FLOW-BASED FRAGMENT IDENTIFICATION VIA CONTRASTIVE LEARNING OF BINDING SITE-SPECIFIC LATENT REPRESENTATIONS

Rebecca M. Neeser[1], **Ilia Igashov**[1], **Arne Schneuing**[1],
Micheal Bronstein[1,2,3], **Philippe Schwaller**[1,4], **Bruno Correia**[1]

[1]Ecole Polytechnique Fédérale de Lausanne, Switzerland
[2]Oxford University, UK
[3]VantAI, USA
[4]National Centre of Competence in Research (NCCR) Catalysis EPFL, Switzerland
{rebecca.neeser,philippe.schwaller,bruno.correia}@epfl.ch

ABSTRACT

Fragment-based drug design is a promising strategy leveraging the binding of individual fragments, potentially yielding ligands with multiple key interactions, surpassing the efficiency of full ligand screening. The initial step of fragment identification remains challenging, as fragments often bind weakly and non-specifically. We propose a protein-fragment encoder, a new contrastive learning approach that captures protein-fragment interactions. Its latent space allows to perform virtual screening as well as generative design. In the latter case, fragment embeddings are generated conditioned on the protein surface. Our method locates protein-fragment interactions with high sensitivity and can be directly applied to virtual screening for which we observed competitive fragment recovery rates. The generative method outperforms common methods such as virtual screening providing a valuable starting point for fragment hit discovery. Together, these approaches contribute to advancing fragment identification and could provide valuable tools for fragment-based drug discovery. All the code and data can be found on https://github.com/rneeser/LatentFrag.

1 INTRODUCTION

Hit identification remains a critical challenge in drug discovery, despite advances in screening techniques and computational tools (Jalencas et al., 2024; Hasselgren & Oprea, 2024). High-throughput screening (HTS) struggles to explore vast chemical space efficiently and often fails to identify strong starting points for drug design (Edfeldt et al., 2011). Fragment-Based Drug Design (FBDD) offers a promising alternative, leveraging smaller fragments with higher ligand efficiency that can be combined to form potent ligands (Congreve et al., 2008; Hubbard & Murray, 2011; Yu et al., 2020).

Recent machine learning (ML) approaches enable rapid exploration of chemical space, yet many FBDD methods do not account for a protein structure, rely on prior knowledge of fragment hits (McCorkindale et al., 2022), or depend on molecular docking[1], which often struggles to localize fragment binding within protein pockets (Bian & Xie, 2018; Marchand & Caflisch, 2018). Structure-Based Drug Design (SBDD) incorporates geometric information but frequently generates unrealistic or synthetically inaccessible molecules when designing full ligands (Buttenschoen et al., 2024; Yang et al., 2024). While FBDD constrains chemical space using known fragments, challenges remain in merging, linking, and growing fragments into complete ligands (Imrie et al., 2020; Neeser et al., 2023; Igashov et al., 2024; Ferla et al., 2025).

To address these limitations, we introduce a novel structure-based fragment screening approach that learns a protein-fragment representation through contrastive training. Our encoder maps protein

[1]Docking is a computational technique used to predict the binding pose of a molecule within a pocket.

surfaces and molecular fragments into a shared latent space, capturing not just fragment chemistry but also binding interactions. Inspired by Igashov et al. (2022), who performed ligand and fragment screening based on pocket representations but using a model trained in protein-protein interactions, our model is specifically tailored for fragment screening and does not depend on the availability of fragments in crystal structures. Additionally, we endow our encoder with a generative framework for fragment identification. We use a flow matching-based model to sample fragment embeddings conditioned on the protein pocket, combined with a fragment library that is queried based on the sampled embeddings. This approach does not require a decoder, ensures chemically realistic fragments, and allows to flexibly change libraries without retraining.

To summarize, our contributions are the following:

- **Novel Protein-Fragment Contrastive Learning:** We introduce a protein-fragment encoder trained in a contrastive fashion that jointly learns representations of protein surfaces and molecular fragments in a shared latent space. Our approach captures aspects of interaction while maintaining chemical relevance through fragment similarity-based penalties.
- **Structure-Based Fragment Identification:** We propose a flow matching framework for fragment identification that operates directly in the protein-fragment latent space. Unlike existing methods, our approach is explicitly conditioned on protein structure and guarantees chemically valid outputs through library-based sampling.
- **Evaluation Framework:** By assessing both "hard" recovery metrics and "soft" pharmacophoric similarity, we provide insights into the model's ability to identify meaningful fragment hits. Our analysis demonstrates improvements over virtual screening baselines while highlighting current limitations in exact fragment matching.

2 METHODS

2.1 PROTEIN-FRAGMENT CONTRASTIVE LEARNING

The protein-fragment encoder is trained contrastively and is designed to produce expressive latent embeddings for both fragments and protein surfaces. Thus, the latent vectors capture critical features for binding interactions and are uniquely suited to the task of fragment identification.

Training involves maximizing the cosine similarity between embeddings of fragments and nearby protein surface points (positive examples), while minimizing the similarity to other surface points elsewhere on the protein (negative examples). Negative examples are selected to include diverse protein surface geometries from the pocket—convex, concave, and random regions—to prevent overfitting to pocket-specific features. The surface curvature is predicted on the fly by a concurrently trained classifier. Protein surface embeddings are parametrized by a geodesic convolutional neural network similar to dMaSIF (Sverrisson et al., 2021) while fragments are processed by a graph transformer (Dwivedi & Bresson, 2020; Vignac et al., 2022; Igashov et al., 2023) (Appendix B.1).

To ensure chemical relevance, a fragment similarity penalty (FSP) is incorporated via a hinge loss. This loss discourages molecules with low Tanimoto similarity from having similar embeddings. An additional classification loss is used to train the model to predict the type of non-covalent interaction (NCI), if present, that each protein surface point can engage in, integrating interaction-specific information. Notably, the relative positions of the molecules are only used to select positive and negative examples during training, and do not directly influence the embeddings as fragments are represented as 2D graphs. This design promotes position-agnostic representations of fragment and surface features, enabling their broad applicability in fragment-based drug design (for details see Appendix B).

2.2 FRAGMENT IDENTIFICATION VIA FLOW MATCHING

We employ a generative modeling approach using flow matching (Lipman et al., 2022), representing proteins as surface point clouds and ligands as coarse fragment graphs. Protein surface points are featurized by latent vectors and ligand nodes representing fragments are defined by a latent embedding (fragment type) and the arithmetic mean of their coordinates (centroid). The latent vectors

are learned embeddings from our protein-fragment encoder. A schematic overview of the fragment identification process is shown in Figure 2A, and the neural network is illustrated in Figure C.1.

Flow matching is a generative modeling approach that learns to map noise to structured data by matching probability flows, enabling efficient sampling from complex distributions. Two flows are employed: a spherical flow for the latent fragment embeddings, which assumes a unit sphere prior, and a Euclidean flow for the centroids, with a Gaussian prior. This method avoids the need for a decoder by selecting fragments from a precomputed library based on cosine similarity of fragment embeddings, ensuring chemical plausibility of the resulting fragments. This facilitates exchanging or expanding fragment library even after sampling. We use a library with 41,224 unique fragments extracted from PDB (Appendix D). More information on the generative framework is in Appendix C.

3 Results & Discussion

3.1 Latent Representation

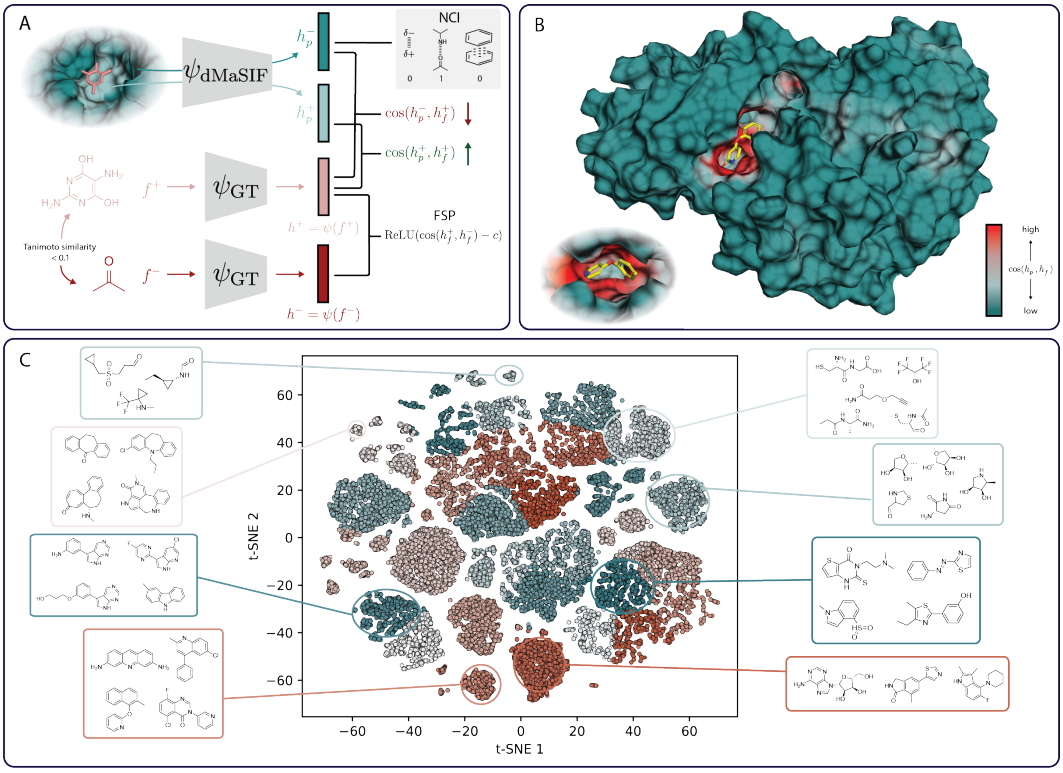


Figure 1: *A:* Encoder pipeline, maximizing the cosine similarity $\cos(\cdot)$ between fragment and close pocket surface embeddings while minimizing the similarity between negative pairs. Additionally, a fragment similarity penalty (FSP) and loss on classifying non-covalent interactions (NCI) from protein surfaces are incorporated. *B:* Protein surface colored by cosine similarity to the fragment (PDB ID: 6Q4I). *C:* t-SNE plot of the fragment library in latent space with representative molecules.

