nature machine intelligence

Article

https://doi.org/10.1038/s42256-024-00815-9

Equivariant 3D-conditional diffusion model for molecular linker design

Received: 31 May 2023

Accepted: 27 February 2024

Published online: 11 April 2024

Check for updates

Ilia Igashov[1], Hannes Stärk[2], Clément Vignac[1], Arne Schneuing[1], Victor Garcia Satorras[3], Pascal Frossard[1], Max Welling[3,5], Michael Bronstein[4] & Bruno Correia[1]

Fragment-based drug discovery has been an effective paradigm in early-stage drug development. An open challenge in this area is designing linkers between disconnected molecular fragments of interest to obtain chemically relevant candidate drug molecules. In this work, we propose DiffLinker, an E(3)-equivariant three-dimensional conditional diffusion model for molecular linker design. Given a set of disconnected fragments, our model places missing atoms in between and designs a molecule incorporating all the initial fragments. Unlike previous approaches that are only able to connect pairs of molecular fragments, our method can link an arbitrary number of fragments. Additionally, the model automatically determines the number of atoms in the linker and its attachment points to the input fragments. We demonstrate that DiffLinker outperforms other methods on the standard datasets, generating more diverse and synthetically accessible molecules. We experimentally test our method in real-world applications, showing that it can successfully generate valid linkers conditioned on target protein pockets.

The space of pharmacologically relevant molecules is estimated to exceed 10^{60} structures[1], and searching in that space poses substantial challenges for drug design. A successful approach to reduce the size of this space is to start from 'fragments', smaller molecular compounds that usually have no more than 20 heavy (non-hydrogen) atoms. This strategy is known as fragment-based drug design (FBDD)[2]. Given a protein pocket (a site on the target protein that has suitable properties for ligand binding), computationally determining fragments that interact with the pocket is a cheaper and more efficient alternative to experimental screening methods[2]. Once the relevant fragments have been identified and docked to the target protein, it remains to combine them into a single connected chemical compound. As has been shown in various applications, including FBDD[3], scaffold hopping (that is, discovery of structurally novel compounds starting from a known active molecule by modifying its core)[4] and proteolysis targeting chimera (PROTAC) design[5], the geometries of the identified fragments are crucial for the effective design of relevant and potent molecules. In addition, consideration of the structure of the protein pocket during the linker design process can remarkably improve the affinity of the generated compound leads[6]. In this work, we address the problem of linking fragments placed in a three-dimensional (3D) context with the possibility of conditioning the design process to the target protein pocket. Since we address several possible application scenarios, we note that the term 'linker' denotes any chemical matter that can connect starting molecular fragments and does not relate to any aspects of the terminology specific for any of the discussed domains.

Early computational methods for molecular linker design were based on database search and physical simulations[7], both of which are computationally intensive. Therefore, there is increasing interest in machine learning methods that can go beyond the available data and generate diverse linkers more efficiently. Existing approaches are based either on syntactic pattern recognition[8] or on autoregressive models[9–11]. While the former method operates solely on SMILES[12], the latter takes into account 3D positions and orientations of the input

[1]École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland. [2]Massachusetts Institute of Technology, Cambridge, MA, USA. [3]Microsoft Research AI4Science, Amsterdam, The Netherlands. [4]University of Oxford, Oxford, UK. [5]Present address: University of Amsterdam, Amsterdam, The Netherlands. e-mail: bruno.correia@epfl.ch

https://doi.org/10.1038/s42256-024-00815-9

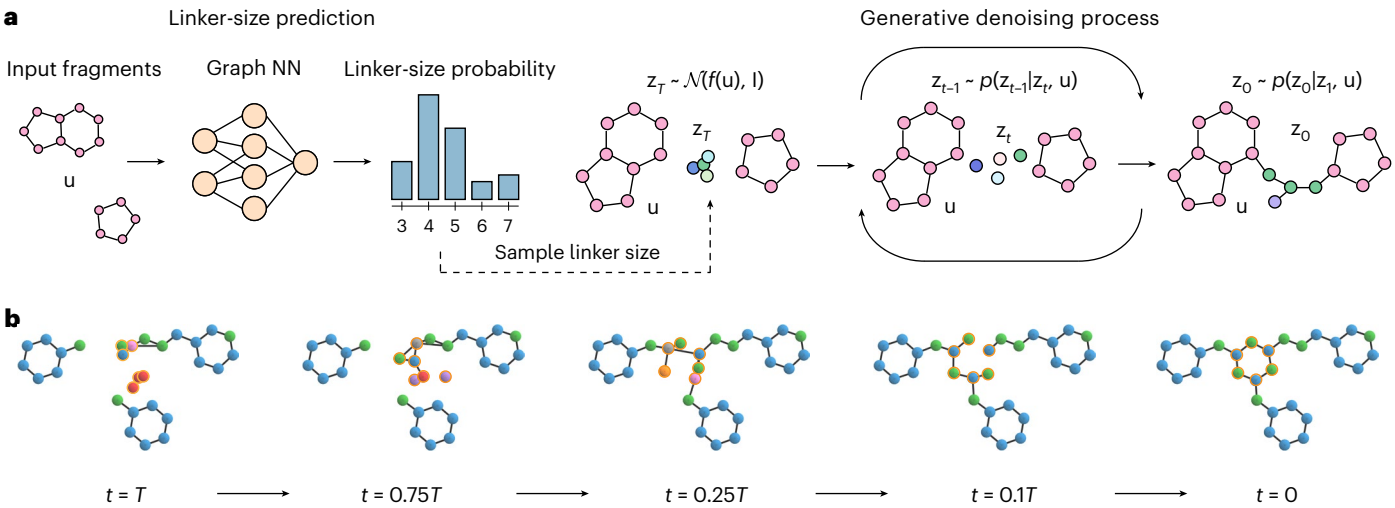


Fig. 1 | Overview of the molecular linker generation process. a, Probabilities of linker sizes are computed for the input fragments, and linker atoms are sampled and denoised using our fragment-conditioned equivariant diffusion model. **b**, Example of the linker generation process. Linker atoms are highlighted in orange.

fragments, as this information is essential for designing valid and stable molecules in various applications (see Supplementary Information for details). However, these methods are not equivariant with respect to the permutation of atoms and can only combine pairs of fragments. Finally, to date, there is no computational method for molecular linker design that takes the target protein pocket into account.

In this work, we introduce DiffLinker, a conditional diffusion model that generates molecular linkers for a set of input fragments represented as a 3D atomic point cloud. First, our model generates the size of the prospective linker and then samples initial linker atom types and positions from the normal distribution. Next, the linker atom types and coordinates are iteratively updated using a neural network that is conditioned on the input fragments. Ultimately, the denoised linker atoms and the input fragment atoms form a single connected molecule, as shown in Fig. 1.

DiffLinker has several desirable properties: it is equivariant to translations, rotations, reflections and permutations; it is not limited by the number of input fragments, does not require information on the attachment atoms and generates linkers with no predefined size. Moreover, we propose a new 3D conditioning mechanism for Euclidean diffusion models, which makes DiffLinker a versatile and state-of-the-art generative method applicable to various structure-based drug design tasks.

We show that DiffLinker has performance superior to that of previous methods in generating chemically relevant linkers between pairs of fragments. Our method achieves state-of-the-art results in synthetic accessibility and drug-likeness, which makes it useful in drug design pipelines. Besides, DiffLinker remarkably outperforms other methods in the chemical diversity of the generated linkers. We further propose a more challenging benchmark and show that our method is able to successfully link more than two fragments, which cannot be done by the other methods. We also demonstrate that DiffLinker can be conditioned on the target protein pocket; our model respects geometric constraints imposed by the surrounding protein atoms and generates molecules that are structurally compatible with the corresponding pockets. To demonstrate the relevance of DiffLinker in practical drug design applications, we provide three case studies where our method can be integrated into the fragment-based design of ligands to target heat shock protein 90 (Hsp90) and inosine 5′-monophosphate dehydrogenase (IMPDH), and scaffold hopping for improving selectivity for c-Jun N-terminal kinases (JNKs). To the best of our knowledge, DiffLinker is the first method that is not limited by the number of input fragments and accounts the information about pockets. The overall goal of this work is to provide practitioners with an effective tool for molecular linker generation in realistic drug design scenarios.

Results

We evaluate our method on four benchmarks in several different scenarios. First, we report the performance of DiffLinker on ZINC[13] and CASF[14] datasets that contain only pairs of fragments to be connected. Next, we introduce a new dataset based on GEOM molecules[15], where each entry contains two or more separate fragments. For all three sets we experiment with different modalities of our method: with predefined or sampled linker size and with known or unknown anchor points. Additionally, we assess the ability of DiffLinker to design relevant linkers in the presence of the protein pocket. For that, we introduce another dataset based on Binding MOAD[16]. Besides standard metrics used in the previous benchmarks, we measure the number of steric clashes between generated linkers and surrounding protein atoms. Finally, we demonstrate the applicability of DiffLinker in fragment-based design of Hsp90 and IMPDH inhibitors and in scaffold hopping for improving selectivity for JNKs. More details on datasets, baselines and metrics can be found in Methods.

Connecting fragment pairs

While DiffLinker shows greater flexibility and applicability in different scenarios than other methods, we show below that it also outperforms them on standard benchmarks ZINC and CASF in terms of chemical relevance (namely, the quantitative estimate of drug-likeness (QED), synthetic accessibility (SA) and number of rings) of the generated molecules. As shown in Table 1, molecules generated by DiffLinker are predicted to be more synthetically accessible and demonstrate higher drug-likeness, which is important for drug design applications. Moreover, our molecules usually share higher chemical and geometric similarity with the reference molecules as demonstrated by the SC_{RDKit} scores given in Supplementary Table 5. In terms of validity, our models perform on par with the other methods. Note that both DeLinker and 3DLinker are autoregressive approaches that explicitly employ valency rules at each generation step, while our model is shown to learn these rules from the data. Remarkably, the validity of the reference molecules from CASF with covalent bonds computed by OpenBabel is 92.2% while our model generated molecules with 90.2% validity. Notably, sampling the size of the linker substantially improves novelty and uniqueness of the generated linkers without serious degradation of the most important metrics.

Table 1 | Performance metrics on ZINC, CASF and GEOM test sets

	Method	QED ↑	SA ↓	No. of rings ↑	Valid, %	Unique, %	Novel, %
ZINC	DeLinker+ConfVAE+MMFF	0.64	3.11	0.21	**98.3**	44.2	**47.1**
	3DLinker (given anchors)	0.65	3.11	0.23	**99.3**	29.0	41.2
	3DLinker	0.65	3.14	0.24	71.5	29.2	41.9
	DiffLinker	**0.68**	**3.01**	0.25	93.8	24.0	30.3
	DiffLinker (given anchors)	**0.68**	**3.03**	0.26	97.6	22.7	32.4
	DiffLinker (sampled size)	0.65	3.19	**0.32**	90.6	**51.4**	42.9
	DiffLinker (given anchors, sampled size)	0.65	3.24	**0.36**	94.8	**50.9**	**47.7**
CASF	DeLinker+ConfVAE+MMFF	0.35	4.05	0.35	**95.7**	51.6	**55.6**
	DiffLinker	**0.41**	**4.00**	0.34	85.3	40.5	41.8
	DiffLinker (given anchors)	0.40	**4.03**	**0.38**	**90.2**	37.3	48.4
	DiffLinker (sampled size)	**0.40**	4.06	0.30	63.7	**60.0**	49.3
	DiffLinker (given anchors, sampled size)	0.40	4.10	**0.38**	68.4	**57.1**	**56.9**
GEOM	DeLinker+ConfVAE+MMFF	**0.76**	3.59	0.00	0.1	**74.5**	—
	3DLinker	0.36	3.56	0.00	16.3	**73.7**	—
	DiffLinker	0.48	**2.98**	0.78	**93.5**	36.7	70.7
	DiffLinker (given anchors)	**0.49**	**3.01**	**0.82**	**93.4**	37.3	70.5
	DiffLinker (sampled size)	0.46	3.24	0.76	87.4	63.1	**76.3**
	DiffLinker (given anchors, sampled size)	0.47	3.30	**0.84**	88.8	64.4	**76.6**

The first three metrics, average QED[42], average SA[26] and average number of rings in the linker, assess the chemical relevance of the generated molecules. The last three metrics, validity, uniqueness and novelty, evaluate the standard generative properties of the methods. For all three datasets, we compare DiffLinker with two state-of-the-art baselines: 3DLinker[11] and DeLinker[9]. To obtain 3D conformations for the molecules generated by DeLinker, we apply a pretrained ConfVAE[45] followed by a force field relaxation procedure using MMFF[46]. The top two best results for each metric are highlighted in bold.

