
DiffHopp: A Graph Diffusion Model for Novel Drug Design via Scaffold Hopping

Anonymous Authors¹

Abstract

Scaffold hopping is a drug discovery strategy to generate new chemical entities by modifying the core structure, the *scaffold*, of a known active compound. This approach preserves the essential molecular features of the original scaffold while introducing novel chemical elements or structural features to enhance potency, selectivity, or bioavailability. However, there is currently a lack of generative models specifically tailored for this task, especially in the pocket-conditioned context. In this work, we present DiffHopp, a conditional E(3)-equivariant graph diffusion model tailored for scaffold hopping given a known protein-ligand complex.

1. Introduction

Scaffold hopping (Böhm et al., 2004) is a widely used strategy in drug discovery that involves modifying the core structure or ‘scaffold’ of a known active compound whilst preserving the functional groups which can be seen as the ‘business-end’ of the molecule which interacts with the target (Figure 1). The aim of scaffold hopping is to retain the essential molecular features (also known as pharmacophoric features (Yang, 2010)) of the original scaffold while introducing new chemical elements or structural features that can improve the desired properties, such as potency, selectivity, or bioavailability, whilst designing molecule of novel structure.

Recently, there has been considerable excitement on the application of deep generative models for many areas within drug discovery (Tong et al., 2021; Xie et al., 2022; Isert et al., 2023; Baillif et al., 2023), particularly using diffusion models (Ho et al., 2020). There are a number of diffusion models that have been proposed for structure-based drug design (Schneuing et al., 2022), fragment-linking (Igashov et al., 2022) and molecular docking (Corso et al., 2022).

¹Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the 2023 ICML Workshop on Computational Biology. Do not distribute.

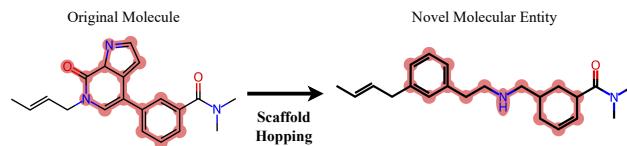


Figure 1: Scaffold hopping: Scaffold atoms highlighted in red. The scaffold holds the functional groups in place for binding. Scaffold hopping refers to interchanging scaffolds while leaving the functional groups unchanged. Compared to inpainting, scaffold hopping uses a fixed definition of scaffold-vs-rest and in contrast to fragment linking, which links large existing fragments with small molecular bridges, scaffold hopping typically redesigns the majority of a molecule.

While diffusion models for drug design can in principle be repurposed for scaffold hopping by using an inpainting formulation (appendix C), there are no diffusion models specifically designed for scaffold hopping and it is unclear how inpainting with existing models (Schneuing et al., 2022) would compare to a tailored approach.

In this work, we introduce DiffHopp, an E(3)-equivariant graph diffusion model specifically trained to perform scaffold hopping on known active compounds within protein pockets. Here, we seek to learn the conditional probability distribution of molecular scaffolds given a target pharmacophore. In summary, our main contributions are:

1. We train a 3D diffusion generative model specifically for the case of scaffold hopping that is conditioned on whole protein pockets, rather than some desired shape. We further repurpose general pocket conditioned diffusion models (Schneuing et al., 2022) via inpainting (appendix C) and observe that specific training for scaffold hopping outperforms comparable general models used via inpainting.
2. We find that using more powerful geometric graph neural networks provides a cure for low connectivity, a key limitation in current pocket-conditioned molecule generation via diffusion (Schneuing et al., 2022).

055 2. Background and Related Work

056 **Traditional Scaffold Hopping Methods** Traditionally,
057 scaffold hopping can be accomplished in different ways
058 (Sun et al., 2012). Pharmacophore-based methods define a
059 pharmacophore model which captures common features in
060 known bioactive molecules and then screen large databases
061 for active molecules of novel structure (Hessler & Bar-
062 inghaus, 2010). Fragment-based methods aim to replace
063 problematic fragments or scaffolds by searching fragment
064 databases based on a simplified chemical similarity (Birchall
065 & Gillet, 2011). However, these non-generative approaches
066 rely on similarity functions, which might not capture the
067 whole spectrum of scaffold relationships. (Hu et al., 2017)

068 **Deep Learning-based Scaffold Hopping** Early work
069 treated scaffold hopping as sequence translation problem
070 using SMILES (Zheng et al., 2021). However, this does not
071 allow reasoning about the 3D chemistry. SQUID (Adams &
072 Coley, 2022) introduces the first 3D generative model for
073 scaffold hopping, but condition on a desired chemical shape
074 rather than the full receptor chemistry. When used in an
075 inpainting formulation (Lugmayr et al. (2022), appendix C),
076 DiffSBDD (Schneuing et al., 2022), a diffusion model for
077 pocket conditioned ligand generation, can be seen as the
078 closest work to ours. Further, DiffLinker (Igashov et al.,
079 2022), which is trained to generate linkers between molec-
080 ular fragments using a conditional diffusion model, could
081 in principle be repurposed for scaffold hopping. However,
082 fragment linking typically redesigns small linkers between
083 large fragments, while scaffold hopping normally requires
084 redesigning most of the molecule and is therefore out-of-
085 distribution for the training of fragment linking models.

