotted for 3 design tasks. For this experiment we used 20 randomly selected targets from the test set.

Panels C and F of Figure 2 present a representative selection of molecules for one target from each test set. The selection is filtered to contain examples which are drug-like ($\text{QED} > 0.5$) and similar to purchasable molecules from the Enamine Screening Collection. These filters represent favorable properties one might look for in a drug design campaign. The target with Protein Data Bank (PDB) identifier 6c0b, for example, is a human receptor which is involved in microbial infection [34] and possibly tumor suppression [35]. The reference molecule, a long fatty acid (see Figure 2F, bottom panel) that aids receptor binding [34], has too high a number of rotatable bonds and low a number of hydrogen bond donors/acceptors to be considered a suitable drug (QED of 0.36). Our model however, generates drug-like ($\text{QED} = 0.87$ in the first example) and suitably sized molecules by adding aromatic rings connected by a few rotatable bonds, which allows the molecules to adopt a complementary binding geometry and is entropically favourable by reducing the degrees of freedom, a classic approach in medicinal chemistry [36]. Larger random samples of generated molecules are presented in the Appendix (Figure F6).

4 Generating novel chemical matter from known substructures

In drug discovery it is very common to design molecules around previously identified potent substructures. For example, we may wish to design a scaffold around a set of functional groups (scaffold

Table 2: Evaluation of molecular inpainting for fragment linking, scaffold hopping and scaffold elaboration across the whole Binding MOAD test set. Percentage value next to task name denotes the proportion of atoms fixed during the design process. DiffSBDD-*baseline* generates a whole new molecule from scratch, DiffSBDD-*diversify* is elaborating around a fixed substructure of an existing molecule and DiffSBDD-*de novo* is designing new motifs around a fixed substructure from scratch. t and r are the number of partial noising steps or resamplings for *diversify* and *de novo* respectively. Note, here we use a subset of our original test set for which we can calculate masks for all design tasks ($n = 55$).

	Vina (↓)	Validity (↑)	Connectivity (↑)	QED (↑)	SA (↓)	Diversity (↑)
Test set	-9.86 ± 1.6	1	1	0.543 ± 0.16	3.86 ± 1.2	—
DiffSBDD- <i>baseline</i>	-5.69 ± 6	0.964	0.589	0.419 ± 0.2	4.99 ± 1.1	0.701 ± 0.09
Fragment linking (73.43% atoms fixed)						
DiffSBDD- <i>de novo</i> ($r = 20$)	-7.74 ± 1.8	0.796	0.558	0.322 ± 0.17	5.38 ± 0.76	0.486 ± 0.09
DiffSBDD- <i>diversify</i> ($t = 100$)	-8.72 ± 1.7	0.991	0.968	0.476 ± 0.14	4.26 ± 1.1	0.35 ± 0.089
DiffLinker [41]	-6.92 ± 2.7	0.947	0.840	0.349 ± 0.19	4.72 ± 0.97	0.453 ± 0.13
Scaffold hopping (27.32% atoms fixed)						
DiffSBDD- <i>de novo</i> ($r = 20$)	-7.6 ± 2.5	0.782	0.663	0.39 ± 0.18	5.29 ± 0.72	0.612 ± 0.074
DiffSBDD- <i>diversify</i> ($t = 100$)	-8.95 ± 1.8	0.977	0.948	0.492 ± 0.18	4.39 ± 1.0	0.479 ± 0.1
Scaffold elaboration (72.68% atoms fixed)						
DiffSBDD- <i>de novo</i> ($r = 20$)	-8.1 ± 1.8	0.852	0.445	0.388 ± 0.2	5.2 ± 0.7	0.397 ± 0.11
DiffSBDD- <i>diversify</i> ($t = 100$)	-9.32 ± 1.7	0.995	0.971	0.516 ± 0.18	4.12 ± 1.1	0.282 ± 0.14

hopping) or extend an existing fragment to make a whole molecule (fragment growing). Generating compounds, or parts thereof, conditioned on a given molecular context is reminiscent of inpainting, a technique originally introduced for completing missing parts of images [37, 38] but also adopted in other domains, including biomolecular structures [39]. We can realise a number of drug discovery sub-tasks via an inpainting technique known as the ‘replacement method’ [38, 40], whereby we add new atoms in and around fixed regions of the substructure to design whole molecules (Fig. 1C and Methods Section 7.4). Unlike previous methods, using DiffSBDD in this way does not require retraining a model on any specialized or synthetic datasets. Curating such datasets is often time and labor-intensive, and typically relies on potentially sub-optimal assumptions (e.g. definition of fragments) to convert a general dataset of small molecules into a task-specific dataset that can be used to train specialised models. With our proposed approach, by contrast, the simple definition of an arbitrary binary mask is sufficient for the diffusion model to generalize to any inpainting task whilst using a neural network trained on all available protein-ligand data in raw form.

Examples of molecules designed with inpainting for scaffold hopping, scaffold elaboration, fragment merging, fragment growing, and fragment linking are shown in Figure 3. All molecular inpainting experiments use a version of DiffSBDD trained on Binding MOAD. Figure 3A shows an example of scaffold hopping a design for a mitotic kinesin Eg5 inhibitor (PDB code 2gm1) [42] where we fix the functional groups mediating the binding to the pocket whilst designing a new scaffold structure. Figure 3B shows the opposite case of scaffold elaboration for a rationally designed oncology inhibitor targeting a phosphoprotein (PDB code 5ndu) [43] where we fix the scaffold and design new functional groups. Figure 3C shows an example of fragment merging. We successfully replicate the results of Gahbauer et al. [44], which performed fragment merging of two fragments (PDB code 5rsw and 5rue) identified by experimental screening [45] for the SARS-CoV-2 non-structural protein 3 (Nsp3) using the chemoinformatics-based method Fragmentenstein¹. Figure 3D shows an example of fragment growing around the central motif of PDB entry 5ndu. Finally, Figure 3E gives an example of fragment linking between two outer fragments of PDB entry 5ndu. Note that here we are not only designing a small linker made of a few atoms but rather an entirely new fragment with two connecting linkers. Such a linker design would be substantially out-of-distribution for previous methods, due to the size and complexity of the linker relative to those used during training [46]. Further implementation details of the fragment merging experiment are given in Appendix Section G.1.

Depending on the use case, we find it desirable to perform molecular inpainting within two regimes; (i) designing a completely new inpainted region *de novo* (DiffSBDD-*de novo*) as to explore the entire chemical fitness landscape, or (ii) redesigning an existing region (e.g. a scaffold) via partial noising

¹www.github.com/matteoferla/Fragmentenstein

then denoising (see Section H), thus locally exploring desired properties by exploitation (DiffSBDD-*diversify*). The first case is more amenable to situations in which we have *no prior information* other than the fixed substructure (e.g. fragment linking after a fragment screen), meaning that unconstrained exploration of the chemical fitness landscape is the preferred approach for the majority of SBDD. The second case is more relevant in scenarios where we *have prior information* about the desired chemical and topological composition of the designed region which we can use to bias generation (with the choice of t being a hyperparameter). This is particularly relevant in the case of scaffold hopping, where we are trying to keep the properties of a molecule relatively unchanged whilst designing a new topology [47]. To investigate the differences between both approaches quantitatively, we further tested our method systematically for the whole Binding MOAD test set in the tasks of linker design, scaffold hopping and scaffold elaboration (Table 2). As baselines, we provide the metrics for the molecules in the test set, molecules only conditioned on the pocket with no fixed substructure (DiffSBDD-*baseline*) and the performance of DiffLinker [41] in the fragment linking task. Due to the fact that we can only calculate fragment and scaffold masks for larger molecules, we have reduced the size of the test set used in Table 2 to $n = 55$ to ensure fair comparison between methods.

Constraining fixed regions to highly complementary substructures within the protein pocket significantly enhances Vina scores using DiffSBDD-*de novo* compared to the DiffSBDD-*baseline*. For molecules generated without substructure conditioning under DiffSBDD-*baseline*, the average Vina score is -5.69. In contrast, DiffSBDD-*de novo* sees improved scores, achieving -7.74 kcal/mol when used for linker design from starting fragments and achieves results comparable to the specialist model DiffLinker. In the case of scaffold elaboration, DiffSBDD-*de novo* significantly boosts docking scores from -5.69 kcal/mol to -8.10 kcal/mol over the baseline, by focusing on the optimal placement of functional groups to facilitate key residue binding on a pre-existing scaffold. Moreover, there is a notable improvement in average docking scores for scaffold hopping, with scores rising from -5.69 kcal/mol to -7.60 kcal/mol when using DiffSBDD-*de novo*. This enhancement is achieved despite a relatively small proportion of atoms being fixed, about 27.32%, compared to other tasks. The significant increase in docking scores is attributed to the nature of the fixed atoms, primarily functional groups that form the pharmacophore, which are crucial for binding affinity. For detailed implementation, see Appendix G.2.

Similar to earlier findings, the replacement method often produces poor and inconsistent outcomes. Following the approach by Lugmayr et al. [38], we enhanced the sample quality by refining intermediate states iteratively before progressing in the denoising process, a technique called resampling, as detailed in the Methods Section 7.4. This technique is crucial for seamlessly integrating modified and original areas and proves essential in the molecular context. Minimal resampling resulted in chemically valid but disjointed structures, while increased iterations led to coherent molecules, even in complex scenarios with extensive modifications needed (Fig. 3F). Our results indicate that the effect of resampling on molecular connectivity is particularly pronounced (Fig. 3F) but it also significantly impacts another metrics (Appendix Figure G7B-D).

5 Molecule optimization: iterative search for better molecule candidates

For hit identification and optimization of lead molecules in real use cases, it is not enough to just sample molecules from the whole training data distribution. Instead, we are usually interested in the better performing tail of the distribution, and only want to pursue the most promising candidates. Since we could show that DiffSBDD recapitulates the chemical space of the training set including high-scoring molecules, we should always find promising drug candidates with strong docking scores, synthetic accessibility and other desired properties. Here we propose a simple protocol to access them efficiently (Figure 1D).

We first noise a molecule from an experimental protein-ligand complex for t steps, where $t \ll T$, using the forward diffusion process. From this partially noised sample, we can then denoise the appropriate number of steps with the reverse process until $t = 0$. The stochasticity in this quick noise/denoise

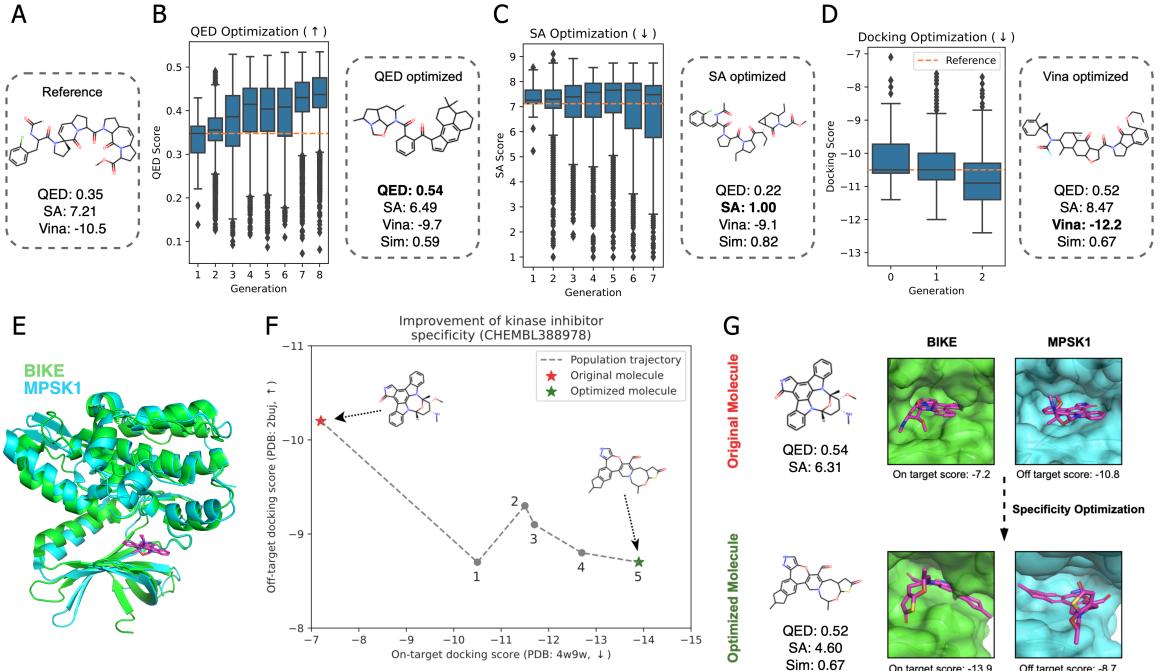


Fig. 4: Results on molecular optimization using DiffSBDD. (A-D) Experiments on single property molecular optimization. (A) Starting molecule from PDB code 5NDU. (B) QED optimization over 8 generations. (C) SA optimization over 7 generations. (D) docking score optimization over 3 generations. We found that optimization over subsequent generations continuously optimized the docking score, but that was at expense of molecular quality. (E-G) Kinase inhibitor specificity optimization experiment. (E) Cartoon representation showing the high degree of structural similarity between our two kinases of interest (BIKE and MPSK1). (F) Trajectory plot showing the highest scoring molecule at each iteration during kinase inhibitor optimization. (G) Visual representation of the molecular graphs and bound conformations of the native and final molecules with corresponding Vina docking scores. Boxes in panels (B-D) represent the upper and lower quartile as well as the median of the data. Whiskers denote 1.5 times the interquartile range. Outliers outside this range are shown as flier points.

process allows us to sample new and diverse candidates of various properties whilst staying in the same region of chemical space, assuming t is small (see Appendix Figure H8). This approach is inspired by [48] but note this does not allow for direct optimization of specific properties. Instead, it can be regarded as an exploration around the local chemical space whilst maintaining high shape and chemical complementary via the conditional denoising model.

We extend this idea by combining the partial noising/denoising procedure with a simple evolutionary algorithm that optimizes for specific molecular properties (Figure 1D). We find that our model performs well at this task out of the box without requiring additional fine-tuning. At every stage in the optimization process, we generate 100 new molecules (from either the previous generation or the original molecule in the first case). Molecules are modified via partial noising/denoising with a randomly chosen t between 10 and 150. The new molecules are then passed to an oracle/score function (e.g. docking program or synthetic accessibility predictor) to be ranked. The top k molecules are then selected to seed the new population. In our study, we use $k = 10$.

