Chemical Science

ROYAL SOCIETY OF CHEMISTRY

EDGE ARTICLE

View Article Online
View Journal | View Issue

Cite this: *Chem. Sci.*, 2025, **16**, 1417

All publication charges for this article have been paid for by the Royal Society of Chemistry

Received 3rd September 2024
Accepted 3rd December 2024

DOI: 10.1039/d4sc05894a

rsc.li/chemical-science

DiffBP: generative diffusion of 3D molecules for target protein binding†

Haitao Lin,‡[ab] Yufei Huang,‡[ab] Odin Zhang,‡[a] Siqi Ma,[b] Meng Liu,[c] Xuanjing Li,[d] Lirong Wu,[ab] Jishui Wang,[c] Tingjun Hou *[a] and Stan Z. Li*[b]

Generating molecules that bind to specific proteins is an important but challenging task in drug discovery. Most previous works typically generate atoms autoregressively, with element types and 3D coordinates of atoms generated one by one. However, in real-world molecular systems, interactions among atoms are global, spanning the entire molecule, leading to pair-coupled energy function among atoms. With such energy-based consideration, modeling probability should rely on joint distributions rather than sequential conditional ones. Thus, the unnatural sequential auto-regressive approach to molecule generation is prone to violating physical rules, yielding molecules with unfavorable properties. In this study, we propose DiffBP, a generative diffusion model that generates molecular 3D structures, leveraging target proteins as contextual constraints at the full-atom level in a non-autoregressive way. Given a designated 3D protein binding site, our model learns to denoise both element types and 3D coordinates of an entire molecule using an equivariant network. In experimental evaluations, DiffBP demonstrates competitive performance against existing methods, generating molecules with high protein affinity, appropriate molecule sizes, and favorable drug-like profiles. Additionally, we developed a website server for medicinal chemists interested in exploring the art of molecular generation, which is accessible at https://www.manimer.com/moleculeformation/index.

Introduction

Deep learning (DL) is revolutionizing various fields, including biology[1–3] and molecular science.[4,5] In the realm of micro-molecule design, a number of works have emerged on generating chemical formulas[6–12] or conformations of molecules.[4,13,14] Similarly, in macro-molecule design, AlphaFold and other protein structure prediction methods have had a profound and longstanding impact on computational biochemistry.[15–19]

The success of reverse pharmacology has proved that structure-based drug design (SBDD) is a promising approach for discovering lead compounds more rapidly and cost-effectively.[20,21] The method guides the identification of lead compounds with potent target inhibitory activity, leveraging molecular-level understanding of the disease. However, the application of machine learning (ML) techniques to design molecules that specifically bind to a target protein remains underexplored. One obstacle is the requirement for extensive data to develop effective ML approaches, although such datasets are now becoming available.[22] Another challenge lies in the complexity of the task itself, which can be attributed to three key factors. Firstly, the protein binding site, as the conditional context, is complicated, as it involves not only the 3D geometric structures of target proteins but also other informative contexts such as amino acid types that must be considered to generate molecules with high affinities. Secondly, the desired distribution across molecular chemistry and coordinates has vast support sets. Unlike the conformation generation task, the chemical formulas as 2D graph constraints are unknown, necessitating a well-designed model that can capture the intricate coupling of element types, continuous 3D coordinates, and other chemical properties or geometries. Finally, the geometric symmetries of molecules should be considered for generalization. In the physical 3D space, these symmetries include translations and rotations from the Euclidean group, suggesting that if symmetry operations are performed on a binding site, the generated molecules should undergo corresponding rotations or translations.

Recently, a line of DL-based methods has been proposed for SBDD.[23–27] Initially, grid-based models, such as LiGAN[23] and 3DSBDD,[24] were introduced to predict whether grid points are occupied by specific atoms. However, these models regard the position of molecules in a discretized space, which contradicts

[a]Zhejiang University, Hangzhou 310058, Zhejiang, China. E-mail: tingjunhou@zju.edu.cn

[b]AI Lab, School of Engineering, Westlake University, Hangzhou 310024, Zhejiang, China. E-mail: Stan.ZQ.Li@westlake.edu.cn

[c]Texas A&M University, Texas, TX 77843, USA

[d]Jingdong, Beijing 101111, China

† Electronic supplementary information (ESI) available. See DOI: https://doi.org/10.1039/d4sc05894a

‡ Equal contributions.

the continuous movement of atoms in 3-dimensional (3D) space. Subsequently, the development of neural networks for point clouds has advanced DL-based SBDD methods, with auto-regressive models like Pocket2Mol[25] and GraphBP[26] established. These methods aim to generate atoms auto-regressively and directly model the probability of the next atom's type as a discrete categorical attribute and its position as continuous geometry. In these methods, the distributions of the next atom's position and type are determined by the previously generated ones and the given protein context. However, there are several issues with this auto-regressive framework. Firstly, the sequential modeling of a molecule contradicts real-world physical scenarios, where atomic interactions are global. In other words, the position and element type of each atom are affected by all other atoms within the molecular system. Secondly, auto-regressive sampling for molecules usually suffers from the 'early stopping' problem. Specifically, the model tends to generate molecules with a small number of atoms (or small molecule size), failing to accurately capture the true distribution of atom numbers in drug-like molecules.

To address these significant limitations, recently proposed EDM[28] and GCDM[27] utilize the diffusion denoising probabilistic models[29] as a one-shot molecule generation solution, while in SBDD tasks, the problem is not fully resolved. In comparison, we propose DiffBP, a target-aware molecular diffusion model for protein binding. By harnessing the exceptional capacity of diffusion denoising generative models to generate high-quality samples,[29–31] combined with the high expressivity of equivariant graph neural networks,[32,33] our model can generate molecules that exhibit favorable drug-like properties and high affinity toward the target protein. We first analyze the issues inherent in auto-regressive models from a physics and probabilistic perspective. Motivated by these observations, we propose DiffBP, which directly models all atoms during target-aware molecule generation. Experimental evaluation demonstrate that DiffBP outperforms previous methods, exhibiting promising performance in terms of appropriate molecule size, higher ligand efficiency with target protein, and other drug-like properties.

Results and discussion

Property evaluation and performance comparison

To evaluate our model, we follow previous works[23,26] and use the CrossDocked2020 (ref. 22) dataset to generate ligand molecules that specifically bind to target protein pockets based on the pocket structures. Consistent with previous studies, we adopt the same split for the training and test sets. Three state-of-the-art (SOTA) methods including 3DSBDD,[24] Pocket2Mol[25] and GraphBP[26] are employed as the baseline models for comparison. For each protein pocket, we generate 100 molecules to calculate metrics for a comprehensive evaluation. To calculate the affinity score, we adopt Gnina,[34,35] an ensemble of CNN scoring functions, which has also been used to evaluate GraphBP and LiGAN. We report two affinity score metrics: Ligand Efficiency (LE) and Mean Percentage Binding Gap (MPBG). Additionally, we calculate several chemical metrics, including QED, SA, Sim and LPSK. To provide a more granular analysis, we categorize the metrics into three groups based on the generated molecule sizes: small, medium, and large. The size range within each group is defined relative to the sizes of the reference molecules.

Table 1 presents the two quantitative binding metrics for the four methods, where 'Ratio' is the proportion of molecules of different sizes compared to the total number of generated molecules. Notably, 3DSBDD and Pocket2Mol suffer from the 'early-stopping' issue in their auto-regressive generating process, resulting in a higher small ratio of small molecules compared to the other two methods. In contrast, DiffBP generates more medium-sized molecules, accounting for 75.19% of the total, which contributes the lowest MPBG and the highest LE of DiffBP. It is notable that the calculation of binding scores often assigns higher affinity to larger molecules, because larger molecules are more likely to form more interactions with target proteins. This bias is a common issue among existing scoring functions. Therefore, GraphBP benefits from a high proportion of large molecules but also suffers from a distribution shift in atom sizes, with a low medium ratio and a significant proportion of large molecules. In contrast, DiffBP generates a high proportion of molecules of appropriate size. However, considering the bias of the binding score metric towards molecular size, it is crucial to consider additional metrics for a fair comparison. Therefore, we

Table 2 Additional drug-like properties for the molecules generated by different methods (bold values are the top-2 metrics)

	3DSBDD	Pocket2Mol	GraphBP	DiffBP
QED (↑)	0.3811	**0.5106**	0.3830	**0.4431**
SA (↑)	0.5185	**0.5430**	0.4828	**0.5377**
Sim (↓)	0.3485	0.3485	**0.2707**	**0.3290**
LPSK (↑)	0.6678	**0.8134**	0.5961	**0.7042**

Table 1 Comparison on the affinity score metrics of the molecules generated by different methods

	3DSBDD			Pocket2Mol			GraphBP			DiffBP		
	Ratio	MPBG	LE	Ratio	MPBG	LE	Ratio	MPBG	LE	Ratio	MPBG	LE
Small	41.45%	27.92%	4.90%	36.62%	25.18%	4.10%	27.72%	35.16%	5.19%	5.22%	17.61%	10.25%
Medium	54.06%	19.78%	14.84%	59.02%	5.38%	32.53%	32.03%	18.68%	15.30%	75.19%	2.36%	40.20%
Large	4.48%	−7.53%	48.56%	4.36%	−11.21%	75.42%	37.97%	−10.13%	60.21%	19.59%	−4.11%	52.64%
Overall		21.92%	12.22%		11.90%	23.98%		12.30%	29.54%		1.88%	41.07%

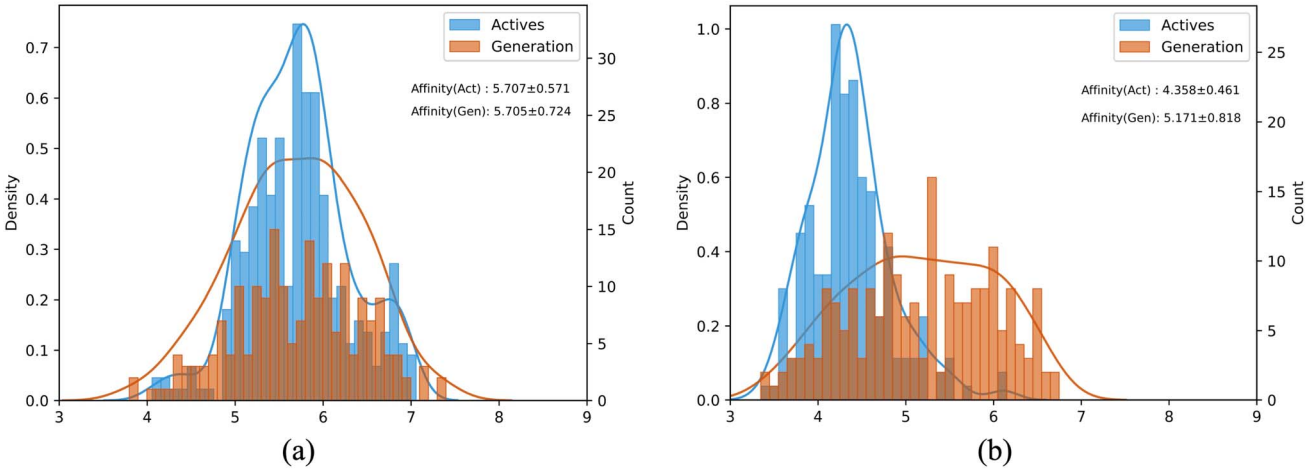

Fig. 1 KDE (kernel density estimation) distribution of binding affinity of generated samples v.s. active molecules for (a) AKT1 and (b) CDK. Affinity (Act) and affinity (Gen) means the average binding affinity obtained by active and generated molecules, with standard deviation.

Table 3 Other properties of drug-like molecules for comparison between the generated and active molecules toward AKT1 and CDK2

	AKT1 (Gen)	AKT1 (Act)	CDK2 (Gen)	CDK2 (Act)
QED (↑)	0.5289	0.5047	0.5908	0.5095
SA (↑)	0.5550	0.6903	0.5164	0.5908
Sim (↓)	0.5399	0.2872	0.4385	0.2962
LPSK (↑)	0.8446	0.7428	0.9300	0.7330

introduce LE, a concept in drug discovery, to compare the binding scores among ligands of the same size. Table 1 demonstrates the superiority of DiffBP in terms of LE. For medium-size molecules, DiffBP achieves a remarkable 40.20% compared to GraphBP's 15.30%. Overall, DiffBP scores 41.07%, significantly surpassing GraphBP's 29.54%, implying that the molecules generated by DiffBP exhibit higher efficiency in targeting specific proteins.

Although reference molecules may not serve as the gold benchmark in SBDD, they can still reflect certain properties of the binding site. For instance, the site of the binding site can be approximated by the size of the reference molecule. Therefore, it is reasonable to use the size of the reference molecule to define the suitable number of atoms for generated molecules. By comparing the metrics across different size groups, we can identify the scoring functions' preferences for specific molecule sizes when evaluating drug properties. Additional metrics for drug properties on generated molecules are shown in Table 2,

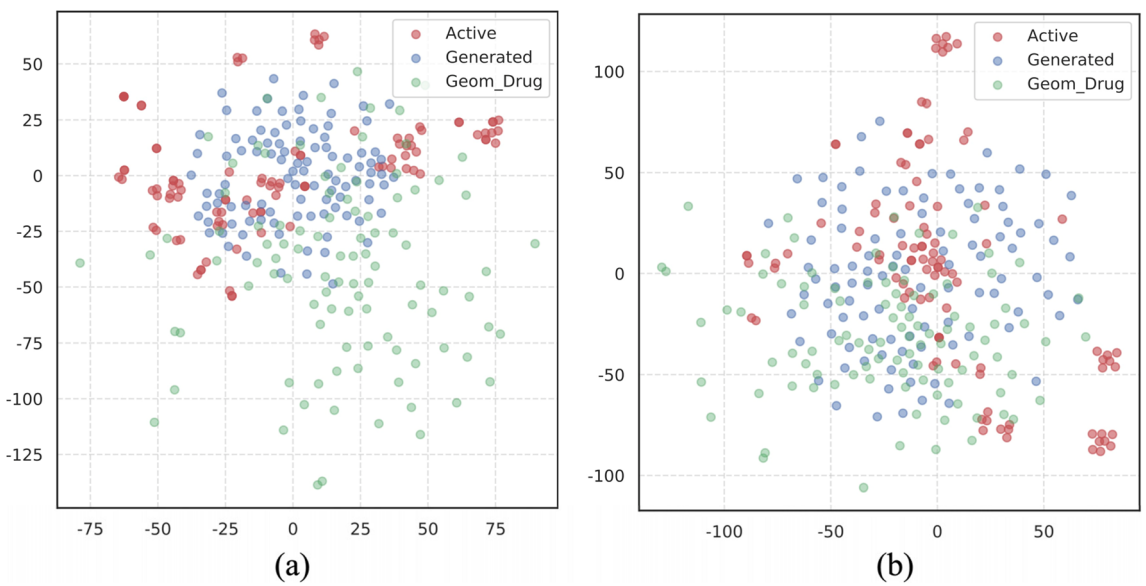

Fig. 2 T-SNE plots of generated molecules, randomly-selected molecules, and active molecules for (a) AKT1 and (b) CDK2. The Morgan Fingerprint is used as chemical descriptor for encoding molecules. The encoded values are standardized and transformed into 2D features with T-SNE.

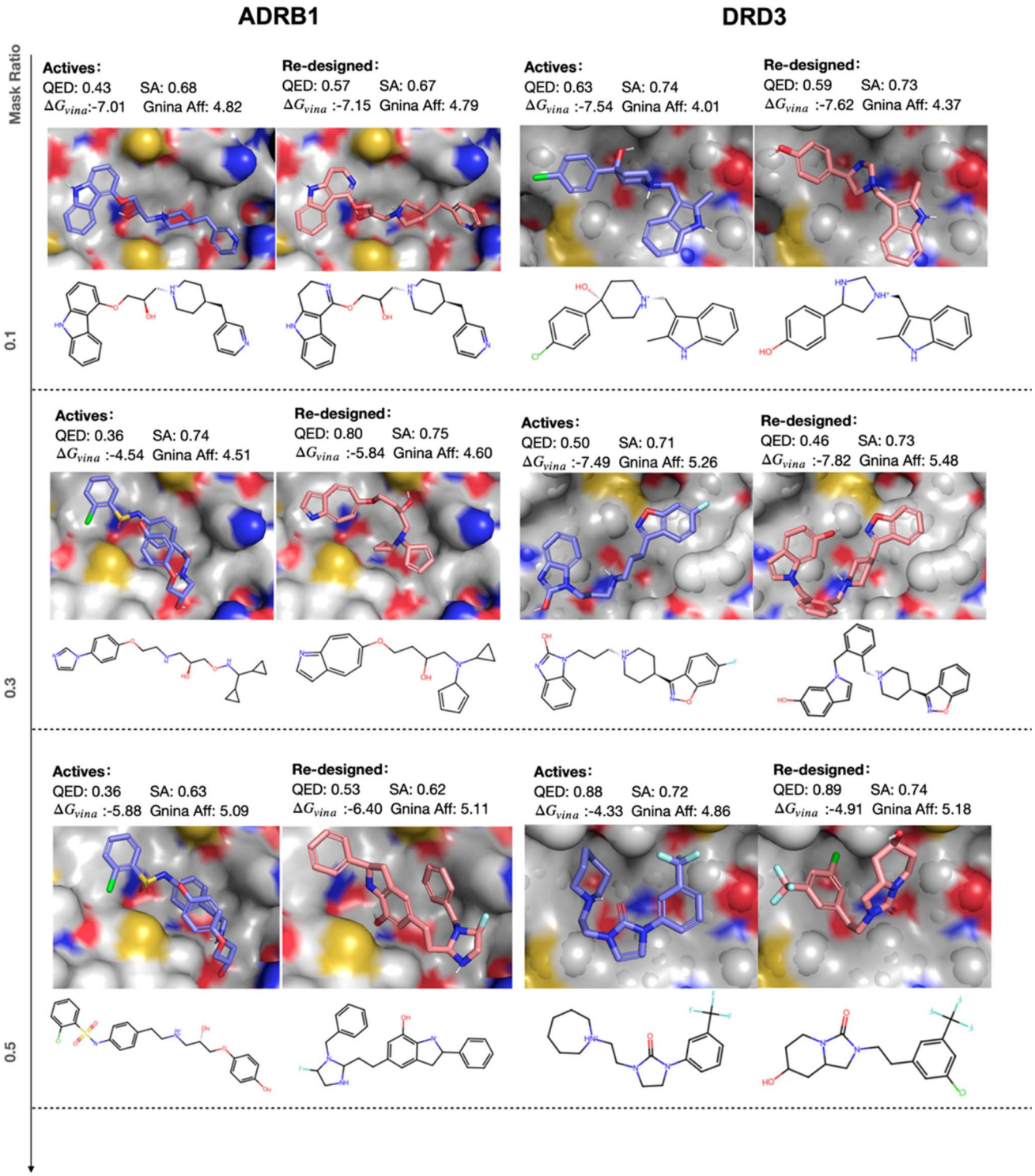

Fig. 3 Visualization of molecules which are controllably designed by DiffBP for target ADRB1 and DRD3. As the ratio increases, the differences between active molecules and re-designed molecules are more significant.

highlighting the competitiveness of DiffBP-generated molecules across all metrics. Although PocketMol performs better in terms of the QED and SA metrics, we have previously demonstrated its tendency to favor small molecules, which naturally exhibit high QED and SA scores. To further illustrate this bias, we provide the details in Appendix Table S2.†

Generation on targets AKT1 and CDK2

To verify whether DiffBP can generalize to real-world pharmaceutic targets, we choose AKT1 (protein kinase B alpha) and CDK2 (cyclin-dependent kinases 2) as case studies. These two kinases are representative targets that have been extensively studied in

previous work.[36] AKT1 and CDK2 play pivotal roles in cellular processes, including survival, growth, metabolism, and cycle regulation. Dysregulation of AKT1 function can lead to diseases such as cancer, diabetes, and neurological issues, making its pathway a focus for cancer drug development. CDK2 is also critical for cell cycle control, highlighting its importance in cancer research and the therapeutic potential of CDK inhibitors.

Regarding these two targets, 100 molecules have been experimentally confirmed as actives, while 100 molecules have been generated using DiffBP. The comparison of the kernel density estimation of the binding affinity histograms between the generated molecules and the experimentally active ones is shown in Fig. 1. Obviously, the distribution of the binding affinity scores for the DiffBP-generated molecules is close to that of the experimentally determined actives, with some generated molecules exhibiting even higher binding affinity towards the target. This finding proves the capacity of DiffBP to generate molecules with favorable binding energies for the specific target. Table 3 reveals that DiffBP performs slightly better than real active ligands in terms of two drug-likeness metrics, QED and LPSK, indicating the DiffBP has effectively learned the characteristics of pharmaceutic molecules during the training phase. But it should be noted that the synthesized accessibility score, SA, for the generated molecules is relatively lower than that for the experimental actives, which may be attributed to the common observation that AI-generated molecules are not fully optimized and may require refinement by medicinal chemists.[37] In addition, the generated molecules exhibit lower internal similarity, implying that DiffBP has a high potential for lead discovery. The higher diversity also contributes to the different distribution shapes observed, where the active distribution has a high peak and short tail, while the DiffBP distribution shows a longer tail. In conclusion, the experiment demonstrates that DiffBP is capable of generating molecules with active-like properties.

The chemical distributions of the molecules generated by DiffBP for AKT1 and CDK2 are visually plotted in Fig. 2. Both the active and generated molecules are represented by 100 samples each. For comparison, we randomly select 100 molecules from the GEOM-DRUG dataset.[38] The results indicate that the chemical distribution of the molecules generated by DiffBP is closer to that of the active molecules. For AKT1, the active molecules show several clusters, and DiffBP not only shots the cluster center occupied by the active molecules but also explores a border chemical space. Furthermore, several generated molecules deviate from the cluster centers, suggesting DiffBP's potential in discovering unexplored chemical space for drug design. In comparison, the molecules from GEOM-Drugs occupy a different chemical space from both the actives and DiffBP-generated molecules, verifying that DiffBP-generated molecules are aware of protein structures and mimic the active molecules more closely than random ones for both AKT1 and CDK2 targets. In conclusion, DiffBP can generate bound molecules with similar chemical characteristics to active molecules, thus enhancing its credibility and practical utility in real-world drug design.

Controllable design on targets ADRB1 and DRD3

In this section, we consider two proteins that belong to the G-Protein-Coupled Receptor (GPCR) family: ARDB1 (beta-1 adrenergic receptor) and DRD3 (dopamine receptor D3). ARDB1 plays a pivotal role in regulating various physiological processes, responding to the neurotransmitter epinephrine (adrenaline) and norepinephrine. Drugs that selectively activate or block this receptor are commonly used in treating various cardiovascular conditions, such as heart failure and certain arrhythmias. On the other hand, DRD3 is primarily expressed in the brain, particularly in regions such as the limbic system and the ventral striatum, mediating the effects of the neurotransmitter dopamine within the central nervous system.

To preserve the integrality of certain active components while controllably designing others as a conditional generation task, we employ a perturbed strategy with DiffBP as a diffusion-based method, following a recent protocol for antibody design.[39] Specifically, we use DiffBP to firstly mask atoms randomly, with a masking ratio varying from 0.1 to 0.5. Subsequently, we adjust the molecule sizes by adding or removing these masked atoms. DiffBP then generates the masked atom types and their corresponding positions within the molecule. For each target, we randomly select 100 actives and generate the 100 optimized molecules accordingly. Fig. 3 visually illustrates the re-designed active molecules by DiffBP using different masking ratios. The binding poses are calculated by Vina-Dock,[40] and the binding affinity is estimated by Gnina. The results suggest that DiffBP allows for controllable generation of molecules with improved properties and structures resembling existing leads. Fig. 4 demonstrates the LE of the optimized molecules under different masking ratios. It can be shown that at lower ratios, the binding affinities are comparable to those of the actives, with only a small fraction of generated molecules exhibiting superior binding affinity. As the masking ratio increases, a greater percentage of generated molecules show better affinity towards the target proteins, indicating that DiffBP

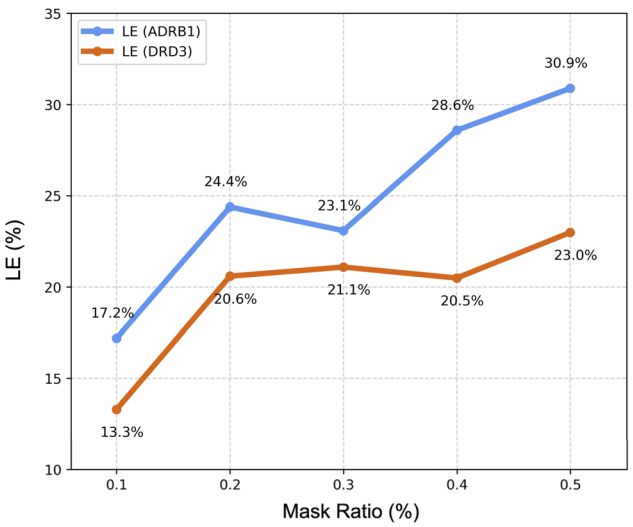

Fig. 4 The change of LE calculated by the re-designed molecules from the selected actives for target ADRB1 and DRD3.

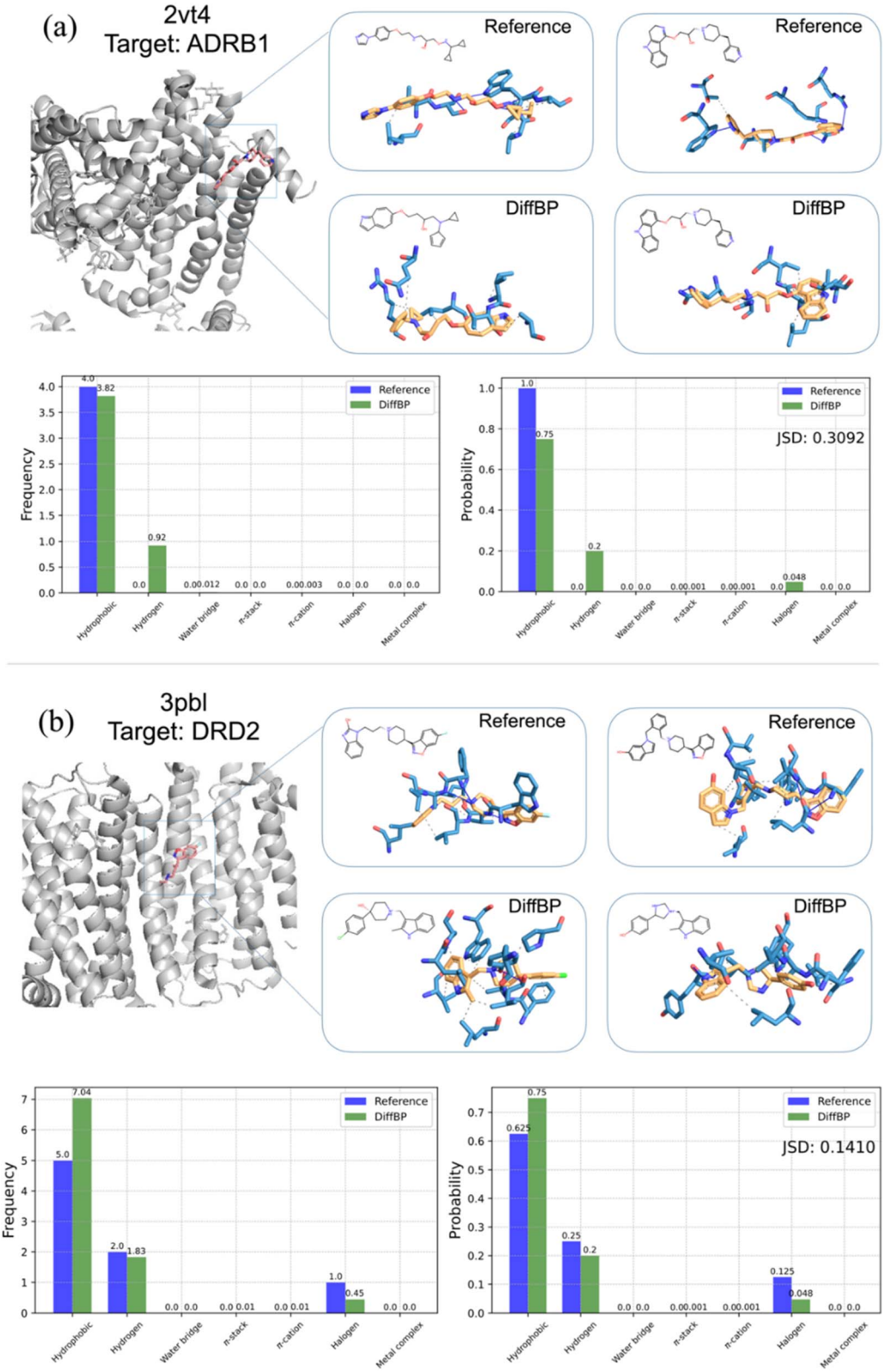

Fig. 5 Visualization of protein–ligand interaction patterns and the frequencies and distributions for different interaction types, on (a) ADRB1 and (b) DRD2 targets.

can generate molecular structures that fit better into protein pockets.

Interaction pattern analysis

We here try to figure out whether protein-conditioned 3D generative models can capture the microscopic interaction patterns present in the 3D conformations of protein–ligand complexes. We analyze two targets, ADRB1 and DRD3, using the controllable-designed molecules. With PLIP, we characterized the protein–ligand interactions between these protein targets and the generated ligands. We compare these results with the interaction patterns of the referenced active ligands. As shown in Fig. 5(a), the generated molecules from DiffBP closely match the reference ligands in their interaction patterns, indicating that DiffBP effectively learns how a hit localizes within the pocket. Besides, we set the mask ratio as 1.0, enabling *de novo* generation of molecules, and analyze the interactions between the pockets (ADRB1 and DRD3) and 100 active molecules as a reference, alongside 100 generated molecules. We define the frequency as the number of interactions of each type per molecule and their corresponding interaction distribution. Fig. 5(b) shows that active molecules tend to prioritize forming hydrophobic interactions with the target. This tendency is further enhanced in DRD2 by the generated molecules, which exhibit a higher frequency of such interactions. Additionally, for some interactions that do not exist in active molecules, such as π-stack interactions, the molecules generated by DiffBP also show a small probability of forming these types of interactions. Distribution analysis reveals that, overall, the molecules generated by DiffBP maintain the distribution of different types of interactions of active molecules for these two targets, which is also reflected in the Jensen–Shannon divergence (JSD).

Table 4 Sub-structure analysis on ratios, which is defined as how many sub-structures exist in one molecule on average. Bold values are the top-2 ratios close to reference molecules in training and test sets

		Train	Test	3DSBDD	Pocket2Mol	GraphBP	DiffBP
Bond	Single	18.76	18.24	**14.37**	**16.50**	27.16	23.37
	Double	6.67	5.24	2.42	**2.98**	0.94	**3.43**
	Triple	0.05	0.02	0.01	0.01	0.01	**0.02**
Ring	Tri	0.04	0.04	1.99	**0.83**	2.48	**0.91**
	Quad	0.01	0.00	0.46	**0.01**	0.93	**0.22**
	Pent	0.80	0.71	0.31	0.43	**0.47**	**0.58**
	Hex	1.93	1.58	0.59	**1.43**	0.38	**0.89**

Sub-structure analysis

To compare the substructural generation capabilities of different methods, we evaluate the ratio of different bond types and molecules containing different rings in the generated molecules and reference molecules, as shown in Table 4. For a comprehensive atom type analysis, please refer to Appendix Table S3.† Notably, DiffBP tends to generate more bonds generally, as well as 'Pent' rings, which can be attributed to the increased number of atoms in the generated molecules. Conversely, Pocket2Mol has the advantage of generating fewer 'Tri' rings and more 'Hex' rings, in coordination with the experimental analysis reported by Peng *et al.*[25] The main reason for the suboptimal generative performance of substructures is that DiffBP does not consider chemical bond information during training and generation. Instead, it directly models the coordinates and element types of the atoms with the software used to reconstruct based on the generated targets. While autoregressive methods find it relatively easier to utilize information about chemical bonds, directly modeling this information for diffusion methods is more challenging, which will be the focus of our future research.

Physical illustration

Previous methods typically employ auto-regressive strategies to generate molecules, whereas our model naturally simulates the process using physical dynamics principles. Fig. 7 illustrates the workflow of how the positions and types of atoms are updated

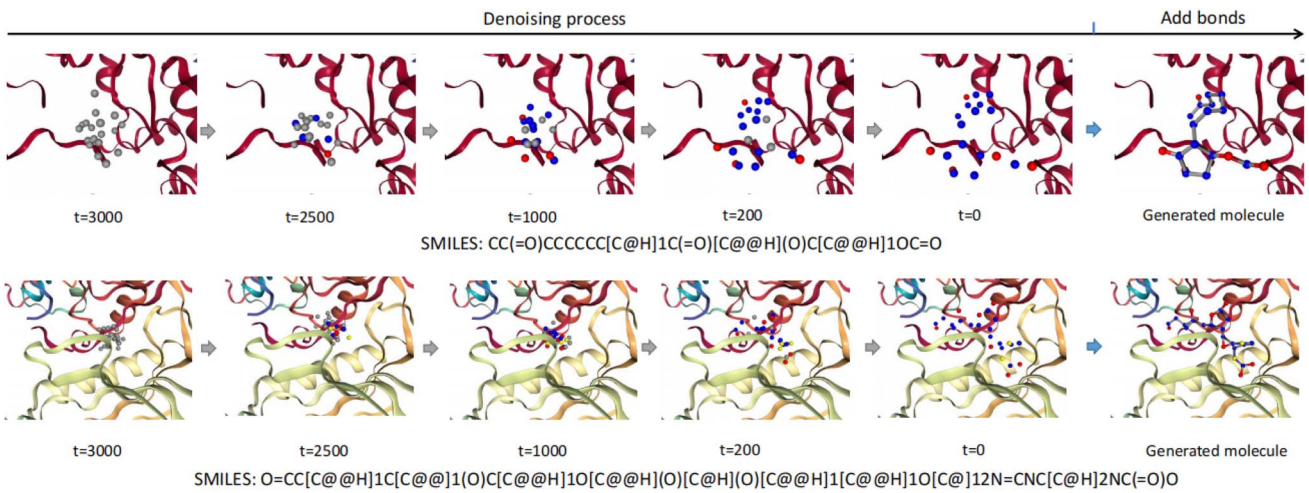

Fig. 6 Visualization on generation process two molecules (affinity score = 4.583 and 5.682), which are binding to the protein '1afs_A_rec' and '4azf_A_rec' respectively.

by DiffBP. Guided by physical rules, the Langevin dynamic updates the positions of atoms according to the energy function E:

$$\boldsymbol{x}_{i,t-1} = \boldsymbol{x}_{i,t} - \frac{\lambda_t}{2} E\left(\left\{\left(\boldsymbol{a}_{j,t}, \boldsymbol{x}_{j,t}\right)\right\}_{j=1}^{N+M}\right) + \mathcal{N}\left(0, \boldsymbol{\sigma}_t{}^2 \boldsymbol{I}\right), \quad (1)$$

where λ_t is the step length for updating the atom positions at t.

In our generative denoising process, the positions are updated as

$$\mu(\hat{\boldsymbol{x}}_{i,0}, \boldsymbol{x}_{i,t}) = \frac{1}{\boldsymbol{\alpha}_{t|s}} \boldsymbol{x}_{i,t} - \frac{\boldsymbol{\sigma}_{t|s}{}^2}{\boldsymbol{\alpha}_{t|s} \boldsymbol{\sigma}_t} \widehat{\boldsymbol{\epsilon}}_{i,t} + \mathcal{N}\left(0, \boldsymbol{\sigma}_t{}^2 \boldsymbol{I}\right) \quad (2)$$

where here $\widehat{\boldsymbol{\epsilon}}_{i,t} = [\phi_\theta{}^k(\{(\boldsymbol{a}_{j,t}, \boldsymbol{x}_{j,t})\}_{j=1}{}^{N+M}, t)]_{k=1,2,3}$. By this mean, the learned position denoiser $[\phi_\theta{}^k]_{k=1,2,3}$ can be regarded as an approximated energy function when $\alpha_{t|s} = 1$. Besides, from a probabilistic perspective, since $\widehat{\boldsymbol{\epsilon}}_{i,t}$ is calculated across all atoms, DiffBP models the joint probability rather than a sequence of conditional distributions, which is the focus of previous auto-regressive-based methods, as discussed in our Problem statement in method. Two generative denoising processes for binding molecules to proteins are shown in Fig. 6, where the grey atoms represent 'dummy' or 'absorbing' types. During the generative process, positions are updated, and element types are recovered by the graph denoisers. Finally, Openbabel[41] is used to construct chemical bonds among the atoms. This process is like throwing a handful of particles into a protein pocket. The model learns how to position these atoms in appropriate places within the pocket to form a strong binding. The entire process is determined by the potential energy statistics learned by the model.

Further analysis of modules

In our workflow, several modules are not necessary to fill the task, but we are interested in exploring whether they can enhance performance. One such module is the intersection loss, proposed as a regularization term in the optimization objective section of the Method. This regularization term significantly affects the success rate, also known as validity. Fig. 7 provides the validity of the molecules generated by the trained model based on different loss parameters. The results indicate that the regularization loss can slightly improve validity. However, when the two coefficients are excessively large, the validity decreases. The rationale for it is that the loss term can be viewed as a repulsive force between protein and molecule atoms. If the repulsion is too strong, the atomic positions are likely to collapse into a small region, resulting in an excessive number of neighbors for each atom, effectively exceeding the allowed valences.

Secondly, within our workflows, the CoMs and atom numbers obtained by pre-generation models can be replaced by those provided by reference molecules, leading to the variant of DiffBP(Pre-Ref). If the generated CoMs deviate significantly from the binding sites or the molecule sizes become excessively large, the contextual information of protein pockets will hardly affect the generative denoising process. As shown in Table 5, our pipeline of DiffBP(PreGen) consisting of a pre-generation model and a diffusion generative model performs effectively since the molecules generated by DiffBP(Pre-Gen) achieve comparable affinity scores to those of DiffBP(Pre-Ref).

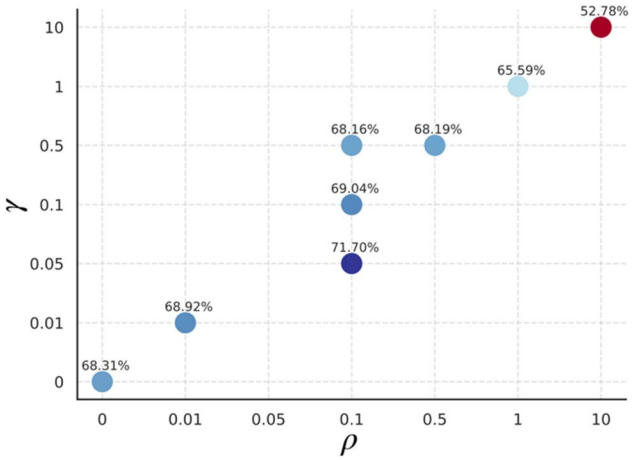

Fig. 7 Validity of generated samples *v.s.* ρ and γ.

Table 5 Binding affinity of DiffBP(Pre-Gen) and DiffBP(Pre-Ref)

	DiffBP(Pre-Gen)			DiffBP(Pre-Ref)		
	Ratio	MPBG	LE	Ratio	MPBG	LE
Small	5.22%	17.61%	10.25%	0.00%	0.00%	0.00%
Medium	75.19%	2.36%	40.20%	100.00%	1.56%	42.18%
Large	19.59%	−4.11%	52.64%	0.00%	0.00%	0.00%
Overall		1.88%	41.07%		−1.56%	42.18%

Finally, we compare the distributions of molecule sizes generated by different methods. Fig. 8 presents the atom number distribution of each method, approximated by kernel density estimation (KDE). We calculate the Jensen–Shannon divergence (JSD) between the distributions generated by different methods and the reference (Table 6). This analysis provides an intuition that, thanks to the pre-generation models, DiffBP typically generates molecules of appropriate sizes.

Platform

The DiffBP server is supported by a Linux Ubuntu cluster with 4 nodes, where each node consists of two Intel Xeon Gold 6348

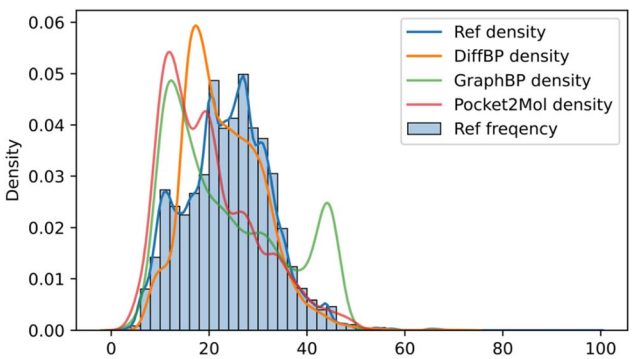

Fig. 8 Atom number distribution of different methods *v.s.* the reference.

Table 6 JSD of the distribution of generated molecules' atom number by the evaluated methods *vs.* reference

Method	JSD
GraphBP	0.4479
Pocket2Mol	0.2872
DiffBP	0.2138

processors/2.60 GHz/28 cores and 512 GB of RAM, 4 NVIDIA A40 GPUs. The web server is built on Apache HTTP, HTML and JS Vue applet. For DL implementation, python relies on PyTorch 2.2.0, with MySQL serving as the database and RabbitMQ as the message queue for scheduling different nodes. Besides, Gnina[35] and AutoDock Vina[40] are implemented for binding energy calculation and pose optimization. Pymol[42] is used for visualization. To speed up inference, tasks run in parallel with 10 CPU cores and a single NVIDIA-A40 GPU. The web service is freely accessible without registration, but users can create a separate account to save task results, ensuring privacy for all users. For a quick start, we provide tutorial pdb files to guide users through the process.

Input. Multiple modes can be available for molecule generation. In the FP (full protein) mode, users must upload a complete pdb file for the target protein. Besides, the center of the pocket or binding site with the box size as the boundary should be specified to guide the model to accurately locate the target site. In the IP (intercepted pocket) mode, a pdb file, which is intercepted with amino acids forming pockets included, is required for the generation of molecules, representing the intercepted pocket as the target site.

Output. Firstly, the bound structures of the generated molecules will be stored as a series SDF files. After filtering the invalid molecules with low SA and QED scores, the molecule with the highest probability of becoming a drug candidate would be sent to the registered email address. Besides, four metrics including binding affinity, SA, QED, and LPSK will be provided to evaluate the quality of the generated molecules. In an empirical test, we found that the IP mode usually generate molecules of higher quality.

Conclusion

In this work, we propose a 3D molecule generative model based on diffusion denoising models, DiffBP, to generate drug-like molecules of high binding affinities with the target proteins. The generation process is in line with the laws of physics, *i.e.* generate molecules at a full atom level within protein pockets. Based on the comprehensive analysis, we outline the problem previous methods exist and propose a potential solution, a metric related to ligand efficiency. In summary, DiffBP offers chemists a powerful tool to perform structure-based drug design under the current popular Artificial Intelligence Generated Context (AIGC) scheme.

Method

Problem statement

For a protein–molecule (also called protein–ligand) binding system as $\mathcal{C}$, which contains $N + M$ atoms, we represent the index set of molecules as $\mathcal{J}_{\text{mol}}$ and proteins as $\mathcal{J}_{\text{pro}}$, where $|\mathcal{J}_{\text{mol}}| = N$ and $|\mathcal{J}_{\text{pro}}| = M$. To be specific, let $\boldsymbol{a}_i$ be the K-dimensional one-hot vector indicating the atom element type of the i-th atom in the binding system and $\boldsymbol{x}_i$ be its 3D Cartesian coordinate, and then $\mathcal{C} = \{(\boldsymbol{a}_i, \boldsymbol{x}_i)\}_{i=1}^{N+M}$ can be split into two sets as $\mathcal{C} = \mathcal{M} \cup \mathcal{P}$, where $\mathcal{M} = \{(\boldsymbol{a}_i, \boldsymbol{x}_i) : i \in \mathcal{J}_{\text{mol}}\}$ and $\mathcal{P} = \{(\boldsymbol{a}_j, \boldsymbol{x}_j) : j \in \mathcal{J}_{\text{pro}}\}$. For protein–aware molecule generation, our goal is to establish a probabilistic model to learn the conditional distribution of molecules conditioned on the target proteins, "*i.e.*" $p(\mathcal{M}|\mathcal{P})$.

Problems in auto-regressive models. Recently proposed auto-regressive models sequentially generate $(\boldsymbol{a}_i, \boldsymbol{x}_i)$ by modeling the conditional probability $p(\boldsymbol{a}_i, \boldsymbol{x}_i|\mathcal{C}_{i-1})$, where $\mathcal{C}_{i-1} = \{(\boldsymbol{a}_j, \boldsymbol{x}_j)\}_{j=1}^{i-1} \cup \mathcal{P}$ is the intermediate binding system at the step i. By this means, the desired probability is modeled as a sequence of conditional distributions, as $p(\mathcal{M}|\mathcal{P}) = \prod_{i=1}^{N} p(\boldsymbol{a}_i, \boldsymbol{x}_i|\mathcal{C}_{i-1})$, where $\mathcal{C}_0 = \mathcal{P}$. By contrast, in real-world protein–molecule systems, there are force interactions between any pair (or even higher order) of atoms such that the energy function can be decomposed as $E(\mathcal{C}) = \sum_{i \neq j} E(\boldsymbol{a}_i, \boldsymbol{x}_i, \boldsymbol{a}_j, \boldsymbol{x}_j)$. The stable system reaches an energy-minimal state. From the perspective of energy-based generative models, the corresponding Boltzmann distribution is written as $p(\mathcal{C}) = \exp\left(-\frac{E(\mathcal{C})}{\kappa\tau}\right) = \prod_{i \neq j} p(\boldsymbol{a}_i, \boldsymbol{x}_i, \boldsymbol{a}_j, \boldsymbol{x}_j)$, where κ is Boltzmann constant and τ is the temperature, indicating that the modeling of probability is based on joint distributions at a full atom level, instead of sequential conditional distributions. Thus, auto-regressive models are likely to violate physical rules. To address it, we employ the following diffusion models for molecule generation at a full-atom level.

Diffusion models

Diffusion on continuous variables. Diffusion models[29,43] for continuous variables learn the data distribution by manually constructing the forward diffusion process and using a denoising model to gradually remove the noise added in the diffusion process. The latter process is called the reverse denoising process. Denote the input data point by $z = z_0$, and the diffusion process adds multivariate Gaussian noise to z_t for $t = 0, \ldots, T$, so that

$$q(z_t|z_0) = \mathcal{N}\left(z_t; \alpha_t z_0, {\sigma_t}^2 \boldsymbol{I}\right), \tag{3}$$

where $\alpha_t \in \mathbb{R}^+$ is usually monotonically decreasing from 1 to 0, and $\boldsymbol{\sigma}^2$ is increasing, which means the retained input signals are gradually corrupted by the Gaussian noise along t-axis, leading to $q(z_T) = \mathcal{N}(z_T; 0, \boldsymbol{I})$. For variance-preserving diffusion process,[29], $\boldsymbol{\alpha}_t = \sqrt{1 - {\boldsymbol{\sigma}_t}^2}$; for variance-exploding process,[30,44] $\boldsymbol{\alpha}_t = 1$. Following recently proposed variational diffusion models,[28,45] where signal-to-noise ratio is defined as

$$\text{SNR}(t) = \frac{{\boldsymbol{\alpha}_t}^2}{{\boldsymbol{\sigma}_t}^2}, \tag{4}$$

The Markov representation of the diffusion process can be equivalently written with transition distribution as

$$q(z_t|z_s) = \mathcal{N}\left(\boldsymbol{z}_t; \boldsymbol{\alpha}_{(t|s)z_s}, {\boldsymbol{\sigma}_{(t|s)}}^2 \boldsymbol{I}\right), \tag{5}$$

where $s = t - 1$, $\boldsymbol{\alpha}_{(t|s)} = \dfrac{\boldsymbol{\alpha}_t}{\boldsymbol{\alpha}_s}$ and ${\boldsymbol{\sigma}_{(t|s)}}^2 = {\boldsymbol{\sigma}_t}^2 - {\boldsymbol{\alpha}_{(t|s)}}^2 {\boldsymbol{\sigma}_s}^2$. In the true denoising process, when the input signals are given, the transition distribution is also normal, and given by

$$q(z_s|z_0, z_t) = \mathcal{N}\left(z_s; \boldsymbol{\mu}(z_0, z_t), {\boldsymbol{\sigma}'_{(t|s)}}^2 \boldsymbol{I}\right) \tag{6}$$

where $\boldsymbol{\mu}(z_0, z_t) = \dfrac{\alpha_{(t|s)}\sigma_{s^2}}{{\sigma_t}^2} z_t + \dfrac{\alpha_s {\sigma_{(t|s)}}^2}{{\sigma_t}^2} z_0$, $\boldsymbol{\sigma}'_{(t|s)} = \dfrac{\sigma_{(t|s)\sigma_s}}{\sigma_t}$.

In the generative denoising process, because the input signals are not given, it firstly uses a neural network to approximate z_0 by $\hat{z}_0$, and then the learned transition distribution which is similar to the eqn (6) is written as

$$p(z_s|z_t) = \mathcal{N}\left(z_s; \boldsymbol{\mu}(\hat{z}_0, z_t), {\sigma'_{t|s}}^2 \boldsymbol{I}\right). \tag{7}$$

Besides, by rewriting the variational lower bound on the likelihood of z_0, the loss function can be simplified to

$$L_{\text{cont}} = \sum_{t=1}^{T} \text{KL}(q(z_s|z_0, z_t)|p(z_s|z_t)). \tag{8}$$

Instead of directly predicting $\hat{z}_0$, using neural networks, diffusion models try to predict the added noise in a score-matching way. Specifically, according to eqn (3), $z_t = \alpha_t z_0 + \sigma_t \in_t$, where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \boldsymbol{I})$, and then the neural network φ predicts $\widehat{\boldsymbol{\epsilon}} = \phi(\boldsymbol{z}_t, \boldsymbol{t})$, so that $\hat{z}_0 = \dfrac{1}{\alpha_t} \boldsymbol{z}_t - \dfrac{\sigma_t}{\alpha_t} \widehat{\boldsymbol{\epsilon}}_t$. Further, the simplified loss function can is written as

$$L_{\text{cont}} = \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \boldsymbol{I})} \left[\frac{1}{2}\left(1 - \frac{\text{SNR}(s)}{\text{SNR}(t)}\right) \|\boldsymbol{\epsilon}_t - \widehat{\boldsymbol{\epsilon}}_t\|^2\right]. \tag{9}$$

Diffusion on discrete variables. Differing from Gaussian diffusion processes that operate in continuous spaces, diffusion models for discrete variables[46,47] firstly define the diffusion process of random variables $z_t \in \{1, \ldots, K\}$ with K categories as

$$q(z_t|z_0) = \text{Cat}(z_0 \bar{\boldsymbol{Q}}_t), \tag{10}$$

where z_t is the one-hot vector of z_t, $\bar{\boldsymbol{Q}}_t = \boldsymbol{Q}_1 \boldsymbol{Q}_2 \ldots \boldsymbol{Q}_t$ with $[\boldsymbol{Q}_t]_{ij} = q(z_t = j|z_s = i)$ denoting diffusion transition probabilities, and $[z_0 \bar{\boldsymbol{Q}}_t]_i$ is the probability of $z_t = i$. The true denoising process is given as

$$q(z_s|z_t, z_0) = \text{Cat}\left(\frac{\boldsymbol{z}_t \boldsymbol{Q}_t^{\top} \odot \boldsymbol{z}_0 \overline{\boldsymbol{Q}}_{t-1}^{\top}}{\boldsymbol{z}_0 \overline{\boldsymbol{Q}}_t \boldsymbol{z}_t^{\top}}\right), \tag{11}$$

Here we employ the absorbing discrete diffusion model, which parameterizes the transition matrix $\boldsymbol{Q}_t$ as

$$[\boldsymbol{Q}_t]_{ij} = \begin{cases} 1 & \text{if} \quad i = j = K+1 \\ 1 - \beta_t & \text{if} \quad i = j \neq K+1 \\ \beta_t & \text{if} \quad j = K+1,\ i \neq K+1 \end{cases}, \tag{12}$$

where $K + 1$ is an absorbing state, usually denoted as [MASK] token in text generation. β_t monotonically increases from 0 to 1, means that when $t = T$, all the discrete variables are absorbed into the $K + 1$ category.

Due to the effectiveness of BERT-style training, a neural network is used to directly predict $p(z_0|z_t)$ rather than $p(z_s|z_t)$, leading to a BERT-like training objective as

$$L_{\text{disc}} = \sum_{t=1}^{T} \lambda(t) \mathbb{E}_{z_t \sim q(z_t|z_0)}[\log p(z_0|z_t)], \tag{13}$$

where $\lambda(t)$ is the weights at different time steps. For example, $\lambda(t) = \dfrac{1}{T}$ leads to equal weights

Symmetry in physics

For molecular systems, the atoms' positions are represented as 3D Cartesian coordinates, so both equivariance of the transition distributions and invariance of the data distribution of 3D structure are required for generalization w.r.t. the SE(3) group. Formally, $f: \mathbb{R}^3 \to \mathbb{R}^3$ is an equivariant function w.r.t. SE(3) group, if for any rotation and translation transformation in the group, which is represented by R as orthogonal matrices and $t \in \mathbb{R}^3$ respectively, $f(R\boldsymbol{x} + t) = Rf(\boldsymbol{x}) + t$. If $f : \mathbb{R}^3 \to \mathbb{D}$ is invariant w.r.t. SE(3) group, then $f(R\boldsymbol{x} + t) = f(\boldsymbol{x})$, where $\mathbb{D}$ can be any domain.

In the setting of generative models, the learned distribution $p(\boldsymbol{x})$ should be invariant to SE(3) group.[28] Köhler *et al.*[48] showed that an invariant distribution composed with an equivariant invertible function results in an invariant distribution. In the setting of diffusion models, the transition distribution is defined to be SE(3)-equivariant if $p(\boldsymbol{x}_s|\boldsymbol{x}_t) = p(R\boldsymbol{x}_s + t|R\boldsymbol{x}_t + t)$. Xu *et al.*[49] showed if $p(\boldsymbol{x}_T)$ is an SE(3)-invariant distribution, and for any t, $p(\boldsymbol{x}_s|\boldsymbol{x}_t)$is equivariant, then $p(\boldsymbol{x}_0)$ is also SE(3)-invariant. However, because the translation equivariance cannot be preserved in the diffusion process (see Appendix), we fix the center of mass (CoM) to zero to avoid all the translation transformation, and diffuse and denoise the atoms' coordinates in the linear subspace with $\sum_{i \in \mathcal{J}_{\text{mol}}} \boldsymbol{x}_i = 0$. Further, in the reverse process, because the initial distribution of $p(\boldsymbol{x}_T)$ is set as standard Gaussian, the distribution naturally satisfies invariance to rotation transformation. The zero CoM trick in the corresponding denoising process also circumvents that $p(\boldsymbol{x}_T + \boldsymbol{t}) = p(\boldsymbol{x}_T)$, which makes it impossible for p to be a distribution.

Target-aware diffusion process in DiffBP

By using the notations of diffusion models, we write the input binding system as $\mathcal{C}_0$, and the intermediate noisy system as $\mathcal{C}_t$, where $t = 1, \ldots, T$. Note that both the diffusion and denoising processes are all performed on $\mathcal{M}$, while the protein system $\mathcal{P}$ keeps unchanged in the processes.

Therefore, for notation simplicity, we omit the subscript $i \in \mathcal{J}_{\text{mol}}$ in the following and assume that the forward diffusion process as

$$q(\boldsymbol{x}_{i,t}, \boldsymbol{a}_{i,t}|\mathcal{C}_0) = q(\boldsymbol{x}_{i,t}|\boldsymbol{x}_{i,0}, \mathcal{P}) q(\boldsymbol{a}_{i,t}|\boldsymbol{a}_{i,0}, \mathcal{P}). \tag{14}$$

Diffusion on continuous positions. For the continuous 3D coordinate x_i of each atom in the molecule, the forward diffusion process is written as

$$\boldsymbol{x}_{i,t} = \alpha_t \boldsymbol{x}_{i,0} + \sigma_t \epsilon_{i,t}, \tag{15}$$

where $\boldsymbol{\epsilon}_{i,t} \sim \mathcal{N}(0, \boldsymbol{I})$, where $0 \in \mathbb{R}^3, \boldsymbol{I} \in \mathbb{R}^{3\times 3}$. The noise schedule of $\{\alpha_t\}_{t=1}^{T}$ and $\{\sigma_t\}_{t=1}^{T}$ are chosen as a simple polynomial scheme (see Appendix). To fix the CoM of all the intermediate molecules to zero, we translate $\{\boldsymbol{x}_{i,t}\}_{i=1}^{N}$ so that $\sum_i x_{i,t} = 0$ for $t = 1, \ldots, T$.

Diffusion on discrete types. For the discrete atom element type, we use the absorbing diffusion model, where the noise schedule is chosen as uniform. In detail, assume $K + 1$ is the absorbing state, and the atom element type $a_{i,t} \in \{1, \ldots, K + 1\}$, corresponding to its one-hot vector $\boldsymbol{a}_{i,t} \in \{0,1\}^{K+1}$, then

$$\begin{aligned} q(a_{i,t} = a_{0,t} | a_{0,t}) &= 1 - \frac{t}{T}; \\ q(a_{i,t} = K + 1 | a_{0,t}) &= \frac{t}{T}. \end{aligned} \tag{16}$$

Equivariant graph denoiser in DiffBP

To learn the transition distribution $p(\boldsymbol{x}_{i,t-1}, \boldsymbol{a}_{i,t-1} | \mathcal{C}_t)$, we use the EGNN[32] satisfying rotational equivariance w.r.t. $\boldsymbol{x}_i$and invariance w.r.t. $\boldsymbol{a}_i$. Specifically,

$$p(a_{i,t-1}, \boldsymbol{x}_{i,t-1} | \mathcal{C}_t) = p\left(a_{i,t-1}, R\boldsymbol{x}_{i,t-1} \middle| \left\{\left(\boldsymbol{a}_{j,t}, R\boldsymbol{x}_{j,t}\right)\right\}_{j=1}^{N+M}\right). \tag{17}$$

Then a commonly-used SE(3)-EGNN reads

$$\left[\widehat{\boldsymbol{\epsilon}}_{i,t}, \hat{\boldsymbol{p}}_{i,0}\right] = \phi_\theta\left(\left\{\left(\boldsymbol{a}_{j,t}, \boldsymbol{x}_{j,t}\right)\right\}_{j=1}^{N+M}, t\right), \tag{18}$$

where the l-th equivariant convolutional layer is defined as

$$\begin{aligned} \boldsymbol{v}_{i,j} &= \psi_{\text{hid}}\left(\left\|\boldsymbol{x}_i^{(l)} - \boldsymbol{x}_j^{(l)}\right\|^2, \boldsymbol{h}_i^{(l)}, \boldsymbol{h}_j^{(l)}, \boldsymbol{e}_i, \boldsymbol{e}_j, \boldsymbol{e}_t\right); \\ \boldsymbol{h}_i^{(l+1)} &= \boldsymbol{h}_i^{(l)} + \sum_{j\in\nu(i)} \boldsymbol{v}_{i,j}\ ; \\ \boldsymbol{u}_{i,j} &= \psi_{\text{eqv}}\left(\left\|\boldsymbol{x}_i^{(l)} - \boldsymbol{x}_j^{(l)}\right\|^2, \boldsymbol{h}_i^{(l+1)}, \boldsymbol{h}_j^{(l+1)}, \boldsymbol{e}_i, \boldsymbol{e}_j, \boldsymbol{e}_t\right); \\ \boldsymbol{x}_i^{(l+1)} &= \boldsymbol{x}_i^{(l)} + \sum_{j\in\nu(i)} \frac{\left(\boldsymbol{x}_i^{(l)} - \boldsymbol{x}_j^{(l)}\right)}{\left\|\boldsymbol{x}_i^{(l)} - \boldsymbol{x}_j^{(l)}\right\|^2} \boldsymbol{u}_{i,j}, \end{aligned} \tag{19}$$

where $i \in \mathcal{J}_{\text{mol}}$, and $j \in \mathcal{J}_{\text{mol}} \cup \mathcal{J}_{\text{pro}}$. The input e_i is the atom type embedding of $\boldsymbol{a}_i$, $\boldsymbol{e}_t$ is the time embedding of t, and $\nu(i)$ is the neighborhood of i established with KNN according to $\mathcal{C}_t$. $\boldsymbol{h}_i^{(l)} \in \mathbb{R}^{D_h}$ is the i-th atom's hidden state in the l-th layer which is rotational-invariant.

Another alternative architecture of the graph denoiser is geometric vector perceptron.[33,50] In general, we find these two architectures have close performance empirically. In the final layer, a softmax function following a multi-layer perceptron $f_{\text{mlp}}: \mathbb{R}^{D_h} \to \mathbb{R}^K$ is used to transform the logits $f_{\text{mlp}}(\boldsymbol{h}_i^{(L)})$ into atom element type probabilities $\hat{p}_{i,0}$.

Optimization objective in DiffBP

Denoising continuous positions. As we set the final layer's output as $\widehat{\boldsymbol{\epsilon}}_{i,t} = \boldsymbol{x}_{i,t}^{(L)}$, and $\mathbb{E}[\boldsymbol{\epsilon}_{i,t}] = 0$, we firstly translate $\widehat{\boldsymbol{\epsilon}}_{i,t}$, so that $\sum_i \widehat{\boldsymbol{\epsilon}}_{i,t} = 0$, and then employ eqn (9) as our loss on atom's continuous position, which reads

$$L_{\text{pos}} = \sum_{t=1}^{T} \sum_{i=1}^{N} \mathbb{E}_{\boldsymbol{\epsilon}_{i,t}\sim\mathcal{N}(0,\boldsymbol{I})}\left[\frac{1}{2}\left(1 - \frac{\text{SNR}(s)}{\text{SNR}(t)}\right)\left\|\boldsymbol{\epsilon}_{i,t} - \widehat{\boldsymbol{\epsilon}}_{i,t}\right\|^2\right]. \tag{20}$$

Denoising discrete types. The rotational-invariant probability vector $\hat{p}_{i,0} = \text{softmax}(f_{\text{mlp}}(\boldsymbol{h}_i))$ gives the distribution of the i-th atom's element type. We use eqn (13) as the training loss, to recover the atoms of the absorbing type $K + 1$ back to its true type with the uniform weights of time.

$$L_{\text{type}} = \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{a_{i,t}\sim q\left(a_{i,t}\middle|a_{i,0}\right)}\left[\sum_{i=1, a_{i,t}=K+1}^{N} \text{CE}\left(\boldsymbol{a}_{i,0}, \hat{\boldsymbol{p}}_{i,0}\right)\right], \tag{21}$$

where $\text{CE}(\cdot,\cdot)$ is the cross-entropy loss, which is only calculated on the absorbing types.

Avoiding intersection for binding. In the binding site, chances are that the noisy atom positions x_t go across the surface of a protein, leading to the intersection of proteins and molecules. In the generative denoising process, we hope to include the inductive bias of non-intersection. To achieve it, we add an intersection loss as a regularization term in protein docking.[51] We turn to previous works on the surface of proteins and point cloud reconstruction,[52,53] where the surface of a protein point cloud $\{\boldsymbol{x}_j : j \in \mathcal{J}_{\text{pro}}\}$ is firstly defined as $\{\boldsymbol{x} \in \mathbb{R}^3 : S(\boldsymbol{x}) = \gamma\}$, where $S(x) = -\boldsymbol{\rho}\ln\left(\sum_{j\in\mathcal{I}_{\text{pro}}} \exp(-\|\boldsymbol{x} - \boldsymbol{x}_j\|^2/\boldsymbol{\rho})\right)$. In this way, $\{\boldsymbol{x} \in \mathbb{R}^3 : S(\boldsymbol{x}) < \gamma\}$ is the interior of the protein, and the atoms of the binding molecule should be forced to lay in $\{\boldsymbol{x} \in \mathbb{R}^3 : S(\boldsymbol{x}) > \gamma\}$. As a result, the inductive bias as a regularization loss function reads

$$L_{\text{reg}} = \sum_{i=1}^{N} \max(0, \gamma - S(\hat{\boldsymbol{x}}_{i,0})), \tag{22}$$

where $\hat{x}_{i,0}$ is the approximated positions at $t = 0$, as $\hat{\boldsymbol{x}}_{i,0} = \frac{1}{\alpha_t}\boldsymbol{x}_{i,t} - \frac{\sigma_t}{\alpha_t}\widehat{\boldsymbol{\epsilon}}_{i,t}$. γ and ρ are predefined.

Reconstruction on other attributes. In the denoising step of $t = 1$, our equivariant graph denoiser can also be used to recover other attributes of the binding systems, which is based on $\boldsymbol{x}_{i,0} \approx \boldsymbol{x}_{i,1}$ as the $\text{SNR}(1) \to \infty$. For example, in order to predict the binary atom attribute z^{aro} of 'is_aromatic', another prediction head $g_{\text{mlp}}: \mathbb{R}^{D_h} \to \mathbb{R}$ can be defined, such that $p(z_i^{\text{aro}} = 1 | \mathcal{C}_1) = p_i^{\text{aro}} = \text{sigmoid}(g_{\text{mlp}}(\boldsymbol{h}_{i,1}^{(l)}))$. And the binary cross-entropy loss can be used to train the denoiser, leading to

$$L_{\text{rec}} = \sum_{i=1}^{N} \text{BCE}\left(p_i^{\text{aro}}, y_i^{\text{aro}}\right), \tag{23}$$

where y_i^{aro} is the i-th atom's ground-truth label on the attribute of 'is_aromatic'.

To sum up, the overall loss function used in the training process reads

$$L = L_{\text{pos}} + L_{\text{type}} + L_{\text{reg}} + L_{\text{rec}}. \tag{24}$$

Generative denoising process in DiffBP

Generating atom positions. In generating the atom positions, we firstly sample $\boldsymbol{x}_{i,T} \sim \mathcal{N}(0, \boldsymbol{I})$. Then $\boldsymbol{x}_{i,s}$ is drawn from

$p(\boldsymbol{x}_{i,s}|\boldsymbol{x}_{i,t}) = \mathcal{N}(\boldsymbol{x}_{i,s}; \boldsymbol{\mu}(\hat{\boldsymbol{x}}_{i,0}, \boldsymbol{x}_{i,t}), \sigma'^{2}_{t|s}\boldsymbol{I})$, where the parameters is calculated by

$$\begin{aligned}\mu(\hat{\boldsymbol{x}}_{i,0}, \boldsymbol{x}_{i,t}) &= \frac{\alpha_{t|s}\sigma_s^2}{\sigma_t^2}\boldsymbol{x}_{i,t} + \frac{\alpha_s \sigma_{t|s}^2}{\sigma_t^2}\hat{\boldsymbol{x}}_{i,0} \\ &= \frac{1}{\alpha_{t|s}}\boldsymbol{x}_{i,t} - \frac{\sigma_{t|s}^2}{\alpha_{t|s}\sigma_t}\hat{\boldsymbol{\epsilon}}_{i,t}; \\ \sigma'_{t|s} &= \frac{\sigma_{t|s}\sigma_s}{\sigma_t}.\end{aligned} \tag{25}$$

$\mathbb{E}[\boldsymbol{x}_{i,T}] = 0$, we can easily obtain that $\mathbb{E}[\boldsymbol{x}_{i,t}] \approx 0$. Therefore, we choose the mass center of the binding molecules as zero by translating the protein–molecule binding system such that the origin of the global coordinate system coincides with the molecule's CoM. Otherwise, it is challenging for the generated molecules to be located in the binding site.

Another problem raised by this translation is that the CoM of the molecules is unknown in the generation process, since only the information of proteins is the pre-given context as

Algorithm1 Training DiffBP (Sampling algorithm is given in Appendix. Algorithm. S1)

Input: Zero-centered molecule $\{\boldsymbol{a}_j, \boldsymbol{x}_j\}_{j=1}^{N}$, protein $\{\boldsymbol{a}_j, \boldsymbol{x}_j\}_{j=1}^{M}$, and graph denoiser $\boldsymbol{\phi}_\theta$

Sample $t \sim \mathcal{U}(0, \ldots, T)$, $\boldsymbol{\epsilon}_{i,t} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$

Subtract center of mass from $\boldsymbol{\epsilon}_{i,t}$

Compute $\boldsymbol{x}_{i,t} = \alpha_t \boldsymbol{x}_{i,0} + \sigma_t \boldsymbol{\epsilon}_{i,t}$

Sample uniformly $\mathcal{I}_{\text{abs}}$, $|\mathcal{I}_{\text{abs}}|/M = t/T$

If $i \in \mathcal{I}_{\text{abs}} \cap \mathcal{I}_{\text{mol}}$ **then**

Set $a_{i,t} = K + 1$

end if

Compute $[\hat{\boldsymbol{\epsilon}}_{i,t}, \hat{\boldsymbol{p}}_{i,0}] = \boldsymbol{\phi}_\theta(\{(\boldsymbol{a}_{j,t}, \boldsymbol{x}_{j,t})\}_{j=1}^{N+M}, t)$

Minimize L as defined in Eq.(24)

Note that $\mathcal{I}_{\text{abs}}$ is the index set of atoms of absorbing types.

Generating element types. For atoms' element types, $\{a_{i,T}\}$ are all set as the absorbing type of $K+1$ at time T. Then, in each step at t, we randomly select $(T-t)/T$ of atoms to predict their element types. Besides, if the element type of an atom has been recovered in the past steps, it would not change, which indicates that the recovery of the element type of each atom is only performed once.

Pre-generation models. It is noted that the translational invariance cannot be satisfied without the zero CoM trick. Moreover, there are several mass centers of the system, such as the protein's CoM and the protein–molecule CoM. Choosing a proper CoM of the system is a matter of careful design. We propose that for the diffusion model, the zero CoM should be performed according to the molecule. As shown in eqn (25), it can be derived that $\mathbb{E}[\boldsymbol{x}_{i,s}] = \frac{1}{\alpha_{t|s}}\mathbb{E}[\boldsymbol{x}_{i,t}] - \frac{\sigma_{t|s}^2}{\alpha_{t|s}\sigma_t}\mathbb{E}[\hat{\boldsymbol{\epsilon}}_{i,t}]$. For a well-trained equivariant graph neural network, it is expected that $\mathbb{E}\|\hat{\boldsymbol{\epsilon}}_{i,t} - \boldsymbol{\epsilon}_{i,t}\|^2 \to 0$. By Jenson's inequality, $(\mathbb{E}[\|\hat{\boldsymbol{\epsilon}}_{i,t} - \boldsymbol{\epsilon}_{i,t}\|])^2 \le \mathbb{E}[\|\hat{\boldsymbol{\epsilon}}_{i,t} - \boldsymbol{\epsilon}_{i,t}\|^2]$, which indicates that $\mathbb{E}[\hat{\boldsymbol{\epsilon}}_{i,t}] \approx 0$ for any $i \in \mathcal{J}_{\text{mol}}$ and $0 \le t \le T$. Combining with conditions. Besides, as a full-atom generation method, it is necessary to assign the atom numbers of the binding molecules before both diffusion and denoising process. To address these problems, an additional graph neural network $\varphi_\omega(\mathcal{P}) \in \mathbb{R}^3 \times \mathbb{Z}^+$ with the same intermediate architectures as shown in eqn (19) is pre-trained as a pre-generation model, which aims to generate the atom numbers as well as the molecules' CoM before the denoising process. In detail,

$$\left[\boldsymbol{x}_{\text{c}}^{\text{mol}}, N^{\text{mol}}\right] = \varphi_\omega\left(\left\{\boldsymbol{a}_j, \boldsymbol{x}_j\right\}_{j \in \mathcal{J}_{\text{pro}}}\right), \tag{26}$$

where $\boldsymbol{x}_{\text{c}}^{\text{mol}}$ is finally obtained by an SE(3)-equivariant pooling as $\text{mean}(\{\boldsymbol{x}_j^{(L)}\}_{j=1}^{M}) = \frac{1}{M}\sum_{j=1}^{M}\boldsymbol{x}_j^{(L)}$. Moreover, an SE(3)-invariant feature $\max(u_{\text{mlp}}(\{\boldsymbol{h}_j^{(L)}\}_{j=1}^{M}))$ is used to predict the distribution of atom numbers with $u_{\text{mlp}}: \mathbb{R}^{D_h} \to \mathbb{R}^{N_s}$, where N_s is the number of different molecule sizes which can be statistically obtained by the training set, and u_{mlp} transforms the latent features into logits for calculating the probability. A dictionary is used to map the predicted class $\hat{n}_s$ to $\mathbb{Z}^+$. For example, dict $= \{1:18, 2:27\}$

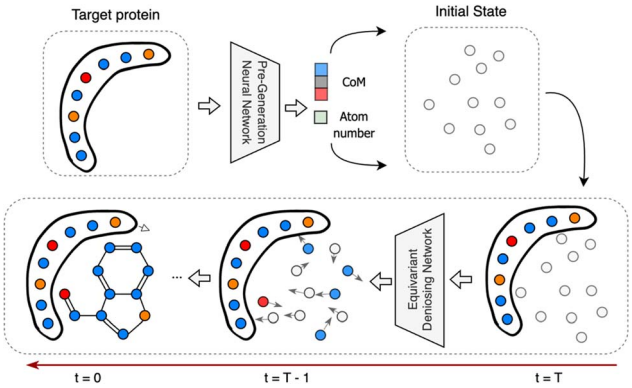

Fig. 9 Overall framework as an illustration of the workflows of DiffBP. The pre-generation model first predicts the center of mass and atom number of the generated molecules. Then, in the generative denoising process (t goes from T to 0), the equivariant denoising network outputs both 3D-positions and element types of each atom.

means when $\hat{n}_s = 1$, the atom number N^{mol} will be assigned as 18. Once the molecule's CoM and atom number are obtained, we translate the protein system by $\boldsymbol{x}_j := \boldsymbol{x}_j - \boldsymbol{x}_c^{\text{mol}}$ for $j = 1, \ldots, M$, and set $N := N^{\text{mol}}$ as the molecule size for generation of the diffusion model. The overall workflow of DiffBP is presented in Fig. 9.

Detailed experimental setup

Datasets. For evaluation of our model, we follow the previous works[23,26] and use the CrossDocked2020 (ref. 22) dataset to generate ligand molecules that bind to target proteins' pockets based on the pocket structures, with the same split of training and test set. For atom-level contexts of protein pockets, we employ element types and the amino acids the pocket atoms belong to as their SE(3)-invariant features, and atoms' positions as SE(3) equivariant features. No additional contexts on ligands are used except atom element types and positions in training because in generating process, the only given contexts are pocket atoms' features.

Baselines. Three state-of-the-art methods including 3DSBDD,[24] Pocket2Mol[25] and GraphBP[26] are employed as benchmarks. These three methods are all auto-regressive generative models. Pocket2Mol and GraphBP directly generate the continuous coordinates of atoms. In comparison, 3DSBDD generates the atoms' position on regular grids. We generate 100 molecules for each protein pocket to calculate metrics for further evaluation.

Metrics. Following GraphBP and LiGAN, for the calculation of affinity score, we adopt Gnina[34,35] which is an ensemble of CNN scoring functions that were trained on the CrossDocked2020 dataset. Empirical studies show such CNN predicted affinity is more accurate than Autodock Vina empirical scoring function. Quantitatively, two metrics on affinity score are reported: (i) LE (Ligand Efficiency) referring to a comparison of the binding score among same-size ligands, which is calculated as the percentage of generated molecules that have higher predicted binding affinity than the corresponding reference molecules. It measures the maximum performance that the model can achieve, and the larger the better. (ii) Mean Percentage Binding Gap (MPBG) which is proposed in our paper to measure the mean performance of the model. For a single pair of a protein pocket and its reference molecule, it is calculated by $\frac{1}{N_{\text{gen}}}\sum_{i=1}^{N_{\text{gen}}}\frac{\text{Aff}^{\text{ref}} - \text{Aff}_i^{\text{gen}}}{\text{Aff}_{\text{ref}}} \times 100\%$. A smaller MPBG represents the better binding affinity of the generated molecules on average. Besides, other scores are used for further comparison including (i) QED as quantitative estimation of drug-likeness; (ii) SA as normed synthetic accessibility score, ranging from 0 to 1; (iii) Sim as the average Tanimoto similarities of the multiple generated molecules in one pocket, reviling the diversity of generation; (iv) LPSK as the ratio of the generated drug molecules satisfying the Lipinski's rule of five. These chemical properties are calculated by RDKit package.[54] After the molecules are generated by the model, Openbabel[41] is used to construct chemical bonds between atoms, and Universal Force Field (UFF) minimization[55] is used for refinement before the metric calculation. Note that in Pocket2Mol, refinement with UFF is not performed, so we report the metric of Pocket2Mol-without-UFF in Appendix Table S1† and the reported metrics in this part are all based on Pocket2Mol-with-UFF.

Data availability

The data and source code of this study are freely available at GoogleDrive (**https://drive.google.com/drive/folders/1bSDoQBERjXXvwAFRscuZXcyiKAyjRHuI?usp=sharing**) to allow replication of the results.

Author contributions

Stan. Z. Li and Tingjun Hou proposed the research topics, made an overall plan, organized collaboration, and guided the writing of this paper and the experiments. Haitao Lin and Meng Liu constructed the overall idea of the diffusion model. Yufei Huang and Lirong Wu proposed and solved the problem in the pre-generation models, and assisted to run the results for baseline models. Haitao Lin wrote the manuscripts, which are revised by Yufei Huang, Meng Liu, Odin Zhang, Xuanjing Li, Shuiwang Ji and Tingjun Hou. Odin Zhang and Tingjun Hou's expertise in computational chemistry helped to improve the paper in terms of drug design. Siqi Ma develop the platform sever for public access. Meng Liu and Jishui Wang's expertise in graph neural networks helps the method perform better in practice.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

This work was supported by National Key R&D Program of China (2022ZD0115100), National Natural Science Foundation

of China Project (U21A20427, 22220102001), and Project (WU2022A009) from the Center of Synthetic Biology and Integrated Bioengineering of Westlake University.

References

1 J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King and D. Baker, Robust deep learning-based protein sequence design using ProteinMPNN, *Science*, 2022, **378**, 49–55.

2 Z. Gao, C. Tan and S. Z. Li, Alphadesign: A graph protein design method and benchmark on alphafolddb, *arXiv*, 2022, preprint, arXiv:2202.01079, DOI: **10.48550/arXiv.2202.01079**.

3 Y. Sun, S. Selvarajan, Z. Zang, W. Liu, Y. Zhu, H. Zhang, W. Chen, H. Chen, L. Li and X. Cai, Artificial intelligence defines protein-based classification of thyroid nodules, *Cell Discovery*, 2022, **8**, 85.

4 C. Shi, S. Luo, M. Xu and J. Tang, *presented in part at the Proceedings of the 38th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2021.

5 X. Zeng, F. Wang, Y. Luo, S.-g. Kang, J. Tang, F. C. Lightstone, E. F. Fang, W. Cornell, R. Nussinov and F. Cheng, Deep generative molecular design reshapes drug discovery, *Cell Rep. Med.*, 2022, **3**, 100794.

6 C. Shi, M. Xu, Z. Zhu, W. Zhang, M. Zhang and J. Tang, Graphaf: a flow-based autoregressive model for molecular graph generation, *arXiv*, 2020, preprint, arXiv:2001.09382, DOI: **10.48550/arXiv.2001.09382**.

7 D. Polykovskiy, A. Zhebrak, B. Sanchez-Lengeling, S. Golovanov, O. Tatanov, S. Belyaev, R. Kurbanov, A. Artamonov, V. Aladinskiy and M. Veselov, Molecular sets (MOSES): a benchmarking platform for molecular generation models, *Front. Pharmacol*, 2020, **11**, 565644.

8 C. Tan, Z. Gao and S. Z. Li, Target-Aware Molecular Graph Generation, *Joint European Conference on Machine Learning and Knowledge Discovery in Databases: Applied Data Science and Demo Track (ECML PKDD 2023)*, 2023.

9 X. Yang, J. Zhang, K. Yoshizoe, K. Terayama and K. Tsuda, ChemTS: an efficient python library for *de novo* molecular generation, *Sci. Technol. