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

Jos Torge [1]   Charles Harris [1]   Simon V. Mathis [1]   Pietro Lió [1]

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
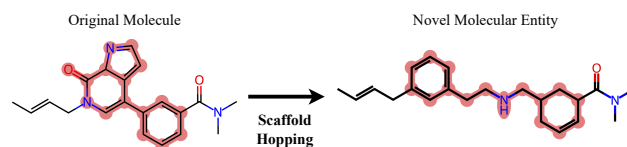

Figure 1: Scaffold hopping: Scaffold atoms highlighted in red. The scaffold holds the functional groups in place for binding. Scaffold hopping refers to interchanging scaffolds while leaving the functional groups unchanged. Compared to inpainting, scaffold hopping uses a fixed definiton of scaffold-vs-rest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

et al., 2022) and molecular docking (Corso et al., 2022).

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-conditionend molecule generation via diffusion (Schneuing et al., 2022).

[1] Department of Computer Science and Technology, University of Cambridge. Correspondence to: Jos Torge <jst59@cam.ac.uk>, Charles Harris <cch57@cam.ac.uk>, Simon Mathis <simon.mathis@cl.cam.ac.uk>.

2. Background and Related Work

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

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

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

3. Methods

Dataset We train our model on 19,378 protein-ligand complexes from PDBBind, filtered for QED ≥ 0.3 and split as in Corso et al. (2022). From these complexes, we define the Murko-Bemis scaffold (Bemis & Murcko, 1996) for each ligand using RDKit[1]. Here, we treat atoms not in the scaffold as functional groups.

---
[1] www.rdkit.org

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

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

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

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

where $[\boldsymbol{\epsilon}_x, \boldsymbol{\epsilon}_h] = \boldsymbol{\epsilon}$ and $[\boldsymbol{\epsilon}'_x, \boldsymbol{\epsilon}'_h] = \varepsilon_\theta(\mathbf{z}_t, t, \mathbf{u})$ denote the true and predicted noise respectively. Our sampling procedure follows previous work on equivariant diffusion models (Hoogeboom et al., 2022) and is given in Algorithm 2 (see appendix).
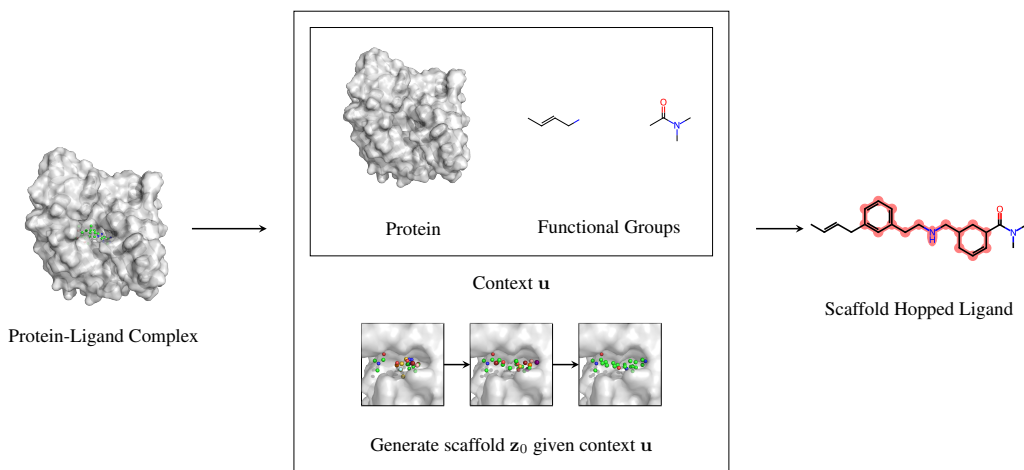
Figure 2: Visual abstract. Given a protein-ligand complex, an equivariant diffusion model is used to sample a scaffold from a scaffold distribution conditioned on functional groups and protein pocket. The resulting scaffold is merged with the functional groups to lead a scaffold hopped ligand.