In Figure 4A-D, we optimize an inhibitor molecule found in PDB entry 5ndu [43], which has poor SA and QED scores, 7.21 and 0.35 respectively, but high binding affinity. Over a number of rounds of optimization, we can observe substantial increases in QED, from 0.35 to mean of 0.43, whilst still maintaining high similarity to the original molecule. We can also rescue the low synthetic accessibility score of the seed molecule by producing a battery of highly accessible molecules when selecting for SA. Finally, we observe that we can perform significant optimization of binding affinity after only a few rounds of optimization.

To demonstrate the power of molecular optimization with DiffSBDD, we consider the challenging case of *highly selective kinase inhibitor design* (Figure 4E-G). In our experiment, we perform positive design against our on-target kinase BIKE (PDB code 4w9w) [49] whilst simultaneously performing negative design against the structurally similar off-target kinase MPSK1 (PDB code 2buj) [50] (Fig. 4E) by optimizing for L . Furthermore, before selecting the top molecules, we pruned any candidates that regress with regard to the on- and off-target docking scores of the original molecule (i.e. those above or left of the red star in Fig. 4F) in order to bias the molecules to have high affinity to the on-target kinase as well as specificity. The starting molecule (ChEMBL identifier CHEMBL388978) had an on- and off-target docking score of -7.2 kcal/mol and -10.8 kcal/mol respectively. After 5 rounds of optimization, we had improved the on- and off-target docking scores for the top molecule to -13.9 kcal/mol and -8.7 kcal/mol respectively (Fig 4G), demonstrating substantially improved specificity and reduced off-target activity. Furthermore, all of this is achieved whilst maintaining a high QED score (from 0.54 to 0.52) and improving the SA score from 6.31 to 4.60.

6 Conclusion

Many machine learning methods for structure-based drug design create molecules one atom at a time, thereby imposing an arbitrary order on the generative process and preventing them from reasoning about molecules holistically. Moreover, most existing approaches focus exclusively on the *de novo* generation of new ligands from scratch, which often limits their sample quality and synthesizability, and ultimately hinders lab validation of designs. In this work, we investigated the distribution learning capabilities of 3D-conditional diffusion models as an alternative to the autoregressive paradigm. To this end, we developed DiffSBDD, an $SE(3)$ -equivariant 3D-conditional diffusion model that generates small molecule ligands for given target pocket structures, with chemical properties that closely match those of native ligands. Since, on the application side, medicinal chemists typically have concrete design specifications in mind, we studied different substructure redesign and optimization techniques, and showed how these can be used to improve a range of desirable properties of candidate compounds, such as computational docking and drug-likeness scores. Constraining the problem to realistic substructures like fragments or scaffolds leads to better designs because it prevents the neural network from overly hallucinating. Retaining substructures of previously synthesized molecules facilitates chemical synthesis and experimental testing. Moreover, the capability to further ‘locally’ optimize designed ligands is important in real-world drug discovery and effectively improves the quality of the initial designs. For similar applications, previous works typically resorted to specialised models that were trained on tailored datasets and performed well only on narrowly defined tasks. Here, we provided evidence that a powerful general diffusion model can be used as a drop-in replacement for these specialised models if the sampling procedure is modified appropriately. This means we can expect better performance in all discussed sub-tasks, solely by improving the distribution learning capabilities and sample quality of the main model.

In general, data-driven techniques are emerging as a suitable tool to tackle the immense complexity of the biochemical design space, which is extremely hard to navigate with mechanistic approaches. While the purely *de novo* design of novel chemical matter remains challenging, we could show that learning-based tools are ready to be incorporated in drug development pipelines if additional design constraints are enforced. In the future, we foresee an increasing importance of machine learning models in this domain as they are improved further. Developing more informative metrics and more reliable benchmarks will be an important step towards reaching this goal because they reduce the reliance on visual inspection of examples and expert judgment and thus accelerate progress. Lastly, we envision that other iterative sampling techniques, including flow matching and Schrödinger bridge models, will benefit from the presented inference strategies as well.

7 Methods

7.1 Denoising Diffusion Probabilistic Models

Denoising diffusion probabilistic models (DDPMs) [23, 51] are a class of generative models inspired by non-equilibrium thermodynamics. Briefly, they define a Markovian chain of random diffusion steps by slowly adding noise to sample data and then learning the reverse of this process (typically via a neural network) to reconstruct data samples from noise.

In this work, we closely follow the framework developed by Hoogeboom et al. [24]. In our setting, data samples are atomic point clouds $\mathbf{z}_{\text{data}} = [\mathbf{x}, \mathbf{h}]$ with 3D geometric coordinates $\mathbf{x} \in \mathbb{R}^{N \times 3}$ and categorical features $\mathbf{h} \in \mathbb{R}^{N \times d}$, where N is the number of atoms. A fixed noise process

$$q(\mathbf{z}_t | \mathbf{z}_{\text{data}}) = \mathcal{N}(\mathbf{z}_t | \alpha_t \mathbf{z}_{\text{data}}, \sigma_t^2 \mathbf{I}) \quad (1)$$

adds noise to the data \mathbf{z}_{data} and produces a latent noised representation \mathbf{z}_t for $t = 0, \dots, T$. α_t controls the signal-to-noise ratio $\text{SNR}(t) = \alpha_t^2 / \sigma_t^2$ and follows either a learned or pre-defined schedule from $\alpha_0 \approx 1$ to $\alpha_T \approx 0$ [52]. We choose a variance-preserving noising process [37] with $\alpha_t = \sqrt{1 - \sigma_t^2}$.

Since the noising process is Markovian, we can write the denoising transition from time step t to $s < t$ in closed form as

$$q(\mathbf{z}_s | \mathbf{z}_{\text{data}}, \mathbf{z}_t) = \mathcal{N}\left(\mathbf{z}_s \mid \frac{\alpha_{t|s} \sigma_s^2}{\sigma_t^2} \mathbf{z}_t + \frac{\alpha_s \sigma_{t|s}^2}{\sigma_t^2} \mathbf{z}_{\text{data}}, \frac{\sigma_{t|s}^2 \sigma_s^2}{\sigma_t^2} \mathbf{I}\right) \quad (2)$$

with $\alpha_{t|s} = \frac{\alpha_t}{\alpha_s}$ and $\sigma_{t|s}^2 = \sigma_t^2 - \alpha_{t|s}^2 \sigma_s^2$ following the notation of Hoogeboom et al. [24]. This true denoising process depends on the data sample \mathbf{z}_{data} , which is not available when using the model for generating new samples. Instead, a neural network ϕ_θ is used to approximate the sample $\hat{\mathbf{z}}_{\text{data}}$. More specifically, we can reparameterize Equation (1) as $\mathbf{z}_t = \alpha_t \mathbf{z}_{\text{data}} + \sigma_t \epsilon$ with $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and directly predict the Gaussian noise $\hat{\epsilon}_\theta = \phi_\theta(\mathbf{z}_t, t)$. Thus, $\hat{\mathbf{z}}_{\text{data}}$ is simply given as $\hat{\mathbf{z}}_{\text{data}} = \frac{1}{\alpha_t} \mathbf{z}_t - \frac{\sigma_t}{\alpha_t} \hat{\epsilon}_\theta$.

The neural network is trained to maximize the likelihood of observed data by optimizing a variational lower bound on the data, which is equivalent to the simplified training objective [23, 52] $\mathcal{L}_{\text{train}} = \frac{1}{2} \|\epsilon - \phi_\theta(\mathbf{z}_t, t)\|^2$ up to a scale factor (see Appendix A for details).

7.2 Equivariance

Structural biology remains a rather data-sparse domain. It is therefore common practice to encode known geometric constraints, typically equivariance to rotations and translations, directly into the neural network architecture, thereby facilitating the learning task because possible neural operations are limited to a meaningful subset. In the 3D molecule generation setting, we explicitly exclude reflection-equivariant operations because they would make the model blind to some aspects of stereochemistry. It is known that different stereoisomers can have significantly different therapeutic effects (e.g., [53], Figure 1E) and might even lead to unforeseen off-target activity and hence toxicity. We therefore developed a reflection-sensitive system that is $SE(3)$ - rather than $E(3)$ -equivariant although the latter is more commonly adopted in related works [18, 24, 54].

Technically, we ensure $SE(3)$ -equivariance in the following sense²: evaluating the likelihood of a molecule $\mathbf{x}^{(L)} \in \mathbb{R}^{3 \times N_L}$ given the three-dimensional representation of a protein pocket $\mathbf{x}^{(P)} \in \mathbb{R}^{3 \times N_P}$ should not depend on global $SE(3)$ -transformations of the system, i.e. $p(\mathbf{Rx}^{(L)} + \mathbf{t} | \mathbf{Rx}^{(P)} + \mathbf{t}) = p(\mathbf{x}^{(L)} | \mathbf{x}^{(P)})$ for orthogonal $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ with $\mathbf{R}^T \mathbf{R} = \mathbf{I}$, $\det(\mathbf{R}) = 1$ and $\mathbf{t} \in \mathbb{R}^3$ added column-wise. At the same time, it should be possible to generate samples $\mathbf{x}^{(L)} \sim p(\mathbf{x}^{(L)} | \mathbf{x}^{(P)})$ from this conditional probability distribution so that equivalently transformed ligands $\mathbf{Rx}^{(L)} + \mathbf{t}$ are sampled with the same

²We ignore node type features, which transform invariantly, for simpler notation.

probability if the input pocket is rotated and translated and we sample from $p(\mathbf{R}\mathbf{x}^{(L)} + \mathbf{t} | \mathbf{R}\mathbf{x}^{(P)} + \mathbf{t})$. This definition explicitly excludes reflections which are connected with chirality and can alter the biomolecule's properties.

In our set-up, equivariance to the orthogonal group $O(3)$ (comprising rotations and reflections) is achieved because we model both prior and transition probabilities with isotropic Gaussians where the mean vector transforms equivariantly with respect to rotations of the context (see Hoogeboom et al. [24] and Appendix D). Ensuring translation equivariance, however, is harder because the transition probabilities $p(\mathbf{z}_{t-1} | \mathbf{z}_t)$ are not inherently translation-equivariant. In order to circumvent this issue, we follow previous works [24, 55, 56] by limiting the whole sampling process to a linear subspace where the center of mass (CoM) of the system is zero. In practice, this is achieved by subtracting the center of mass of the system before performing likelihood computations or denoising steps. Since equivariance of the transition probabilities depends on the parameterization of the noise predictor $\hat{\epsilon}_\theta$, we can make the model sensitive to reflections with a simple additive cross-product term in the EGNN's coordinate update as discussed in Section 7.3 and Appendix E.

7.3 $SE(3)$ -equivariant Graph Neural Networks

A function $f : \mathcal{X} \rightarrow \mathcal{Y}$ is said to be *equivariant* w.r.t. the group G if $f(g.\mathbf{x}) = g.f(\mathbf{x})$, where g denotes the action of the group element $g \in G$ on \mathcal{X} and \mathcal{Y} [57]. Graph Neural Networks (GNNs) are learnable functions that process graph-structured data in a permutation-equivariant way, making them particularly useful for molecular systems where nodes do not have an intrinsic order. Permutation invariance means that $\text{GNN}(\mathbf{\Pi X}) = \mathbf{\Pi GNN}(\mathbf{X})$ where $\mathbf{\Pi}$ is an $n \times n$ permutation matrix acting on the node feature matrix.

Since the nodes of the molecular graph represent the 3D coordinates of atoms, we are interested in additional equivariance w.r.t. the Euclidean group $E(3)$ or rigid transformations. An $E(3)$ -equivariant GNN (EGNN) satisfies $\text{EGNN}(\mathbf{\Pi X A} + \mathbf{b}) = \mathbf{\Pi EGNN}(\mathbf{X})\mathbf{A} + \mathbf{b}$ for an orthogonal 3×3 matrix \mathbf{A} with $\mathbf{A}^\top \mathbf{A} = \mathbf{I}$ and some translation vector \mathbf{b} added row-wise.

In our case, since the nodes have both geometric atomic coordinates \mathbf{x} as well as atomic type features \mathbf{h} , we can use a simple implementation of EGNN proposed by Satorras et al. [54], in which the updates for features \mathbf{h} and coordinates \mathbf{x} of node i at layer l are computed as follows:

$$\mathbf{m}_{ij} = \phi_e(\mathbf{h}_i^l, \mathbf{h}_j^l, d_{ij}^2, a_{ij}), \tilde{e}_{ij} = \phi_{\text{att}}(\mathbf{m}_{ij}) \quad (3)$$

$$\mathbf{h}_i^{l+1} = \phi_h(\mathbf{h}_i^l, \sum_{j \neq i} \tilde{e}_{ij} \mathbf{m}_{ij}) \quad (4)$$

$$\mathbf{x}_i^{l+1} = \mathbf{x}_i^l + \sum_{j \neq i} \frac{\mathbf{x}_i^l - \mathbf{x}_j^l}{d_{ij} + 1} \phi_x(\mathbf{h}_i^l, \mathbf{h}_j^l, d_{ij}^2, a_{ij}) \quad (5)$$

where ϕ_e , ϕ_{att} , ϕ_h and ϕ_x are learnable Multi-layer Perceptrons (MLPs) and d_{ij} and a_{ij} are the relative distances and edge features between nodes i and j respectively. Following [41], we do not update the coordinates of nodes that belong to the pocket to ensure the three-dimensional protein context remains fixed throughout the EGNN layers.

We can break the symmetry to reflections and thereby make the GNN layer $SE(3)$ -equivariant by adding a cross product-dependent term to the coordinate update, which changes sign under reflection:

$$\mathbf{x}_i^{l+1} = \mathbf{x}_i^l + \sum_{j \neq i} \frac{\mathbf{x}_i^l - \mathbf{x}_j^l}{d_{ij} + 1} \phi_x^d(\mathbf{h}_i^l, \mathbf{h}_j^l, d_{ij}^2, a_{ij}) \quad (6)$$

$$+ \frac{(\mathbf{x}_i^l - \bar{\mathbf{x}}^l) \times (\mathbf{x}_j^l - \bar{\mathbf{x}}^l)}{\|(\mathbf{x}_i^l - \bar{\mathbf{x}}^l) \times (\mathbf{x}_j^l - \bar{\mathbf{x}}^l)\| + 1} \phi_x^\times(\mathbf{h}_i^l, \mathbf{h}_j^l, d_{ij}^2, a_{ij}). \quad (7)$$

Here, $\bar{\mathbf{x}}^l$ denotes the center of mass of all nodes at layer l . ϕ_x^\times is an additional MLP. This modification is discussed in more detail in Appendix E.

7.4 Inpainting

For molecular inpainting as shown in Figure 1C, a subset of all atoms is fixed and serves as the molecular context we want to condition on. All other atoms are generated by the DDPM. To this end, we diffuse the fixed atoms at each time step and predict a new latent representation z_{t-1}^{gen} with the neural network. We then replace the generated atoms corresponding to fixed nodes with their forward noised counterparts:

$$z_{t-1}^{\text{input}} \sim q(z_{t-1} | z_{\text{data}}) \quad (8)$$

$$z_{t-1}^{\text{gen}} \sim p_\theta(z_{t-1} | z_t) \quad (9)$$

$$z_{t-1} = [z_{t-1}^{\text{input}}, z_{t-1, i \notin \mathcal{M}}^{\text{gen}}], \quad (10)$$

where \mathcal{M} denotes the set of mask indices used to uniquely identify nodes corresponding to fixed atoms. In this manner, we traverse the Markov chain in reverse order from $t = T$ to $t = 0$ to generate conditional samples. Because the noise schedule decreases the noising process's variance to almost zero at $t = 0$ (Equation (1)), the final sample is guaranteed to contain an unperturbed representation of the fixed atoms. This approach can be applied to pocket-conditioned ligand-inpainting by fixing all pocket nodes when sampling from the joint distribution model. However, it is much more general and allows us to mask and replace arbitrary parts of the ligand-pocket system without retraining, and can also be combined with the conditionally trained model—an option we explore in Section 4.

Equivariance

Since the equivariant diffusion process is defined for a CoM-free system, we must ensure that this requirement remains satisfied after the substitution step in Equation (10). To prevent a CoM shift, we therefore translate the fixed atom representation so that its center of mass coincides with the predicted representation: $\tilde{\mathbf{x}}_{t-1}^{\text{input}} = \mathbf{x}_{t-1}^{\text{input}} + \frac{1}{n} \sum_{i \in \mathcal{M}} \mathbf{x}_{t-1,i}^{\text{gen}} - \frac{1}{n} \sum_{i \in \mathcal{M}} \mathbf{x}_{t-1,i}^{\text{input}}$ before creating the new combined representation $z_{t-1} = [\tilde{\mathbf{z}}_{t-1}^{\text{input}}, z_{t-1, i \notin \mathcal{M}}^{\text{gen}}]$ with $\tilde{\mathbf{z}}_{t-1}^{\text{input}} = [\tilde{\mathbf{x}}_{t-1}^{\text{input}}, \mathbf{h}_{t-1}^{\text{input}}]$ and $n = |\mathcal{M}|$.

Resampling

Trippe et al. [58] show that this simple *replacement method* inevitably introduces approximation error that can lead to inconsistent inpainted regions. In our experiments, we observe that the inpainting solution sometimes generates disconnected molecules that are not properly positioned in the target pocket (see Figure C1 for an example). Trippe et al. [58] propose to address this limitation with a particle filtering scheme that upweights more consistent samples in each denoising step. We, however, choose to adopt the conceptually simpler idea of *resampling* [38], where each latent representation is repeatedly diffused back and forth before advancing to the next time step as demonstrated in Algorithm 1. This enables the model to harmonize its prediction for the generated part and the noisy sample from the fixed part, which does not include any information about the generated part. We choose $r = 10$ resamplings per denoising step for our experiments with DiffSBDD-joint based on empirical results discussed in Appendix C.

7.5 Implementation details

Molecule size

As part of a sample's overall likelihood, we compute the empirical joint distribution of ligand and pocket nodes $p(N_L, N_P)$ observed in the training set and smooth it with a Gaussian filter ($\sigma = 1$).

Algorithm 1 Sampling with the replacement method and resampling iterations. r denotes the number of resampling steps and \mathcal{M} is a set of indices of all atoms we want to fix. Note that samples from the generative process $p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t)$ are assumed to be CoM-free.

```

Require:  $r, \mathcal{M}$ 
 $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
for  $t = T, \dots, 1$  do
    for  $k = 1, \dots, r$  do
         $\mathbf{z}_{t-1}^{\text{input}} \sim q(\mathbf{z}_{t-1}|\mathbf{z}_{\text{data}})$   $\triangleright$  Sample known context
         $\mathbf{z}_{t-1}^{\text{gen}} \sim p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t)$   $\triangleright$  Sample generated part
         $\tilde{\mathbf{x}}_{t-1}^{\text{input}} = \mathbf{x}_{t-1}^{\text{input}} + \frac{1}{n} \sum_{i \in \mathcal{M}} \mathbf{x}_{t-1,i}^{\text{gen}} - \frac{1}{n} \sum_{i \in \mathcal{M}} \mathbf{x}_{t-1,i}^{\text{input}}$   $\triangleright$  Adjust center of mass
         $\mathbf{z}_{t-1} = [\mathbf{z}_{t-1}^{\text{input}}, \mathbf{z}_{t-1, i \notin \mathcal{M}}^{\text{gen}}]$   $\triangleright$  Combine
        if  $k < r$  then
             $\mathbf{z}_t \sim q(\mathbf{z}_t|\mathbf{z}_{t-1})$   $\triangleright$  Apply noise and repeat
        end if
    end for
end for
return  $\mathbf{z}_0$ 

```

In the conditional generation scenario, we derive the distribution $p(N_L|N_P)$ and use it for likelihood computations.

For sampling, we can either fix molecule sizes manually or sample the number of ligand nodes from the same distribution given the number of nodes in the target pocket:

$$N_L \sim p(N_L|N_P). \quad (11)$$

For the experiments discussed in Section 3, we increase the mean size of sampled molecules by 5 (CrossDocked) and 10 (Binding MOAD) atoms, respectively, to approximately match the sizes of molecules found in the test set. This modification makes the reported Vina scores more comparable as the *in silico* docking score is highly correlated with the molecular size, which is demonstrated in Figure F4. Average molecule sizes after applying the correction are shown in Table F4 together with corresponding values for generated molecules from other methods.

Preprocessing

All molecules are expressed as graphs. The full atom model uses the same one hot encoding of atom types for ligand and protein nodes. For the C_α only model the node features of the protein are set as the one hot encoding of the amino acid type instead. We refrain from adding a categorical feature for distinguishing between protein and ligand atoms and use two separate MLPs for embedding the node features instead (see Figure 1F).

Noise schedule

We use the pre-defined polynomial noise schedule introduced in [24]:

$$\tilde{\alpha}_t = 1 - \left(\frac{t}{T} \right)^2, \quad t = 0, \dots, T \quad (12)$$

Following [24, 59], values of $\tilde{\alpha}_{t|s}^2 = (\frac{\tilde{\alpha}_t}{\tilde{\alpha}_s})^2$ are clipped between 0.001 and 1 for numerical stability near $t = T$, and $\tilde{\alpha}_t$ is recomputed as

$$\tilde{\alpha}_t = \prod_{\tau=0}^t \tilde{\alpha}_{\tau|\tau-1}. \quad (13)$$

A tiny offset $\epsilon = 10^{-5}$ is used to avoid numerical problems at $t = 0$ defining the final noise schedule:

$$\alpha_t^2 = (1 - 2\epsilon) \cdot \tilde{\alpha}_t^2 + \epsilon. \quad (14)$$

Feature scaling

We scale the node type features \mathbf{h} by a factor of 0.25 relative to the coordinates \mathbf{x} which was empirically found to improve model performance in previous work [24]. To train joint probability models in the all-atom scenario, it was necessary to scale down the coordinates (and corresponding distance cutoffs) by a factor of 0.2 instead in order to avoid introducing too many edges in the graph near the end of the diffusion process at $t = T$.

Hyperparameters

Hyperparameters for all presented models are summarized in Table 3. Training takes about 2.5 h/3.8 h (conditional/joint) per 100 epochs on a single NVIDIA V100 for Binding MOAD in the C_α scenario and 11.5 h/14.7 h per 100 epochs with full atom pocket representation on two V100 GPUs. For CrossDocked, 100 training epochs take approximately 6 h/8 h in the C_α case and 48 h/60 h per 100 epochs on a single NVIDIA A100 GPU with all atom pocket representation.

Table 3: DiffSBDD hyperparameters.

	CrossDocked				Binding MOAD			
	Cond	Joint	Cond (C_α)	Joint (C_α)	Cond	Joint	Cond (C_α)	Joint (C_α)
No. layers	5	5	6	6	6	6	5	5
Joint embedding dim.	32	32	128	128	128	128	32	32
Hidden dim.	128	128	256	256	192	192	128	128
Learning rate	10^{-3}	10^{-3}	10^{-3}	10^{-3}	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$
Weight decay	10^{-12}	10^{-12}	10^{-12}	10^{-12}	10^{-12}	10^{-12}	10^{-12}	10^{-12}
Diffusion steps	500	500	500	500	500	500	500	500
Edges (ligand-ligand)	fully connected	fully connected	fully connected	fully connected	fully connected	fully connected	fully connected	fully connected
Edges (ligand-pocket)	$< 5 \text{ \AA}$	$< 5 \text{ \AA}$	$< 5 \text{ \AA}$	$< 5 \text{ \AA}$	$< 7 \text{ \AA}$	$< 7 \text{ \AA}$	$< 8 \text{ \AA}$	$< 8 \text{ \AA}$
Edges (pocket-pocket)	$< 5 \text{ \AA}$	$< 5 \text{ \AA}$	$< 5 \text{ \AA}$	$< 5 \text{ \AA}$	$< 4 \text{ \AA}$	$< 4 \text{ \AA}$	$< 8 \text{ \AA}$	$< 8 \text{ \AA}$
Epochs	1000	1000	1000	1000	1000	1000	1000	1000

Postprocessing

For postprocessing of generated molecules, we use a similar procedure as in [60]. Given a list of atom types and coordinates, bonds are first added using OpenBabel [61]. We then use RDKit to sanitise molecules and filter for the largest molecular fragment.

7.6 Experimental Set-up

Datasets

We use the CrossDocked dataset [26] with 100,000 high-quality protein-ligand pairs for training and 100 proteins for testing, following previous works [15, 60]. The data split was done by 30% sequence identity using MMseqs2 [62].

We also evaluate our method on a curated dataset of experimentally determined complexed protein-ligand structures from Binding MOAD [25]. We keep pockets with valid³ and moderately ‘drug-like’ ligands with QED score > 0.3 . We further discard small molecules that contain atom types $\notin \{C, N, O, S, B, Br, Cl, P, I, F\}$ as well as binding pockets with non-standard amino acids. We define binding pockets as the set of residues that have any atom within 8 Å of any ligand atom.

³as defined in <http://www.bindingmoad.org/>

Ligand redundancy is reduced by randomly sampling at most 50 molecules with the same chemical component identifier (3-letter-code). After removing corrupted entries that could not be processed, 40 344 training pairs and 130 testing pairs remain. A validation set of size 246 is used to monitor estimated log-likelihoods during training. The split is made to ensure different sets do not contain proteins from the same Enzyme Commission Number (EC Number) main class.

Since the ResGen baseline model did not successfully generate samples for 21 and 5 targets from the CrossDocked and Binding MOAD test sets, respectively, our analyses are only performed for samples from the remaining pockets.

Baselines

We select two recently published autoregressive deep-learning methods for structure-based drug design. Pocket2Mol [15] and ResGen [30] are sequential schemes relying on graph representations of the protein pocket and previously placed atoms to predict probabilities based on which new atoms are added. They are currently state-of-the-art among this class of models. Similar methods (e.g. 3D-SBDD [60] or GraphBP [17]) consistently under-performed these baselines in our initial experiments and are therefore not included in the final comparison. For Pocket2Mol, we re-evaluate already generated ligands on the CrossDocked dataset kindly provided by the authors. All other results were produced using the official implementations available online⁴ with default sampling parameters. Note that, unlike DiffSBDD, we therefore sample for the Binding MOAD test set with Pocket2Mol and ResGen models that have been trained on CrossDocked. Since these two sets overlap (30 test PDBs from Binding MOAD are found in the CrossDocked training set), there is potential data leakage. In practice, however, we do not observe significantly different results when these targets are excluded from the analysis. We also attempted to train Pocket2Mol on Binding MOAD, but did not manage to robustly train the model on this dataset due to instability during training. While we aimed to sample 100 ligands per pocket for the results in Section 3, the exact number of available molecules varies slightly due to the characteristics of the different methods (see Table F3).

Evaluation metrics

We employ widely-used metrics to assess the quality of our generated molecules [14, 15]: (1) **Vina Score** is an empirical estimation of binding affinity between small molecules and their target pocket; (2) **QED** is a simple quantitative estimation of drug-likeness combining several desirable molecular properties [27]; (3) **SA** estimates synthetic accessibility, i.e. the difficulty of synthesis [28]; (4) **Lipinski** measures how many rules in the Lipinski rule of five [63] are satisfied (in addition to the original four rules we require 10 or fewer rotatable bonds); (5) **Diversity** is computed as the average pairwise dissimilarity (1 - *Tanimoto similarity*) between molecular fingerprints of all generated molecules for each pocket; (6) **Inference Time** is the average sampling time per target. Roughly 100-130 molecules were sampled per target but the exact number varies between targets for Pocket2Mol and ResGen. Chemical properties are calculated with RDKit [64]. Docking scores are obtained after local minimization with an empirical force field using the GNINA implementation [33] or, if specified, after redocking with QuickVina2 [65].

Supplementary Information. Description of method details and additional experimental results are included in the Supplementary Information.

Code Availability. Our source codes are publicly available at <https://github.com/arneschneuing/DiffSBDD>. Model weights can be downloaded from Zenodo: <https://zenodo.org/records/8183747>.