086 **Diffusion Models** Denoising Diffusion Probabilistic Mod-
087 els (DDPMs) (Ho et al., 2020) are a powerful class of gener-
088 ative model used to learn complex probability distributions.
089 In short, DDPMs define a Markovian diffusion process that
090 transforms an observed data distribution into a known prior
091 (typically $\mathcal{N}(\mathbf{0}, \mathbf{I})$). A *score function* (where the score is
092 the gradient of the log probability of the underlying density
093 function $\nabla_{\mathbf{x}} \log p(\mathbf{x})$) is then learnt to reverse this forward
094 diffusion process, meaning we can sample new data from
095 the tractable prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$ (Song & Ermon, 2019).

100 3. Methods

101 **Dataset** We train our model on 19,378 protein-ligand com-
102 plexes from PDBBind, filtered for QED ≥ 0.3 and split as
103 in Corso et al. (2022). From these complexes, we define
104 the Murcko-Bemis scaffold (Bemis & Murcko, 1996) for
105 each ligand using RDKit¹. Here, we treat atoms not in the
106 scaffold as functional groups.

107 ¹www.rdkit.org

Molecule representation All molecules (proteins and lig-
108 ands) are represented as geometric graphs $\mathcal{G} = \{h, x\}$ with
109 node features $h \in \mathbb{R}^{N \times F}$ and coordinates $x \in \mathbb{R}^{N \times 3}$. Lig-
110 ands are represented at an atom level, with h being the one-
111 hot encoding of the atom type. For computational efficiency,
112 protein graphs are subset to the pocket region (defined as
113 all atoms within 8 Å of the ligand) and are represented at
114 a C_{α} granularity with node features h_P being the one-hot
115 encoded residue type. Edges within the ligand are fully
116 connected, whereas all protein-ligand and protein-protein
117 edges are drawn with a radius threshold of 5Å. The edge
118 features between nodes i and j consist of the distance d_{ij}
119 and the normalised direction vector $(\mathbf{x}_i - \mathbf{x}_j)/(d_{ij})$.

DiffHopp architecture We recast the scaffold hopping
120 problem as learning a conditional probability distribution in
121 3D, where we wish to construct a new sample scaffold \mathbf{z}_0
122 given a molecular context \mathbf{u} (\mathbf{u} is the concatenation of the
123 pocket \mathbf{p} and functional groups \mathbf{g}). This is achieved using
124 an equivariant diffusion model $p_{\theta}(\mathbf{z}_0 | \mathbf{u})$ parameterized
125 using a denoising network $\varepsilon_{\theta}(\mathbf{z}_t, t, \mathbf{u})$. We parameterize
126 our denoising network ε_{θ} using a diffusion adaptation of the
127 equivariant Geometric Vector Perceptron (GVP) architec-
128 ture (Jing et al., 2020). Following previous work (Schneuing
129 et al., 2022; Igashov et al., 2022), we embed all features
130 into a shared feature space using separate Multi-Layer Per-
131 ceptrons (MLPs) $\mathbf{h}_{\text{emb}} = [\phi_z(\mathbf{h}_z), \phi_g(\mathbf{h}_g), \phi_p(\mathbf{h}_p)]$ for z , g
132 and p respectively. We then perform 7 layers of message
133 passing on the combined pocket-ligand graph to update the
134 hidden node features h' and x' . The noise estimator for the
135 scaffold is then taken as $\epsilon'_x, \epsilon'_h = \mathbf{x}'_z, \phi_{\text{out}}(\mathbf{h}'_z)$ with ϕ_{out} an
136 MLP to map from embedding space to Gaussian noise.

Training and Sampling We follow the DDPM training
137 procedure (Ho et al., 2020) outlined in detail in Appendix A
138 (Algorithm 1). To ensure equivariance, we employ the zero
139 center of mass trick from previous work (Hoogeboom et al.,
140 2022). We use $T = 500$, AdamW as the optimizer and employ
141 a polynomial variance schedule (Hoogeboom et al., 2022)
142 with $s = 10^{-4}$ and all α_t values clipped to a lower bound of 10^{-3} . We also scale atom features h
143 by 0.25, which was shown in previous work to improve
144 performance empirically (Hoogeboom et al., 2022). We
145 adapt the simplified noise-prediction objective (Schneuing
146 et al., 2022) into a reweighted loss optimizing atom type
147 and coordinate features individually:

$$L_{\text{reweighted}} = \mathbb{E} \left[\frac{1}{4} (\|\epsilon_x - \epsilon'_x\|^2 + \|\epsilon_h - \epsilon'_h\|^2) \right] \quad (1)$$

148 where $[\epsilon_x, \epsilon_h] = \epsilon$ and $[\epsilon'_x, \epsilon'_h] = \varepsilon_{\theta}(\mathbf{z}_t, t, \mathbf{u})$ denote the
149 true and predicted noise respectively. Our sampling pro-
150 cedure follows previous work on equivariant diffusion models
151 (Hoogeboom et al., 2022) and is given in Algorithm 2 (see
152 appendix).