In this experiment, we considered four different versions of DiffLinker depending on the amount of the prior information on anchors and linker length available at the sampling stage. Overall, the information about anchors helps to achieve higher validity and novelty of the generated samples, and this modality is preferred if such information is available. On the other hand, if anchor atoms are unknown, the resulting samples are more diverse as sampled linkers connect different pairs of atoms. Sampling linker length increases the diversity and novelty of the designed molecules while other metrics such as drug-likeness, SA and validity slightly degrade. In many drug design applications, uniqueness plays a crucial role, and chemical diversity provides chemists with more options to consider and test. In such cases, the DiffLinker model with minimum prior information (anchor atoms and linker size are unknown) is preferred. Examples of linkers generated by DiffLinker for different input fragments are shown in Extended Data Fig. 1.

Connecting multiple fragments

One of the major advantages of DiffLinker compared to recently developed autoregressive models DeLinker and 3DLinker is one-shot generation of the linker between any arbitrary number of fragments. This overcomes the limitation of DeLinker and 3DLinker, which can only link two fragments at a time. Although these autoregressive models can be adjusted to connect pairs of fragments iteratively while growing the molecule, the full context cannot be taken into account in this case. Therefore, suboptimal solutions are more likely to be generated. To illustrate this difference, we adapted DeLinker and 3DLinker to iteratively connect pairs of fragments in molecules where more than two fragments should be connected and tested all the methods on the GEOM dataset. As shown in Table 1, 3DLinker fails to construct valid molecules in almost 84% of cases and cannot recover any reference molecule, as shown in Supplementary Table 5. Despite the higher complexity of linkers in this dataset, our models achieve 93% validity and recover more than 85% of the reference molecules. DeLinker fails to generate valid molecules in almost 100% of samples. Besides, molecules generated by 3DLinker have no rings in the linkers, have substantially lower QED and are predicted to be harder to synthesize. Examples of linkers generated by DiffLinker for different input fragments are provided in Extended Data Fig. 2. An example of the DiffLinker sampling process for a molecule from the GEOM dataset is shown in Fig. 1b.

Pocket-conditioned linker design

To illustrate the ability of DiffLinker to leverage the structural information provided by the target's pockets, we trained three models on the Pockets dataset (Methods). These models were conditioned on the full-atom pocket representation, on the backbone atoms only and unconditioned, which serves as a baseline to evaluate the pocket conditioning. We computed the standard metrics reported in Supplementary Tables 6 and 7, as well as the number of steric clashes between generated molecules and the pockets. Clashes between two atoms are defined based on the distance between them and their van der Waals radii. As shown in Fig. 2b, the model conditioned on the full-atom pocket representation generates molecules with similar levels of steric clashes to those of the reference complexes from the test set. There is a clear trend in the number of clashes depending on the level of resolution of the pockets on which DiffLinker is conditioned, where conditioning on full-atom pockets generates molecules with less steric clashes.

To highlight the benefits of a reduced search space when using a fragment-based approach, we also compare the results of our full-atom conditioned model with two fully de novo generation methods. We choose ResGen[17], a 3D autoregressive method, and DiffSBDD[18], a conceptually similar diffusion model, as our baselines and evaluate the predicted binding propensity. In particular, we use GNINA[19] to relax the generated molecules in the pocket and calculate an estimate of the binding affinity. As shown in Fig. 2c, DiffLinker produces molecules with lower predicted binding affinity and poses that agree better with the orthogonal docking method GNINA than those generated without predefined fragments.

https://doi.org/10.1038/s42256-024-00815-9

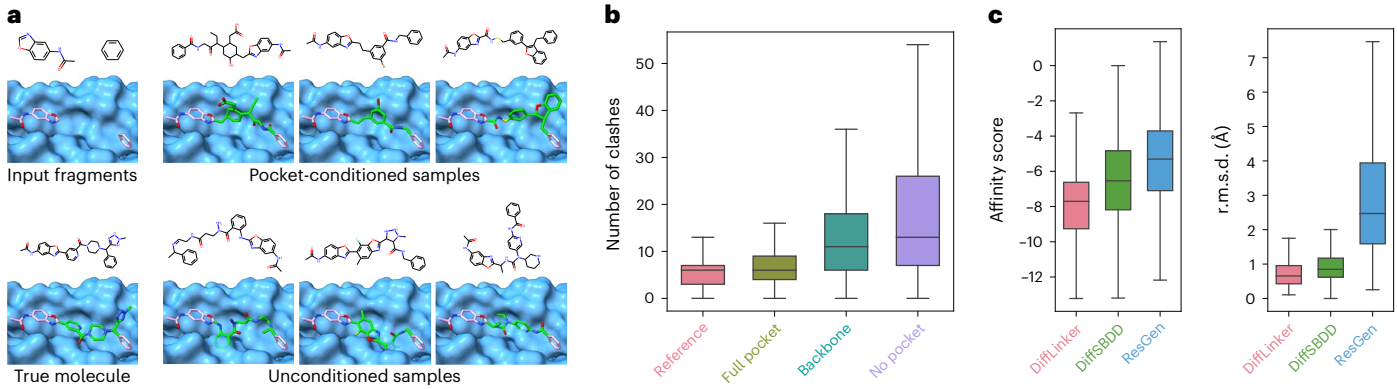


Fig. 2 | Ligand generation with DiffLinker in both pocket-conditioned and unconditioned scenarios. a, Examples of linkers sampled by DiffLinker, conditioned on pocket atoms (top row) and unconditioned (bottom row). Linkers sampled by the unconditioned model present steric clashes with the protein pocket. **b**, Quantification of steric clashes in reference molecules (that is, ground truth from the test set, number of data points $n = 566$) and samples from three DiffLinker models differently conditioned (or unconditioned) on pocket atoms (number of samples $n = 56{,}600$ per model). **c**, Comparison of DiffLinker pocket-conditioned linker design approach with two de novo molecule design baselines. All generated molecules were minimized using the molecular docking software GNINA[19]. We report the resulting docking score and r.m.s.d. between the original and the final conformation. Numbers of DiffLinker, DiffSBDD and ResGen samples are 56,596, 12,720 and 10,652, respectively. All box-and-whisker plots include the median line, the box denotes the interquartile range (IQR) and whiskers denote the rest of the data distribution within ±1.5× IQR.

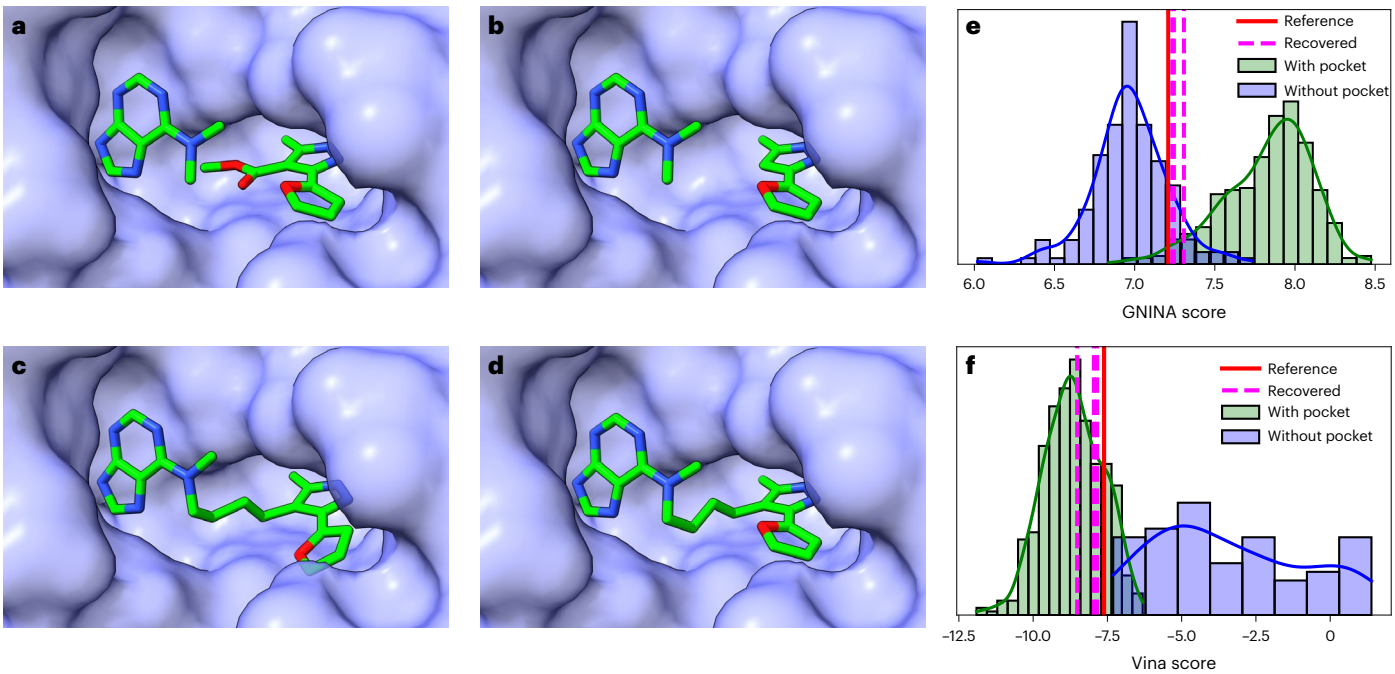


Fig. 3 | Hsp90 as a practical example for fragment-based ligand generation. **a**, Experimentally identified fragments bound to separate subsites within the ATPase pocket of Hsp90 (PDB code 3HZ1). **b**, Starting fragments after removing methyl ester group. **c**, Crystal structure of the inhibitor reported in ref. 20 (PDB code 3HZ5). **d**, DiffLinker generated molecule that reproduces the reference inhibitor. **e**,**f**, Distributions of GNINA (**e**) and Vina (**f**) scores for unique samples generated by DiffLinker conditioned (green, $n = 485$) and unconditioned (blue, $n = 166$ for GNINA (**e**) and $n = 66$ for Vina (**f**) after removing outliers) on the target pocket. Red solid line depicts the score of the reference inhibitor reported in ref. 20. Magenta dashed lines represent scores for three DiffLinker samples that recover the reference inhibitor. For GNINA scores, higher values represent higher affinities.

Case studies

Here we demonstrate how DiffLinker can be integrated in real-world pipelines for drug design and discuss three scenarios taken from the literature: fragment-based design of Hsp90 and IMPDH inhibitors, and scaffold hopping for improving selectivity for JNKs.