Data Availability. The subset of the CrossDocked dataset used in this study was curated in a previous work and is available online: <https://github.com/pengxingang/Pocket2Mol/tree/main/data>. The raw BindingMOAD data can be downloaded from <http://www.bindingmoad.org/>. We

⁴<https://github.com/pengxingang/Pocket2Mol>, <https://github.com/HaotianZhangAI4Science/ResGen>

provide further instructions on how to process these data in our code repository: <https://github.com/arneschneuing/DiffSBDD>. Sampled molecules are available on Zenodo: <https://zenodo.org/records/8239058>.

Acknowledgments. We thank Xingang Peng and Shitong Luo for providing us generated molecules from Pocket2Mol. We thank Hannes Stärk and Joshua Southern for valuable feedback and insightful discussions. This work was supported by the European Research Council (starting grant no. 716058), the Swiss National Science Foundation (grant no. 310030_188744), and Microsoft Research AI4Science. Charles Harris is supported by the Cambridge Centre for AI in Medicine Studentship which is in turn funded by AstraZeneca and GSK. Michael Bronstein is supported in part by ERC Consolidator grant no. 724228 (LEMAN).

References

- [1] Anderson, A.C.: The process of structure-based drug design. *Chemistry & biology* **10**(9), 787–797 (2003)
- [2] Lyne, P.D.: Structure-based virtual screening: an overview. *Drug discovery today* **7**(20), 1047–1055 (2002)
- [3] Shoichet, B.K.: Virtual screening of chemical libraries. *Nature* **432**(7019), 862–865 (2004)
- [4] Irwin, J.J., Shoichet, B.K.: Zinc- a free database of commercially available compounds for virtual screening. *Journal of chemical information and modeling* **45**(1), 177–182 (2005)
- [5] Ferreira, L.G., Dos Santos, R.N., Oliva, G., Andricopulo, A.D.: Molecular docking and structure-based drug design strategies. *Molecules* **20**(7), 13384–13421 (2015)
- [6] Bronstein, M.M., Bruna, J., Cohen, T., Veličković, P.: Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. arXiv preprint arXiv:2104.13478 (2021)
- [7] Atz, K., Grisoni, F., Schneider, G.: Geometric deep learning on molecular representations. *Nature Machine Intelligence* **3**(12), 1023–1032 (2021)
- [8] Khakzad, H., Igashov, I., Schneuing, A., Goverde, C., Bronstein, M., Correia, B.: A new age in protein design empowered by deep learning. *Cell Systems* **14**(11), 925–939 (2023)
- [9] Gaudelet, T., Day, B., Jamasb, A.R., Soman, J., Regep, C., Liu, G., Hayter, J.B.R., Vickers, R., Roberts, C., Tang, J., Roblin, D., Blundell, T.L., Bronstein, M.M., Taylor-King, J.P.: Utilizing graph machine learning within drug discovery and development. *Briefings in Bioinformatics* **22**(6) (2021) <https://doi.org/10.1093/bib/bbab159>
- [10] Lu, W., Wu, Q., Zhang, J., Rao, J., Li, C., Zheng, S.: Tankbind: Trigonometry-aware neural networks for drug-protein binding structure prediction. *bioRxiv* (2022)
- [11] Stärk, H., Ganea, O., Pattanaik, L., Barzilay, R., Jaakkola, T.: Equibind: Geometric deep learning for drug binding structure prediction. In: International Conference on Machine Learning, pp. 20503–20521 (2022). PMLR
- [12] Corso, G., Stärk, H., Jing, B., Barzilay, R., Jaakkola, T.: Diffdock: Diffusion steps, twists, and turns for molecular docking. arXiv preprint arXiv:2210.01776 (2022)
- [13] Ragoza, M., Masuda, T., Koes, D.R.: Generating 3d molecules conditional on receptor binding sites with deep generative models. *Chemical science* **13**(9), 2701–2713 (2022)
- [14] Li, Y., Pei, J., Lai, L.: Structure-based de novo drug design using 3d deep generative models. *Chemical science* **12**(41), 13664–13675 (2021)

- [15] Peng, X., Luo, S., Guan, J., Xie, Q., Peng, J., Ma, J.: Pocket2mol: Efficient molecular sampling based on 3d protein pockets. arXiv preprint arXiv:2205.07249 (2022)
- [16] Drotár, P., Jamasb, A.R., Day, B., Cangea, C., Liò, P.: Structure-aware generation of drug-like molecules. arXiv preprint arXiv:2111.04107 (2021)
- [17] Liu, M., Luo, Y., Uchino, K., Maruhashi, K., Ji, S.: Generating 3d molecules for target protein binding. arXiv preprint arXiv:2204.09410 (2022)
- [18] Guan, J., Qian, W.W., Peng, X., Su, Y., Peng, J., Ma, J.: 3d equivariant diffusion for target-aware molecule generation and affinity prediction. arXiv preprint arXiv:2303.03543 (2023)
- [19] Lin, H., Huang, Y., Liu, M., Li, X., Ji, S., Li, S.Z.: Diffbp: Generative diffusion of 3d molecules for target protein binding. arXiv preprint arXiv:2211.11214 (2022)
- [20] Guan, J., Zhou, X., Yang, Y., Bao, Y., Peng, J., Ma, J., Liu, Q., Wang, L., Gu, Q.: Decompdiff: Diffusion models with decomposed priors for structure-based drug design (2023)
- [21] Xu, M., Powers, A.S., Dror, R.O., Ermon, S., Leskovec, J.: Geometric latent diffusion models for 3d molecule generation. In: International Conference on Machine Learning, pp. 38592–38610 (2023). PMLR
- [22] Weiss, T., Mayo Yanes, E., Chakraborty, S., Cosmo, L., Bronstein, A.M., Gershoni-Poranne, R.: Guided diffusion for inverse molecular design. Nature Computational Science, 1–10 (2023)
- [23] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems **33**, 6840–6851 (2020)
- [24] Hoogeboom, E., Satorras, V.G., Vignac, C., Welling, M.: Equivariant diffusion for molecule generation in 3d. In: International Conference on Machine Learning, pp. 8867–8887 (2022). PMLR
- [25] Hu, L., Benson, M.L., Smith, R.D., Lerner, M.G., Carlson, H.A.: Binding moad (mother of all databases). Proteins: Structure, Function, and Bioinformatics **60**(3), 333–340 (2005)
- [26] Francoeur, P.G., Masuda, T., Sunseri, J., Jia, A., Iovanisci, R.B., Snyder, I., Koes, D.R.: 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)
- [27] Bickerton, G.R., Paolini, G.V., Besnard, J., Muresan, S., Hopkins, A.L.: Quantifying the chemical beauty of drugs. Nature chemistry **4**(2), 90–98 (2012)
- [28] Ertl, P., Schuffenhauer, A.: Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. Journal of cheminformatics **1**(1), 1–11 (2009)
- [29] Wildman, S.A., Crippen, G.M.: Prediction of physicochemical parameters by atomic contributions. Journal of chemical information and computer sciences **39**(5), 868–873 (1999)
- [30] Zhang, O., Zhang, J., Jin, J., Zhang, X., Hu, R., Shen, C., Cao, H., Du, H., Kang, Y., Deng, Y., et al.: Resgen is a pocket-aware 3d molecular generation model based on parallel multiscale modelling. Nature Machine Intelligence, 1–11 (2023)
- [31] Peng, X., Luo, S., Guan, J., Xie, Q., Peng, J., Ma, J.: Pocket2Mol: Efficient Molecular Sampling Based on 3D Protein Pockets. <https://github.com/pengxingang/Pocket2Mol>
- [32] Zhang, O., Zhang, J., Jin, J., Zhang, X., Hu, R., Shen, C., Cao, H., Du, H., Kang, Y., Deng, Y., et al.: ResGen: A Pocket-aware 3D Molecular Generation Model Based on Parallel Multi-scale Modeling. <https://github.com/HaotianZhangAI4Science/ResGen>
- [33] McNutt, A.T., Francoeur, P., Aggarwal, R., Masuda, T., Meli, R., Ragoza, M., Sunseri, J., Koes,

- D.R.: Gnina 1.0: molecular docking with deep learning. *Journal of cheminformatics* **13**(1), 1–20 (2021)
- [34] Chen, P., Tao, L., Wang, T., Zhang, J., He, A., Lam, K.-h., Liu, Z., He, X., Perry, K., Dong, M., *et al.*: Structural basis for recognition of frizzled proteins by clostridium difficile toxin b. *Science* **360**(6389), 664–669 (2018)
- [35] Ding, L.-C., Huang, X.-Y., Zheng, F.-F., Xie, J., She, L., Feng, Y., Su, B.-H., Zheng, D.-L., Lu, Y.-G.: Fzd2 inhibits the cell growth and migration of salivary adenoid cystic carcinomas. *Oncology Reports* **35**(2), 1006–1012 (2016)
- [36] Ritchie, T.J., Macdonald, S.J.: The impact of aromatic ring count on compound developability— are too many aromatic rings a liability in drug design? *Drug discovery today* **14**(21-22), 1011–1020 (2009)
- [37] Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456* (2020)
- [38] Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Van Gool, L.: Repaint: Inpainting using denoising diffusion probabilistic models. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11461–11471 (2022)
- [39] Wang, J., Lisanza, S., Juergens, D., Tischer, D., Watson, J.L., Castro, K.M., Ragotte, R., Saragovi, A., Milles, L.F., Baek, M., *et al.*: Scaffolding protein functional sites using deep learning. *Science* **377**(6604), 387–394 (2022)
- [40] Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., Fleet, D.J.: Video diffusion models. *arXiv:2204.03458* (2022)
- [41] Igashov, I., Stärk, H., Vignac, C., Schneuing, A., Satorras, V.G., Frossard, P., Welling, M., Bronstein, M., Correia, B.: Equivariant 3d-conditional diffusion model for molecular linker design. *Nature Machine Intelligence*, 1–11 (2024)
- [42] Kim, K.S., Lu, S., Cornelius, L.A., Lombardo, L.J., Borzilleri, R.M., Schroeder, G.M., Sheng, C., Rovnyak, G., Crews, D., Schmidt, R.J., *et al.*: Synthesis and sar of pyrrolotriazine-4-one based eg5 inhibitors. *Bioorganic & medicinal chemistry letters* **16**(15), 3937–3942 (2006)
- [43] Barone, M., Müller, M., Chiha, S., Ren, J., Albat, D., Soicke, A., Dohmen, S., Klein, M., Bruns, J., Dinther, M., *et al.*: Designed nanomolar small-molecule inhibitors of ena/vasp evh1 interaction impair invasion and extravasation of breast cancer cells. *Proceedings of the National Academy of Sciences* **117**(47), 29684–29690 (2020)
- [44] Gahbauer, S., Correy, G.J., Schuller, M., Ferla, M.P., Doruk, Y.U., Rachman, M., Wu, T., Diolaiti, M., Wang, S., Neitz, R.J., *et al.*: Iterative computational design and crystallographic screening identifies potent inhibitors targeting the nsp3 macrodomain of sars-cov-2. *Proceedings of the National Academy of Sciences* **120**(2), 2212931120 (2023)
- [45] Schuller, M., Correy, G.J., Gahbauer, S., Fearon, D., Wu, T., Díaz, R.E., Young, I.D., Carvalho Martins, L., Smith, D.H., Schulze-Gahmen, U., *et al.*: Fragment binding to the nsp3 macrodomain of sars-cov-2 identified through crystallographic screening and computational docking. *Science advances* **7**(16), 8711 (2021)
- [46] Imrie, F., Bradley, A.R., Schaar, M., Deane, C.M.: Deep generative models for 3d linker design. *Journal of chemical information and modeling* **60**(4), 1983–1995 (2020)
- [47] Böhm, H.-J., Flohr, A., Stahl, M.: Scaffold hopping. *Drug discovery today: Technologies* **1**(3), 217–224 (2004)
- [48] Luo, S., Su, Y., Peng, X., Wang, S., Peng, J., Ma, J.: Antigen-specific antibody design and

optimization with diffusion-based generative models. bioRxiv (2022)

- [49] Sorrell, F.J., Szklarz, M., Azeez, K.R.A., Elkins, J.M., Knapp, S.: Family-wide structural analysis of human numb-associated protein kinases. *Structure* **24**(3), 401–411 (2016)
- [50] Debreczeni, J., Eswaran, J., Bullock, A., Filippakopoulos, P., Kavanagh, K., Amos, A., Fedorov, O., Sobott, F., Ball, L., Von Delft, F., et al.: Crystal structure of the human serine-threonine kinase 16 in complex with staurosporine [internet]. RCSB PDB Protein Data Bank (2005)
- [51] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International Conference on Machine Learning, pp. 2256–2265 (2015). PMLR
- [52] Kingma, D., Salimans, T., Poole, B., Ho, J.: Variational diffusion models. *Advances in neural information processing systems* **34**, 21696–21707 (2021)
- [53] Lepola, U., Wade, A., Andersen, H.F.: Do equivalent doses of escitalopram and citalopram have similar efficacy? a pooled analysis of two positive placebo-controlled studies in major depressive disorder. *International clinical psychopharmacology* **19**(3), 149–155 (2004)
- [54] Satorras, V.G., Hoogeboom, E., Welling, M.: E (n) equivariant graph neural networks. In: International Conference on Machine Learning, pp. 9323–9332 (2021). PMLR
- [55] Köhler, J., Klein, L., Noé, F.: Equivariant flows: exact likelihood generative learning for symmetric densities. In: International Conference on Machine Learning, pp. 5361–5370 (2020). PMLR
- [56] Xu, M., Yu, L., Song, Y., Shi, C., Ermon, S., Tang, J.: Geodiff: A geometric diffusion model for molecular conformation generation. arXiv preprint arXiv:2203.02923 (2022)
- [57] Serre, J.-P., et al.: Linear Representations of Finite Groups vol. 42. Springer, New York (1977)
- [58] Trippe, B.L., Yim, J., Tischer, D., Broderick, T., Baker, D., Barzilay, R., Jaakkola, T.: Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. arXiv preprint arXiv:2206.04119 (2022)
- [59] Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: International Conference on Machine Learning, pp. 8162–8171 (2021). PMLR
- [60] Luo, S., Guan, J., Ma, J., Peng, J.: A 3d generative model for structure-based drug design. *Advances in Neural Information Processing Systems* **34**, 6229–6239 (2021)
- [61] O’Boyle, N.M., Banck, M., James, C.A., Morley, C., Vandermeersch, T., Hutchison, G.R.: Open babel: An open chemical toolbox. *Journal of cheminformatics* **3**(1), 1–14 (2011)
- [62] Steinegger, M., Söding, J.: MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology* **35**(11), 1026–1028 (2017) <https://doi.org/10.1038/nbt.3988>
- [63] Lipinski, C.A., Lombardo, F., Dominy, B.W., Feeney, P.J.: Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced drug delivery reviews* **64**, 4–17 (2012)
- [64] Landrum, G., et al.: Rdkit: Open-source cheminformatics software (2016)
- [65] Alhossary, A., Handoko, S.D., Mu, Y., Kwoh, C.-K.: Fast, accurate, and reliable molecular docking with quickvina 2. *Bioinformatics* **31**(13), 2214–2216 (2015)
- [66] Adams, K., Pattanaik, L., Coley, C.W.: Learning 3d representations of molecular chirality with invariance to bond rotations. arXiv preprint arXiv:2110.04383 (2021)