To assess whether our encoder captures protein-fragment interaction patterns, we evaluated its embeddings based on several key metrics. Specifically, we examined whether the binding region could be recovered, necessitating that the fragments of both the protein and the ligand exhibit similarity. This evaluation was conducted for both the entire surface and the pocket surface alone (Table F.4). The high EF_1 (enrichment factor in the 1st percentile) of 22.85 for the whole surface demonstrates that the model is capable of clearly distinguishing the pocket from the rest of the surface and a EF_1 of 2.28 for the pocket only highlights that even in the pocket itself the model chooses binding surface points twice as likely as non-interacting areas. The area under the precision recall curve (AUPR) of 0.39 represents a significant improvement over the baseline of 0.16 (fraction of interacting vs. non-

interacting points), further indicating that the model is able to prioritize surface points. These results consistently demonstrate that the model successfully distinguishes interacting from non-interacting surfaces and can identify regions of fragment binding.

In addition to the quantitative assessment, Figure 1B illustrates how our embeddings localize interacting surfaces with relatively high sensitivity, with the highest similarity observed between the binding fragment and its corresponding protein region. This "hotspot" can be found without any information on relative position of the two molecules. This indicates that the combination of these two latent representations captures aspects of the underlying interaction and suggests that it can be used for downstream applications such as virtual screening. We further assess the latent space of the encoder using t-SNE dimensionality reduction demonstrating a structured and interpretable embedding space (Figure 1C). Compared to standard `RDKit` fingerprints (Figure F.5), our embeddings form distinguishable clusters, which correspond to fragments with similar chemical profiles. Further results can be found in Appendix F.1.

3.2 Fragment Identification

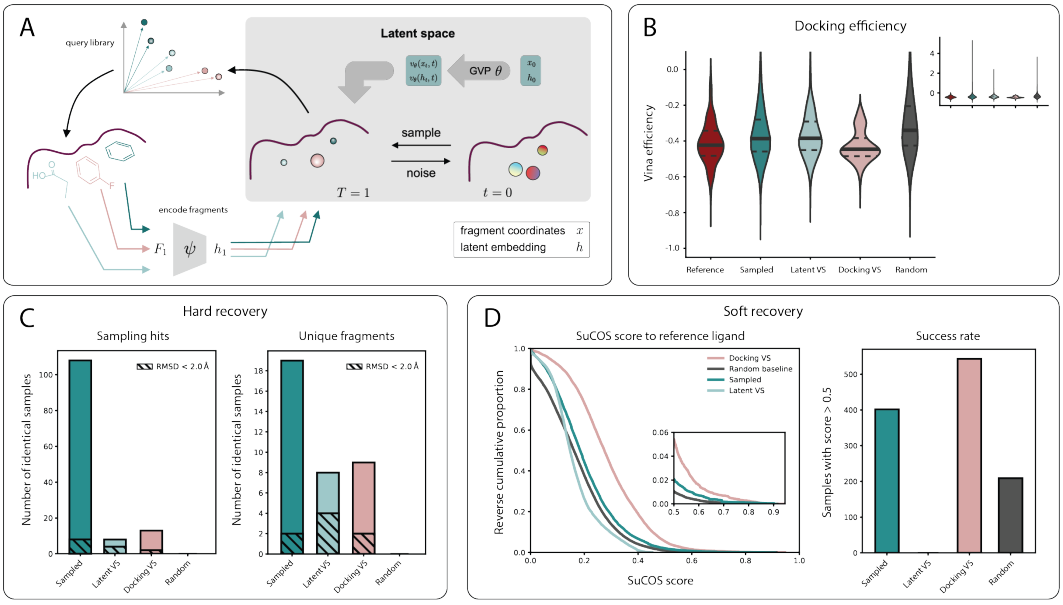


Figure 2: *A:* Generative pipeline starting with encoding and the flow matching model operating in latent space. *B:* Distributions of docking efficiency (docking score normalized by the number of heavy atoms). *C:* Hard recovery metrics based on both number of sampling hits (*left*) (sampled fragments recovering the ground truth) and unique fragment recovery (*right*) (number of unique recovered fragments). The RMSD refers to the distance between centroids. *D:* Soft recovery metrics based on the SuCOS score relative to the full reference ligand are shown as reverse cumulative proportion (*left*) and the success rate with a threshold of 0.5 (*right*).

A straightforward application of our learned latent space is virtual screening (VS), focusing on its ability to identify relevant fragments by similarity to the target pocket surface. However, a natural extension of this approach dubbed Latent VS is the use of generative modeling, which enables the sampling of novel fragment embeddings directly in the learned space. We compare these methods to VS based on docking scores (Docking VS) and a random baseline. For detailed information on evaluation and metrics we refer to Appendix E.2.

Figures 2C and 2D show results for hard and soft recovery. Hard recovery quantifies exact matches to reference fragments and soft recovery tries to assess the overlap in terms of pharmacophoric patterns to the reference. While hard recovery rates of unique fragments were not particularly high, generative modeling consistently outperformed latent and docking VS (recovery rates of 4.33%, 0.02% and 2.05%, respectively). Random sampling did not recover anything. The generative approach is able to find the correct fragments multiple times resulting in a much increased sampling hit rate compared

to latent or docking VS (0.554%, 0.001% and 0.130% respectively). Experimental fragment screen hit rates often vary between 1% and 2%, and are thus slightly higher than our fragment recovery rate, but at the expense of time and resources (Jalencas et al., 2024). These findings demonstrate the potential of our method to obtain initial fragment hits and narrow further experimental search. We further evaluated soft recovery as similarity to the complete reference ligands using the shape and pharmacophore dependent SuCOS score (Leung et al., 2019). Figure 2D shows that generative samples clearly improved over the random baseline by having a bigger fraction of high-scoring molecules that are thus similar to the reference geometry and chemistry. Interestingly, compared to the poor results on true recovery, docking VS fares better than our generative approach on the soft metric with increased number of hits. The higher number of atoms on average per fragment compared to generative sampling (19 and 13 respectively, see Figure F.7) as well as more realistic placement in the pocket might influence this outcome. One major advantage of our approach compared to docking VS is speed: docking the full library to one single target takes approximately 28 hours while generative sampling takes around 10 minutes only. On top of that, once sampled, different libraries or library expansions can be tested retrospectively while for docking this further increases computational cost.

Docking efficiency was evaluated as a proxy for binding affinity (Figure 2). We normalize by number of heavy atoms as the molecule size can have a significant impact especially when dealing with such small structures. While generative samples showed weaker scores than the reference fragments, they outperformed the random baseline. Fragments obtained through docking VS unsurprisingly outperform all other methods as they were filtered based on docking scores.

Lastly, Figure 3 shows three selected samples from our generative pipeline highlighting the ability to recover sensible poses even with our constrained docking approach and match interaction profiles from the reference ligands. The first example recovers the full reference ligand and, given the close overlap, will likely recover also the hydrogen bonds. The second sample has a high SuCOS score to the reference and visually exhibits close overlap with the reference. Lastly, when selecting by docking efficiency, we observe recovering of π-π stacking. For additional results we refer to Appendix F.2.

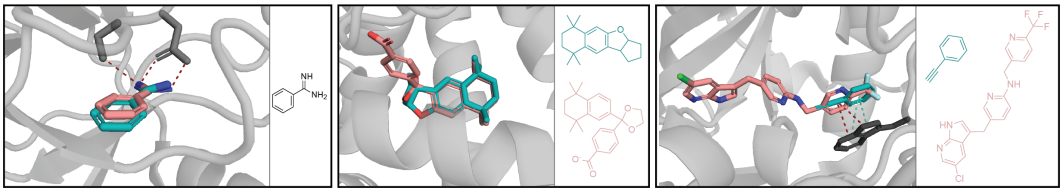


Figure 3: Selected generated samples (*teal*) compared to the reference ligand (*light red*) with corresponding interactions (*red lines*). *Left:* Sample recovering the reference. *Middle:* Sample with high SuCOS score. *Right:* Sample with low docking efficiency and predicted π-π-stacking (*blue lines*).

4 Conclusions

We introduced a novel method that creates a representation capturing key properties of protein-fragment interactions, enabling both virtual screening and generative fragment identification. Our protein-fragment encoder is trained on protein surfaces and molecular fragments in a contrastive fashion encoding them into a joint latent space. The flow matching method correspondingly operates in this latent space sampling both fragment embeddings and their centroids. Our generative approach demonstrated successful fragment recovery, suggesting its potential for providing initial hit hypotheses for experimental validation.