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) (Nittinger et al., 2019).

(a) Reference
Vina: -9.4, QED: 0.79

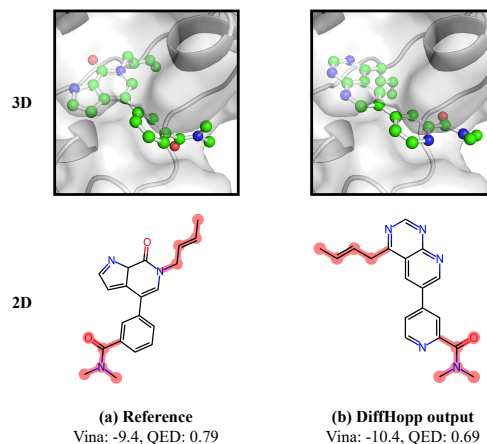**(b) DiffHopp output**
Vina: -10.4, QED: 0.69

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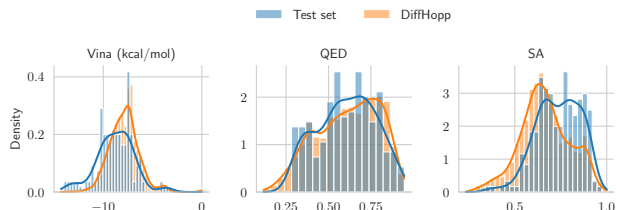

Figure 4: Distribution of selected metrics in the test set and the molecules generated with DiffHopp. The model's samples are comparable in distribution to the test molecules and, while slightly less often than the test set, the model occasionally generates high scoring samples.

We perform dimensionality reduction of the molecular scaffold fingerprints generated by DiffHopp versus the training dataset (Appendix Figure 8) and observe that DiffHopp has successfully learned to generate diverse scaffolds that match the training set for diversity chemotypes.

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.

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 seem 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.

Acknowledgements

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. *MedChemComm*, 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.

Kearsley, S. K., Sallamack, S., Fluder, E. M., Andose, J. D., Mosley, R. T., and Sheridan, R. P. Chemical similarity using physiochemical property descriptors. *Journal of Chemical Information and Computer Sciences*, 36(1):118–127, 1996.

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.

McInnes, L., Healy, J., and Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Nittinger, E., Gibbons, P., Eigenbrot, C., Davies, D. R., Maurer, B., Yu, C. L., Kiefer, J. R., Kuglstatter, A., Murray, J., Ortwine, 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.

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

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

Satorras, V. G., Hoogeboom, E., and Welling, M. E (n) equivariant graph neural networks. In *International conference on machine learning*, pp. 9323–9332. PMLR, 2021.

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

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

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

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

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

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

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

A. Training and sampling algorithm

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

Algorithm 1 Training algorithm
---
repeat
    Sample $\mathbf{z}_0, \mathbf{u}$ from training data
    Subtract center of gravity of $\mathbf{z}_0$ from $\mathbf{z}_0, \mathbf{u}$
    Sample $t \sim \mathcal{U}(0, .., T), \boldsymbol{\epsilon}_x \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \boldsymbol{\epsilon}_h \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    Subtract center of gravity from $\boldsymbol{\epsilon}_x$
    $\boldsymbol{\epsilon} \leftarrow [\boldsymbol{\epsilon}_x, \boldsymbol{\epsilon}_h]$
    $\mathbf{z}_t \leftarrow \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}$
    $[\boldsymbol{\epsilon}'_x, \boldsymbol{\epsilon}'_h] \leftarrow \varepsilon_\theta(\mathbf{z}_t, t, \mathbf{u})$
    Take gradient descent step on $\nabla_\theta(\frac{1}{4}(\|\boldsymbol{\epsilon}_x - \boldsymbol{\epsilon}'_x\|^2 + \|\boldsymbol{\epsilon}_h - \boldsymbol{\epsilon}'_h\|^2))$
until convergence

Algorithm 2 Sampling algorithm

Require: context $\mathbf{u}$
    Sample $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    for t in $T, T-1, \dots, 1$ **do**
      if $t > 1$ **then**
        Sample $\boldsymbol{\epsilon}_x \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \boldsymbol{\epsilon}_h \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
        Subtract center of gravity from $\boldsymbol{\epsilon}_x$
        $\boldsymbol{\epsilon} \leftarrow [\boldsymbol{\epsilon}_x, \boldsymbol{\epsilon}_h]$
      else
        $\boldsymbol{\epsilon} \leftarrow \mathbf{0}$
      end if
      $\mathbf{z}_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}}(\mathbf{z}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(\mathbf{z}_t, t, \mathbf{u})) + \sigma_t\boldsymbol{\epsilon}$
    end for
    return $\mathbf{z}_0$

B. Loss curves

C. Performing scaffold-hopping with inpainting

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

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

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 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 (Lugmayr et al., 2022). To create conditioned samples, it is possible to simply enforce the conditioning in the generative 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 mask indicating the known parts m, we can define

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

$$\mathbf{z}_{t-1}^{\text{unknown}} \sim \mathcal{N}(\boldsymbol{\mu}_\theta(\mathbf{z}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{z}_t, t)) \tag{3}$$

$$\mathbf{z}_{t-1} = m \odot \mathbf{z}_{t-1}^{\text{known}} + (1 - m) \odot \mathbf{z}_{t-1}^{\text{unknown}} \tag{4}$$
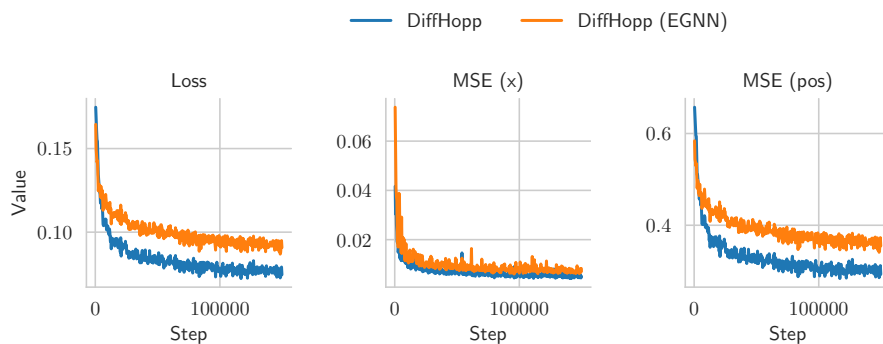
Figure 5: DiffHopp validation losses. Left: total reweighted loss. Middle: node features MSE. Right: Node coordinate MSE.
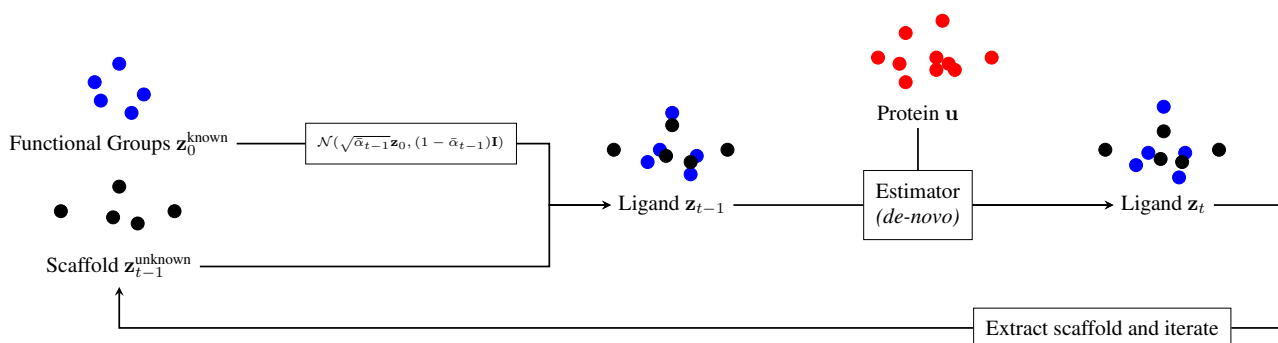

Figure 6: An overview of reverse diffusion process with inpainting. The ligand $\mathbf{z}_{t-1}$ is constructed from known and unknown parts, and then iteratively updated.