- [67] Li, Q.: Application of fragment-based drug discovery to versatile targets. *Frontiers in molecular biosciences* **7**, 180 (2020)
- [68] Gahbauer, S., Correy, G.J., Schuller, M., Ferla, M.P., Doruk, Y.U., Rachman, M., Wu, T., Diolaiti, M., Wang, S., Neitz, R.J., *et al.*: Iterative computational design and crystallographic screening identifies potent inhibitors targeting the nsp3 macrodomain of sars-cov-2. *Proceedings of the National Academy of Sciences* **120**(2), 2212931120 (2023)
- [69] Bemis, G.W., Murcko, M.A.: The properties of known drugs. 1. molecular frameworks. *Journal of medicinal chemistry* **39**(15), 2887–2893 (1996)
- [70] Kong, Z., Ping, W., Huang, J., Zhao, K., Catanzaro, B.: Diffwave: A versatile diffusion model for audio synthesis. In: *International Conference on Learning Representations* (2021)
- [71] Luo, S., Hu, W.: Diffusion probabilistic models for 3d point cloud generation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2837–2845 (2021)
- [72] Du, Y., Fu, T., Sun, J., Liu, S.: Molgensurvey: A systematic survey in machine learning models for molecule design. *arXiv preprint arXiv:2203.14500* (2022)
- [73] Jing, B., Corso, G., Chang, J., Barzilay, R., Jaakkola, T.: Torsional diffusion for molecular conformer generation. *arXiv preprint arXiv:2206.01729* (2022)
- [74] Anand, N., Achim, T.: Protein structure and sequence generation with equivariant denoising diffusion probabilistic models. *arXiv preprint arXiv:2205.15019* (2022)
- [75] Kelley, L.A., Mezulis, S., Yates, C.M., Wass, M.N., Sternberg, M.J.: The phyre2 web portal for protein modeling, prediction and analysis. *Nature protocols* **10**(6), 845–858 (2015)
- [76] Kalyaanamoorthy, S., Chen, Y.-P.P.: Structure-based drug design to augment hit discovery. *Drug discovery today* **16**(17-18), 831–839 (2011)
- [77] Duvenaud, D.K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., Adams, R.P.: Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems* **28** (2015)
- [78] Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: *International Conference on Machine Learning*, pp. 1263–1272 (2017). PMLR
- [79] Lapchekskyi, K., Miller, B., Geiger, M., Smidt, T.: Euclidean neural networks (e3nn) v1. 0. Technical report, Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States) (2020)
- [80] Du, W., Zhang, H., Du, Y., Meng, Q., Chen, W., Zheng, N., Shao, B., Liu, T.-Y.: Se (3) equivariant graph neural networks with complete local frames. In: *International Conference on Machine Learning*, pp. 5583–5608 (2022). PMLR
- [81] Schütt, K.T., Sauceda, H.E., Kindermans, P.-J., Tkatchenko, A., Müller, K.-R.: Schnet—a deep learning architecture for molecules and materials. *The Journal of Chemical Physics* **148**(24), 241722 (2018)
- [82] Klicpera, J., Groß, J., Günnemann, S.: Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123* (2020)
- [83] Duan, C., Du, Y., Jia, H., Kulik, H.J.: Accurate transition state generation with an object-aware equivariant elementary reaction diffusion model. *Nature Computational Science* **3**(12), 1045–1055 (2023)
- [84] Batzner, S., Musaelian, A., Sun, L., Geiger, M., Mailoa, J.P., Kornbluth, M., Molinari, N., Smidt, T.E., Kozinsky, B.: E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications* **13**(1), 1–11 (2022)

- [85] Holdijk, L., Du, Y., Hooft, F., Jaini, P., Ensing, B., Welling, M.: Path integral stochastic optimal control for sampling transition paths. arXiv preprint arXiv:2207.02149 (2022)

Appendix A Note on Variational Lower Bound

To maximise the likelihood of our training data, we aim at optimising the variational lower bound (VLB) [24, 52]

$$-\log p(\mathbf{z}_{\text{data}}) \leq \underbrace{D_{\text{KL}}(q(\mathbf{z}_T|\mathbf{z}_{\text{data}})||p(\mathbf{z}_T))}_{\text{prior loss } \mathcal{L}_{\text{prior}}} - \underbrace{\mathbb{E}_{q(\mathbf{z}_0|\mathbf{z}_{\text{data}})}[\log p(\mathbf{z}_{\text{data}}|\mathbf{z}_0)]}_{\text{reconstruction loss } \mathcal{L}_0} + \underbrace{\sum_{t=1}^T \mathcal{L}_t}_{\text{diffusion loss}} \quad (\text{A1})$$

with

$$\mathcal{L}_t = D_{\text{KL}}(q(\mathbf{z}_{t-1}|\mathbf{z}_{\text{data}}, \mathbf{z}_t)||p_{\theta}(\mathbf{z}_{t-1}|\hat{\mathbf{z}}_{\text{data}}, \mathbf{z}_t)) \quad (\text{A2})$$

$$= \mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[\frac{1}{2} \left(\frac{\text{SNR}(t-1)}{\text{SNR}(t)} - 1 \right) \|\epsilon - \hat{\epsilon}_{\theta}\|^2 \right] \quad (\text{A3})$$

during training. The prior loss should always be close to zero and can be computed exactly in closed form while the reconstruction loss must be estimated as described in Hoogeboom et al. [24]. In practice, however, we simply minimise the mean squared error $\mathcal{L}_{\text{train}} = \frac{1}{2} \|\epsilon - \hat{\epsilon}\|^2$ while randomly sampling time steps $t \sim \mathcal{U}(0, \dots, T)$, which is equivalent up to a multiplicative factor.

Appendix B Note on Equivariance of the Conditional Model

The 3D-conditional model can achieve equivariance without the usual ‘‘subspace-trick’’. The coordinates of pocket nodes provide a reference frame for all samples that can be used to translate them to a unique location (e.g. such that the pocket is centered at the origin: $\sum_i \mathbf{x}_i^{(P)} = \mathbf{0}$). By doing this for all training data, translation equivariance becomes irrelevant and the CoM-free subspace approach obsolete. To evaluate the likelihood of translated samples at inference time, we can first subtract the pocket’s center of mass from the whole system and compute the likelihood after this mapping. Similarly, for sampling molecules we can first generate a ligand in a CoM-free version of the pocket and move the whole system back to the original location of the pocket nodes to restore translation equivariance. As long as the mean of our Gaussian noise distribution $p(\mathbf{z}_t|\mathbf{z}_{\text{data}}^{(P)}) = \mathcal{N}(\boldsymbol{\mu}(\mathbf{z}_{\text{data}}^{(P)}), \sigma^2 \mathbf{I})$ depends equivariantly on the pocket node coordinates $\mathbf{x}^{(P)}$, $O(3)$ -equivariance is satisfied as well (Appendix D). Since this change did not seem to affect the performance of the conditional model in our experiments, we decided to keep sampling in the linear subspace to ensure that the implementation is as similar as possible to the joint model, for which the subspace approach is necessary.

Appendix C Resampling

Here, we briefly recapitulate the resampling algorithm introduced in Ref. [38]. The key intuition is that inpainting with the replacement method combines a generated part with an independently sampled latent representation of the known part. Even though the neural network tries to reconcile these two components in every step of the denoising trajectory, it cannot succeed because the same issue reoccurs in the following step. Lugmayr et al. [38] thus propose to apply the neural network several times before proceeding to the next noise level, allowing the DDPM to preserve more conditional information and move the sample closer to the data distribution again.

Table C1: Evaluation of generated molecules for target pockets from the CrossDocked (C.D.) and Binding MOAD (B.M.) test sets with the inpainting approach and C_α pocket representation for varying numbers of resampling steps r and denoising steps T . Here, the SA scores were mapped to the unit interval using $SA_{\text{norm}} = (10 - SA)/9$.

	r	T	Vina Score (kcal/mol, ↓)	QED (↑)	SA_{norm} (↑)	Lipinski (↑)	Diversity (↑)	Time (s, ↓)
C.D.	1	500	-5.830 ± 2.47	0.403 ± 0.18	0.552 ± 0.13	4.620 ± 0.81	0.808 ± 0.06	97.434 ± 39.79
	5	100	-6.872 ± 2.43	0.444 ± 0.19	0.551 ± 0.12	4.654 ± 0.72	0.766 ± 0.06	96.205 ± 39.22
	10	50	-7.177 ± 3.28	0.556 ± 0.20	0.729 ± 0.12	4.742 ± 0.59	0.718 ± 0.07	94.481 ± 38.86
B.M.	1	500	-5.810 ± 2.00	0.468 ± 0.16	0.627 ± 0.14	4.839 ± 0.49	0.851 ± 0.04	40.298 ± 13.52
	5	100	-6.082 ± 2.01	0.537 ± 0.16	0.701 ± 0.13	4.924 ± 0.31	0.855 ± 0.05	45.074 ± 21.14
	10	50	-6.192 ± 2.24	0.560 ± 0.16	0.737 ± 0.13	4.941 ± 0.27	0.859 ± 0.05	41.490 ± 14.32

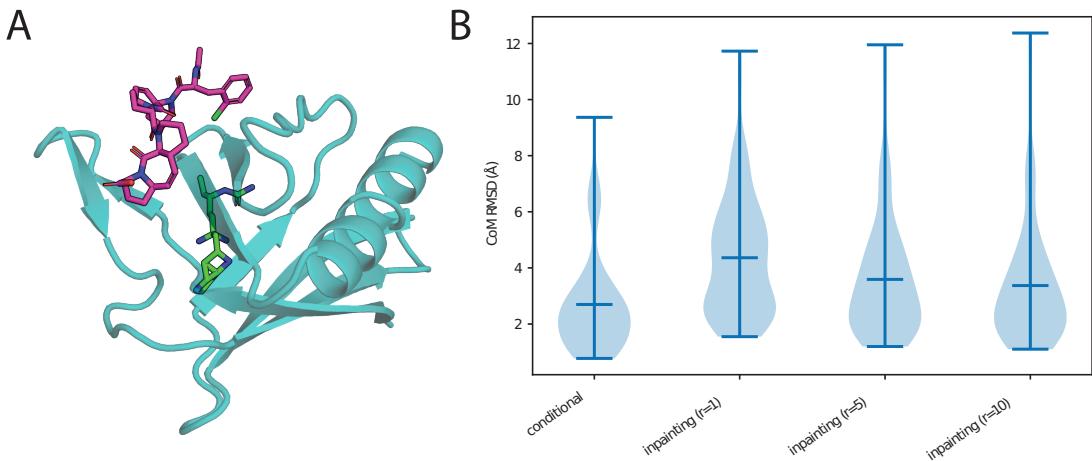


Fig. C1: (A) Example of a generated molecule (green) without additional resampling steps and the reference molecule (magenta) from the target PDB 5ncf. The generated molecule is not placed in the target pocket but in the protein core. (B) RMSD between reference molecules' center of mass and generated molecules' center of mass for the conditional model and inpainting model with varying numbers of resampling steps r . The pocket representation is C_α in all cases.

Number of resampling steps

To empirically study the effect of the number of resampling iterations applied, we generated ligands for all test pockets with $r = 1$, $r = 5$, and $r = 10$ resampling steps, respectively. Because the resampling strategy slows down sampling approximately by a factor of r , we used the striding technique proposed by Nichol and Dhariwal [59] and reduced the number of denoising steps proportionally to r . Nichol and Dhariwal [59] showed that this approach reduces the number of sampling steps significantly without sacrificing sample quality. In our case, it allows us to retain sampling speed while increasing the number of resampling steps.

To gauge the effect of resampling for molecule generation we show the distribution of RMSD values between the center of mass of reference molecules and generated molecules in Figure C1. The unmodified replacement method ($r = 1$) produces molecules that are clearly farther away from the presumed pocket center than the conditional model. Increasing r moves the mean distance closer to the average displacement of molecules from the conditional method. This effect seems to saturate at $r = 10$ which is in line with the results obtained for images [38].

Table C1 shows that neither the additional resampling steps nor the shortened denoising trajectory degrade the performance on the reported molecular metrics. The average docking scores even improve slightly which might reflect better positioning of generated ligands in the pockets prior to docking. The same model trained with $T = 500$ diffusion steps was used in all three cases.

Appendix D Proofs

In the following proofs we do not consider categorical node features \mathbf{h} as only the positions \mathbf{x} are subject to equivariance constraints. Furthermore, we do not distinguish between the zeroth latent representation \mathbf{x}_0 and data domain representations \mathbf{x}_{data} for ease of notation, and simply drop the subscripts.