Virtual screening based on latent fragment embeddings performed on par with docking-based screening but required significantly less computation, while our generative method is even more cost-effective. Furthermore, our method offers the added advantage of flexible choice and expansion of the fragment library even after sampling, increasing the chemical space that is accessed and potentially increasing hit rates. Beyond fragment identification, our surface embeddings could be explored for binding site prediction, addressing the current limitation requiring prior knowledge on the binding pocket.

ACKNOWLEDGMENTS

R.M.N. thanks VantAI (USA) for their support and helpful feedback. I.I. has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 945363. M.B. is partially supported by the EPSRC Turing AI World-Leading Research Fellowship No. EP/X040062/1 and EPSRC AI Hub No. EP/Y028872/1. P.S. acknowledges support from the NCCR Catalysis (grant number 225147), a National Centre of Competence in Research funded by the Swiss National Science Foundation. This work was supported by the Swiss National Science Foundation grant 310030_197724.

REFERENCES

Melissa F Adasme, Katja L Linnemann, Sarah Naomi Bolz, Florian Kaiser, Sebastian Salentin, V Joachim Haupt, and Michael Schroeder. Plip 2021: Expanding the scope of the protein–ligand interaction profiler to dna and rna. *Nucleic acids research*, 49(W1):W530–W534, 2021a.

Melissa F Adasme, Katja L Linnemann, Sarah Naomi Bolz, Florian Kaiser, Sebastian Salentin, V Joachim Haupt, and Michael Schroeder. Plip 2021: expanding the scope of the protein–ligand interaction profiler to dna and rna. *Nucleic Acids Research*, 49(W1):W530–W534, 05 2021b. ISSN 0305-1048. doi: 10.1093/nar/gkab294. URL https://doi.org/10.1093/nar/gkab294.

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

Foivos Alimisis, Peter Davies, Bart Vandereycken, and Dan Alistarh. Distributed principal component analysis with limited communication. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 2823–2834. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/1680e9fa7b4dd5d62ece800239bb53bd-Paper.pdf.

Rasha Atwi, Ye Wang, Simone Sciabola, and Adam Antoszewski. Roshambo: Open-source molecular alignment and 3d similarity scoring. *Journal of Chemical Information and Modeling*, 64(21): 8098–8104, 2024.

Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1): 235–242, 2000.

Yuemin Bian and Xiang-Qun Xie. Computational fragment-based drug design: current trends, strategies, and applications. *The AAPS journal*, 20:1–11, 2018.

Martin Buttenschoen, Garrett M Morris, and Charlotte M Deane. Posebusters: Ai-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science*, 15(9):3130–3139, 2024.

Suman K Chakravarti. Distributed representation of chemical fragments. *Acs Omega*, 3(3):2825–2836, 2018.

Ting Chen, Ruixiang ZHANG, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *The Eleventh International Conference on Learning Representations*, 2023.

Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

Miles Congreve, Gianni Chessari, Dominic Tisi, and Andrew J Woodhead. Recent developments in fragment-based drug discovery. *Journal of medicinal chemistry*, 51(13):3661–3680, 2008.

Jorg Degen, Christof Wegscheid-Gerlach, Andrea Zaliani, and Matthias Rarey. On the art of compiling and using'drug-like'chemical fragment spaces. *ChemMedChem*, 3(10):1503, 2008.

Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*, 2020.

Fredrik NB Edfeldt, Rutger HA Folmer, and Alexander L Breeze. Fragment screening to predict druggability (ligandability) and lead discovery success. *Drug discovery today*, 16(7-8):284–287, 2011.

Matteo P Ferla, Rubén Sánchez-García, Rachael E Skyner, Stefan Gahbauer, Jenny C Taylor, Frank von Delft, Brian D Marsden, and Charlotte M Deane. Fragmenstein: predicting protein–ligand structures of compounds derived from known crystallographic fragment hits using a strict conserved-binding–based methodology. *Journal of Cheminformatics*, 17(1):4, 2025.

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

Bowen Gao, Bo Qiang, Haichuan Tan, Yinjun Jia, Minsi Ren, Minsi Lu, Jingjing Liu, Wei-Ying Ma, and Yanyan Lan. Drugclip: Contrastive protein-molecule representation learning for virtual screening. *Advances in Neural Information Processing Systems*, 36:44595–44614, 2023.

Zhichun Guo, Kehan Guo, Bozhao Nan, Yijun Tian, Roshni G Iyer, Yihong Ma, Olaf Wiest, Xiangliang Zhang, Wei Wang, Chuxu Zhang, et al. Graph-based molecular representation learning. *arXiv preprint arXiv:2207.04869*, 2022.

Catrin Hasselgren and Tudor I Oprea. Artificial intelligence for drug discovery: Are we there yet? *Annual Review of Pharmacology and Toxicology*, 64(1):527–550, 2024.

Roderick E Hubbard and James B Murray. Experiences in fragment-based lead discovery. In *Methods in enzymology*, volume 493, pp. 509–531. Elsevier, 2011.

Ilia Igashov, Arian R Jamasb, Ahmed Sadek, Freyr Sverrisson, Arne Schneuing, Pietro Lio, Tom L Blundell, Michael Bronstein, and Bruno Correia. Decoding surface fingerprints for protein-ligand interactions. *bioRxiv*, pp. 2022–04, 2022.

Ilia Igashov, Arne Schneuing, Marwin Segler, Michael Bronstein, and Bruno Correia. Retrobridge: Modeling retrosynthesis with markov bridges. *arXiv preprint arXiv:2308.16212*, 2023.

Ilia Igashov, Hannes Stärk, Clément Vignac, Arne Schneuing, Victor Garcia Satorras, Pascal Frossard, Max Welling, Michael Bronstein, and Bruno Correia. Equivariant 3d-conditional diffusion model for molecular linker design. *Nature Machine Intelligence*, pp. 1–11, 2024.

Fergus Imrie, Anthony R Bradley, Mihaela van der Schaar, and Charlotte M Deane. Deep generative models for 3d linker design. *Journal of chemical information and modeling*, 60(4):1983–1995, 2020.

Xavier Jalencas, Hannes Berg, Ludvik Olai Espeland, Sridhar Sreeramulu, Franziska Kinnen, Christian Richter, Charis Georgiou, Vladyslav Yadrykhinsky, Edgar Specker, Kristaps Jaudzems, et al. Design, quality and validation of the eu-openscreen fragment library poised to a high-throughput screening collection. *RSC Medicinal Chemistry*, 15(4):1176–1188, 2024.

Bowen Jing, Stephan Eismann, Pratham N Soni, and Ron O Dror. Equivariant graph neural networks for 3d macromolecular structure. *arXiv preprint arXiv:2106.03843*, 2021.

Susan Leung, Michael Bodkin, Frank von Delft, Paul Brennan, and Garrett Morris. Sucos is better than rmsd for evaluating fragment elaboration and docking poses. *ChemRviv preprint 10.26434/chemrxiv.8100203*, 2019.

Juncai Li and Xiaofei Jiang. Mol-bert: An effective molecular representation with bert for molecular property prediction. *Wireless Communications and Mobile Computing*, 2021(1):7181815, 2021.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Frederieke Lohmann, Stephan Allenspach, Kenneth Atz, Carl CG Schiebroek, Jan A Hiss, and Gisbert Schneider. Protein binding site representation in latent space. *Molecular Informatics*, pp. e202400205, 2024.

Jean-Remy Marchand and Amedeo Caflisch. In silico fragment-based drug design with seed. *EUROPEAN JOURNAL OF MEDICINAL CHEMISTRY*, 156:907–917, AUG 5 2018. ISSN 0223-5234. doi: 10.1016/j.ejmech.2018.07.042.

William McCorkindale, Ivan Ahel, Haim Barr, Galen J Correy, James S Fraser, Nir London, Marion Schuller, Khriesto Shurrush, and Alpha A Lee. Fragment-based hit discovery via unsupervised learning of fragment-protein complexes. *bioRxiv*, pp. 2022–11, 2022.

Andrew T McNutt, Paul Francoeur, Rishal Aggarwal, Tomohide Masuda, Rocco Meli, Matthew Ragoza, Jocelyn Sunseri, and David Ryan Koes. Gnina 1.0: molecular docking with deep learning. *Journal of cheminformatics*, 13(1):43, 2021.

Rebecca M Neeser, Mehmet Akdel, Daniel Kovtun, and Luca Naef. Reinforcement learning-driven linker design via fast attention-based point cloud alignment. *arXiv preprint arXiv:2306.08166*, 2023.

RDKit. RDKit: Open-source cheminformatics. `http://www.rdkit.org`.

Michel F Sanner, Arthur J Olson, and Jean-Claude Spehner. Reduced surface: an efficient way to compute molecular surfaces. *Biopolymers*, 38(3):305–320, 1996.

Arne Schneuing, Ilia Igashov, Thomas Castiglione, Michael M Bronstein, and Bruno Correia. Towards structure-based drug design with protein flexibility. In *ICLR 2024 Workshop on Generative and Experimental Perspectives for Biomolecular Design*, 2024.

Arne Schneuing, Ilia Igashov, Adrian W. Dobbelstein, Thomas Castiglione, Michael M. Bronstein, and Bruno Correia. Multi-domain distribution learning for de novo drug design. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=g3VCIM94ke`.

scitkit learn. Histgradientboostingclassifier. `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.HistGradientBoostingClassifier.html`.