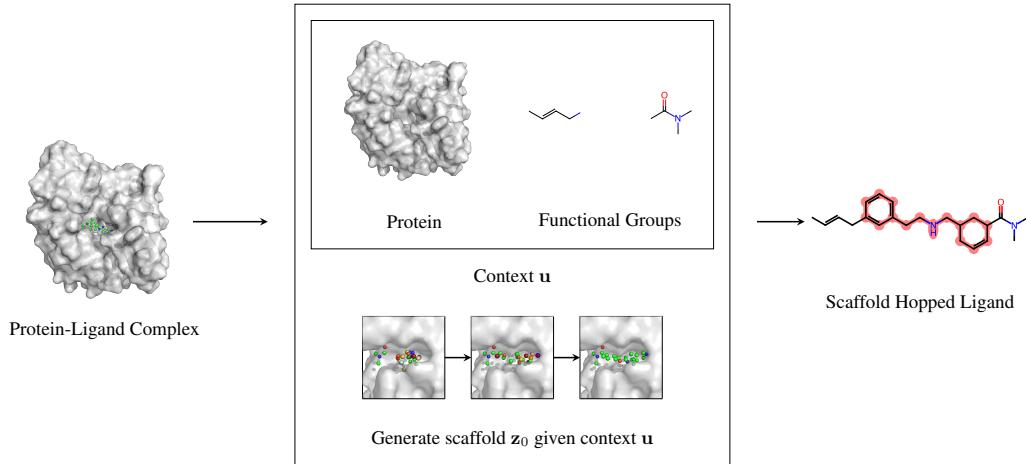


Figure 2: Visual abstract. Given a protein-ligand complex, an equivariant diffusion model is used to sample a scaffold from a scaffold distribution conditioned on functional groups and protein pocket. The resulting scaffold is merged with the functional groups to lead a scaffold hopped ligand.

Postprocessing Following Schneuing et al. (2022), we extract the resulting point cloud (fixed functional groups and designed atoms) and convert it into a molecule with bonds using OpenBabel (O’Boyle et al., 2011). Molecules are then relaxed using 200 steps of force-field relaxation with UFF (Rappé et al., 1992) to remove clashes.

4. Experiments

We set out to answer the following questions: (1) Is a model specifically trained for scaffold hopping much better than a general purpose molecule generation model used with inpainting? (2) What is the effect of using more powerful geometric graph neural networks as denoisers in diffusion for molecule generation?

Evaluation To evaluate the quality of generated molecules, we use metrics established in previous work (Schneuing et al., 2022; Igashov et al., 2022). **Connectivity** measures whether generated molecules are fully connected. **Diversity** is the average pairwise *Tanimoto-dissimilarity* (Bajusz et al., 2015) between all generated molecules for a pocket. **Novelty** is the fraction of molecules different from those in the training set. **QED** (Bickerton et al., 2012) is a measure of drug-likeness. **SA** (Ertl & Schuffenhauer, 2009) estimates ease of synthesis of drug-like molecules. **Vina Score** is an estimate of binding affinity between ligand and target pocket calculated using the docking software QVina2 (Alhossary et al., 2015).

Scaffold-hopping results The main quantitative results for scaffold hopping are presented in Table 1 with distributions of key metrics in Figure 4. Our generated molecules have relatively high chemical diversity, despite the func-

tional groups being fixed in all samples. This indicates that our model can produce molecules of high scaffold/structural diversity. Our mean Vina score of -7.883 is impressive when considering that we often perform drastic topological changes and that the molecules in PDBBind are biased towards high affinity molecules. QED and SA scores are also competitive when compared to the test set and previous work with mean scores of 0.612 (Schneuing et al., 2022) and 0.664 (Adams & Coley, 2022). A graphical example of a DiffHopp output is provided in Figure 3 for a random target in the test set (PDB:6bqd) (Nuttinger et al., 2019).

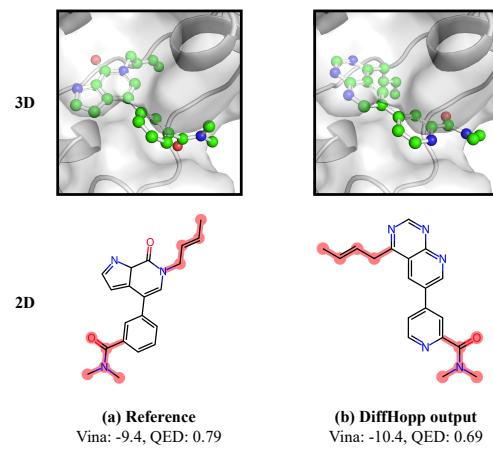


Figure 3: Model output for randomly chosen pocket (PDB:6bqd). Out of 10 generated samples, the best one as measured by Vina score was chosen. Functional groups are highlighted in red. DiffHopp produces more aromatic rings and the molecule has improved binding affinity.

Method	Connectivity (\uparrow)	Diversity (\uparrow)	Novelty (\uparrow)	QED (\uparrow)	SA (\uparrow)	Vina (kcal/mol, \downarrow)
DiffHopp	0.914 \pm 0.28	0.592 \pm 0.21	0.998 \pm 0.05	0.612 \pm 0.18	0.664 \pm 0.13	- 7.883 \pm 1.53
DiffHopp-EGNN	0.757 \pm 0.43	0.644 \pm 0.17	1.000 \pm 0.02	0.514 \pm 0.19	0.604 \pm 0.13	-7.240 \pm 1.47
GVP-inpainting	0.652 \pm 0.48	0.668 \pm 0.18	0.997 \pm 0.06	0.547 \pm 0.20	0.680 \pm 0.11	-7.552 \pm 1.77
EGNN-inpainting	0.793 \pm 0.41	0.667 \pm 0.18	0.999 \pm 0.03	0.467 \pm 0.20	0.644 \pm 0.11	-7.163 \pm 1.52
Test set	1.000 \pm 0.00	-	1.000 \pm 0.00	0.606 \pm 0.17	0.736 \pm 0.12	-8.767 \pm 1.92

Table 1: Mean and standard deviation of the common molecular metrics for the molecules from both the test set and the DiffHopp models. Furthermore, results using inpainting on molecule generation models are shown. Best metrics are in bold.

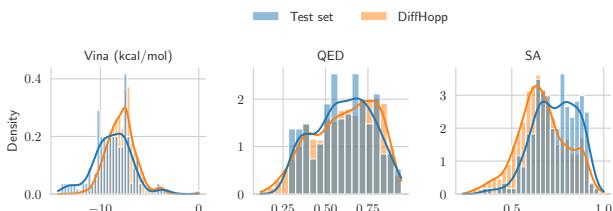


Figure 4: Distribution of selected metrics in the test set and the molecules generated with DiffHopp. The model’s samples are comparable in distribution to the test molecules and, while slightly less often than the test set, the model occasionally generates high scoring samples.

Comparison to scaffold-hopping via inpainting We compared DiffHopp to a general DiffSBDD-like (Schneuing et al., 2022) molecule generation model, trained with the same denoiser. Unlike DiffHopp, this inpainting model was trained without providing functional groups of a ligand as context. To sample scaffolds we fix the functional groups and perform sampling with inpainting (details in App. C). DiffHopp showcases clearly superior performance for connectivity, QED, and Vina scores than the inpainting model, while matching its performance in other metrics, barring diversity (Table 1). Consequently, our findings affirm that a custom scaffold-hopping model outperforms a repurposed general model via inpainting. The price to pay for the extra performance is the rigid definition of scaffold vs rest, which has to be chosen before training.

Ablation study of more powerful denoiser To test the effect of the more powerful GVP-denoiser, we conducted an ablation study (see Table 1), replacing the DiffHopp GVP-encoder with an E(3)-Equivariant Graph Neural Network (EGNN) (Satorras et al., 2021). Both models were tuned through extensive hyperparameter optimization.

Our ablation shows that switching from EGNN to GVP significantly improved connectivity, addressing a common problem in EGNN-based works (Schneuing et al., 2022), where low molecular connectivity due to small coordinate errors causes bond omissions in postprocessing. We believe this improvement is because GVP is a more expressive

model (Joshi et al., 2023), which in contrast to EGNN can also reason about angles. This was also reflected in training through reduced coordinate loss compared to the EGNN ablation (Appendix Fig. 5). Additionally, mean QED and Vina scores improved markedly, underlining GVP’s superiority. Surprisingly, however, the GVP resulted in lower connectivity for the inpainting models in our experiments.

Limitations Full atom representations were shown to improve pocket-conditioned diffusion modeling (Schneuing et al., 2022), however, training such was beyond the computational budget of this project. Future work could investigate whether full-atom representations allow generated scaffolds to better mediate protein-ligand interactions. A related issue is our definition of functional groups as any atom not in the scaffold, which may not capture key pharmacophoric properties contained in the original scaffold (e.g. oxygen bound to the ring in Figure 3).

Whilst the exact size of the medicinally relevant scaffold shape is uncertain, Hu & Bajorath (2010) found that for the majority of targets, between 5-49 structurally distinct scaffolds are available in public databases. Further work will analyse whether DiffHopp is able to enrich the diversity of chemotypes available for targeting a given protein and whether it generalises to scaffolds beyond those seen in the training set.

5. Conclusion

In this work, we have demonstrated that DiffHopp, an equivariant graph diffusion model, is highly capable of performing the medicinally important task of scaffold hopping to design molecules of potent activity whilst generating novel structures. We found that it outperforms generalist molecule diffusion models used via inpainting and that the expressivity of the denoiser correlated directly with high molecular connectivity. We would thus recommend future work to use more expressive architectures such as GVP. Code will be made available upon acceptance.

References

- Adams, K. and Coley, C. W. Equivariant shape-conditioned generation of 3d molecules for ligand-based drug design. *arXiv preprint arXiv:2210.04893*, 2022.
- Alhossary, A., Handoko, S. D., Mu, Y., and Kwoh, C.-K. Fast, accurate, and reliable molecular docking with quickvina 2. *Bioinformatics*, 31(13):2214–2216, 2015.
- Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- Baillif, B., Cole, J., McCabe, P., and Bender, A. Deep generative models for 3d molecular structure. *Current Opinion in Structural Biology*, 80:102566, 2023.
- Bajusz, D., Rácz, A., and Héberger, K. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of cheminformatics*, 7(1):1–13, 2015.
- Bemis, G. W. and Murcko, M. A. The properties of known drugs. 1. molecular frameworks. *Journal of medicinal chemistry*, 39(15):2887–2893, 1996.
- Bickerton, G. R., Paolini, G. V., Besnard, J., Muresan, S., and Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nature Chemistry*, 4(2):90–98, January 2012. doi: 10.1038/nchem.1243. URL <https://doi.org/10.1038/nchem.1243>.
- Birchall, K. and Gillet, V. J. Reduced graphs and their applications in chemoinformatics. *Chemoinformatics and computational chemical biology*, pp. 197–212, 2011.
- Böhm, H.-J., Flohr, A., and Stahl, M. Scaffold hopping. *Drug discovery today: Technologies*, 1(3):217–224, 2004.
- Corso, G., Stärk, H., Jing, B., Barzilay, R., and Jaakkola, T. Diffdock: Diffusion steps, twists, and turns for molecular docking. *arXiv preprint arXiv:2210.01776*, 2022.
- Elfwing, S., Uchibe, E., and Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018.
- Ertl, P. and Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1:1–11, 2009.
- He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- Hessler, G. and Baringhaus, K.-H. The scaffold hopping potential of pharmacophores. *Drug Discovery Today: Technologies*, 7(4):e263–e269, 2010.
- Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.
- Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, pp. 8867–8887. PMLR, 2022.
- Hu, Y. and Bajorath, J. Global assessment of scaffold hopping potential for current pharmaceutical targets. *Med-ChemComm*, 1(5):339–344, 2010.
- Hu, Y., Stumpfe, D., and Bajorath, J. Recent advances in scaffold hopping: miniperspective. *Journal of medicinal chemistry*, 60(4):1238–1246, 2017.
- Igashov, I., Stärk, H., Vignac, C., Satorras, V. G., Frossard, P., Welling, M., Bronstein, M., and Correia, B. Equivariant 3d-conditional diffusion models for molecular linker design. *arXiv preprint arXiv:2210.05274*, 2022.
- Isert, C., Atz, K., and Schneider, G. Structure-based drug design with geometric deep learning. *Current Opinion in Structural Biology*, 79:102548, 2023.
- Jing, B., Eismann, S., Suriana, P., Townshend, R. J., and Dror, R. Learning from protein structure with geometric vector perceptrons. *arXiv preprint arXiv:2009.01411*, 2020.
- Joshi, C. K., Bodnar, C., Mathis, S. V., Cohen, T., and Liò, P. On the expressive power of geometric graph neural networks. *arXiv preprint arXiv:2301.09308*, 2023.
- Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., and 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.
- Nuttinger, E., Gibbons, P., Eigenbrot, C., Davies, D. R., Maurer, B., Yu, C. L., Kiefer, J. R., Kuglstatter, A., Murray, J., Ortwinne, D. F., et al. Water molecules in protein–ligand interfaces. evaluation of software tools and sar comparison. *Journal of computer-aided molecular design*, 33: 307–330, 2019.
- O’Boyle, N. M., Banck, M., James, C. A., Morley, C., Vandermeersch, T., and Hutchison, G. R. Open babel: An open chemical toolbox. *Journal of cheminformatics*, 3(1): 1–14, 2011.

275 Ramachandran, P., Zoph, B., and Le, Q. V. Searching for
276 activation functions. *arXiv preprint arXiv:1710.05941*,
277 2017.

278 Rappé, A. K., Casewit, C. J., Colwell, K., Goddard III,
279 W. A., and Skiff, W. M. Uff, a full periodic table force
280 field for molecular mechanics and molecular dynamics
281 simulations. *Journal of the American chemical society*,
282 114(25):10024–10035, 1992.

283 Satorras, V. G., Hoogeboom, E., and Welling, M. E (n)
284 equivariant graph neural networks. In *International con-*
285 *ference on machine learning*, pp. 9323–9332. PMLR,
286 2021.

287 Schneuing, A., Du, Y., Harris, C., Jamasb, A., Igashov, I.,
288 Du, W., Blundell, T., Lió, P., Gomes, C., Welling, M., et al.
289 Structure-based drug design with equivariant diffusion
290 models. *arXiv preprint arXiv:2210.13695*, 2022.

291 Song, Y. and Ermon, S. Generative modeling by estimating
292 gradients of the data distribution. *Advances in neural*
293 *information processing systems*, 32, 2019.

294 Sun, H., Tawa, G., and Wallqvist, A. Classification of
295 scaffold-hopping approaches. *Drug discovery today*, 17
296 (7-8):310–324, 2012.

297 Tong, X., Liu, X., Tan, X., Li, X., Jiang, J., Xiong, Z., Xu,
298 T., Jiang, H., Qiao, N., and Zheng, M. Generative models
299 for de novo drug design. *Journal of Medicinal Chemistry*,
300 64(19):14011–14027, 2021.

301 Xie, W., Wang, F., Li, Y., Lai, L., and Pei, J. Advances
302 and challenges in de novo drug design using three-
303 dimensional deep generative models. *Journal of Chemical*
304 *Information and Modeling*, 62(10):2269–2279, 2022.

305 Yang, S.-Y. Pharmacophore modeling and applications in
306 drug discovery: challenges and recent advances. *Drug*
307 *discovery today*, 15(11-12):444–450, 2010.

308 Zheng, S., Lei, Z., Ai, H., Chen, H., Deng, D., and Yang,
309 Y. Deep scaffold hopping with multimodal transformer
310 neural networks. *Journal of cheminformatics*, 13:1–15,
311 2021.

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330 A. Training and sampling algorithm

331
332 We use the same training and sampling algorithms as in Hoogeboom et al. (2022) and Schneuing et al. (2022), which is a
333 slight adaptation of the original DDPM sampling (Ho et al., 2020). The main difference to Schneuing et al. (2022) is the
334 separation of coordinate and atom-type loss in the training.

335 Algorithm 1 Training algorithm

336
337 **repeat**
338 Sample \mathbf{z}_0, \mathbf{u} from training data
339 Subtract center of gravity of \mathbf{z}_0 from \mathbf{z}_0, \mathbf{u}
340 Sample $t \sim \mathcal{U}(0, \dots, T)$, $\epsilon_x \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\epsilon_h \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
341 Subtract center of gravity from ϵ_x
342 $\epsilon \leftarrow [\epsilon_x, \epsilon_h]$
343 $\mathbf{z}_t \leftarrow \sqrt{\bar{\alpha}_t} \mathbf{z}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon$
344 $[\epsilon'_x, \epsilon'_h] \leftarrow \varepsilon_\theta(\mathbf{z}_t, t, \mathbf{u})$
345 Take gradient descent step on $\nabla_\theta (\frac{1}{4} (\|\epsilon_x - \epsilon'_x\|^2 + \|\epsilon_h - \epsilon'_h\|^2))$
346 **until** convergence

347 Algorithm 2 Sampling algorithm

348
349 **Require:** context \mathbf{u}
350 Sample $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
351 **for** t in $T, T-1, \dots, 1$ **do**
352 **if** $t > 1$ **then**
353 Sample $\epsilon_x \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\epsilon_h \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
354 Subtract center of gravity from ϵ_x
355 $\epsilon \leftarrow [\epsilon_x, \epsilon_h]$
356 **else**
357 $\epsilon \leftarrow \mathbf{0}$
358 $\mathbf{z}_{t-1} \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}} (\mathbf{z}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \varepsilon_\theta(\mathbf{z}_t, t, \mathbf{u})) + \sigma_t \epsilon$
359 **return** \mathbf{z}_0

360 B. Loss curves

361 C. Performing scaffold-hopping with inpainting

362 Another approach to scaffold generation is via inpainting: Lugmayr et al. (2022) introduce an inpainting method for existing
363 diffusion models to condition their output on known parts. They demonstrate the potential and applicability of the technique
364 by using diffusion models pre-trained for image generation to inpaint images - filling in missing regions.
365

366 It is possible to view the scaffold hopping problem as an inpainting task - using a model trained on de-novo ligand generation,
367 it is possible to consider the scaffold as a missing region while providing the known functional groups of the molecule.
368

369 The inpainting method is based on the observation that each step in the reverse diffusion process $p_\theta(\mathbf{z}_{t-1} | \mathbf{z}_t)$ depends only
370 on \mathbf{z}_t . Thus, it is possible to change \mathbf{z}_t as long as the correct properties of the corresponding distribution are maintained
371 (Lugmayr et al., 2022). To create conditioned samples, it is possible to simply enforce the conditioning in the generative
372 process by replacing parts of the predicted \mathbf{z}_{t-1} with the correct $\mathbf{z}_{t-1}^{\text{known}}$. Formally, given a known \mathbf{z}_0 , a current \mathbf{z}_t and a
373 mask indicating the known parts m , we can define

$$\mathbf{z}_{t-1}^{\text{known}} \sim \mathcal{N}(\sqrt{\bar{\alpha}_{t-1}} \mathbf{z}_0, (1 - \bar{\alpha}_{t-1}) \mathbf{I}) \quad (2)$$

$$\mathbf{z}_{t-1}^{\text{unknown}} \sim \mathcal{N}(\mu_\theta(\mathbf{z}_t, t), \Sigma_\theta(\mathbf{z}_t, t)) \quad (3)$$

$$\mathbf{z}_{t-1} = m \odot \mathbf{z}_{t-1}^{\text{known}} + (1 - m) \odot \mathbf{z}_{t-1}^{\text{unknown}} \quad (4)$$

374 As the diffusion model attempts to harmonise the input as the diffusion process progresses, this should naturally result in the
375 model generating in-distribution samples with the desired known parts. The process of inpainting is shown in Figure 6.
376

377
378
379
380
381
382
383
384

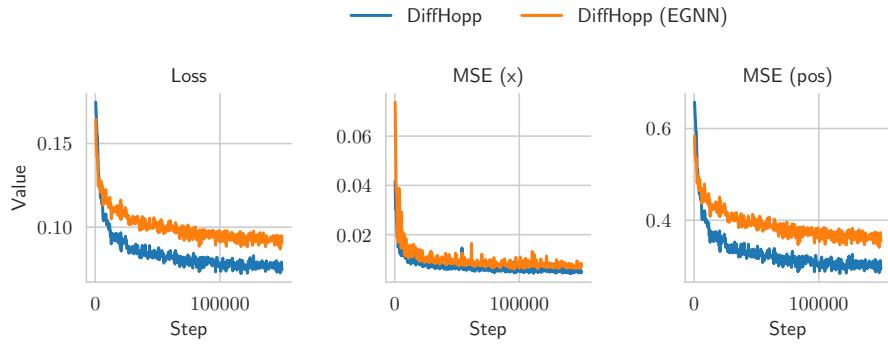


Figure 5: DiffHopp validation losses. Left: total reweighted loss. Middle: node features MSE. Right: Node coordinate MSE.

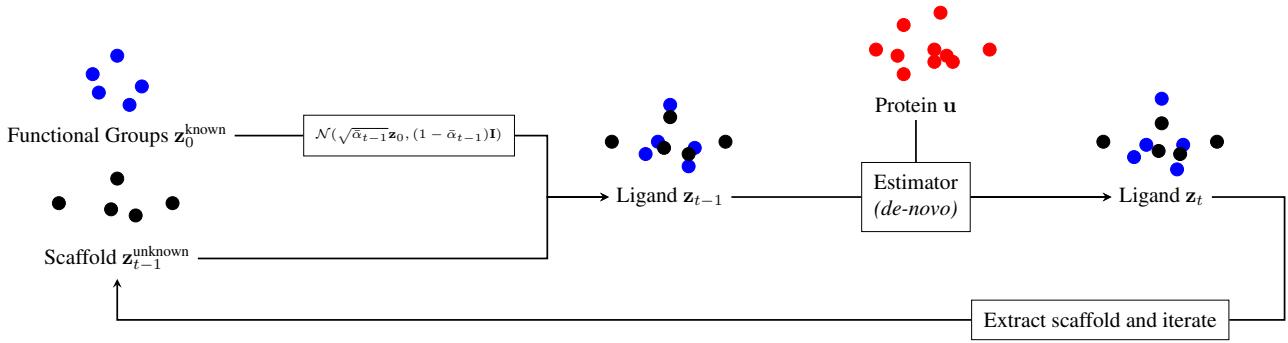


Figure 6: An overview of reverse diffusion process with inpainting. The ligand \mathbf{z}_{t-1} is constructed from known and unknown parts, and then iteratively updated.

However, Lugmayr et al. note that direct application of this method leads to locally harmonised results that struggle to incorporate the global context² (Lugmayr et al., 2022). They theorise that the model is limited in how much it can harmonise the sample \mathbf{z}_t at each step because it does not know about $\mathbf{z}_t^{\text{known}}$ when making the prediction for $\mathbf{z}_t^{\text{unknown}}$. They compensate for this by not directly following the reverse Markov chain in a linear fashion, but instead moving back and forth in the diffusion process to enable the model to properly incorporate the known parts. This movement is parameterised by j and r , where the jump length j indicates the length of each of the r resamples. An example of a repaint schedule is shown in Figure 7.

We extend the training procedure to support the training of diffusion models for de-novo ligand generation. In practice, this is done by restricting the context \mathbf{u} to contain only the protein pocket. Furthermore, \mathbf{z}_0 represents a complete ligand in the training procedure, not just a scaffold. We thus support training models that approximate $p_\theta(\mathbf{z}_0|\mathbf{u})$, where \mathbf{z}_0 is a ligand and \mathbf{u} is a protein pocket, similar to the setup of DiffSBDD (Schneuing et al., 2022).

D. Model details

This section details more specific architectural details. Following previous work (Schneuing et al., 2022), the Swish activation function with $\beta = 1$ (Ramachandran et al., 2017), defined as SiLU (Elfwing et al., 2018)

$$f(x) = x \cdot \text{sigmoid}(x) \quad (5)$$

is used for all non-linearities, except where explicitly detailed otherwise.

²In the paper, the authors describe a case where inpainting the face of a dog leads to a furry texture, not to a face.

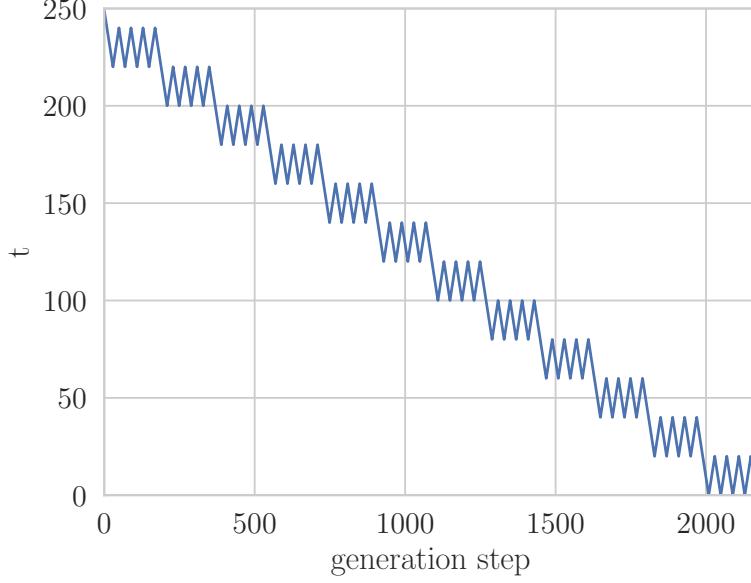


Figure 7: An example repainting schedule with $T = 250$, $r = 5$ and $j = 20$. Note that resampling always happens r times with length j .

D.1. Encoding and Decoding Functions for the Graph Embedding

The learnable functions ϕ_z , ϕ_g , ϕ_p which encode the respective node features for the shared graph are implemented as Multi Layer Perceptron (MLP). Each function consists of two linear layers, the former mapping from n to $2n$ features and the latter from $2n$ to m features, with one non-linearity in between. n denotes the number of original features and m the size of the joint embedding space (without time appended).

ϕ_{out} is a 2-layer MLP with the inverted structure of ϕ_z .

D.2. EGNN

The learnable functions ϕ_e , ϕ_h , ϕ_{att} and ϕ_x are used in each EGNN layer (Satorras et al., 2021).

Given a value for hidden features h , ϕ_e is a two-layer MLP mapping from the features of the input to h and then from h to h using two linear layers, with a non-linearity after each of them. The final output is divided by a normalisation factor $C = 100$, to prepare for the sum aggregation.

The learnable function ϕ_h is another two-layer MLP with a hidden layer size h with a single non-linearity between both layers.

The attention mechanism ϕ_{att} is defined as a single linear layer with a single output, followed by a sigmoid function.

Finally, the position update $\phi_x = R \tanh(\phi_{x'})$, where $R = 15$ limits the range of movement. $\phi_{x'}$ is a 3-layer MLP with hidden sizes h , where the last layer maps to a scalar and has no bias.

D.3. GVP-GNN

This section describes the architecture of the constructed GNN using GVPs (Jing et al., 2020) in more detail. σ_g is a sigmoid function in all GVPs mentioned. Unless explicitly stated otherwise, σ is the SiLU activation function and σ^+ the identify function.

The inputs of the GVP are nodes with scalar features s , representing the embedded input scalars in the graph space, and no vector features. The input edges simply consist of a normed direction vector and the distance between the two respective nodes, as detailed in the paper.

495 Both edge attributes and node attributes are passed through embedding layers. The edges are embedded in a two step process,
 496 first normalising the inputs using a layer normalisation (Ba et al., 2016) and then passing them through a GVP with σ , σ^+
 497 being the identity function, which outputs a scalars of hidden size $h/2$ and a single vector.

498 The nodes are embedded in a similar fashion, however outputting $h/2$ scalars and $h/2$ vectors, leading to h features in total.
 499

500 The message passing layers are defined similar to the EGNN:

$$\begin{aligned} \mathbf{m}'_{vw} &= \phi_e(\mathbf{h}_v, \mathbf{h}_w, \mathbf{e}_{vw}) \\ \mathbf{m}'_v &= \sum_{w \in \mathcal{N}_v} \tilde{e}_{vw} \mathbf{m}'_{vw} \\ \mathbf{h}'_v &= \phi_h(\mathbf{h}_v, \mathbf{m}'_v) \end{aligned} \quad (6)$$

501 where $\tilde{e}_{vw} = \phi_{\text{att}}(\mathbf{m}_{vw})$ is an attention mechanism to learn a soft estimations of the edges, similar to the EGNN.
 502

503 ϕ_e is a composition of three GVPs with hidden sizes ($h/2, h/2$) and the last one having σ has identity function. ϕ_{att} is a
 504 single GVP mapping to a single scalar with σ being the sigmoid activation function. The final output is normalised by
 505 $C = 100$, similar to the EGNN.

506 $\phi_h(\mathbf{h}_v, \mathbf{m}'_v) = \text{norm}(\mathbf{h}_v + \phi'_h(\text{norm}(\mathbf{h}_v + \mathbf{m}'_v)))$, a residual (He et al., 2016) architecture with ϕ'_h being a composition of
 507 two GVPs with ($h/2, h/2$) has input, hidden and output size. The last layer again has σ as identity function. norm denotes a
 508 layer normalisation(Ba et al., 2016). The norm is not learned for vectors.
 509

510 E. Hyperparameter tuning and settings

511 We considered the following hyperparameter settings. The best model was chosen by taking the model with the lowest
 512 validation set loss. Hidden Features denote the number of features for each node between the GNN Message Passing layers.
 513 Embedding Size denotes the size of the node embeddings in the input graph. GNN Layers denotes the number of message
 514 passing layers in the GNN architecture.
 515

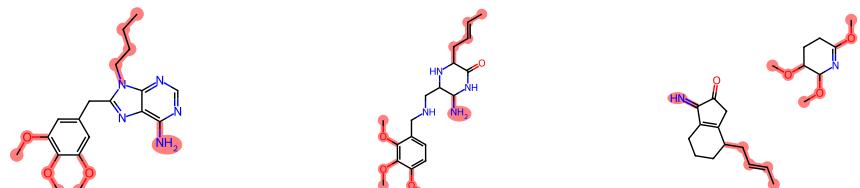
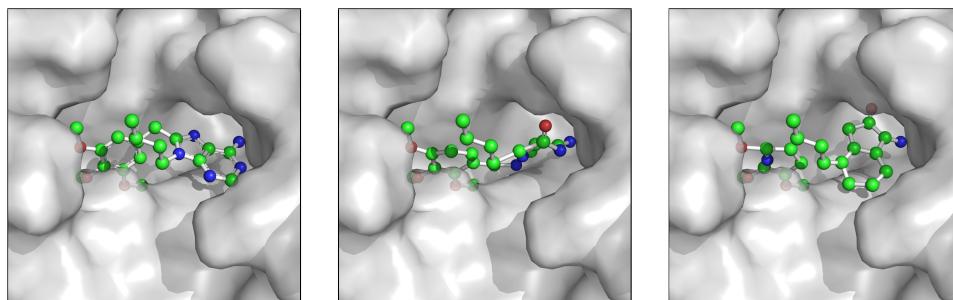
516 Parameters	517 Search space	518 DiffHopp-EGNN	519 EGNN-inp.	520 DiffHopp	521 GVP-inp.
522 Static					
523 Batch Size	524	525 32			
526 Diffusion steps T	527	528 500			
529 Number of steps	530	150000			
Seed		1			
531 Tuned					
Attention mechanism	True, False	False	False	True	False
Hidden Features	32, 64, 128, 256	64	256	256	256
Embedding Size	32, 64, 128, 256	128	256	64	32
Learning rate	5e-3, 2e-3, 1e-3, 5e-4	1e-3	1e-3	5e-4	5e-4
GNN Layers	4,5,6,7	7	7	7	6

532 F. Further examples

533 To help understand the merits and shortcomings of DiffHopp, we cherrypicked three examples below that illustrate typical
 534 successes and shortcomings of DiffHopp (Figure 8).

550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585

PDB:6olx



(a) Reference
Vina: -8.0, QED: 0.65

(b) Connected sample
Vina: -8.5, QED: 0.69

(c) Unconnected sample
Vina: -8.4, QED: 0.46

586 Figure 8: Cherry-picked samples to illustrate capabilities and shortcomings. It can be seen that the model is able to produce
587 scaffolds that perform better on key metrics. However, not all samples are connected. In the chemical structure, functional
588 groups are highlighted in red.

589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604