D.1 $O(3)$ -equivariance of the prior probability

The isotropic Gaussian prior $p(\mathbf{x}_T^{(L)} | \mathbf{x}^{(P)}) = \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}^{(P)}), \sigma^2 \mathbf{I})$ is equivariant to rotations and reflections represented by an orthogonal matrix $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ as long as $\boldsymbol{\mu}(\mathbf{R}\mathbf{x}^{(P)}) = \mathbf{R}\boldsymbol{\mu}(\mathbf{x}^{(P)})$ because:

$$\begin{aligned} p(\mathbf{R}\mathbf{x}_T^{(L)} | \mathbf{R}\mathbf{x}^{(P)}) &= \frac{1}{\sqrt{(2\pi)^{N_L} \sigma^2}} \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{R}\mathbf{x}_T^{(L)} - \boldsymbol{\mu}(\mathbf{R}\mathbf{x}^{(P)})\|^2\right) \\ &= \frac{1}{\sqrt{(2\pi)^{N_L} \sigma^2}} \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{R}\mathbf{x}_T^{(L)} - \mathbf{R}\boldsymbol{\mu}(\mathbf{x}^{(P)})\|^2\right) \\ &= \frac{1}{\sqrt{(2\pi)^{N_L} \sigma^2}} \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{R}(\mathbf{x}_T^{(L)} - \boldsymbol{\mu}(\mathbf{x}^{(P)}))\|^2\right) \\ &= \frac{1}{\sqrt{(2\pi)^{N_L} \sigma^2}} \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x}_T^{(L)} - \boldsymbol{\mu}(\mathbf{x}^{(P)})\|^2\right) \\ &= p(\mathbf{x}_T^{(L)} | \mathbf{x}^{(P)}). \end{aligned}$$

Here we used $\|\mathbf{R}\mathbf{x}\|_2 = \|\mathbf{x}\|_2$ for orthogonal \mathbf{R} .

D.2 $O(3)$ -equivariance of the transition probabilities

The denoising transition probabilities from time step t to $s < t$ are defined as isotropic normal distributions:

$$p_\theta(\mathbf{x}_{t-1}^{(L)} | \mathbf{x}_t^{(L)}, \hat{\mathbf{x}}^{(L)}, \mathbf{x}^{(P)}) = \mathcal{N}(\mathbf{x}_{t-1}^{(L)} | \boldsymbol{\mu}_{t \rightarrow s}(\mathbf{x}_t^{(L)}, \hat{\mathbf{x}}^{(L)}, \mathbf{x}^{(P)}), \sigma_{t \rightarrow s}^2 \mathbf{I}). \quad (\text{D4})$$

Therefore, $p_\theta(\mathbf{x}_{t-1}^{(L)} | \mathbf{x}_t^{(L)}, \hat{\mathbf{x}}^{(L)}, \mathbf{x}^{(P)})$ is $O(3)$ -equivariant by a similar argument to Section D.1 if $\boldsymbol{\mu}_{t \rightarrow s}$ is computed equivariantly from the three-dimensional context.

Recalling the definition of $\boldsymbol{\mu}_{t \rightarrow s} = \frac{\alpha_{t|s} \sigma_s^2}{\sigma_t^2} \mathbf{x}_t^{(L)} + \frac{\alpha_s \sigma_{t|s}^2}{\sigma_t^2} \hat{\mathbf{x}}^{(L)}$, we can prove its equivariance as follows:

$$\begin{aligned} \boldsymbol{\mu}_{t \rightarrow s}(\mathbf{R}\mathbf{x}_t^{(L)}, \mathbf{R}\mathbf{x}^{(P)}) &= \frac{\alpha_{t|s} \sigma_s^2}{\sigma_t^2} \mathbf{R}\mathbf{x}_t^{(L)} + \frac{\alpha_s \sigma_{t|s}^2}{\sigma_t^2} \hat{\mathbf{x}}^{(L)}(\mathbf{R}\mathbf{x}_t^{(L)}, \mathbf{R}\mathbf{x}^{(P)}) \\ &= \frac{\alpha_{t|s} \sigma_s^2}{\sigma_t^2} \mathbf{R}\mathbf{x}_t^{(L)} + \frac{\alpha_s \sigma_{t|s}^2}{\sigma_t^2} \mathbf{R}\hat{\mathbf{x}}^{(L)}(\mathbf{x}_t^{(L)}, \mathbf{x}^{(P)}) \quad (\text{equivariance of } \hat{\mathbf{x}}^{(L)}) \\ &= \mathbf{R}\left(\frac{\alpha_{t|s} \sigma_s^2}{\sigma_t^2} \mathbf{x}_t^{(L)} + \frac{\alpha_s \sigma_{t|s}^2}{\sigma_t^2} \hat{\mathbf{x}}^{(L)}(\mathbf{x}_t^{(L)}, \mathbf{x}^{(P)})\right) \\ &= \mathbf{R}\boldsymbol{\mu}_{t \rightarrow s}(\mathbf{x}_t^{(L)}, \mathbf{x}^{(P)}), \end{aligned}$$

where $\hat{\mathbf{x}}^{(L)}$ defined as $\hat{\mathbf{x}}^{(L)} = \frac{1}{\alpha_t} \mathbf{x}_t^{(L)} - \frac{\sigma_t}{\alpha_t} \hat{\epsilon}$ is equivariant because:

$$\hat{\mathbf{x}}^{(L)}(\mathbf{R}\mathbf{x}_t^{(L)}, \mathbf{R}\mathbf{x}^{(P)}) = \frac{1}{\alpha_t} \mathbf{R}\mathbf{x}_t^{(L)} - \frac{\sigma_t}{\alpha_t} \hat{\epsilon}(\mathbf{R}\mathbf{x}_t^{(L)}, \mathbf{R}\mathbf{x}^{(P)}, t)$$

$$\begin{aligned}
&= \frac{1}{\alpha_t} \mathbf{R} \mathbf{x}_t^{(L)} - \frac{\sigma_t}{\alpha_t} \mathbf{R} \hat{\epsilon}(\mathbf{x}_t^{(L)}, \mathbf{x}^{(P)}, t) \quad (\hat{\epsilon} \text{ predicted by equivariant neural network}) \\
&= \mathbf{R} \left(\frac{1}{\alpha_t} \mathbf{x}_t^{(L)} - \frac{\sigma_t}{\alpha_t} \hat{\epsilon}(\mathbf{x}_t^{(L)}, \mathbf{x}^{(P)}, t) \right) \\
&= \mathbf{R} \hat{\mathbf{x}}^{(L)}(\mathbf{x}_t^{(L)}, \mathbf{x}^{(P)}).
\end{aligned}$$

D.3 $O(3)$ -equivariance of the learned likelihood

Let $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ be an orthogonal matrix representing an element g from the general orthogonal group $O(3)$. We obtain the marginal probability density of the Markovian denoising process as follows

$$\begin{aligned}
p_\theta(\mathbf{x}_0^{(L)} | \mathbf{x}^{(P)}) &= \int p(\mathbf{x}_T^{(L)} | \mathbf{x}^{(P)}) p_\theta(\mathbf{x}_{0:T-1}^{(L)} | \mathbf{x}_T^{(L)}, \mathbf{x}^{(P)}) d\mathbf{x}_{1:T} \\
&= \int p(\mathbf{x}_T^{(L)} | \mathbf{x}^{(P)}) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}^{(L)} | \mathbf{x}_t^{(L)}, \mathbf{x}^{(P)}) d\mathbf{x}_{1:T}
\end{aligned}$$

and the sample's likelihood is $O(3)$ -equivariant:

$$\begin{aligned}
p_\theta(\mathbf{R} \mathbf{x}_0^{(L)} | \mathbf{R} \mathbf{x}^{(P)}) &= \int p(\mathbf{R} \mathbf{x}_T^{(L)} | \mathbf{R} \mathbf{x}^{(P)}) \prod_{t=1}^T p_\theta(\mathbf{R} \mathbf{x}_{t-1}^{(L)} | \mathbf{R} \mathbf{x}_t^{(L)}, \mathbf{R} \mathbf{x}^{(P)}) d\mathbf{x}_{1:T} \\
&= \int p(\mathbf{x}_T^{(L)} | \mathbf{x}^{(P)}) \prod_{t=1}^T p_\theta(\mathbf{R} \mathbf{x}_{t-1}^{(L)} | \mathbf{R} \mathbf{x}_t^{(L)}, \mathbf{R} \mathbf{x}^{(P)}) d\mathbf{x}_{1:T} \quad (\text{equivariant prior}) \\
&= \int p(\mathbf{x}_T^{(L)} | \mathbf{x}^{(P)}) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}^{(L)} | \mathbf{x}_t^{(L)}, \mathbf{x}^{(P)}) d\mathbf{x}_{1:T} \quad (\text{equivariant transition probabilities}) \\
&= p_\theta(\mathbf{x}_0^{(L)} | \mathbf{x}^{(P)})
\end{aligned}$$

Appendix E $SE(3)$ -equivariant Graph Neural Network

Chiral molecules cannot be superimposed by any combination of rotations and translations. Instead they are mirrored along a stereocenter, axis, or plane. As chirality can significantly alter a molecule's chemical properties, we use a variant of the $E(3)$ -equivariant graph neural networks [54] presented in Equations (3)-(5) that is sensitive to reflections and hence $SE(3)$ -equivariant. We change the coordinate update equation, Equ. (5), of standard EGNNs in the following way

$$\mathbf{x}_i^{l+1} = \mathbf{x}_i^l + \sum_{j \neq i} \frac{\mathbf{x}_i^l - \mathbf{x}_j^l}{d_{ij} + 1} \phi_x^d(\mathbf{h}_i^l, \mathbf{h}_j^l, d_{ij}^2, a_{ij}) + \frac{(\mathbf{x}_i^l - \bar{\mathbf{x}}^l) \times (\mathbf{x}_j^l - \bar{\mathbf{x}}^l)}{\|(\mathbf{x}_i^l - \bar{\mathbf{x}}^l) \times (\mathbf{x}_j^l - \bar{\mathbf{x}}^l)\| + 1} \phi_x^\times(\mathbf{h}_i^l, \mathbf{h}_j^l, d_{ij}^2, a_{ij}), \quad (E5)$$

where $\bar{\mathbf{x}}^l$ denotes the center of mass of all nodes at layer l . This modification makes the EGNN layer sensitive to reflections while staying close to the original formalism. Since the resulting graph neural networks are only equivariant to the $SE(3)$ group, we will hereafter call them $SE(3)$ GNNs for short.

E.1 Discussion of Equivariance

Here we study how the suggested change in the coordinate update equation breaks reflection symmetry while preserving equivariance to rotations. Messages and scalar feature updates (Equations (3) and (4)) remain $E(3)$ -invariant as in the original model and are therefore not considered in this section. We analyze transformations composed of a translation by $\mathbf{t} \in \mathbb{R}^3$ and a rotation/reflection

by an orthogonal matrix $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ with $\mathbf{R}^T \mathbf{R} = \mathbf{I}$. The output at layer $l + 1$ given the transformed input $\mathbf{R}\mathbf{x}_i^l + \mathbf{t}$ at layer l is calculated as:

$$\mathbf{R}\mathbf{x}_i^l + \mathbf{t} + \sum_{j \neq i} \frac{\mathbf{R}\mathbf{x}_i^l + \mathbf{t} - (\mathbf{R}\mathbf{x}_j^l + \mathbf{t})}{d_{ij} + 1} \phi_x^d(\cdot) + \frac{(\mathbf{R}\mathbf{x}_i^l + \mathbf{t} - (\mathbf{R}\bar{\mathbf{x}}^l + \mathbf{t})) \times (\mathbf{R}\mathbf{x}_j^l + \mathbf{t} - (\mathbf{R}\bar{\mathbf{x}}^l + \mathbf{t}))}{Z_{ij}^\times + 1} \phi_x^\times(\cdot) \quad (\text{E6})$$

$$= \mathbf{R}\mathbf{x}_i^l + \mathbf{t} + \sum_{j \neq i} \frac{\mathbf{R}(\mathbf{x}_i^l - \mathbf{x}_j^l)}{d_{ij} + 1} \phi_x^d(\cdot) + \frac{(\mathbf{R}\mathbf{x}_i^l - \mathbf{R}\bar{\mathbf{x}}^l) \times (\mathbf{R}\mathbf{x}_j^l - \mathbf{R}\bar{\mathbf{x}}^l)}{Z_{ij}^\times + 1} \phi_x^\times(\cdot) \quad (\text{E7})$$

$$= \mathbf{R}\mathbf{x}_i^l + \mathbf{t} + \sum_{j \neq i} \frac{\mathbf{R}(\mathbf{x}_i^l - \mathbf{x}_j^l)}{d_{ij} + 1} \phi_x^d(\cdot) + \frac{\det(\mathbf{R})\mathbf{R}((\mathbf{x}_i^l - \bar{\mathbf{x}}^l) \times (\mathbf{x}_j^l - \bar{\mathbf{x}}^l))}{Z_{ij}^\times + 1} \phi_x^\times(\cdot) \quad (\text{E8})$$

$$= \mathbf{R}\mathbf{x}_i^{l+1} + \mathbf{t} + (\det(\mathbf{R}) - 1) \sum_{j \neq i} \frac{\mathbf{R}((\mathbf{x}_i^l - \bar{\mathbf{x}}^l) \times (\mathbf{x}_j^l - \bar{\mathbf{x}}^l))}{Z_{ij}^\times + 1}. \quad (\text{E9})$$

This result shows that the output coordinates are only equivariantly transformed if \mathbf{R} is orientation preserving, i.e. $\det(\mathbf{R}) = 1$. If \mathbf{R} is a reflection ($\det(\mathbf{R}) = -1$), coordinates will be updated with an additional summand that breaks the symmetry.

The learnable coefficients $\phi_x^d(\mathbf{h}_i^l, \mathbf{h}_j^l, d_{ij}^2, a_{ij})$ and $\phi_x^\times(\mathbf{h}_i^l, \mathbf{h}_j^l, d_{ij}^2, a_{ij})$ only depend on relative distances and are therefore $E(3)$ -invariant. Their arguments are represented with the “.” symbol for brevity. Likewise, the normalization factor $\|(\mathbf{x}_i^l - \bar{\mathbf{x}}^l) \times (\mathbf{x}_j^l - \bar{\mathbf{x}}^l)\|$ is abbreviated as Z_{ij}^\times . Already in the first line we used the fact that the mean transforms equivariantly. Furthermore, we use $\mathbf{Ra} \times \mathbf{Rb} = \det(\mathbf{R})\mathbf{R}(\mathbf{a} \times \mathbf{b})$ in the second step, which can be derived as follows:

$$\mathbf{x}^T(\mathbf{Ra} \times \mathbf{Rb}) = \det(\underbrace{[\mathbf{x}, \mathbf{Ra}, \mathbf{Rb}]}_{\in \mathbb{R}^{3 \times 3}}) \quad (\text{E10})$$

$$= \det(\mathbf{R}[\mathbf{R}^T \mathbf{x}, \mathbf{a}, \mathbf{b}]) \quad (\text{E11})$$

$$= \det(\mathbf{R}) \det([\mathbf{R}^T \mathbf{x}, \mathbf{a}, \mathbf{b}]) \quad (\text{E12})$$

$$= \det(\mathbf{R})(\mathbf{x}^T \mathbf{R}(\mathbf{a} \times \mathbf{b})) \quad (\text{E13})$$

$$= \mathbf{x}^T(\det(\mathbf{R})\mathbf{R}(\mathbf{a} \times \mathbf{b})) \quad (\text{E14})$$

The stated property of the cross product follows because this derivation is true for all $\mathbf{x} \in \mathbb{R}^3$.