Bonggun Shin, Sungsoo Park, Keunsoo Kang, and Joyce C Ho. Self-attention based molecule representation for predicting drug-target interaction. In *Machine learning for healthcare conference*, pp. 230–248. PMLR, 2019.

Ken Shoemake. Animating rotation with quaternion curves. *SIGGRAPH Comput. Graph.*, 19(3): 245–254, July 1985. ISSN 0097-8930. doi: 10.1145/325165.325242. URL `https://doi.org/10.1145/325165.325242`.

Vignesh Ram Somnath, Charlotte Bunne, and Andreas Krause. Multi-scale representation learning on proteins. *Advances in Neural Information Processing Systems*, 34:25244–25255, 2021.

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

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. *arXiv preprint arXiv:2209.14734*, 2022.

Hongwei Wang, Weijiang Li, Xiaomeng Jin, Kyunghyun Cho, Heng Ji, Jiawei Han, and Martin D Burke. Chemical-reaction-aware molecule representation learning. *arXiv preprint arXiv:2109.09888*, 2021.

J Michael Word, Simon C Lovell, Jane S Richardson, and David C Richardson. Asparagine and glutamine: using hydrogen atom contacts in the choice of side-chain amide orientation. *Journal of molecular biology*, 285(4):1735–1747, 1999.

Bo Yang, Chijian Xiang, and Jianing Li. 3d structure-based generative small molecule drug design: Are we there yet? *bioRxiv*, pp. 2024–12, 2024.

Haoyu S Yu, Kalyan Modugula, Osamu Ichihara, Kimberly Kramschuster, Simon Keng, Robert Abel, and Lingle Wang. General theory of fragment linking in molecular design: why fragment linking rarely succeeds and how to improve outcomes. *Journal of Chemical Theory and Computation*, 17(1):450–462, 2020.

Gengmo Zhou, Zhifeng Gao, Qiankun Ding, Hang Zheng, Hongteng Xu, Zhewei Wei, Linfeng Zhang, and Guolin Ke. Uni-mol: A universal 3d molecular representation learning framework. In *The Eleventh International Conference on Learning Representations*, 2023.

A Previous Work

Molecular representation learning (MRL) is a well-established field, with numerous studies refining and adapting methods for specific tasks such as property and reaction prediction (Guo et al., 2022). Various MRL approaches leverage language models as encoders (Shin et al., 2019; Chithrananda et al., 2020; Li & Jiang, 2021) or introduce specialized representations, such as UniMol, which incorporates 3D conformers (Zhou et al., 2023), and MolR, which is tailored for reaction-based learning (Wang et al., 2021). However, these methods typically do not explicitly account for protein targets and are not primarily designed for hit screening or drug discovery. Gao et al. (2023) proposed DrugCLIP, which contrastively learns pocket and ligand representations for the task of VS based on the UniMol encoder architecture (Zhou et al., 2023). This method conceptually shares many aspects with out work but encodes the pocket globally and full ligands not allowing the task of fragment placement. While none of the aforementioned approaches focus specifically on molecular fragments, Chakravarti (2018) propose a fragment-based method, though it remains centered on chemical properties relevant to tasks like property prediction rather than interaction-driven applications. A concurrent work that was published at the time of writing by Lohmann et al. (2024) proposes a similar encoder. They encode both proteins and ligands but then use the protein embeddings and for the task of affinity prediction.

The task of computational fragment identification has been mostly dependent on fragment docking (Bian & Xie, 2018; Marchand & Caflisch, 2018), which is costly and less accurate than ligand docking. FRESCO (McCorkindale et al., 2022) is a ML-based method that implicitly considers target structure through pharmacophore distributions. The extraction of those, however, requires known hits from fragment screens and respective crystal structures, which is often not available in a drug discovery campaign. The closest approach to ours by Igashov et al. (2022) matches protein pocket embeddings in order to find related fragment hits, making it also limited by the availability of crystal structures.

B Latent Encoder

B.1 Model Architecture

Protein Encoding The protein encoding approach is similar to the dMaSIF (Sverrisson et al., 2021) method with some notable exceptions: a smaller receptive field r (geodesic radius), higher embedding dimension d and surface computation as in the original MaSIF (Gainza et al., 2020). The protein surface is first computed using the MSMS program (Sanner et al., 1996) with a density of 3 Å, a water probe radius of 1.5 Å and subsequent downsampling to a resolution of 1 Å. Input features computed on this surface include geometric properties (shape index, curvature) and chemical features (electrostatics, hydrogen bond donors/acceptors, hydropathy). The architecture is based on a special convolutional neural network which uses Gaussian kernels defined in a local geodesic coordinate system. We refer to Gainza et al. (2020) for detailed information.

Fragment Encoding Fragments are represented as 2D graphs. Node features are initialized by one-hot encoded atom types considering {C, N, O, S, B, Br, Cl, P, I, F, NH, N+, O-} with NH corresponding to a Nitrogen with an explicit Hydrogen and +/- represent formal charges. Additionally, a suite of properties are computed for atoms, bonds and on the global level, which are summarized in Table B.1. The fragment encoding is parametrized using a Graph Transformer (GT) (Dwivedi & Bresson, 2020; Vignac et al., 2022; Igashov et al., 2023) directly using the output of the learned global embedding y. We largely follow the GT implementation of Igashov et al. (2023) with one notable exception of linearly projecting the input features before adding them to the network output instead of simply cropping the vector.

Table B.1: Features for initializing nodes, edges and global features of the fragment graph. Non-binary atom and bond properties were one-hot encoded and global continuous values min-max scaled. These properties were determined using `RDKit`. (RDKit)

Localization	Feature	Values	Size
atom	atom type	{C, N, O, S, B, Br, Cl, P, I, F, NH, N+, O-}	13
	formal charge	[-1,3]	5
	degree	[0,6]	7
	is in ring	binary	1
	is aromatic	binary	1
	hybridization	{sp, sp2, sp3, sp2d, sp3d, sp3d2, unspecified}	7
	chiral tag	[S, R, unspecified, other]	4
bond	bond type	{single, double, triple, aromatic, no bond}	5
	conjugated	binary	1
	is in ring	binary	1
global	# atoms	continuous	1
	# bonds	continuous	1
	# rings	continuous	1
	# aromatic rings	continuous	1
	MW	continuous	1
	logP	continuous	1
	TPSA	continuous	1
	# HBD	continuous	1
	# HBA	continuous	1
	ring sizes	[3, 18], no ring, other ring size	18

B.2 Contrastive Training

The encoder is trained contrastively maximizing the cosine similarity for positive protein-fragment pairs while minimizing it for negative pairs sampled from the same protein pocket.

Positive and Negative Pairs Positive surface points $\mathcal{P}^+$ are defined as all surface points within a threshold distance d_{bind} of any fragment atom. Negative surface points $\mathcal{P}^-$ are sampled from various regions within the pocket, excluding positive points. These include:

- Random pocket points: Points sampled uniformly from the pocket.
- Concave points: Predicted by a curvature classifier as regions with inward curvature.
- Convex points: Predicted as regions with outward curvature.

To classify surface curvature as concave or convex, we train a Histogram-based Gradient Boosting Classifier from `scikit-learn` (scitkit learn) on the same dataset using curvatures at different smoothing scales as described in dMaSIF (Sverrisson et al., 2021). The classifier labels concave regions based on their proximity to fragments ($d < d_{bind}$), assuming such regions are more likely to host binding interactions, while convex points are those outside this threshold. The classifier is trained and run concurrently with the encoder training, enabling on-the-fly curvature prediction.

Contrastive Loss For the contrastive training loss, the cosine similarity between the protein surface embedding h_p and fragment embedding h_f is computed as follows:

$$\cos(h_p, h_f) = \frac{h_p \cdot h_f}{\|h_p\| \|h_f\|}. \tag{B.1}$$

Positive pairs maximize $\cos(h_p, h_f)$ while negative minimize it

$$\mathcal{L}_{pos} = -\mathbb{E}_{(p,f)\in\mathcal{P}^+} \left[w(p,f) \log \sigma(\cos(h_p, h_f)) \right] \tag{B.2}$$

$$\mathcal{L}_{neg} = -\mathbb{E}_{(p,f)\in\mathcal{P}^-} \left[\log \sigma(-\cos(h_p, h_f)) \right]. \tag{B.3}$$

Here, σ is the sigmoid function, and $w(p, f)$ scales the contribution of positive pairs by their distances d, defined as:

$$w(p, f) = \frac{1}{1 + \exp(-\alpha(\frac{d - d_{min}}{d_{max}} - 0.5))} \tag{B.4}$$

where d_{min} and d_{max} normalize the distances, and α controls the sharpness of the weighting. This up-weights positive points that are further away, encouraging the model to focus on points at the edge of the binding regions. This was shown to work much better than the more intuitive way of penalizing far points by inverting the sign before α in Equation B.4, which surprisingly resulted in approximately random ROC AUC scores on the test set. This weighting scheme also slightly improves over the binary case (no weights) especially with respect to discriminative metrics such as EF_1 or top-k Accuracy also on the test set. One potential explanation for this is that interactions further away are weaker albeit more numerous, which in return can result in them being "drowned out" and the model not taking this collective effect of interactions into account.