Design of Hsp90 inhibitors. Hsp90 is a molecular chaperone involved in enabling the folding of numerous proteins, including those participating in oncogenic transformations. The authors of ref. 20 proposed a potent inhibitor for Hsp90 using fragment-based screening and structure-based design techniques. First, using biochemical screening followed by X-ray crystallography, ref. 20 identified fragments bound to separate subsites within the ATPase pocket of Hsp90 (Protein Data Bank (PDB) code 3HZ1), as shown in Fig. 3a. The authors report that by linking these fragments, compounds with more than 1,000-fold improvement in affinity over the initial fragment hit were generated. A crystal structure of the reported inhibitor bound to Hsp90 is shown in Fig. 3c.

In our experiment, we follow the overall procedure reported in ref. 20 and integrate DiffLinker in the fragment-linking step.

https://doi.org/10.1038/s42256-024-00815-9

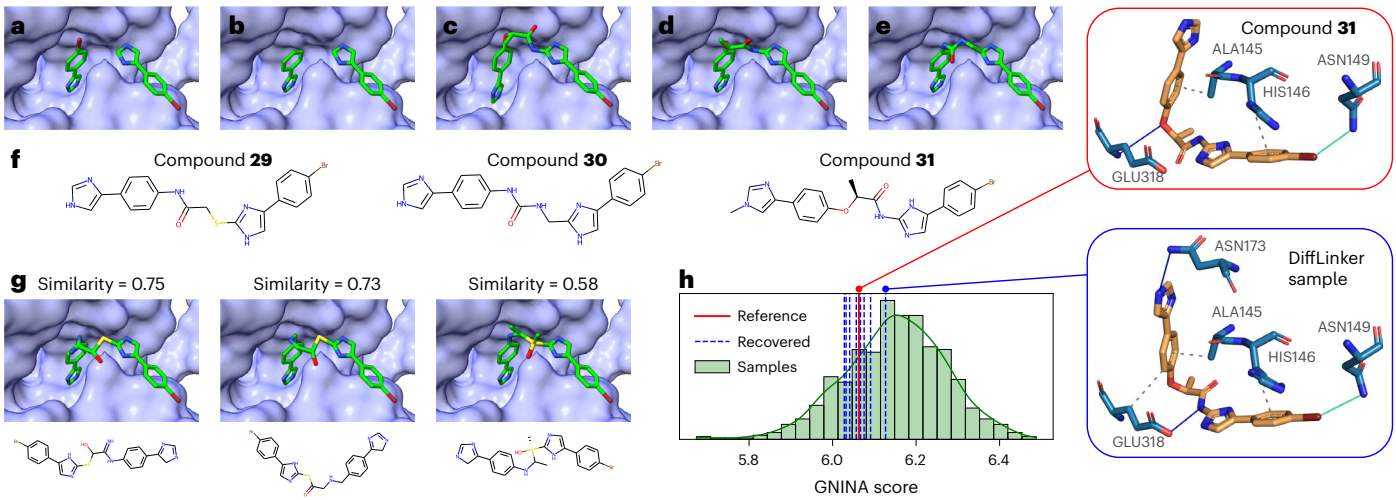


Fig. 4 | Case study for fragment-based design of IMPDH inhibitors. **a**, Experimentally identified fragment hits by ref. 6 (PDB code 5OU2), with IMPDH shown in blue. **b**, Starting fragments after removing bromine. **c**, Crystal structure of the most potent inhibitor (compound **31**) reported in ref. 6 (PDB code 5OU3). **d**, DiffLinker sample that reproduces compound **31**. **e**, DiffLinker sample that reproduces compound **30**. **f**, Chemical structures of three potent inhibitors reported in ref. 6. **g**, Three DiffLinker samples with highest chemical similarity (Tanimoto distance) to compound **29**. **h**, Distribution of GNINA scores for unique samples (n = 800) generated by DiffLinker. Red solid line depicts the score of experimentally validated compound **31**. Blue dashed lines represent scores for eight DiffLinker samples that recover compound **31**. On the right, we represent the interactions between a molecule (top, reference; bottom, DiffLinker sample with the highest score) and the target IMPDH pocket computed by PLIP[22].

We consider two experimentally observed fragments bound to the ATPase pocket of Hsp90 (Fig. 3a), remove the methyl ester group from one of them (Fig. 3b) and generate 1,000 linkers using the pocket-conditioned model. To predict the size of the linker, we use a graph neural network (GNN) trained on the ZINC dataset. We note that the inhibitor reported in ref. 20 was not included in the Pockets and ZINC training sets. Additionally, none of the relevant crystal structures was included in the Pockets training set.

DiffLinker successfully recovers the inhibitor reported in ref. 20. Among 1,000 samples, three have the same chemical structure as the reference ligand. The molecule with the highest SC_{RDKit} score, which captures the highest geometric and chemical similarity to the reference compound, is shown in Fig. 3d.

Additionally, we generated 1,000 linkers with the model trained on the ZINC dataset (without pocket conditioning). Having the reference molecule and samples generated by two different DiffLinker models, we scored the protein-ligand complexes with GNINA[19] and Vina[21], as implemented in the GNINA package. We use GNINA and Vina as proxies for binding energy, as these methods are fast, and their predictions present some level of correlation with experimentally determined binding affinities, as shown in Extended Data Fig. 3 and discussed in more detail in Supplementary Information. As shown in Fig. 3e,f, docking scores of the molecules sampled by the model conditioned on the protein pocket are improved relative to those by DiffLinker trained on the ZINC dataset only (P values of a two-sided Kolmogorov–Smirnov test are 1.832×10^{-124} and 1.460×10^{-175} for GNINA and Vina scores, respectively). Notably, some of the sampled molecules have docking scores superior to those of the best pose of the reference compound. We additionally note that docking scores of all three DiffLinker samples that reproduce the reference inhibitor molecule are comparable with scores of the reference, as depicted by dashed and solid lines in Fig. 3e,f.

Design of IMPDH inhibitors. IMPDH is an attractive tuberculosis drug target which plays an important role in de novo synthesis of guanine nucleotides. Using fragment-based screening and structure-based design techniques, ref. 6 identified potent IMPDH inhibitors. Having started with two initial fragment hits shown in Fig. 4a (PDB code 5OU2), the authors reported three successful compounds obtained through fragment linking. These compounds are represented in Fig. 4f. Notably, the authors achieved more than 1,000-fold improvement in affinity over the initial fragment hits with the most potent candidate, compound **31**. The crystal structure of the protein complexed with the compound is shown in Fig. 4c (PDB code 5OU3).

We generated 1,000 linkers of length 5 and 6 using the pocket-conditioned model. DiffLinker recovered compound **30** and compound **31**, which are some of the most potent inhibitors among those reported in ref. 6. Sampled molecules that reproduce these compounds with the highest SC_{RDKit} score are shown in Fig. 4d,e. Even though DiffLinker did not reproduce compound **29**, it generated similar molecules in terms of Tanimoto distance. In Fig. 4g, we provide the top three closest samples with their Tanimoto distances.

Finally, following our previous experiment with Hsp90 inhibitors, we compute GNINA docking scores for DiffLinker samples and represent them also relative to the score of the reference crystallized compound **31** in Fig. 4h. Vina scores for the same molecules are provided in Extended Data Fig. 4a. We highlight the scores of eight samples that reproduce compound **31**. We note that all eight samples show similar docking scores to the reference crystal structure. To better understand the differences between the reference and sampled molecules, we computed the interactions between the reference molecule and the IMPDH pocket residues using PLIP[22]. We also computed the interactions between the DiffLinker sample that reproduces compound **31** with the highest docking score and the target pocket. As shown in Fig. 4h, the reference and sampled linkers interact differently with the pocket. While the reference linker interacts with the pocket through the acceptor oxygen that forms a hydrogen bond with the nitrogen of Glu-318, the sampled linker interacts with the pocket through the nitrogen donor that forms a hydrogen bond with the oxygen of Glu-318. This difference in the interactions and docking scores suggests that our model explores the space of possible ligand conformations trying to find favourable interactions with the protein pocket.

Improving selectivity of JNK inhibitors. JNKs constitute an important protein family of mitogen-activated protein kinases that regulate various cellular processes, including cell proliferation, apoptosis, autophagy and inflammation[23]. Kamenecka et al.[24] designed

https://doi.org/10.1038/s42256-024-00815-9

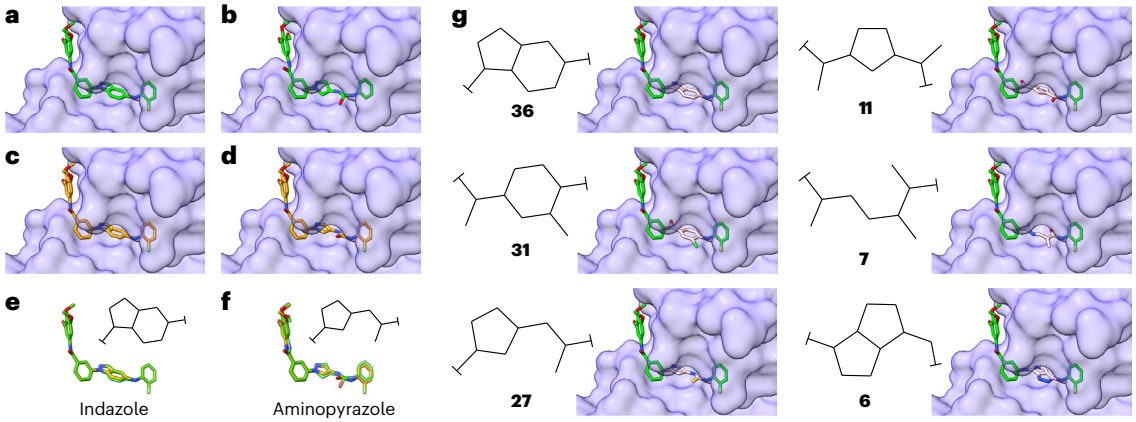


Fig. 5 | Exploring chemical diversity for improving selectivity of JNK inhibitors. a,**b**, Crystal structures of compounds with indazole (PDB code 3FI3) (**a**) and aminopyrazole (PDB code 3FI2) (**b**) scaffolds, respectively, with JNK3 shown in blue. **c**,**d**, DiffLinker samples that reproduce indazole (**c**) and aminopyrazole (**d**) scaffolds of the compounds reported in ref. 24. **e**,**f**, Overlay of real (green) and sampled (orange) indazole (**e**) and aminopyrazole (**f**) structures. **g**, Six distinct linking moieties along with the corresponding exemplary DiffLinker samples. For each of the shown topologies, we also provide the number of unique chemical structures employing this topology to demonstrate how frequently each moiety was sampled.

JNK3-selective inhibitors that had more than 1,000-fold selectivity over p38, another closely related mitogen-activated protein kinase family member. Starting with the indazole class of compounds and by changing the compound's scaffold, the authors obtained an aminopyrazole scaffold that resulted in compounds with over 2,800-fold JNK-selectivity. Crystal structures of compounds with indazole and aminopyrazole scaffolds reported in ref. 24 are shown in Fig. 5a,b.

Here, we study the ability of DiffLinker to generate a set of diverse scaffolds. We input the structure of fragments with the missing core (taken from indazole crystal structure, PDB code 3FI3) and generate 1,000 scaffolds with 8 and 9 atoms using our pocket-conditioned model. DiffLinker recovered both indazole and aminopyrazole scaffolds, as observed in the ground-truth compounds. Following the previous experiments, we provide docking scores of DiffLinker samples in Extended Data Fig. 4b,c. Sampled molecules that reproduce compounds reported in ref. 24 with the highest SC_{RDKit} score are shown in Fig. 5c,d respectively. Overlay of real (green) and sampled (orange) indazole and aminopyrazole structures is shown in Fig. 5e,f. In addition, we identified 238 unique topologies of the generated scaffolds, which suggests that DiffLinker is able to extensively explore the space of potentially relevant scaffolds through the sampling of linker regions. Six most common distinct topologies along with the exemplary DiffLinker samples are represented in Fig. 5g. For each of the represented moieties, we also provide the number of unique sampled chemical structures employing this topology. While none of the relevant crystal structures was included in the training set, we note that indazole and aminopyrazole moieties are among the most commonly sampled ones.

Discussion

In this work, we introduced DiffLinker, a new E(3)-equivariant 3D conditional diffusion model for molecular linker design. Our method showed several desirable and practical features that have the potential to help accelerate the development of prospective drug candidates using FBDD strategies.

However, several aspects remain for further improvement; for instance, chemical validity of the sampled compounds is a necessary requirement for a successful molecule design method. As explained in Supplementary Information, lower validity of DiffLinker samples is caused by the fact that our model generates raw point clouds, which are then processed by OpenBabel[25] to compute covalent bonds. In contrast, other methods construct bonds and employ valency rules at each generation step explicitly. While our model clearly demonstrates the ability to effectively learn fundamental chemistry from the raw geometric data, several options that could be beneficial remain to be tested. One possible direction is incorporating the information on covalent bonds to the model (that is, adding edge features) and generating chemical bonds along with atom types and coordinates.

Another important property of the sampled molecules is high SA. This quality plays a crucial role in real-world drug discovery pipelines. In the current work, we report SA score[26] and show that DiffLinker produces more synthetically accessible molecules, compared to other linker design methods; however, there still remains room for improvement. While the current model gets a notion of SA only from the raw training data, one may explicitly employ this concept in the method by guiding the denoising process with, for instance, SA score[26].

While DiffLinker effectively suggests diverse and valid chemical structures in tasks like fragment linking and scaffold hopping, we have observed that generating relevant linkers for PROTAC-like molecules poses a greater challenge. The main difference between these problems lies on the linker length and the distance between the input fragments. While the average linker size in our training sets is around 8 atoms (5 for ZINC, 10 for GEOM, 10 for Pockets), a typical linker in a PROTAC varies between 12 and 20 atoms[27]. It means that the distribution of linkers in PROTACs has different characteristics compared to the distributions of linkers provided in our training sets. Therefore, to improve the performance of DiffLinker in PROTAC design, one may consider retraining the model using more suitable PROTAC data.

Finally, although the current work focuses on molecular linker design, DiffLinker can facilitate other stages of fragment-based drug discovery, as there are no fundamental limitations in applying our model to molecule growing or de novo generation of molecular fragments.

Methods

Here we describe DiffLinker, an E(3)-equivariant diffusion model for generating molecular linkers conditioned on 3D fragments. First, we provide an overview of diffusion models and discuss the data representation and equivariance. Next, we formulate equivariance requirements for the underlying denoising distributions and propose an appropriate learnable dynamic function. We also discuss the strategy of sampling the size of a linker and conditioning on protein pockets. Finally, we provide information on datasets, evaluation methodology, baselines and sampling efficiency of DiffLinker. The full linker generation workflow is schematically represented in Fig. 1, and the pseudocode of DiffLinker's training and sampling procedures is provided in Supplementary Information.

Diffusion models

Diffusion models[28] are a class of generative methods that consist of a 'diffusion process', which progressively transforms a data point $\mathbf{x}$ into noise and a 'generative denoising model', which approximates the reverse of the diffusion process.

In this paper, we consider Gaussian diffusion: at a time step $t = 0, \ldots, T$, the conditional distribution of the intermediate data state $\mathbf{z}_t$ given previous state $\mathbf{z}_{t-1}$ is defined by the multivariate normal distribution,

$$q(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t; \bar{\alpha}_t\mathbf{z}_{t-1}, \bar{\sigma}_t^2 I), \tag{1}$$

where I is an identity matrix, parameter $\bar{\alpha}_t \in \mathbb{R}^+$ controls how much signal is retained and parameter $\bar{\sigma}_t \in \mathbb{R}^+$ controls how much noise is added. The noise model is chosen to be Markovian, such that the probability of a trajectory can be written as:

$$q(\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_T|\mathbf{x}) = q(\mathbf{z}_0|\mathbf{x}) \prod_{t=1}^{T} q(\mathbf{z}_t|\mathbf{z}_{t-1}), \tag{2}$$

where $\mathbf{z}_T$ is the data state at time step T. As the distribution q is normal, a simple formula for the distribution of $\mathbf{z}_t$ given $\mathbf{x}$ can be derived:

$$q(\mathbf{z}_t|\mathbf{x}) = \mathcal{N}(\mathbf{z}_t|\alpha_t\mathbf{x}, \sigma_t^2 I), \tag{3}$$

where $\bar{\alpha}_t = \alpha_t/\alpha_{t-1}$ and $\bar{\sigma}_t^2 = \sigma_t^2 - \bar{\alpha}_t^2\sigma_{t-1}^2$. This closed-form expression shows that noise does not need to be added iteratively to $\mathbf{x}$ to achieve an intermediate state $\mathbf{z}_t$.

Another key property of Gaussian noise is that the reverse process of the diffusion, referred to as the true denoising process, also admits a closed-form solution when conditioned on the original data point $\mathbf{x}$:

$$q(\mathbf{z}_{t-1}|\mathbf{x}, \mathbf{z}_t) = \mathcal{N}(\mathbf{z}_{t-1}; \boldsymbol{\mu}_t(\mathbf{x}, \mathbf{z}_t), \varsigma_t^2 I), \tag{4}$$

where distribution parameters $\boldsymbol{\mu}_t$ and ς_t can be derived analytically:

$$\boldsymbol{\mu}_t(\mathbf{x}, \mathbf{z}_t) = \frac{\bar{\alpha}_t\sigma_{t-1}^2}{\sigma_t^2}\mathbf{z}_t + \frac{\alpha_s\bar{\sigma}_t^2}{\sigma_t^2}\mathbf{x} \text{ and } \varsigma_t = \frac{\bar{\sigma}_t\sigma_{t-1}}{\sigma_t}. \tag{5}$$

This formula describes that if a diffusion trajectory starts at $\mathbf{x}$ and ends at $\mathbf{z}_T$, then the expected value of any intermediate state is an interpolation between $\mathbf{x}$ and $\mathbf{z}_T$.

The second component of a diffusion model is the generative denoising process, which aims to invert the diffusion trajectory by approximating the original data point $\mathbf{x}$ using a neural network. The generative transition distribution is then defined as:

$$p(\mathbf{z}_{t-1}|\mathbf{z}_t) = q(\mathbf{z}_{t-1}|\hat{\mathbf{x}}, \mathbf{z}_t), \tag{6}$$

where $\hat{\mathbf{x}}$ is an approximation of the data point $\mathbf{x}$ computed by a neural network φ. Instead of predicting $\mathbf{x}$ directly, ref. 29 has empirically shown that it is more effective to first predict the Gaussian noise $\hat{\boldsymbol{\epsilon}}_t = \varphi(\mathbf{z}_t, t)$ and then estimate $\hat{\mathbf{x}}$ based on equation (3):

$$\hat{\mathbf{x}} = (1/\alpha_t)\mathbf{z}_t - (\sigma_t/\alpha_t)\hat{\boldsymbol{\epsilon}}_t. \tag{7}$$

The neural network is trained to maximize an evidence lower bound on the likelihood of the data under the model. Up to a prefactor that depends on t, this objective is equivalent to the mean squared error between predicted and true noise[29,30]. Therefore, we use the simplified objective $\mathcal{L}(t) = ||\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_t||^2$ that can be optimized by mini-batch gradient descent using an estimator $\mathbb{E}_{t\sim\mathbf{u}(0,\ldots,T)}[T\mathcal{L}(t)]$.

Finally, once the network is trained, it can be used to sample new data points. For this purpose, one first samples the Gaussian noise: $\mathbf{z}_T \sim \mathcal{N}(0, I)$. Then, for $t = T, \ldots, 1$, one should iteratively sample $\mathbf{z}_{t-1} \sim p(\mathbf{z}_{t-1}|\mathbf{z}_t)$ and finally sample $\mathbf{x} \sim p(\mathbf{x}|\mathbf{z}_0)$, where $\mathbf{z}_0$ is the data state at the time step $t = 0$.

Molecule representation and equivariance

In our model, molecules are represented as 3D atomic point clouds. A molecule is represented by the coordinates of its M atoms $\mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_M) \in \mathbb{R}^{M\times 3}$ and their corresponding feature vectors, $\mathbf{h} = (\mathbf{h}_1, \ldots, \mathbf{h}_M) \in \mathbb{R}^{M\times \mathrm{nf}}$, which are one-hot encoded atom types. We refer to this point cloud as $\mathbf{x} = [\mathbf{r}, \mathbf{h}]$.

While atomic coordinates are continuous, atom types are discrete variables that need to be handled differently in our diffusion model. Instead of using categorical diffusion models[31,32], we use a simpler strategy[33] that lifts the atom types to a continuous space using one-hot encoding and adding Gaussian noise. The continuous values are then converted back to discrete values through argmax over the different categories during the final transition from $\mathbf{z}_0$ to $\mathbf{x}$. For more details on the structure of the final transition distribution $p(\mathbf{x}|\mathbf{z}_0)$ and likelihood computation, we refer the reader to ref. 33.

To process 3D molecules efficiently, the data symmetries need to be respected. In this work, we consider the Euclidean group E(3) that comprises translations, rotations and reflections of $\mathbb{R}^3$ and the orthogonal group O(3) that includes rotations and reflections of $\mathbb{R}^3$. A function f is E(3)-equivariant if for any point cloud $\mathbf{x}$, orthogonal matrix $R \in \mathbb{R}^{3\times 3}$ and translation vector $\mathbf{t} \in \mathbb{R}^3$ we have: $f(R\mathbf{x} + \mathbf{t}) = Rf(\mathbf{x}) + \mathbf{t}$. Note that for simplicity, we use notation $R\mathbf{x} + \mathbf{t} = [(R\mathbf{r}_1 + \mathbf{t}, \ldots, R\mathbf{r}_M + \mathbf{t})^\top, \mathbf{h}]$. A conditional distribution $p(\mathbf{x}|\mathbf{y})$ is E(3)-equivariant if for any point clouds $\mathbf{x}$, $\mathbf{y}$, $p(R\mathbf{x} + \mathbf{t}|R\mathbf{y} + \mathbf{t}) = p(\mathbf{x}|\mathbf{y})$. Finally, a function f and a distribution p are O(3)-equivariant if $f(R\mathbf{x}) = Rf(\mathbf{x})$ and $p(R\mathbf{x}|R\mathbf{y}) = p(\mathbf{x}|\mathbf{y})$, respectively. We call the function f translation invariant if $f(\mathbf{x} + \mathbf{t}) = f(\mathbf{x})$.

Equivariant 3D conditional diffusion model

Unlike other diffusion models for molecule generation[33,34], our method is conditioned on three-dimensional data. More specifically, we assume that each point cloud $\mathbf{x}$ has a corresponding 'context' $\mathbf{u}$, which is another point cloud consisting of all input fragments and (optionally) protein pocket atoms that remain unchanged throughout the diffusion and denoising processes, as shown in Fig. 1. Hence, we consider the generative process from equation (6) to operate on point cloud $\mathbf{x}$ while being conditioned on the fixed corresponding context:

$$p(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{u}) = q(\mathbf{z}_{t-1}|\hat{\mathbf{x}}, \mathbf{z}_t), \text{ where } \hat{\mathbf{x}} = (1/\alpha_t)\mathbf{z}_t - (\sigma_t/\alpha_t)\varphi(\mathbf{z}_t, \mathbf{u}, t). \tag{8}$$

The presence of a 3D context puts additional requirements on the generative process, as it should be equivariant to its transformations.