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

However, Lugmayr et al. note that direct application of this method leads to locally harmonised results that struggle to incorporate the global context[2] (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)

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

---

[2]In the paper, the authors describe a case where inpainting the face of a dog leads to a furry texture, not to a face.
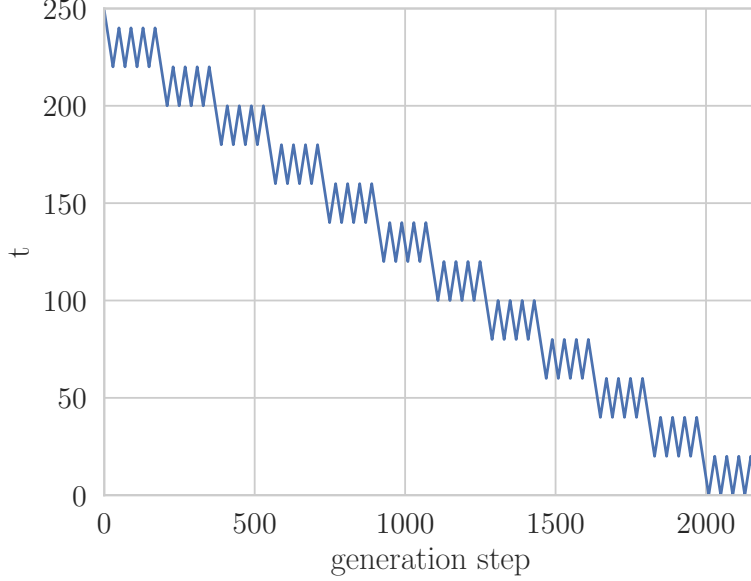
Figure 7: An example repainting schedule with $T = 250$, $r = 5$ and $j = 20$. Note that resampling always happens r times with length j.

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

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 $\mathbf{s}$, 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.

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

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

The message passing layers are defined similar to the EGNN:

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

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.

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 single GVP mapping to a single scalar with σ being the sigmoid activation function. The final output is normalised by $C = 100$, similar to the EGNN.

$\phi_h(\mathbf{h}_v, \mathbf{m}'_v) = \text{norm}(\mathbf{h}_v + \phi'_h(\text{norm}(\mathbf{h}_v + \mathbf{m}'_v)))$, a residual (He et al., 2016) architecture with ϕ'_h being a composition of 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 layer normalisation(Ba et al., 2016). The norm is not learned for vectors.

E. Hyperparameter tuning and settings

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

Parameters	Search space	DiffHopp-EGNN	EGNN-inp.	DiffHopp	GVP-inp.
Static					
Batch Size		32			
Diffusion steps T		500			
Number of steps		150000			
Seed		1			
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