The final contrastive loss is the sum of both terms

$$\mathcal{L}_{contrastive} = \frac{\mathcal{L}_{pos} + \mathcal{L}_{neg}}{2} \tag{B.5}$$

B.3 Fragment Similarity Penalty (FSP)

To incorporate aspects purely dependent on the small molecular fragment and encourage diversity of embeddings we introduce a fragment similarity penalty (FSP). The FSP is a hinge loss between the true positive fragment embedding h_f^+ and the embedding of a randomly sampled fragment from the same training library with Tanimoto similarity below 0.1. This penalizes similar embeddings despite dissimilar chemical structures. The loss is defined like so

$$\mathcal{L}_{FSP} = \text{ReLU}(\cos(h_f^+, h_f^-) - c) \tag{B.6}$$

with margin c.

B.4 Non-covalent Interaction (NCI) Loss

The additional NCI loss is based on a multi-label classifier predicting non-covalent interaction (NCI) types for protein pocket surface points using their latent embeddings. This encourages the differentiation of the pocket surface and aims at increasing sensitivity to different fragments. The NCI classifier module is a lightweight feedforward neural network with one linear layers, surface embeddings as input and $n_{NCI} + 1$ classes as output using a sigmoid activation function. The model is thus tasked with predicting probabilities for the classes {hydrophobic interactions, hydrogen bonds, water bridges, salt bridges, π-stacks, π-cation interactions, halogen bonds, interaction presence}. Interactions are extracted by using the Protein-Ligand Interaction Profiler (PLIP) (Adasme et al., 2021b) and positive labels, as one-hot encoded interaction profile, are assigned when surface points are in a distance of 2 Å of the interacting residue atom. One surface point can have several interactions if it is close to multiple interactions. The loss is the the binary cross entropy (BCE) between predicted probabilities z and true labels y

$$\mathcal{L}_{NCI} = \text{BCE}(z, y) \tag{B.7}$$

B.5 Training Loss

The final training loss is the weighted sum of the previous individual losses including L2 regularization

$$\mathcal{L} = \lambda_{contrastive}\mathcal{L}_{contrastive} + \lambda_{reg}\mathcal{L}_{reg} + \lambda_{FSP}\mathcal{L}_{FSP} + \lambda_{NCI}\mathcal{L}_{NCI}. \tag{B.8}$$

with the regularization term amounting to

$$\mathcal{L}_{reg} = \frac{1}{n + m}\left(\sum_{i=1}^{n} h_{p,i}^2 + \sum_{j=1}^{m} h_{f,j}^2\right). \tag{B.9}$$

B.6 TRAINING

Table B.2: Hyperparameters of the encoder training.

Module	Parameter	Value
training	epochs	25
	batch size	32
	learning rate	0.001
loss	$\lambda_{contrastive}$	2.0
	regularization	0.1
	NCI loss	yes
	d_{min} [Å]	0.5
	d_{max} [Å]	$d_{bind} = 3.0$
	α	5.5
	λ_{NCI}	1.0
	FSP loss	yes
	λ_{FSP}	1.0
	FSP margin c	0.3
	threshold d_{bind} [Å]	3.0
negative samples	random [%]	0.34
	concave [%]	0.33
	convex [%]	0.33
protein encoder dMaSIF	radius r [Å]	3.0
	resolution [Å]	1.0
	layers	2
	hidden dimension	16
	embedding dimension	128
	curvature scales	{1.0, 3.0, 5.0, 7.0 9.0}
fragment encoder GT	layers	4
	input node dimension	37
	input edge dimension	5
	input global dimension	27
	encoder dimensions	X: 256, E: 64, y: 256
	hidden dimensions	dx: 256, de: 32, dy: 256 d_{ffX}: 256, d_{ffE}: 64, d_{ffy}: 256
	heads	8
	output node dimension	128
	output global dimension	128

C GENERATIVE MODELING

The architecture and flow matching framework was inspired by DrugFlow (Schneuing et al., 2024; 2025).

C.1 FLOW MATCHING

Flow matching is a generative method, where a prior p_0 is connected to the data distribution p_1 through a sequence of probability distributions $\{p_t : t \in [0, 1]\}$. This path is defined by a time-dependent vector field that can be approximated by this method. The transform the prior into a data point the vector field $u_t(x)$ is integrated to give the flow

$$\frac{d}{dt}\psi_t(x) = u_t(\psi_t(x)). \tag{C.10}$$

Lipman et al. (2022) showed that rather than defining the true vector field $u_t(x)$, it is easier to parametrize the conditional flow $u_t(x|x_1)$ based on a data point x_1. The conditional flow matching loss is defined as

$$\mathcal{L}_{CFM} = \mathbb{E}_{t,q(x_1),p_t(x_0)} \|v_\theta(t, x_t) - \dot{x}_t\|^2. \tag{C.11}$$

with $\dot{x}_t| = \frac{d}{dt}\psi(x_0|x_1)$, the time derivative of the conditional flow. To generate new samples, we sample x_0 from the prior distribution $p(x_0)$ and simulate the ODE in equation C.10 using the learned vector field $v_\theta(t, x_t)$.

C.2 Centroid coordinates: Euclidean Flow

To parametrize the centroid coordinates we use the Independent-coupling conditional flow matching (Albergo & Vanden-Eijnden, 2022; Tong et al., 2023) with the generative vector field

$$u_t(x|x_1) = \frac{\sigma_t'(x_1)}{\sigma_t(x_1)}(x - \mu_t(x_1)) + \mu_t'(x_1). \tag{C.12}$$

A constant velocity vector field $\dot{x}_t = \frac{x_1 - x_t}{1-t}$ results from a constant $\sigma = \sigma_t(x_1)$ and $\mu_t(x_1) = tx_1 + (1-t)x_0$ to define the Gaussian probability path. The loss for this flow thus amounts to

$$\mathcal{L}_{coord} = \mathbb{E}_{t,q(x_1),p_t(x_0)} \|v_\theta(t, x_t) - (x_1 - x_0)\|^2. \tag{C.13}$$

C.3 Latent Fragment Embeddings: Spherical flow

For the latent embeddings of the fragments, we define a flow on the unit sphere $\mathbb{S}^2 = \{x \in \mathbb{R}^{2+1} : \|x\|_2 = 1\}$. We can interpolate between x_0 and x_1 in the tangent space in which the vector field $v_\theta(t, x_t)$ is also learned, allowing one to avoid simulation. Concurrently, the loss is also computed in tangent space where the local geometry is Euclidean

$$\mathcal{L}_{S^2} = \mathbb{E}_{t,q(x_1),p_t(x_0)} \|v_\theta(t, x_t) - \frac{\log_{x_t}(x_1)}{1-t}\|^2 \tag{C.14}$$

To obtain x_t as the point on the sphere, we employ exponential and logarithmic maps

$$x_t = exp_{x_0}(t\log_{x_0}(x_1)) \tag{C.15}$$

with $\exp_x(u) = \cos(\|u\|)x + \sin(\|u\|)(\frac{u}{\|u\|})$, which maps a tangent vector u to the sphere and $\log_x(y) = \frac{\theta}{\sin(\theta)}(x - \cos(\theta)y)$, which takes point x on the sphere to a vector in the tangent space at base y. $\theta = \arccos(< x, y >)$, the geodesic distance between x and the base y. Equation C.15 corresponds to the spherical linear interpolation (SLERP) (Shoemake, 1985)

$$x_t = \text{SLERP}(x_0, x_1, t) = \frac{\sin((1-t)\theta)}{\sin(\theta)}x_0 + \frac{\sin(t\theta)}{\sin(\theta)}x_1. \tag{C.16}$$

The spherical flow allows consistency in relation to the cosine similarity-based encoder training and library searching and results in a smooth trajectory between x_0 and x_1. A Euclidean flow, while also applicable, exhibits a drastic change in cosine similarity only with high t requiring the model to learn big velocities for later time points only. This flow is similar to the system described by Alimisis et al. (2021).

C.4 Training Loss

Combining the two modalities gives a weighted sum of the previously defined losses:

$$\mathcal{L} = \lambda_{coord}\mathcal{L}_{coord} + \lambda_{S^2}\mathcal{L}_{S^2} \tag{C.17}$$

C.5 BACKBONE ARCHITECTURE

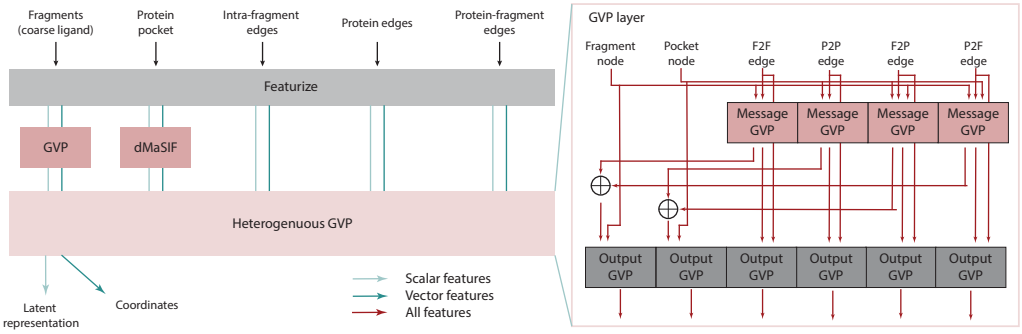


Figure C.1: Schematic overview of backbone architecture of the neural network parametrizing out flows. *Left:* All inputs are featurized individually to scalar and vector features, subsequently the fragment and pocket node features are transformed with a GVP and the pre-trained dMaSIF module from the encoder repsectively before being passed to a shared heterogeneous GVP Jing et al. (2021). Out output consists of velocities for the latent representation (scalar) and fragment centroid coordinates (vector). *Right:* Detailed layer of the GVP with distinct messages computed based on source and destination nodes for every edge feature. All modules are processed separately in output module based on another GVP. Figure adapted from Schneuing et al. (2025)

PROTEIN REPRESENTATION Proteins are represented as pocket surface point clouds, restricted to a 7 Å radius around the ligand. Surface points are featurized with normals as vector features and latent embeddings from the MaSIF module of the encoder as scalar features. For edge construction in the heterograph, k-nearest neighbors are sampled to connect nodes. Edge distances transformed via a radial basis function (RBF) with 16 bases.

FRAGMENTED LIGAND REPRESENTATION Small molecule ligands are represented as coarse graphs, where fragmented ligands are decomposed into nodes and edges. Each node corresponds to a fragment, with its centroid as the position x and a latent embedding serving as the scalar node feature h. No additional vector features are used except from self-conditioning. The graph is fully connected, as the goal is to learn fragment placement and identification rather than reconstruct to the ligand. Edge features are derived from pairwise distances using a RBF with 16 bases.

NEURAL NETWORK Combining pocket and fragments results in a heterogenous graph that is composed of two distinct node groups, fragments and pocket surface points, connected by four types of edges: fragment-to-fragment (F2F), fragment-to-pocket (F2P), pocket-to-fragment (P2F), and pocket-to-pocket (P2P). To accommodate this complexity, we employ a heterogeneous graph neural network architecture using geometric vector perceptron (GVP) layers Jing et al. (2021). This architecture incorporates distinct learnable message functions for each edge type and separate update functions for each node type. The pocket nodes are not parametrized by a GVP but instead by the MaSIF module pre-trained during the contrastive learning (weights are not frozen). Figure C.1 displays a scheme of the network architecture.

C.6 TRAINING

Table C.3: Hyperparameters for the training of the flow matching model. s = scalar feature, V = vector feature

Module	Parameter	Value
training	epochs	44[2]
	learning rate	5e-4
loss	λ_x	1.0
	λ_h	100.0
heterogeneous graph	pocket edge k neighbors	10
	edge cutoff interaction [Å]	10.0
GVP	layers	5
	node hidden dimensions (s, V)	265, 32
	edge hidden dimensions (s, V)	64, 16
simulation parameters	steps	500

C.7 SAMPLING AND POST-PROCESSING

SELF-CONDITIONING Self-conditioning (Chen et al., 2023) is used, in which the model takes previous predictions as input during sampling. This has been shown to improve performance.

NUMBER OF COARSE NODES In order to sample new fragments, the model needs to know how many nodes to place to build a graph. This can be done either by ground truth size, given this information is available, or alternatively is made dependent on the training distribution of number of coarse nodes given a pocket with x number of residues.

LIBRARY QUERYING At the end of sampling, the model outputs fragment embeddings h and centroid coordinates x. We can move from this coarse representation to an all-atom output by querying a fragment library maximum cosine similarity of the respective latent embeddings. This fragment library is independent of sampling or training and thus flexibly interchangeable.

DOCKING In order to obtain a realistic orientation with respect to the protein, all fragments are docked individually. Docking is performed using Gnina (McNutt et al., 2021) with the fragment placed at the sampled centroid defining the bounding box plus a margin of 1 Å.

D DATA

For both the encoder and the generative modeling protein-ligand structures were extracted from the Protein Data Bank (PDB) (Berman et al., 2000). For this, PDB was queried (accessed May 2023) retrieving 122,012 protein-only entries where the structures were determined using one of the following experimental methods: X-ray diffraction, electron microscopy, solid-state NMR, or solution NMR. The entries are filtered to include only those with a refinement resolution of 3 Å or better and at least one distinct non-polymer entity present. Subsequently, all structures are split into individual chains and corresponding ligands. All ligand SMILES present in the PDB are extracted to reassign the correct bond order using `RDKit`. Pairs where this reassignment failed were discarded. All proteins were protonated using Reduced (Word et al., 1999).

All ligands were fragmented using BRICS rules (Degen et al., 2008) with the exception of breaking double bonds. Fragments smaller than 8 heavy atoms got recombined with all possible neighboring fragments (related to graph partitioning), which served as a data augmentation technique.

[2]Training stopped early due to convergence and limiting training time.

D.1 ENCODER

For the encoder, only fragments in the distance of at least 5 Å were kept and the following filtering criteria were applied. Fragments are discarded if:

- Have no interaction as determined by PLIP (Adasme et al., 2021b).
- Have more than 20 heavy atoms.
- An element present is not in {C, N, O, S, B, Br, Cl, P, I, F}
- Molecular weight is more than 500 Da.
- Have a maximal ring size of more than 8.
- Have a phosphate group or an aklyl chain of at least 4 Carbons.

The dataset is split into training, validation and test following the approach used for HoloProt (Somnath et al., 2021), which is based on precomputed 30% sequence similarity. This splitting approach does result in having overlapping fragments but no data leakage in terms of protein sequence similarity.

Each fragment in the dataset is randomly paired with another fragment with a Tanimoto similarity below 0.1 for the FSP.

The fragment library used for VS and the generative pipeline is identical to the training fragments with the exception of removing fragments that have no profiled non-covalent interactions. This results in a fragment library of 41,224 unique fragments and 52,070 conformers (Fig. D.2), with the lowest energy conformer being retained for downstream tasks. The full library consist of 86,927 unique chains. The training set consists of 310,298, the validation set of 33,547 and the test set 45,210 protein:fragment pairs. Figure D.3 shows the distribution of some important molecular properties for the training data and the library.

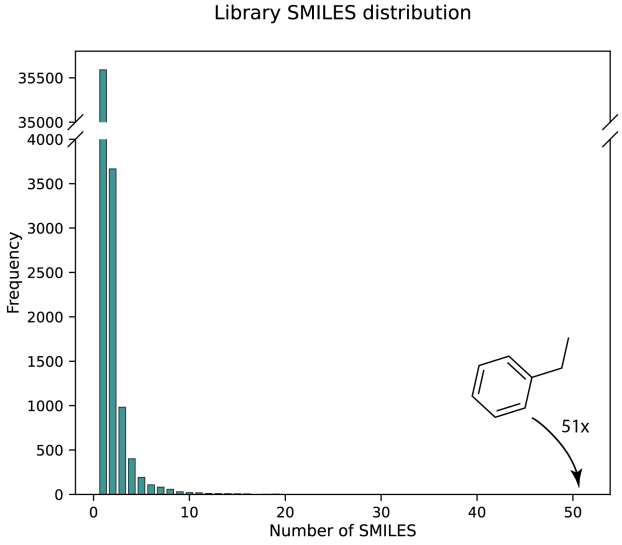


Figure D.2: Distribution of frequency of SMILES in the fragment library. Individual fragments can appear multiple times with different conformations but will have the same latent encoding. Ethylbenzene is the most frequent fragment and appears 51 times..

D.2 FLOW MATCHING

For the training and evaluation of the generative framework we use the same splits based on 30% sequence similarity as described above but randomly subsample the validation and test set to 100 datapoints with unique pockets. We further use the same data augmentation technique as described

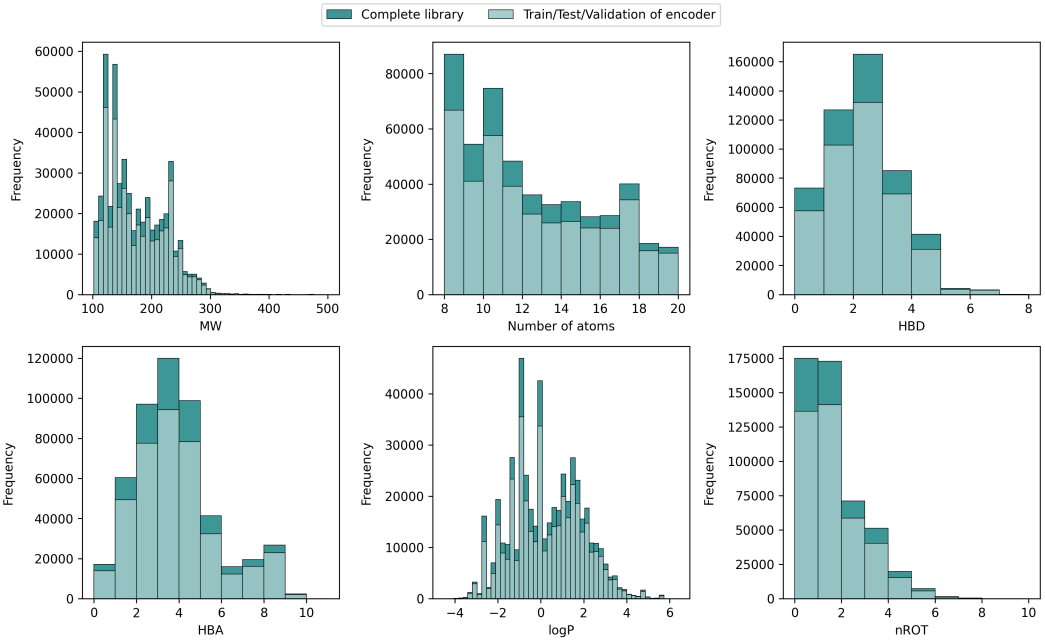


Figure D.3: Molecular properties of fragments extracted from the PDB structures. Both the complete library (*dark teal*) used for querying fragments in the generative pipeline and the set filtered by the presence of interactions (*light teal*) are shown. MW = molecular weight, HBD = hydrogen bond donor, HBA = hydrogen bond acceptor, logP = lipophilicity, nROT = number of rotational bonds

for the encoder for which we recombine fragments smaller than 8 heavy atoms in a combinatorial manner with their neighbors in order to get big enough fragments. One datapoint will have all fragments necessary to make up one ligand and contrary to the encoder, fragments further away are not discarded. The training set comprises of 118,786 pocket:fragment-set pairs.

Pocket surfaces are extracted by removing points further than 7 Å from the full ligand. Pocket surfaces points cloud smaller than 250 points are discarded.

E Metrics and Evaluation

E.1 Encoder

The encoder is evaluated on the full test set by assessing results obtained from both full surfaces and pocket surfaces only. The **ROC AUC** (area under the receiver operating characteristic curve) represents the main training objective of being able to discriminate between positive (interacting) and negative (non-interacting) surface:fragment pairs and is computed based on their cosine similarities across the whole surfaces. The following metrics are all computed on the cosine similarities between a fragment and each protein surface point and true (interacting) points are defined as those within 3 Å of the fragment.

The **enrichment factor (EF)** measures how well high-similarity points are enriched for binding sites compared to random expectation. All points above a similarity threshold given by the specified percentile (we consider top 1^{st}, 5^{th}, and 10^{th} percentile) are considered high-similarity points. The EF is the division of the fraction of interacting points among high similarity points and the fraction of interacting points across all points (whole surface or pocket). The **top-k accuracy** evaluates how well the highest-scoring points correspond to actual binding sites. It first ranks all points based on similarity and selects the top k (1, 10, 100) highest-scoring ones to compute the accuracy. The **Area Under the Precision-Recall Curve (AUPR)** quantifies how well similarity values distinguish binding from non-binding sites. The precision-recall curve is computed by varying the similarity

threshold and measuring the trade-off between precision and recall from which the area under the curve is calculated.

To assess correlation of cosine similarity between latent representation to more interpretable similarities the Tanimoto similarity based on `RDKit` fingerprints (2048 bits) (RDKit) and the ROSHAMBO score were computed. The ROSHAMBO score (Atwi et al., 2024) is a shape and color (pharamcophore) similarity metric for which conformers had to first be aligned in 3D space.

The clusters in the t-SNE plots in Figures 1 and F.5 were obtained using KMeans clustering and the number of clusters was determined by minimizing the silhouette score (29 for latent representation, 18 for fingerprints).

E.2 Fragment Identification

We investigated fragment identification through two approaches: virtual screening (VS) using our latent embeddings (Latent VS) and the generative framework for sampling fragment embeddings and their centroids. These are compared against VS based on docking (Docking VS) and a random baseline, evaluated on 100 protein pockets with $\leq 30\%$ sequence similarity to the training set.

Generative modeling For each surface 100 samples are generated with number of fragments corresponding to the reference number of fragments. In case of multiple datapoints for one pocket:ligand pair due to graph partitioning a random sample is chosen out of those. Every sample consists of one or more nodes with corresponding centroid and latent representation, which is used to query the library for closest fragment based on cosine similarity. All fragments are subsequently docked using Gnina (McNutt et al., 2021) within a restricted volume around predicted centroids as described in Appendix C.7.

Latent Virtual Screening The latent embeddings are used directly for virtual screening by ranking the sum of all cosine similarities between all pocket surface points and fragments in the library. The top 100 per target are selected and docked in the full pocket with the bounding box given by the reference ligand.

Docking Virtual Screening We further perform VS with docking. The full library is docked as described above to every test pocket. The top 100 fragments ranked by docking score are selected as hits.

Random Baseline A random baseline is established by randomly selecting fragments out of the library and noising the ground truth centroid by adding the double of a randomly chosen noise level drawn from the normal distribution. For every target the same number of fragments as in the ground truth pocket are sampled.

E.2.1 Metrics

We focus on a few evaluation metrics describing recovery, and proxies for interaction:

Hard Recovery Hard recovery is defined as exact matches between sampled and reference fragments based on non-isomeric SMILES. We ignore stereochemistry as chiral centers might be altered during docking especially with some fragments missing stereo-assignments. We further distinguish between Sampling hits and Unique Fragments. With the first, describing the total number of samples that have a corresponding match while the latter is the unique number of reference fragments that got recovered. Recovery rates are calculated correspondingly by dividing by number of sampled fragments or number of reference fragments that are also in the library. The VS baselines are divided by the total amount of selected top fragments for the Sampling hits (100 fragments x 100 targets).

Soft Recovery Soft recovery metrics are based on the SuCOS score (Leung et al., 2019), which assess shape and color (pharmacophoric) similarity between a smaller (fragment) and bigger (reference ligand) molecule. We plot the reverse cumulative proportion and assess ”soft hits” as those fragments with a score higher than 0.5. The scores are calculated based on docked poses.

DOCKING EFFICIENCY Docking efficiency distributions are compared in order to assess a proxy for binding affinity. Docking efficiency is the vina score (docking score computed by Gnina (McNutt et al., 2021)) normalized by the number of heavy atoms to reduce the bias introduced by fragments size. The scores are computed based on docked poses except for the reference, which is based on minimized poses.

NON-COVALENT INTERACTIONS Non-covalent interactions are extracted by PLIP (Adasme et al., 2021b) applied to all docked fragments individually. Reference profiles are based on crystal structure poses.

F ADDITIONAL RESULTS

F.1 ENCODER

We observed that without the FSP loss, we have on par sensitivity but the average similarity between all fragments of the library is much higher than with FSP (0.45 vs. 0.25), which we thought crucial for the task of fragment screening (further preliminary data in Fig. F.4). The effect of the NCI loss is less pronounced but does increase the metrics discussed in Section 3.1.

We further investigated different similarity metrics on a random subset of 1000 fragments. The cosine similarity between each latent representation correlates strongly to Tanimoto similarity (Spearman $\rho = 0.45$) based on the `RDKit` fingerprint (2048 bits) while there is only moderate correlation to shape and pharmacophoric similarities (after alignment) as determined by ROSHAMBO (Atwi et al., 2024) (for details see Appendix E). Interestingly, the correlation to the color score is higher than the corresponding shape score (Spearman ρ of 0.38 and 0.25 respectively) indicating that pharmacophoric aspects dominate.

Table F.4: Metrics evaluating the encoder on the test set. All scores are computed on the cosine similarity between fragment and protein surface representations with points labeled as true being 3 Å way from the fragment.

Model	Surface	ROC AUC ↑	avg h_f ↓	EF_1 ↑	EF_5 ↑	EF_{10} ↑	top-1 Acc ↑	top-10 Acc ↑	top-100 Acc ↑	AUPR ↑ (true rate)
ours	whole	0.7	0.25	22.85	14.35	8.61	0.39	0.36	0.33	0.31 (0.01)
	pocket			2.28	2.23	2.2	0.45	0.41	0.39	0.39 (0.16)
no NCI	whole	0.64	0.23	21.37	14.00	8.47	0.33	0.31	0.31	0.28 (0.01)
	pocket			2.22	2.17	2.15	0.41	0.39	0.37	0.37 (0.16)
no FSP	whole	0.73	0.45	24.79	14.29	8.45	0.45	0.43	0.36	0.32 (0.01)
	pocket			3.16	2.70	2.46	0.53	0.51	0.42	0.40 (0.16)

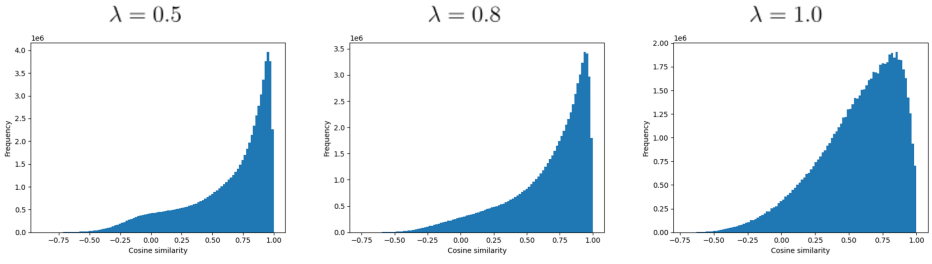


Figure F.4: Preliminary results highlighting the importance of the FSP on fragment representation diversity. λ corresponds to the loss weight.

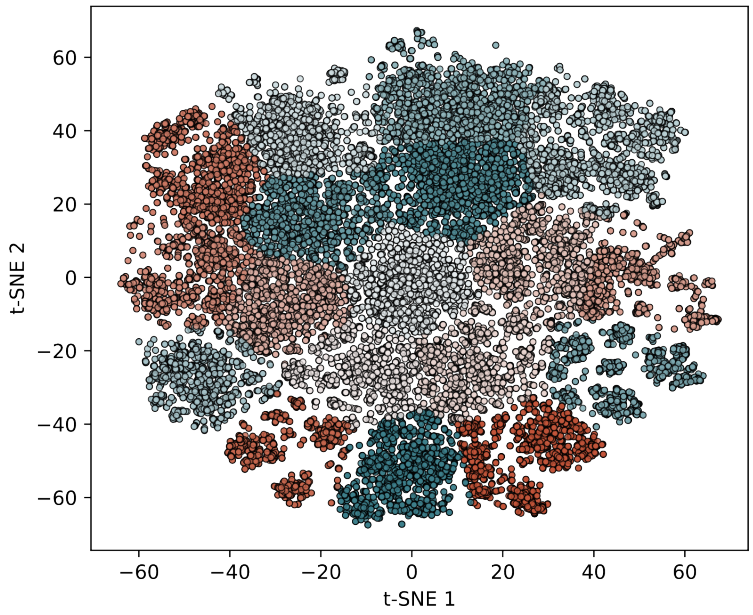


Figure F.5: t-SNE dimensionality reduction of the fragment library represented as `RDKit` fingerprints (2048 bits).

F.2 Fragment Identification

Table F.5: Recovery rates for sampling hits (samples that were an exact match) and unique fragments (unique fragments that were recovered). It should be noted that the different approaches result in hugely different numbers of sampled/top fragments as for latent VS we select the top 100 fragments based on the sum of cosine similarities to the pocket, for docking VS the top 100 based on docking scores, while the baseline and generative modeling we sample 100 times as many fragments as there are in the reference. This is included in the calculation of the recovery rate.

Approach	Sampling hits recovery rate [%]	Unique fragments recovery rate [%]
Generative modeling	4.33	0.554
Latent VS	0.02	0.001
Docking VS	2.05	0.130
Random baseline	0.00	0.00

Given that our latent representations capture protein-fragment interaction and not just chemical similarity we can expect correlation between RMSD and cosine similarity between sampled fragments and reference fragments. This correlation is based on the assumption that the ideal placement of one fragment is unique in a pocket, which is certainly not always the case. Unfortunately, RMSD did not show a strong correlation with cosine similarity between samples and references. However, generative fragments exhibited improved correlations compared to random samples. Additionally, shape and color overlap metrics using ROSHAMBO (Atwi et al., 2024) showed improvements over the random baseline, though absolute values remained low as well. Importantly, it should be noted that fragments often bind promiscuously across multiple binding sites, which may contribute to the observed variability.

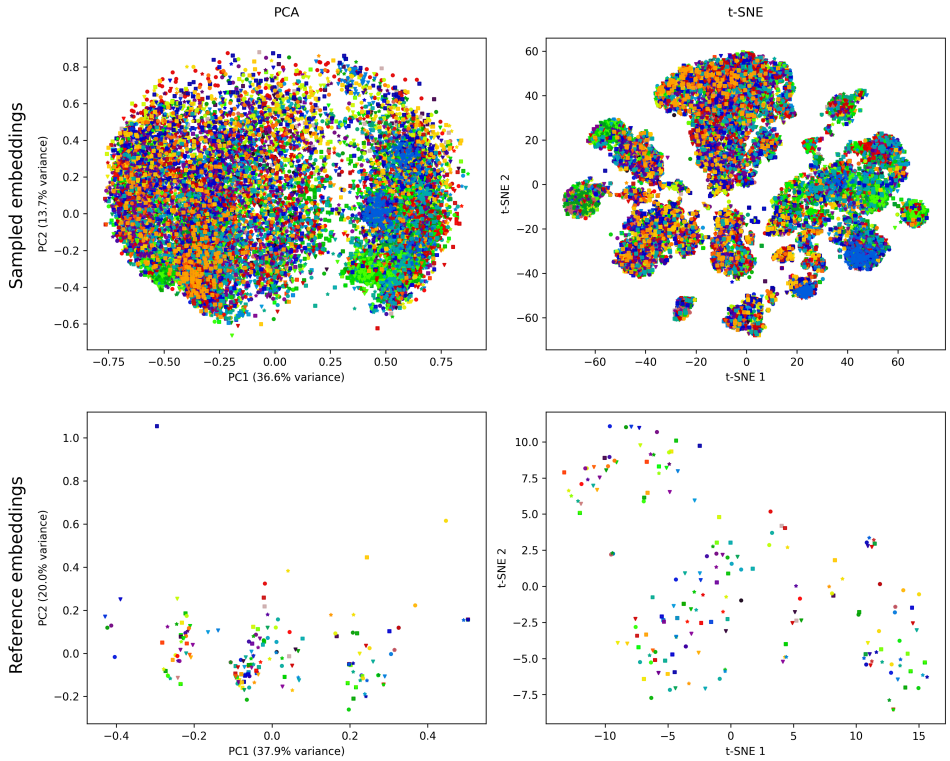


Figure F.6: Dimensionality reduction plot (PCA and t-SNE) of the sampled fragments from the generative pipeline compared to the reference fragments colored and symbolized by target.

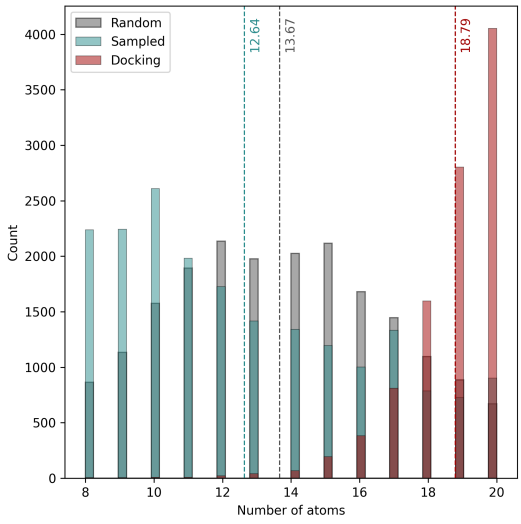
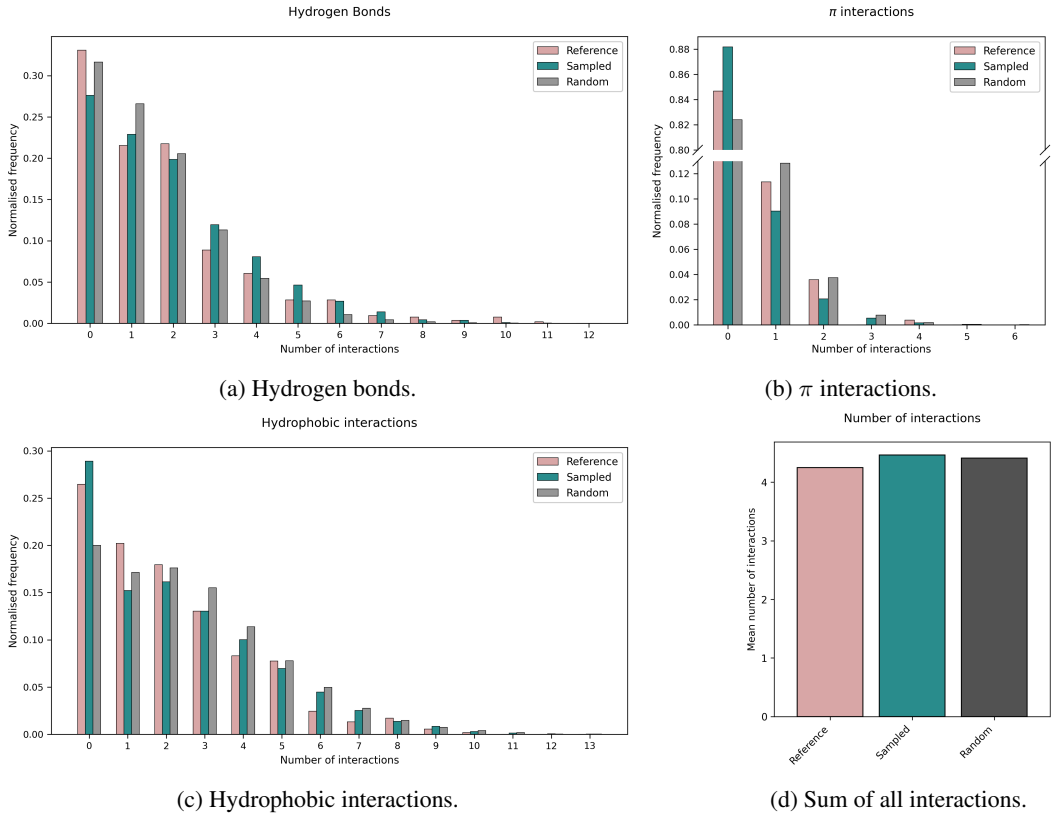


Figure F.7: Distribution of number of heavy atoms for our generated samples, docking VS and the random baseline.

F.2.1 Non-covalent Interactions

Using the Protein-Ligand Interaction Profiler (PLIP) (Adasme et al., 2021a) to compare profiles of non-covalent interactions we observed similar interaction type distributions between the reference

and sampled fragments but a higher average total number of interactions. However, similar results were observed with random samples, with even better profiles than the reference, putting the insightfulness of these results into question. This is likely more an artefact of docking, which aims at maximizing such interactions, than true interactions as we do not dock the reference fragments from crystal structures. Corresponding data is shown in Figure F.8.



(a) Hydrogen bonds.

(b) π interactions.

(c) Hydrophobic interactions.

(d) Sum of all interactions.

Figure F.8: Distribution of non-covalent interactions for sampled, reference and random fragments. π-interactions include π-π-stacking and π-cation interactions. The total number of interactions includes hydrogen bonds, π-interactions, hydrophobic interactions and salt bridges.

G Computational Resources

All models were trained on a single GPU (NVIDIA A100-SXM4-80GB) while development was performed on a NVIDIA GeForce RTX 3090.