Proposition 1. Consider a prior noise distribution $p(\mathbf{z}_T|\mathbf{u}) = \mathcal{N}(\mathbf{z}_T; \boldsymbol{\mu}, I)$, where $\boldsymbol{\mu} = [f(\mathbf{z}_T), \mathbf{0}] \in \mathbb{R}^{M\times(3+\mathrm{nf})}$, and $f : \mathbb{R}^{M\times(3+\mathrm{nf})} \to \mathbb{R}^{M\times 3}$ is a function operating on 3D point clouds. Consider transition distributions $p(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{u}) = q(\mathbf{z}_{t-1}|\hat{\mathbf{x}}, \mathbf{z}_t)$, where q is an isotropic Gaussian and $\hat{\mathbf{x}}$ is an approximation computed by the neural network φ according to equation (8). Let the conditional denoising probabilistic model p be a Markov chain defined as

$$p(\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_T|\mathbf{u}) = p(\mathbf{z}_T|\mathbf{u}) \prod_{t=1}^{T} p(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{u}). \tag{9}$$

If f is O(3)-equivariant and φ is equivariant to joint O(3)-transformations of $\mathbf{z}_t$ and $\mathbf{u}$, then $p(\mathbf{z}_0|\mathbf{u})$ is O(3)-equivariant.

The choice of the function f highly depends on the problem being solved and the available priors. In our experiments, we consider two cases. First, following ref. 9, we make use of the information about atoms that should be connected by the linker. We call these atoms 'anchors' and define $f(\mathbf{u})$ as the anchors' centre of mass. However, in a real-world scenario, it is unlikely to be known which atoms should be the anchors. Here we define $f(\mathbf{u})$ as the centre of mass of the whole

context **u**. We should note that although function f computes a single point in 3D, it outputs its coordinate vector repeated M times along the first dimension (because the noise is further sampled independently for each atom of the point cloud).

We note that the probabilistic model p is not equivariant to translations, as shown in Supplementary Information. To overcome this issue, we construct the network φ to be translation invariant. Then, instead of sampling the initial coordinates noise from $\mathcal{N}(f(\mathbf{u}), I)$ we centre the data at $f(\mathbf{u})$ and sample from $\mathcal{N}(\mathbf{0}, I)$. This makes the generative process independent of translations.

Dynamics

The learnable function φ that models the dynamics of the diffusion model takes as input a noisy version of the linker $\mathbf{z}_t$ at time t and the context **u**. These two parts are modelled as a single fully connected graph where nodes are represented by coordinates **r** and feature vectors **h** that include atom types, time t fragment flags and (optionally) anchor flags. The predicted noise $\hat{\boldsymbol{\epsilon}}$ includes coordinate and feature components: $\hat{\boldsymbol{\epsilon}} = [\hat{\boldsymbol{\epsilon}}^{\mathrm{r}}, \hat{\boldsymbol{\epsilon}}^{\mathrm{h}}]$.

The neural network is built upon the E(3)-equivariant GNN (EGNN)[35]. EGNN consists of the composition of equivariant graph convolutional layers (EGCL) $\mathbf{r}^{l+1}, \mathbf{h}^{l+1} = \mathrm{EGCL}[\mathbf{r}^l, \mathbf{h}^l]$, which are defined as

$$\mathbf{m}_{ij} = \phi_{\mathrm{e}}(\mathbf{h}_i^l, \mathbf{h}_j^l, d_{ij}^2), \quad \mathbf{h}_i^{l+1} = \phi_{\mathrm{h}}\left(\mathbf{h}_i^l, \sum_{j \neq i} \mathbf{m}_{ij}\right), \quad \mathbf{r}_i^{l+1} = \mathbf{r}_i^l + \phi_{\mathrm{vel}}(\mathbf{r}^l, \mathbf{h}^l, i), \quad (10)$$

where $d_{ij} = \| \mathbf{r}_i^l - \mathbf{r}_j^l \|$ and ϕ_{e} and ϕ_{h} are learnable functions parametrized by fully connected neural networks (see Supplementary Information for details).

The latter update for the node coordinates is computed by the learnable function ϕ_{vel}. Note that our graph includes both a noisy linker $\mathbf{z}_t$ and a fixed context **u**, and φ is intended to predict the noise that should be subtracted from the coordinates and features of $\mathbf{z}_t$. Therefore, it is natural to keep the context coordinates unchanged when computing dynamics and to apply non-zero displacements only to the linker part at each EGCL step. Hence, we model the linker node displacements as follows,

$$\phi_{\mathrm{vel}}(\mathbf{r}^l, \mathbf{h}^l, i) = \sum_{j \neq i} \frac{\mathbf{r}_i^l - \mathbf{r}_j^l}{d_{ij} + 1} \phi_{\mathrm{r}}(\mathbf{h}_i^l, \mathbf{h}_j^l, d_{ij}^2), \quad (11)$$

where ϕ_{r} is a learnable function parametrized by a fully connected neural network. Displacements for the context nodes are always set to **0**.

The equivariance of convolutional layers is achieved by construction. The messages ϕ_{e} and the node updates ϕ_{h} depend only on scalar node features and distances between nodes that are E(3)-invariant. Coordinate updates ϕ_{vel} additionally depend linearly on the difference between coordinate vectors $\mathbf{r}_i^l$ and $\mathbf{r}_j^l$, which makes them E(3)-equivariant.

After the sequence of EGCLs is applied, we have an updated graph with new node coordinates $\hat{\mathbf{r}} = [\mathbf{u}^{\mathrm{r}}, \hat{\mathbf{z}}_t^{\mathrm{r}}]$ and new node features $\hat{\mathbf{h}} = [\hat{\mathbf{u}}^{\mathrm{h}}, \hat{\mathbf{z}}_t^{\mathrm{h}}]$. Since we are interested only in the linker-related part, we discard the coordinates and features of context nodes and consider the tuple $[\hat{\mathbf{z}}_t^{\mathrm{r}}, \hat{\mathbf{z}}_t^{\mathrm{h}}]$ to be the EGNN output.

To make the function φ invariant to translations, we subtract the initial coordinates from the coordinate component of the EGNN output following ref. 33:

$$\hat{\boldsymbol{\epsilon}} = [\hat{\boldsymbol{\epsilon}}^{\mathrm{r}}, \hat{\boldsymbol{\epsilon}}^{\mathrm{h}}] = \varphi(\mathbf{z}_t, \mathbf{u}, t) = \mathrm{EGNN}(\mathbf{z}_t, \mathbf{u}, t) - [\mathbf{z}_t^{\mathrm{r}}, \mathbf{0}]. \quad (12)$$

Linker-size prediction

To predict the size of the missing linker between a set of fragments, we represent fragments as a fully connected graph with one-hot encoded atom types as node features and distances between nodes as edge features. From this, a separately trained GNN (see Supplementary Information for details) produces probabilities for the linker size. Our assumption is that relative fragment positions and orientations, along with atom types, contain all the information essential for predicting the most likely size of the prospective linker. When generating a linker, we first sample its size with the predicted probabilities from the categorical distribution over the list of linker sizes seen in the training data, as shown in Fig. 1.

Protein pocket conditioning

In real-world FBDD applications, it often occurs that fragments are obtained by experimental screening followed by structural determination[3] or selected and docked into a target protein pocket[36]. To propose a drug candidate molecule, the fragments have to be linked. When generating the linker, one should take the surrounding pocket into account and construct a linker that is sterically compatible with the protein pocket and, if possible, also contributes to a potent binding affinity. To add pocket conditioning to DiffLinker, we represent a protein pocket as an atomic point cloud and consider it as a part of the context **u**. We also extend node features with an additional binary flag marking atoms that belong to the protein pocket. Finally, as the new context point cloud contains much more atoms, we modify the joint representation of the data point $\mathbf{z}_t$ and the context **u** that are passed to the neural network φ. Instead of considering fully connected graphs, we assign edges between nodes based on a 4 Å distance cutoff, as it makes the resulting graphs less dense and counterbalances the increase in the number of nodes.

Datasets

ZINC. We follow ref. 9 and consider a subset of 250,000 molecules randomly selected by Gómez-Bombarelli et al.[37] from the ZINC database[13]. First, we generate 3D conformers using RDKit[38] and define a reference 3D structure for each molecule by selecting the lowest energy conformation. Then, these molecules are fragmented by enumerating all double cuts of acyclic single bonds that are not within functional groups. The resulting splits are filtered by the number of atoms in the linker and fragments, SA[26], ring aromaticity and pan-assay interference compounds (PAINS)[39] criteria. One molecule can therefore result in various combinations of two fragments with a linker between. The resulting dataset is randomly split into train (438,610 examples), validation (400 examples) and test (400 examples) sets. Atom types considered for this dataset are: C, O, N, F, S, Cl, Br and I.

CASF. Another evaluation benchmark used by ref. 9 is taken from the CASF-2016 dataset[14]. In contrast to ZINC, where molecule conformers were generated computationally, CASF includes experimentally verified 3D conformations. Following the same preprocessing procedures as for the ZINC dataset, ref. 9 obtained an additional test set of 309 examples, which we use in our work. Atom types considered for this dataset are: C, O, N, F, S, Cl, Br and I.

GEOM. ZINC and CASF datasets used in previous works only contain pairs of fragments. However, real-world applications often require connecting more than two fragments with one or more linkers[36]. To address this case, we construct a new dataset based on GEOM molecules[15], which we decompose into three or more fragments with one or two linkers connecting them. To achieve such splits, we use RDKit implementations of two fragmentation techniques—a matched molecular pair analysis (MMPA) based algorithm[40] and BRICS[41]—and combine results, removing duplicates. Overall, we obtain 41,907 molecules and 285,140 fragmentations that are randomly split in train (282,602 examples), validation (1,250 examples) and test (1,288 examples) sets. Atom types considered for this dataset are: C, O, N, F, S, Cl, Br, I and P.

Pockets dataset. To assess the ability of DiffLinker to generate valid linkers given additional information about protein pockets, we use the protein-ligand dataset curated by Schneuing et al.[18] from Binding

MOAD[16]. To define pockets, we consider amino acids that have at least one atom closer than 6 Å to any atom of the ligand. All atoms belonging to these residues constitute the pocket. We split molecules into fragments using RDKit's implementation of an MMPA-based algorithm[40]. We randomly split the resulting data into train (185,678 examples), validation (490 examples) and test (566 examples) sets, taking into account Enzyme Commission numbers of the proteins. Atom types considered for this dataset are: C, O, N, F, S, Cl, Br, I and P.

Metrics

First, we report several chemical properties of the generated molecules that are especially important in drug design applications: average QED[42], average SA[26] and average number of rings in the linker. Next, following ref. 9, we measure validity, uniqueness and novelty of the samples. We then determine if the generated linkers are consistent with the 2D filters used to produce the ZINC training set. These filters are explained in detail in Supplementary Information. In addition, we record the percentage of the original molecules that were recovered by the generation process. To compare the 3D shapes of the sampled and ground-truth molecules, we estimate the root mean squared deviation (r.m.s.d.) between the generated and real linker coordinates in the cases where true molecules are recovered. We also compute the SC_{RDKit} metric that evaluates the geometric and chemical similarity between the ground-truth and generated molecules[43,44].