F. Scaffold clustering

Figure 8 shows a dimensionality reduction plot for the scaffolds from the training dataset and those generated using DiffHopp. For each molecule, we select the Murko-Bemis scaffold (Bemis & Murcko, 1996) and calculate its fingerprints using RDKit's topological fingerprints (Kearsley et al., 1996), dimensionality reduction is then performed with UMAP (McInnes et al., 2018). We observe that DiffHobb is able to generate diversity scaffolds for a variety of molecules with varying functional groups.
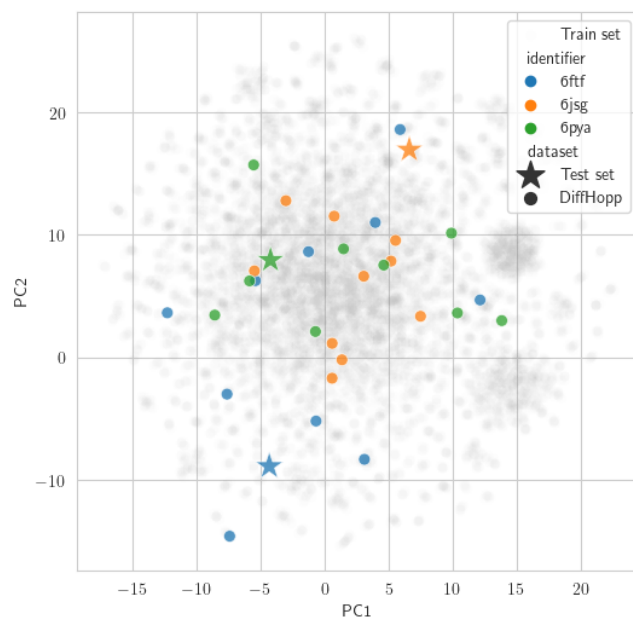
Figure 8: UMAP dimensionality reduction of molecular scaffolds from various sources. Grey: scaffolds in the PDBBinding training dataset. Three molecualrs are chosen at random from the test set (PDB 6FTF, 6JSG and 6PYA in blue, orange and green respectively) and the reduced mapping for the original scaffold (large star) and DiffHopp generated scaffolds (small dot) are shown. DiffHopp is able to generate a large diversity of scaffolds, regardless of the original chemotype specified by the functional groups in the test set molecule.

G. Further examples

To help understand the merits and shortcomings of DiffHopp, we cherrypicked three examples below that illustrate typical successes and shortcomings of DiffHopp (Figure 9).

PDB:6olx

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

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

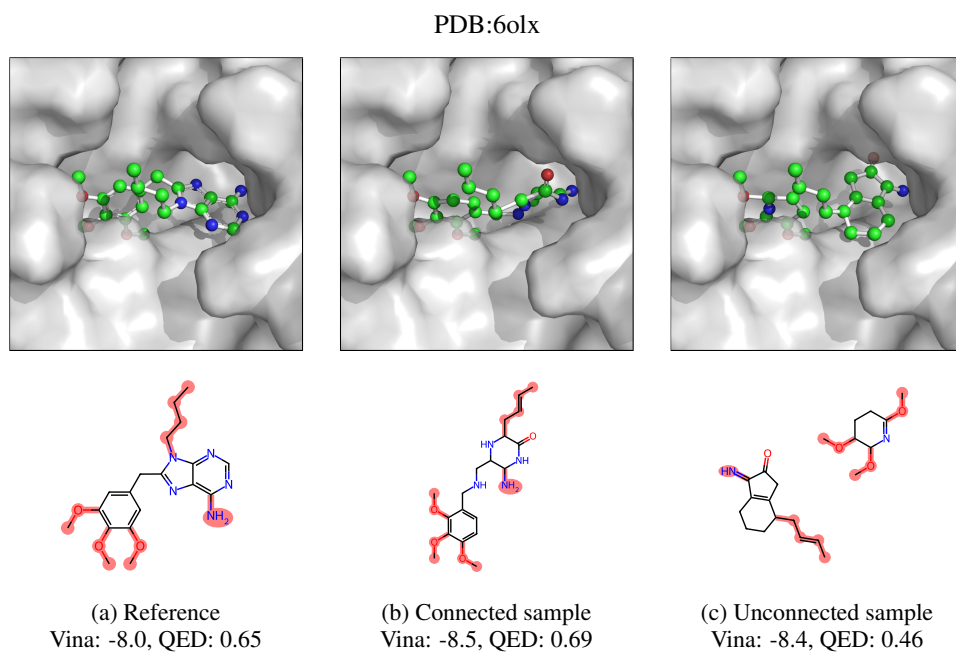(c) Unconnected sample
Vina: -8.4, QED: 0.46

Figure 9: Cherry-picked samples to illustrate capabilities and shortcomings. It can be seen that the model is able to produce scaffolds that perform better on key metrics. However, not all samples are connected. In the chemical structure, functional groups are highlighted in red.