E.2 Empirical Results

To show the effectiveness of this architecture on a simple toy example, we repeat the classification experiment by Adams et al. [66] who train neural networks to classify tetrahedral chiral centers as right-handed (*rectus*, ‘R’) or left-handed (*sinister*, ‘S’). We closely follow their data split and experimental set-up and only replace the classifier with EGNN and SE(3)GNNs, respectively. The results in Table E2 clearly demonstrate that the $SE(3)$ -equivariant EGNN is capable of solving this task (without any hyperparameter optimization) whereas the $E(3)$ -equivariant version does not do better than random guessing.

Table E2: Accuracy on the R/S classification task. Results in the first section are taken from [66] and included for reference.

Model	R/S Accuracy (%)
ChIRo	98.5
SchNet	54.4
DimeNet++	65.7
SphereNet	98.2
EGNN	50.4
SE(3)GNN	83.4

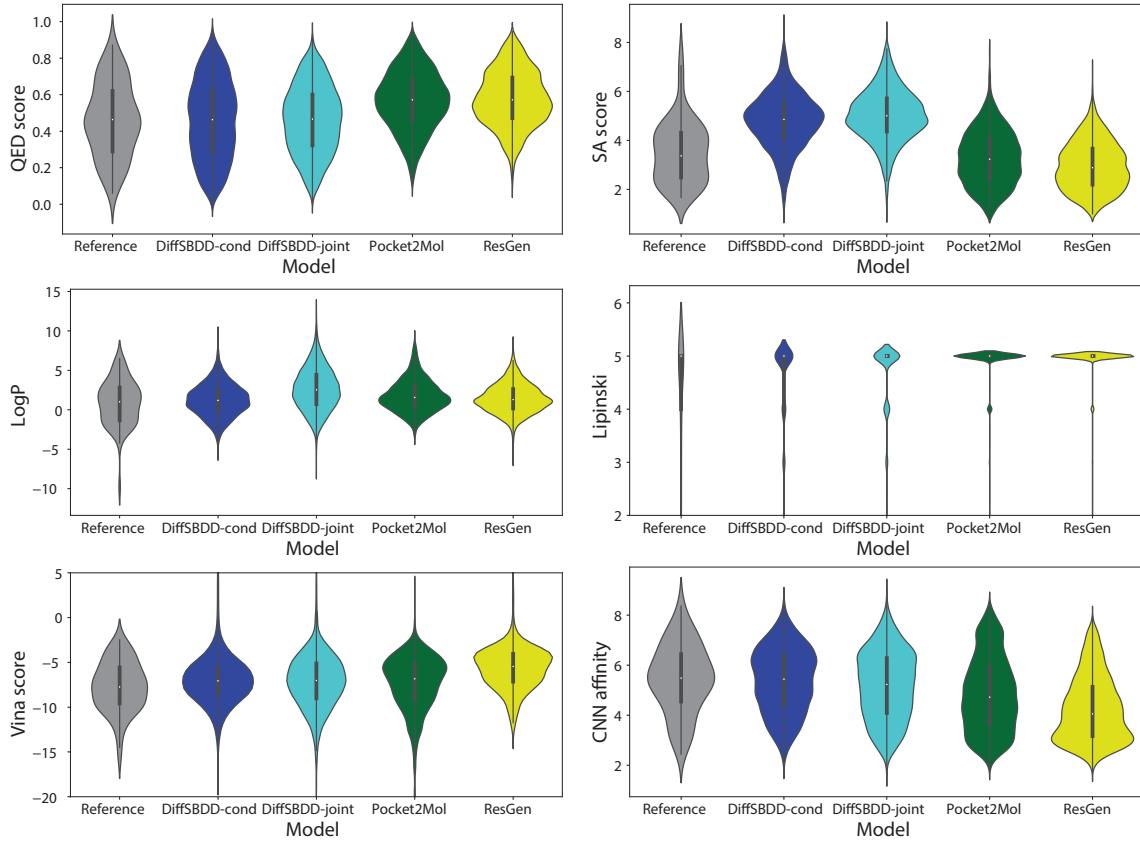


Fig. F2: Distributions of computational scores for generated molecules and reference ligands from the CrossDocked test set.

Appendix F Extended results

F.1 Sampling statistics

Table F3 summarizes the number of generated molecules available for the analyses in Section 3.

Table F3: Average number of molecules per test set pocket.

Method	CrossDocked	Binding MOAD
Test set	1	1
Pocket2Mol [15]	98.01	132.14
ResGen [30]	114.41	109.83
DiffSBDD-cond	100	100
DiffSBDD-joint	100	100

F.2 Distributions of Molecular Properties

The distributions used to compute the Wasserstein distances in Table 1 are visualized in Figures F2 and F3.

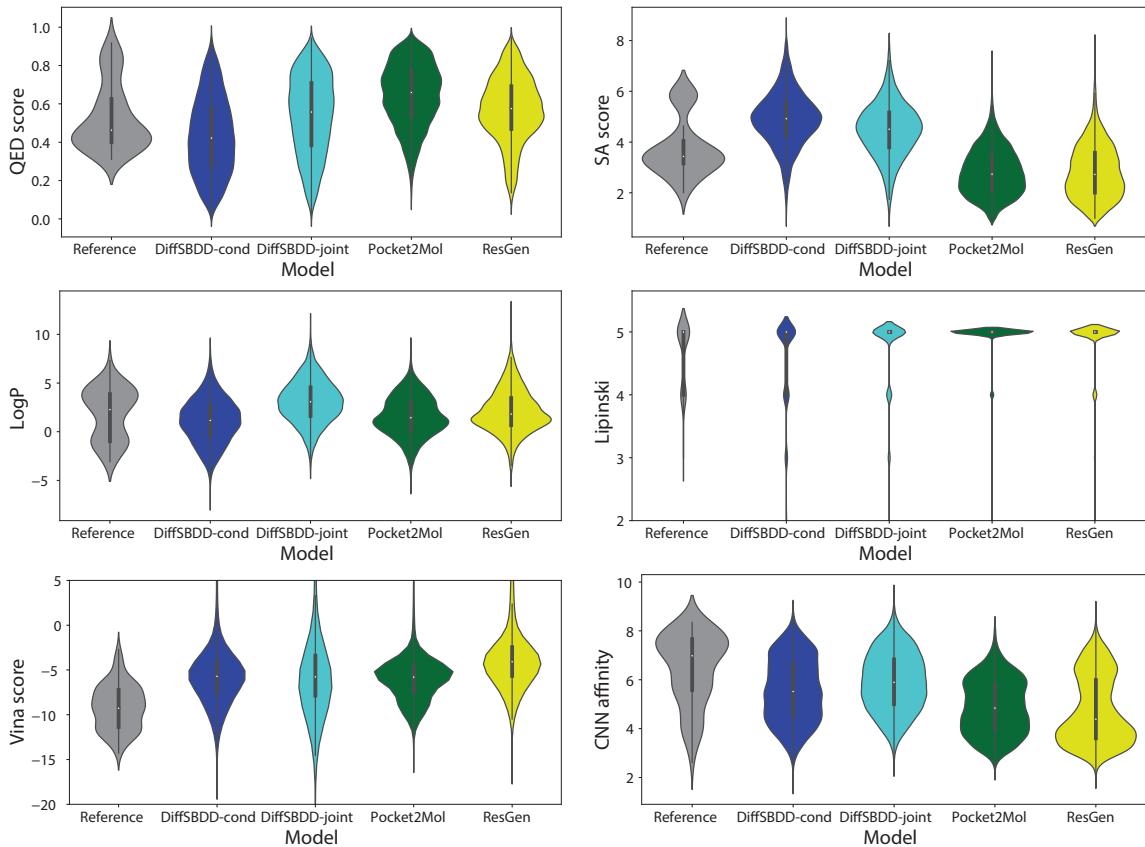


Fig. F3: Distributions of computational scores for generated molecules and reference ligands from the Binding MOAD test set.

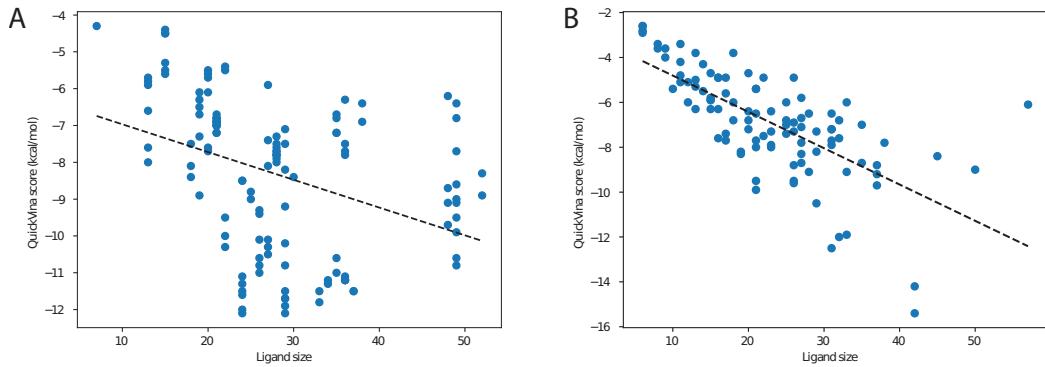


Fig. F4: Correlation between ligand size and QuickVina score for reference molecules from the Binding MOAD (A) and CrossDocked (B) test sets.

F.3 Dependence of Vina scores on molecule size

Figure F4 shows how strongly the empirical Vina score is correlated with the number of heavy atoms in the ligands. For this analysis, we used Vina scores computed by the QuickVina2 [65] software after re-docking. Since we want to match the distribution of scores of reference molecules as closely as possible with our generated molecules, we expect that their sizes should roughly match as well. However, our diffusion model operates on point clouds with fixed sizes, which we determine at the beginning of sampling as explained in Section 7.5. By biasing the procedure, we match the sizes of reference ligands more closely as shown in Table F4.

Table F4: Average number of heavy atoms of generated molecules.

Method	CrossDocked	Binding MOAD
Test set	23.9	28.0
Pocket2Mol [15]	19.0	16.8
ResGen [30]	16.2	18.4
DiffSBDD-cond	24.9	24.5
DiffSBDD-joint	24.5	25.2

F.4 Additional Molecular Metrics

In addition to the molecular properties discussed in Section 7.6 we assess the models’ ability to produce novel and valid molecules using four simple metrics: validity, connectivity, uniqueness, and novelty. **Validity** measures the proportion of generated molecules that pass basic tests by RDKit—mostly ensuring correct valencies. **Connectivity** is the proportion of valid molecules that do not contain any disconnected fragments. We convert every valid and connected molecule from a graph into a canonical SMILES string representation, count the number unique occurrences in the set of generated molecules and compare those to the training set SMILES to compute **uniqueness** and **novelty** respectively.

Table F5 shows that only a small fraction of all generated molecules is invalid and must be discarded for downstream processing. A much larger percentage of molecules is fragmented but, since we can simply select and process the largest fragments in these cases, low connectivity does not necessarily affect the efficiency of the generative process. Moreover, all models produce diverse sets of molecules unseen in the training set.

Table F5: Basic molecular metrics for generated small molecules given a C_α and full atom representation of the protein pocket.

Model	Validity	Connectivity	Uniqueness	Novelty
CrossDocked test set	100%	100%	96%	96.88%
DiffSBDD-cond (C_α)	95.52%	79.52%	99.99%	99.97%
DiffSBDD-inpaint (C_α)	99.18%	98.25%	99.52%	99.97%
DiffSBDD-cond	97.10%	78.27%	99.98%	99.99%
DiffSBDD-inpaint	92.99%	67.52%	100%	100%
Binding MOAD test set	97.69%	100%	38.58%	77.55%
DiffSBDD-cond (C_α)	94.41%	77.38%	100%	100%
DiffSBDD-inpaint (C_α)	98.36%	91.60%	99.99%	99.98%
DiffSBDD-cond	96.32%	63.37%	100%	100%
DiffSBDD-inpaint	93.88%	75.60%	100%	100%

F.5 Results with a coarse-grained pocket representation

Here we discuss the advantages and disadvantages of coarse-grained C_α protein representations as context for the generative model. Table F6 shows that full-atom models outperform their coarse-grained counterparts on the Vina metric, which is the only reported metric that captures interactions with the protein. Ligand-centric metrics do not seem to depend on the protein representation as could be expected. The main advantage of the C_α models is their significantly faster training and inference time, a fact we made use of during model development for fine-tuning and preliminary analyses.

To further demonstrate the limitations of the coarse-grained models, we compare the generated raw conformations to the best scoring QuickVina docking pose after re-docking and plot the distribution

Table F6: Evaluation of generated molecules for targets from the CrossDocked and Binding MOAD test sets. We compare all-atom and coarse-grained (C_α) pocket representations. Note that the SA score has been normalized so that higher is better. Here, the SA scores were mapped to the unit interval using $SA_{norm} = (10 - SA)/9$.

	Vina (All) (↓)	Vina (Top-10%) (↓)	QED (↑)	SA _{norm} (↑)	Lipinski (↑)	Diversity (↑)	Time (s, ↓)
Test set	-6.871 ± 2.32	—	0.476 ± 0.20	0.728 ± 0.14	4.340 ± 1.14	—	—
C.D.	DiffSBDD-cond (C_α)	-6.770 ± 2.73	-8.796 ± 1.75	0.475 ± 0.22	0.612 ± 0.12	4.536 ± 0.91	0.725 ± 0.06
	DiffSBDD-inpaint (C_α)	-7.177 ± 3.28	-9.233 ± 1.82	0.556 ± 0.20	0.729 ± 0.12	4.742 ± 0.59	0.718 ± 0.07
	DiffSBDD-cond	-6.950 ± 2.06	-9.120 ± 2.16	0.469 ± 0.21	0.578 ± 0.13	4.562 ± 0.89	0.728 ± 0.07
	DiffSBDD-inpaint	-7.333 ± 2.56	-9.927 ± 2.59	0.467 ± 0.18	0.554 ± 0.12	4.702 ± 0.64	0.758 ± 0.05
Test set	-8.412 ± 2.03	—	0.522 ± 0.17	0.692 ± 0.12	4.669 ± 0.49	—	—
B.M.	DiffSBDD-cond (C_α)	-6.863 ± 1.59	-8.587 ± 1.34	0.480 ± 0.20	0.554 ± 0.11	4.662 ± 0.68	0.714 ± 0.05
	DiffSBDD-inpaint (C_α)	-6.926 ± 3.39	-9.124 ± 1.35	0.548 ± 0.19	0.580 ± 0.13	4.757 ± 0.51	0.709 ± 0.05
	DiffSBDD-cond	-7.171 ± 1.89	-9.184 ± 2.23	0.436 ± 0.20	0.568 ± 0.12	4.542 ± 0.79	0.714 ± 0.08
	DiffSBDD-inpaint	-7.309 ± 4.03	-9.840 ± 2.18	0.542 ± 0.21	0.615 ± 0.12	4.777 ± 0.53	0.739 ± 0.05

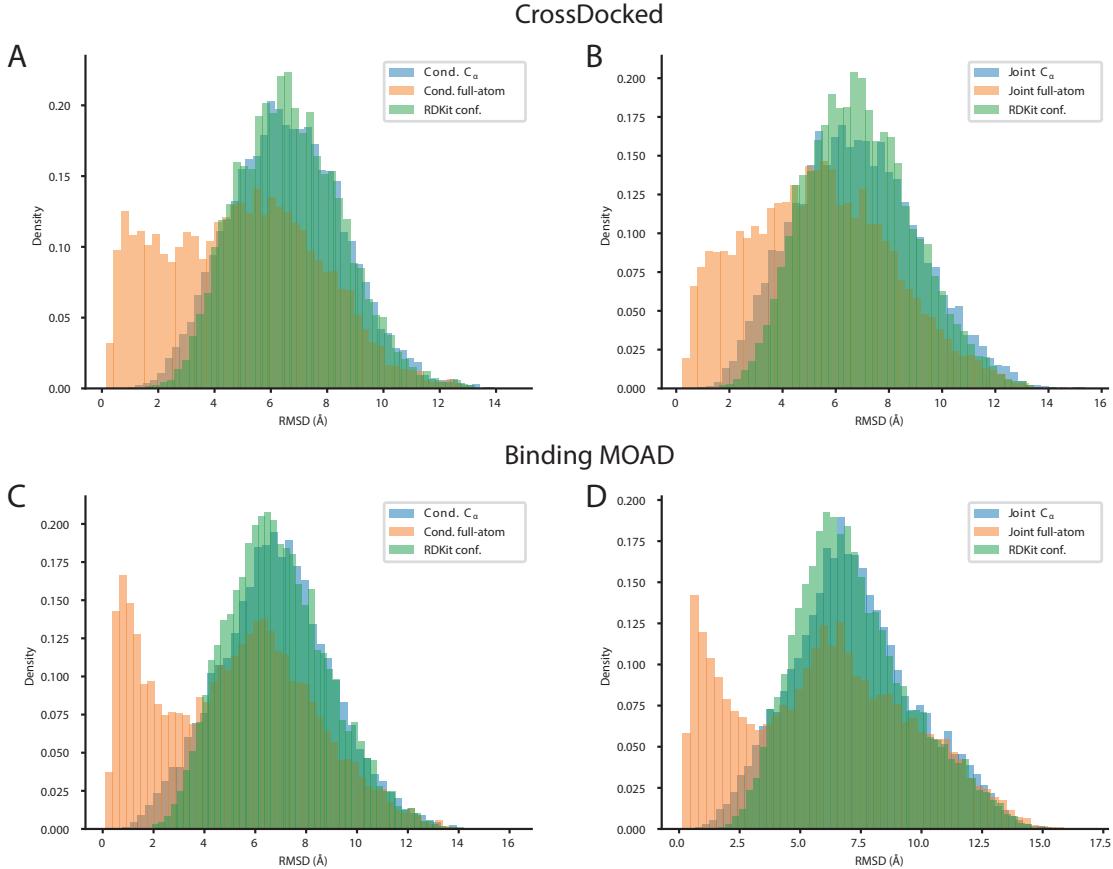


Fig. F5: RMSD between generated and docked conformations for the CrossDocked (A, B) and Binding MOAD (C, D) datasets. Full-atom models are compared to C_α models as well as a baseline of random RDKit conformers of the molecules generated by the C_α -model. (A, C) DiffSBDD-cond. (B) DiffSBDD-joint.

of resulting RMSD values in Figure F5. As a baseline, the procedure is repeated for RDKit conformers of the same molecules with identical center of mass. For a large percentage of molecules generated by the all-atom models, QuickVina agrees with the predicted bound conformations, leaving them almost unchanged (RMSD below 2 Å). This demonstrates successful conditioning on the geometry of the given protein pockets. For the C_α -only models results are less convincing. They produce poses that barely improve upon conformers lacking pocket-context. Likely, this is caused by atomic clashes with the proteins' side chains that QuickVina needs to resolve.

	DiffSBDD-cond	DiffSBDD-joint
CrossDocked		
Binding MOAD		

Fig. F6: Randomly selected samples of generated molecules.

F.6 Random generated molecules

Randomly selected molecules generated with different models are presented in Figure F6.

Appendix G Inpainting implementation

G.1 Implementation of fragment merging experiments

Fragment merging is the task of combining fragments with an overlapping binding site [67]. For this example, instead of masking existing molecules, we instead take fragments from an experimental fragment screen against the non structural protein 3 (NSP3) from SARS-CoV-2 [45]. Using two fragments as input (PDB entries 5rue and 5rsw) we successfully replicate the fragment merge performed in Gahbauer et al. [68] which was accomplished using the chemoinformatics-based approach *Fragmenstein*⁵. To accomplish this, instead of masking out and reinserting atoms, we instead choose

⁵<https://github.com/matteoferla/Fragmenstein>

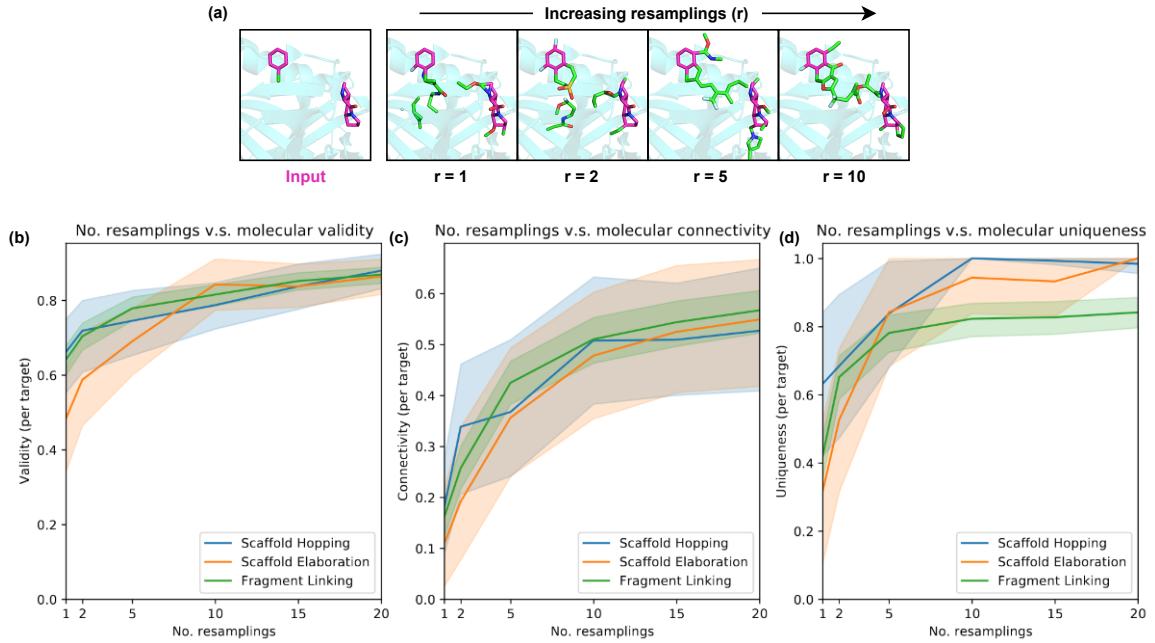


Fig. G7: (a) Importance of high resamplings. Effect of the number of resamplings on molecular validity (b), connectivity (c) and uniqueness (d).

to fix all atoms during generation except the atom on each fragment closest to the other. We need perform $t = 200$ steps of the DiffSBDD-*diversify* procedure to allow the model to arrange the atom positions as well as change the atom types. All PDB files were already structurally aligned.

G.2 Quantitative evaluation of inpainting for the whole Binding MOAD test set

For all experiments across the whole test set, we perform automatic masking of atoms which are to be fixed. For scaffold elaboration, we extract the Bemis-Murcko scaffold [69] using RDKit and compute a binary mask to fix the scaffold, while functional groups are redesigned. For scaffold hopping, we simply take the inverse of the mask used for scaffold elaboration. For linker design, we fragment each molecule in the test set in multiple ways as in Igashov et al. [41], Imrie et al. [46]. To benchmark against DiffLinker, we use the model weights and protocol as described in Igashov et al. [41] except we give the ground-truth linker size as input, rather than predict it using the auxillary model, for fairness. In small-scale experiments where finer control is desirable (e.g. as in the fragment merging example described above), the binary mask can be defined manually.

Appendix H Optimization

We demonstrate the effect the number of noising/denoising steps (t) has on various molecular properties in Figure H8. We test all values of t at intervals of 10 steps and 200 molecules are sampled at every timestep. Note this does not allow for explicit optimization of any particular property unless combined with the evolutionary algorithm, as shown in Figure 1D (all plots are for PDB entry 5NDU [43]).

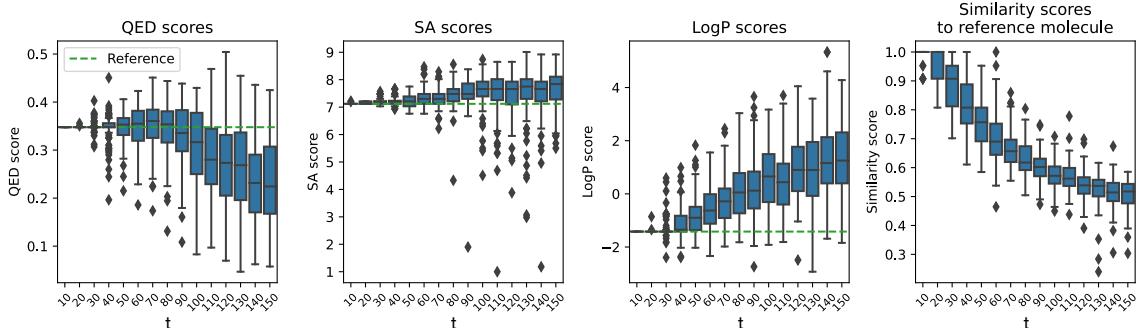


Fig. H8: Effect of number of denoising steps on molecule properties.

Appendix I Related Work

Diffusion Models for Molecules

Inspired by non-equilibrium thermodynamics, diffusion models have been proposed to learn data distributions by modeling a denoising (reverse diffusion) process and have achieved remarkable success in a variety of tasks such as image, audio synthesis and point cloud generation [52, 70, 71]. Recently, efforts have been made to utilize diffusion models for molecule design [72]. Specifically, Hoogeboom et al. [24] propose a diffusion model with an equivariant network that operates both on continuous atomic coordinates and categorical atom types to generate new molecules in 3D space. Torsional Diffusion [73] focuses on a conditional setting where molecular conformations (atomic coordinates) are generated from molecular graphs (atom types and bonds). Similarly, 3D diffusion models have been applied to generative design of larger biomolecular structures, such as antibodies [48] and other proteins [58, 74].

Structure-based Drug Design

Structure-based Drug Design (SBDD) [1, 5] relies on the knowledge of the 3D structure of the biological target obtained either through experimental methods or high-confidence predictions using homology modelling [75]. Candidate molecules are then designed to bind with high affinity and specificity to the target using interactive software [76] and often human-based intuition [5]. Recent advances in deep generative models have brought a new wave of research that model the conditional distribution of ligands given biological targets and thus enable *de novo* structure-based drug design. Most of previous work consider this task as a sequential generation problem and design a variety of generative methods including autoregressive models, reinforcement learning, etc., to generate ligands inside protein pockets atom by atom [14–16, 60]. Most recent work explore the use of diffusion models in structure-based drug design [18–20].

Geometric Deep Learning for Drug Discovery

Geometric deep learning refers to incorporating geometric priors in neural architecture design that respects symmetry and invariance, thus reduces sample complexity and eliminates the need for data augmentation [6]. It has been prevailing in a variety of drug discovery tasks from virtual screening to *de novo* drug design as symmetry widely exists in the representation of drugs. One line of work introduces graph and geometry priors and designs message passing neural networks and equivariant neural networks that are permutation-, translation-, rotation-, and reflection-equivariant, respectively [54, 77–80]. These architectures have been widely used in representing biomolecules from small molecules to proteins [7] and solving downstream tasks such as molecular property prediction [81, 82], binding pose prediction [11], transition state sampling [83], and molecular dynamics [84, 85]. Another line of work focuses on generative design of new molecules [72]. More specifically, molecule design is

formulated as a graph or geometry generation problem, following either a one-shot generation strategy that generates graphs (atom and bond features) in one step or attempting to generate atoms and bonds sequentially.