Baselines

We compare our method with DeLinker[9] and 3DLinker[11] on the ZINC test set and with DeLinker on the CASF dataset. We adapted DeLinker and 3DLinker to connect more than two fragments (see Supplementary Information for details) and evaluate its performance on the GEOM dataset. To obtain 3D conformations for the molecules generated by DeLinker, we applied a pretrained ConfVAE[45] followed by a force field relaxation procedure MMFF[46]. For all methods, including ours, we generate linkers with the ground-truth size unless explicitly noted otherwise. To obtain SMILES representations of atomic point clouds generated by our models, we utilize OpenBabel[25] to compute covalent bonds between atoms. We also use OpenBabel to rebuild covalent bonds for the molecules in test sets to correctly compute the recovery rate, r.m.s.d. and SC_{RDKit} scores for our models. In ZINC and CASF experiments, we sample 250 linkers for each input pair of fragments. For the GEOM dataset and in experiments with pocket conditioning, we sample 100 linkers for each input set of fragments. In our experiments with protein pockets as additional context, we compare DiffLinker with two de novo generation methods, ResGen[17] and DiffSBDD[18]. In both cases, we obtained trained model weights from the publicly available repositories and sample molecules with the default settings as described in the online documentation. We sample 120 new molecules for each target with a version of DiffSBDD that uses the full-atomic pocket context. ResGen produced 100 molecules per target on average (minimum 19, maximum 149).

Sampling

For all the experiments discussed in the main text, we sampled with the same number of denoising steps $T = 500$ as used in training. Sampling time for all the datasets is provided in Supplementary Table 10. Although the time reported in Supplementary Table 10 is more than affordable for applying our method in practice, we explored the capability of DiffLinker to sample even faster without performance loss. Following ref. 47, we conducted an additional evaluation of DiffLinker with the reduced number of denoising steps $T = 500$ in sampling, considering $T/2$, $T/5$, $T/10$, $T/20$, $T/50$ and $T/100$ values. Extended Data Fig. 5 shows how the performance metrics obtained on the ZINC test set depend on the number of denoising steps performed in sampling. In all cases, we used DiffLinker pretrained on ZINC with $T = 500$ denoising steps. As shown in Extended Data Fig. 5, our model is robust to a notable reduction of the number of denoising steps in sampling resulting in 10-fold gain in sampling speed without any performance degradation. Effectively, one can reduce the sampling speed from 0.365 to 0.036 seconds per molecule with no substantial performance metrics loss.

Software

Dataset processing was done in Python (v.3.10.5) using RDKit (v.2022.03.2) for generating molecular conformers and splitting them in fragments and linkers, scikit-learn (v.1.0.1) for splitting datasets and BioPython (v.1.79) for processing protein structures. The MMPA-based algorithm[40] and BRICS[41] used for molecule fragmentation, as well as force field relaxation procedure MMFF[46], are components of The RDKit package. Central packages used for writing DiffLinker as well as training and sampling scripts include NumPy (v.1.22.3), PyTorch (v.1.11.0), PyTorch Lightning (v.1.6.3), WandB (v.0.12.16), RDKit (v.2022.03.2) and OpenBabel (v.3.0.0). For sampling molecules with baseline methods, we used pretrained models and sampling scripts available at the corresponding repositories: 3DLinker (https://github.com/YinanHuang/3DLinker)[48], DeLinker (https://github.com/oxpig/DeLinker)[49], DiffSBDD (https://github.com/arneschneuing/DiffSBDD)[50] and ResGen (https://github.com/HaotianZhangAI4Science/ResGen)[51]. None of these repositories provide version releases. Data analysis and vizualization was done in Python (v.3.10.5) using RDKit (v.2022.03.2), imageio (v.2.19.2), NetworkX (v.2.8.4), SciPy (v.1.7.3), matplotlib (v.3.5.2), seaborn (v.0.11.2) and GNINA (v.1.0.3, https://github.com/gnina/gnina)[52].

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

All the processed datasets, as well as pretrained models, are available at Zenodo. Datasets: ZINC (https://doi.org/10.5281/zenodo.7121271)[53], CASF (https://doi.org/10.5281/zenodo.7121264)[54], GEOM (https://doi.org/10.5281/zenodo.7121278)[55], Pockets (https://doi.org/10.5281/zenodo.7121280)[56]. Models: https://doi.org/10.5281/zenodo.7775568 (ref. 57). Molecules used in the ZINC dataset are available at the ZINC database (https://zinc.docking.org/). Molecules used in the CASF dataset were taken from the CASF-2016 benchmark package (http://www.pdbbind.org.cn/download/CASF-2016.tar.gz) of the PDBbind database (http://www.pdbbind.org.cn/). Molecules used in the GEOM dataset are available at the repository of the original GEOM dataset (https://github.com/learningmatter-mit/geom)[58]. Molecules used in the Pockets dataset were taken from Binding MOAD (http://www.bindingmoad.org/). Crystal structures of the Hsp90 inhibitor and initially bound fragments are available at Protein Data Bank under the access codes 3HZ5 and 3HZ1, respectively. Crystal structures of the initial fragment hits and the reported inhibitor for IMPDH are available at Protein Data Bank under the access codes 5OU2 and 5OU3, respectively. Crystal structures of JNK inhibitors with indazole and minopyrazole scaffolds are available at Protein Data Bank under the access codes 3FI3 and 3FI2, respectively.

Code availability

The source code of this study is freely available at GitHub (https://github.com/igashov/DiffLinker)[59,60].

References

1. Virshup, A. M., Contreras-García, J., Wipf, P., Yang, W. & Beratan, D. N. Stochastic voyages into uncharted chemical space produce a representative library of all possible drug-like compounds. *J. Am. Chem. Soc.* **135**, 7296–7303 (2013).
2. Erlanson, D. A., Fesik, S. W., Hubbard, R. E., Jahnke, W. & Jhoti, H. Twenty years on: the impact of fragments on drug discovery. *Nat. Rev. Drug Discov.* **15**, 605–619 (2016).

3. Bancet, A. et al. Fragment linking strategies for structure-based drug design. *J. Med. Chem.* **63**, 11420–11435 (2020).
4. Sun, H., Tawa, G. & Wallqvist, A. Classification of scaffold-hopping approaches. *Drug Discovery Today* **17**, 310–324 (2012).
5. Bai, N. et al. Rationalizing PROTAC-mediated ternary complex formation using Rosetta. *J. Chem. Inf. Model.* **61**, 1368–1382 (2021).
6. Trapero, A. et al. Fragment-based approach to targeting inosine-5′-monophosphate dehydrogenase (IMPDH) from Mycobacterium tuberculosis. *J. Med.Chem.* **61**, 2806–2822 (2018).
7. Sheng, C. & Zhang, W. Fragment informatics and computational fragment-based drug design: an overview and update. *Med. Res. Rev.* **33**, 554–598 (2013).
8. Yang, Y. et al. Syntalinker: automatic fragment linking with deep conditional transformer neural networks. *Chem. Sci.* **11**, 8312–8322 (2020).
9. Imrie, F., Bradley, A. R., Schaar, M. & Deane, C. M. Deep generative models for 3D linker design. *J. Chem. Inf. Model.* **60**, 1983–1995 (2020).
10. Imrie, F., Hadfield, T. E., Bradley, A. R. & Deane, C. M. Deep generative design with 3D pharmacophoric constraints. *Chem. Sci.* **12**, 14577–14589 (2021).
11. Huang, Y., Peng, X., Ma, J. & Zhang, M. 3DLinker: an E(3) equivariant variational autoencoder for molecular linker design. In *Proc. 39th International Conference on Machine Learning* (eds Chaudhuri, K. et al.) 9280–9294 (PMLR, 2022).
12. Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* **28**, 31–36 (1988).
13. Irwin, J. J. & Shoichet, B. K. ZINC – a free database of commercially available compounds for virtual screening. *J. Chem. Inf. Model.* **45**, 177–182 (2005).
14. Su, M. et al. Comparative assessment of scoring functions: the CASF-2016 update. *J. Chem. Inf. Model.* **59**, 895–913 (2018).
15. Axelrod, S. & Gómez-Bombarelli, R. GEOM, energy-annotated molecular conformations for property prediction and molecular generation. *Sci. Data* **9**, 185 (2022).
16. Hu, L., Benson, M. L., Smith, R. D., Lerner, M. G. & Carlson, H. A. Binding MOAD (mother of all databases). *Proteins* **60**, 333–340 (2005).
17. Zhang, O. et al. ResGen is a pocket-aware 3D molecular generation model based on parallel multiscale modelling. *Nat. Mach. Intell.* **5**, 1020–1030 (2023).
18. Schneuing, A. et al. Structure-based drug design with equivariant diffusion models. Preprint at https://arxiv.org/abs/2210.13695 (2022).
19. McNutt, A. T. et al. GNINA 1.0: molecular docking with deep learning. *J. Cheminform.* **13**, 43 (2021).
20. Barker, J. J. et al. Discovery of a novel Hsp90 inhibitor by fragment linking. *ChemMedChem* **5**, 1697–1700 (2010).
21. Trott, O. & Olson, A. J. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J.Comput. Chem.* **31**, 455–461 (2010).
22. Salentin, S., Schreiber, S., Haupt, V. J., Adasme, M. F. & Schroeder, M. PLIP: fully automated protein–ligand interaction profiler. *Nucleic Acids Res.* **43**, 443–447 (2015).
23. Chen, J. et al. The roles of c-Jun N-terminal kinase (JNK) in infectious diseases. *Int. J. Mol. Sci.* **22**, 9640 (2021).
24. Kamenecka, T. et al. Structure–activity relationships and X-ray structures describing the selectivity of aminopyrazole inhibitors for c-Jun N-terminal kinase 3 (JNK3) over p38. *J. Biol. Chem.* **284**, 12853–12861 (2009).
25. O'Boyle, N. M. et al. Open Babel: an open chemical toolbox. *J. Cheminform.* **3**, 33 (2011).
26. Ertl, P. & Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *J. Cheminform.* **1**, 8 (2009).
27. Cyrus, K. et al. Impact of linker length on the activity of PROTACs. *Mol. Biosyst.* **7**, 359–364 (2011).
28. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N. & Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proc. 32nd International Conference on Machine Learning* (eds Bach, F. & Blei, D.) 2256–2265 (PMLR, 2015).
29. Ho, J., Jain, A. & Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems 33* (eds Larochelle, H. et al.) 6840–6851 (Curran Associates, 2020).
30. Kingma, D., Salimans, T., Poole, B. & Ho, J. Variational diffusion models. In *Advances in Neural Information Processing Systems 34* (eds Ranzato, M. et al.) 21696–21707 (Curran Associates, 2021).
31. Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P. & Welling, M. Argmax flows and multinomial diffusion: learning categorical distributions. In *Advances in Neural Information Processing Systems 34* (eds Ranzato, M. et al.) 12454–12465 (Curran Associates, 2021).
32. Austin, J., Johnson, D. D., Ho, J., Tarlow, D. & Berg, R. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems 34* (eds Ranzato, M. et al.) 17981–17993 (Curran Associates, 2021).
33. Hoogeboom, E., Satorras, V. G., Vignac, C. & Welling, M. Equivariant diffusion for molecule generation in 3D. In *Proc. 39th International Conference on Machine Learning* (eds Chaudhuri, K. et al.) 8867–8887 (PMLR, 2022).
34. Xu, M. et al. GeoDiff: a geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations* (OpenReview.net, 2022); https://openreview.net/forum?id=PzcvxEMzvQC
35. Satorras, V. G., Hoogeboom, E., Fuchs, F. B., Posner, I. & Welling, M. E(n) equivariant normalizing flows. In *Advances in Neural Information Processing Systems 34* (eds Ranzato, M. et al.) 4181–4192 (Curran Associates, 2021).
36. Igashov, I. et al. Decoding surface fingerprints for protein-ligand interactions. Preprint at *bioRxiv* https://doi.org/10.1101/2022.04.26.489341 (2022).
37. Gómez-Bombarelli, R. et al. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Cent. Sci.* **4**, 268–276 (2018).
38. RDKit: open-source cheminformatics software. *RDKit* https://rdkit.org (2013).
39. Baell, J. B. & Holloway, G. A. New substructure filters for removal of pan assay interference compounds (PAINS) from screening libraries and for their exclusion in bioassays. *J. Med. Chem.* **53**, 2719–2740 (2010).
40. Dossetter, A. G., Griffen, E. J. & Leach, A. G. Matched molecular pair analysis in drug discovery. *Drug Discov. Today* **18**, 724–731 (2013).
41. Degen, J., Wegscheid-Gerlach, C., Zaliani, A. & Rarey, M. On the art of compiling and using 'drug-like' chemical fragment spaces. *ChemMedChem* **3**, 1503–1507 (2008).
42. Bickerton, G. R., Paolini, G. V., Besnard, J., Muresan, S. & Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nat. Chem.* **4**, 90–98 (2012).
43. Putta, S., Landrum, G. A. & Penzotti, J. E. Conformation mining: an algorithm for finding biologically relevant conformations. *J. Med.Chem.* **48**, 3313–3318 (2005).
44. Landrum, G. A., Penzotti, J. E. & Putta, S. Feature-map vectors: a new class of informative descriptors for computational drug discovery. *J.Comput. Aided Mol. Des.* **20**, 751–762 (2006).
45. Xu, M., Luo, S., Bengio, Y., Peng, J. & Tang, J. Learning neural generative dynamics for molecular conformation generation. In *International Conference on Learning Representations* (OpenReview.net, 2021); https://openreview.net/forum?id=pAbm1qfheGk

46. Halgren, T. A. Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94. *J. Comput. Chem.* **17**, 490–519 (1996).
47. Nichol, A. Q. & Dhariwal, P. Improved denoising diffusion probabilistic models. In *Proc. 38th International Conference on Machine Learning* (eds Meila, M. & Zhang, T.) 8162–8171 (PMLR, 2021).
48. Huang, Y. 3DLinker. *GitHub* https://github.com/YinanHuang/3DLinker (2022).
49. Oxford Protein Informatics Group. DeLinker. *GitHub* https://github.com/oxpig/DeLinker (2019).
50. Schneuing, A. DiffSBDD. *GitHub* https://github.com/arneschneuing/DiffSBDD (2022).
51. Zhang, O. ResGen. *GitHub* https://github.com/HaotianZhangAI4Science/ResGen (2022).
52. McNutt, A. et al. gnina. *GitHub* https://github.com/gnina/gnina (2021).
53. Igashov, I. et al. DiffLinker ZINC Dataset. *Zenodo* https://doi.org/10.5281/zenodo.7121271 (2022).
54. Igashov, I. et al. DiffLinker CASF Dataset. *Zenodo* https://doi.org/10.5281/zenodo.7121264 (2022).
55. Igashov, I. et al. DiffLinker GEOM Dataset. *Zenodo* https://doi.org/10.5281/zenodo.7121278 (2022).
56. Igashov, I. et al. DiffLinker Pockets Dataset. *Zenodo* https://doi.org/10.5281/zenodo.7121280 (2022).
57. Igashov, I. et al. DiffLinker Models. *Zenodo* https://doi.org/10.5281/zenodo.7775568 (2022).
58. Axelrod, S. & Gomez-Bombarelli, R. learningmatter-mit/geom. *GitHub* https://github.com/learningmatter-mit/geom (2022)
59. Igashov, I. et al. DiffLinker v.1.0. *GitHub* https://github.com/igashov/DiffLinker (2024).
60. Igashov, I. & Stärk, H. DiffLinker: v1.0 *Zenodo* https://doi.org/10.5281/zenodo.10515727 (2024).

Acknowledgements

We thank Y. Du, J. Southern and V. Oleinikovas for helpful feedback and insightful discussions. 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. C.V. thanks the Swiss Data Science Center for supporting him through the PhD fellowship programme (grant P18-11). M.B. is partially funded by the EPSRC Turing AI World-Leading Research Fellowship (grant EP/X040062/1).

Author contributions

I.I. contributed to the main idea, conceptualization, code and manuscript writing. H.S. contributed to the main idea, code reorganization and docking experiments. C.V. contributed to the mathematical conceptualization of the 3D conditional diffusion model and manuscript writing. A.S. contributed to the experiments with the methods for de novo molecule generation and manuscript writing. V.G.S. and M.W. contributed to instruction and providing essential expertise in Euclidean diffusion models. P.F. and M.B. contributed to manuscript revision and financial support. B.C. contributed to the main idea, experimental design, manuscript revision and funding acquisition.

Funding

Open access funding provided by EPFL Lausanne.

Competing interests

The authors declare no competing interests.

Additional information

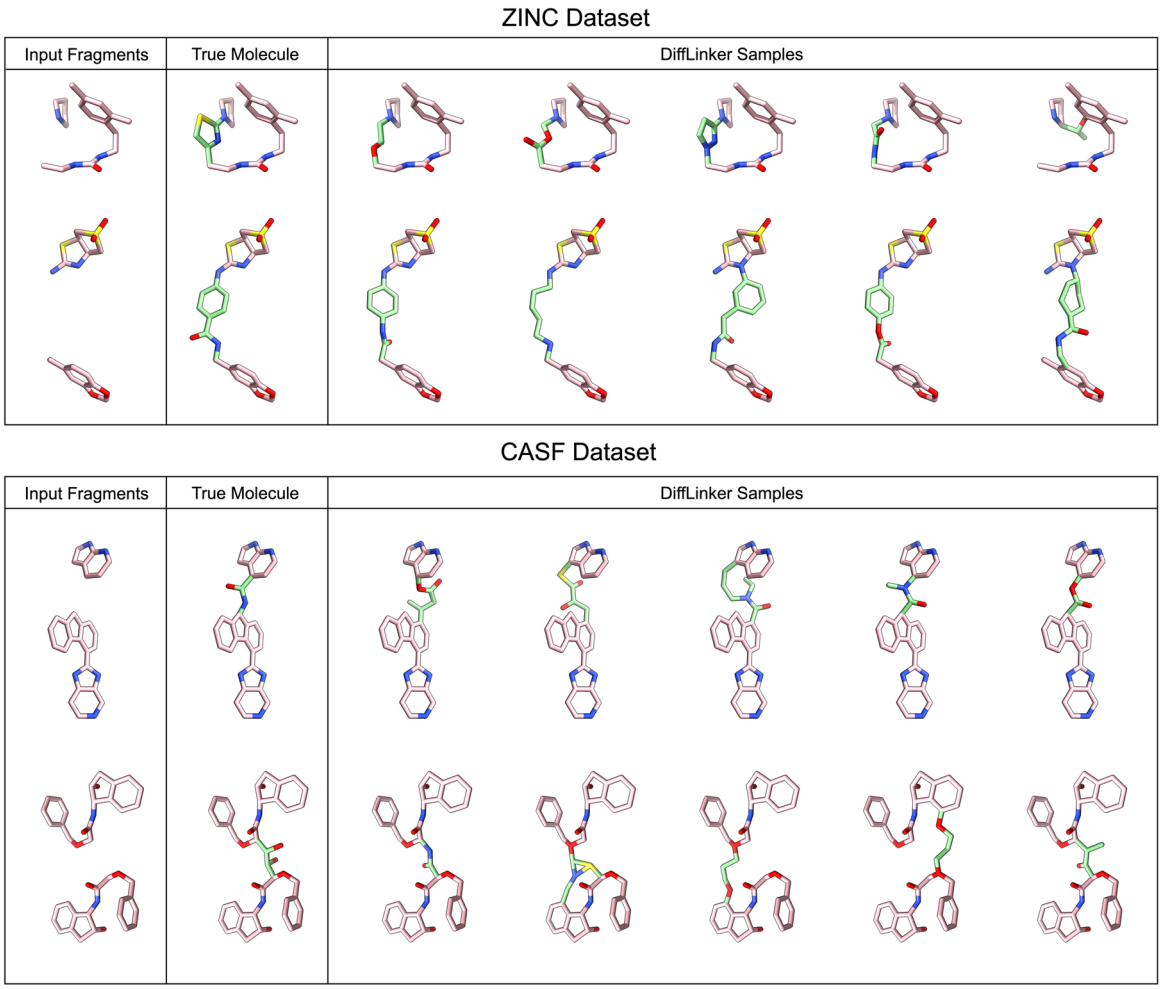
Extended data is available for this paper at https://doi.org/10.1038/s42256-024-00815-9.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-024-00815-9.

Correspondence and requests for materials should be addressed to Bruno Correia.

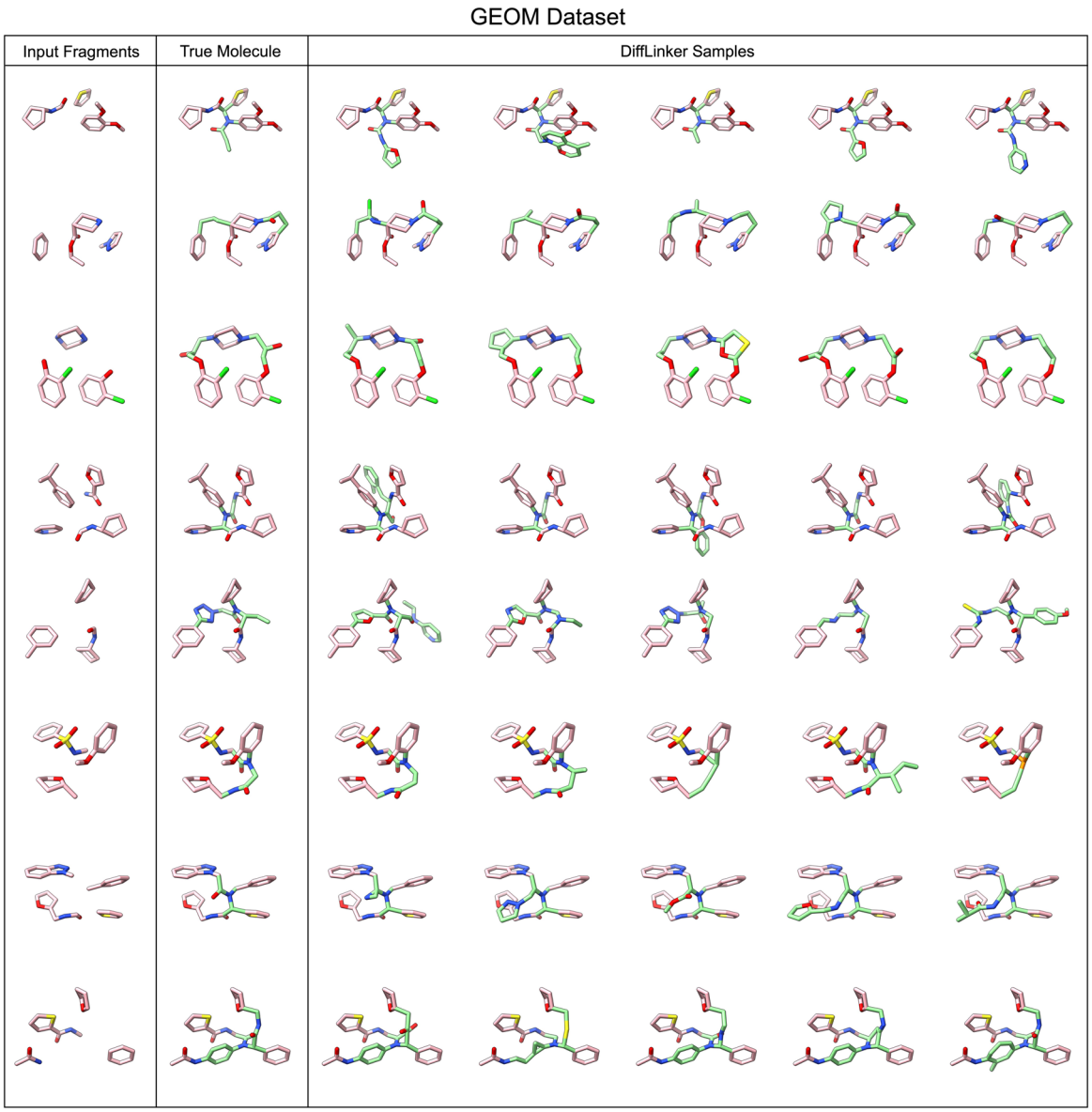
Peer review information *Nature Machine Intelligence* thanks Jihan Kim and Tiago Rodrigues for their contribution to the peer review of this work.

https://doi.org/10.1038/s42256-024-00815-9



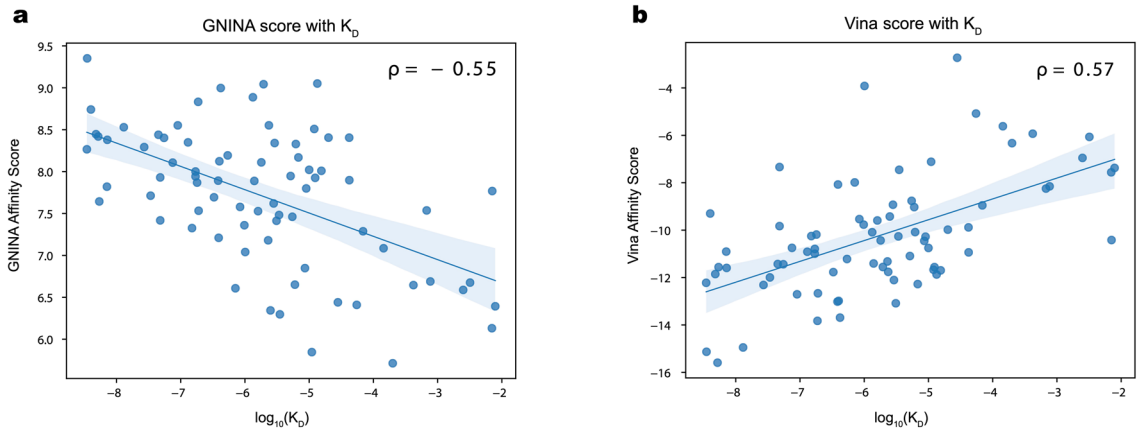
Extended Data Fig. 1 | Examples of DiffLinker samples on ZINC and CASF datasets. Examples of linkers generated by DiffLinker (sampled size) for fragments from CASF and ZINC datasets.

https://doi.org/10.1038/s42256-024-00815-9



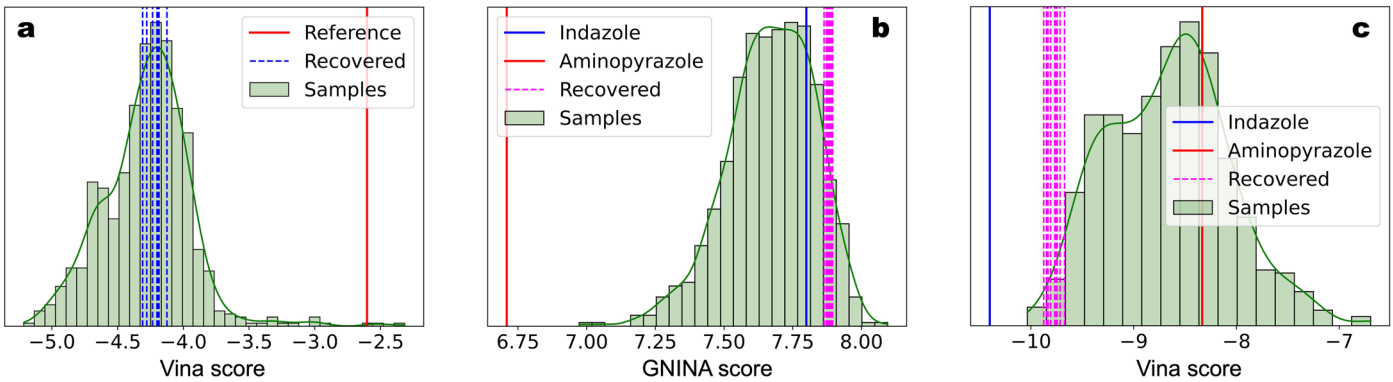
Extended Data Fig. 2 | Examples of DiffLinker samples on GEOM dataset. Examples of linkers generated by DiffLinker (sampled size) for fragments from GEOM datasets.

https://doi.org/10.1038/s42256-024-00815-9



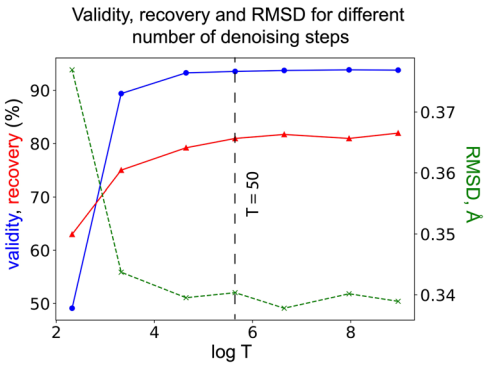
Extended Data Fig. 3 | Correlation of GNINA scores and experimentally determined binding affinities. Predicted GNINA (**a**) and Vina (**b**) scores versus experimental K_D values for Hsp90 proteins and their ligands ($n = 76$) found in PDBbind database. Error bands show 95% confidence intervals using 1000 bootstrap samples.

https://doi.org/10.1038/s42256-024-00815-9



Extended Data Fig. 4 | Distributions of docking scores for DiffLinker samples for IMPDH and JNK. Distributions of Vina and GNINA scores for samples generated by DiffLinker. **a**, Vina scores of unique samples (n = 800) for IMPDH. Red solid line depicts the score of an experimentally validated compound **31** and blue dashed lines represent scores for eight DiffLinker samples that recover compound **31**. **b**-**c**, Distributions of GNINA and Vina scores correspondingly of unique samples (n = 755) for JNK. Blue and red solid lines depict scores of experimentally validated compounds with indazole and aminopyrazole scaffolds. Dashed magenta lines represent scores of eleven DiffLinker samples that recover compound with the indazole scaffold.

https://doi.org/10.1038/s42256-024-00815-9



Extended Data Fig. 5 | Dependency of DiffLinker performance on the number of sampling steps. Dependency of validity, recovery and RMSD on the number of denoising steps in sampling shows that DiffLinker is robust to reducing the number of denoising steps. The robustness of DiffLinker allows for 10-fold gain in sampling speed without any performance degradation. For all experiments we used DiffLinker trained on ZINC with 500 steps and performed evaluation on ZINC test set sampling 250 linkers for each input set of fragments.

nature portfolio

Corresponding author(s): Bruno Correia

Last updated by author(s): Jan 16, 2024

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
☐	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐	A description of all covariates tested
☒	☐	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	☐	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	Dataset processing was done in Python (v3.10.5) using RDKit (v2022.03.2) for generating molecular conformers and splitting them in fragments and linkers, scikit-learn (v1.0.1) for splitting datasets, BioPython (v1.79) for processing protein structures. MMPA-based algorithm and BRICS used for molecule fragmentation, as well as force field relaxation procedure MMFF, are components of RDKit package. Central packages used for writing DiffLinker as well as training and sampling scripts include NumPy (v1.22.3), PyTorch (v1.11.0), PyTorch Lightning (v1.6.3), WandB (v0.12.16), RDKit (v2022.03.2) and OpenBabel (v3.0.0). For sampling molecules with baseline methods, we used pre-trained models and sampling scripts available at the corresponding repositories: 3DLinker (https://github.com/YinanHuang/3DLinker), DeLinker (https://github.com/oxpig/DeLinker), DiffSBDD (https://github.com/arneschneuing/DiffSBDD), ResGen (https://github.com/HaotianZhangAI4Science/ResGen). None of these repositories provide version releases. All custom algorithms, scripts and dependencies are available on GitHub (https://github.com/igashov/DiffLinker).
Data analysis	Data analysis and vizualization was done in Python (v3.10.5) using RDKit (v2022.03.2), imageio (v2.19.2), NetworkX (v2.8.4), SciPy (v1.7.3), matplotlib (v3.5.2), seaborn (v0.11.2), and GNINA v1.0.3 (https://github.com/gnina/gnina). All custom algorithms, scripts and dependencies used for data analysis are available on GitHub (https://github.com/igashov/DiffLinker).

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

All the processed datasets, as well as pre-trained models are available at Zenodo. Datasets: ZINC (https://doi.org/10.5281/zenodo.7121271), CASF (https://doi.org/10.5281/zenodo.7121264), GEOM (https://doi.org/10.5281/zenodo.7121278), Pockets (https://doi.org/10.5281/zenodo.7121280). Models: https://doi.org/10.5281/zenodo.7775568. Molecules used in ZINC dataset are available at ZINC database (https://zinc.docking.org/). Molecules used in CASF dataset were taken from the CASF-2016 benchmark package (http://www.pdbbind.org.cn/download/CASF-2016.tar.gz) of the PDBbind database (http://www.pdbbind.org.cn/). Molecules used in GEOM dataset are available at the repository of the original GEOM dataset (https://github.com/learningmatter-mit/geom). Molecules used in Pockets dataset were taken from Binding MOAD (http://www.bindingmoad.org/). Crystal structures of the Hsp90 inhibitor and initially bound fragments are available at Protein Data Bank under the access codes PDB-3HZ5 and PDB-3HZ1 respectively. Molecules inactive to Hsp90 were collected from three binding assays reported in PubChem under identifiers 754 (657 molecules), 687006 (81 molecules), and 1803875 (18 molecules). Crystal structures of the most potent IMDPH inhibitor and initially bound fragments are available at Protein Data Bank under the access codes PDB-5OU3 and PDB-5OU2 respectively. Crystal structures of JNK inhibitors with indazole and aminopyrazole scaffolds are available at Protein Data Bank under the access codes PDB-3FI3 and PDB-3FI2 respectively.

Human research participants

Policy information about studies involving human research participants and Sex and Gender in Research.

Reporting on sex and gender	N/A
Population characteristics	N/A
Recruitment	N/A
Ethics oversight	N/A

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	For ZINC and CASF datasets, we used the same train/validation/test splits and sizes as in previous works. Sizes of our own GEOM and Pocket datasets are the maximum possible given the available data and chosen fragmentation procedures. The number of samples for evaluation was chosen according to the previous works for ZINC, CASF and GEOM datasets. For Pockets dataset we used smaller sample size due to a higher computational complexity of the conditioned model. The number of samples in case studies was chosen as a trade-off between high sample diversity and speed (to be feasible in the real-world applications).
Data exclusions	Box-and-whisker plots in Figure 2 do not include outlier data points falling beyond the interval of ±1.5xIQR. Figure 3f does not include outlier measurements (higher than 2) of Vina scores for unconditioned samples.
Replication	By design of the algorithm, it contains non-deterministic elements that can be however fixed via random seed. Weights of all the trained models along with data splits, as well as evaluation scripts for replication of the paper results are available in the DiffLinker repository.
Randomization	GEOM and Pocket datasets were randomly split in train/validation/test sets under the conditions preventing data leakage. DiffLinker by design contains non-deterministic elements.
Blinding	The data splits were performed in the blind fashion using randomized split procedure.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging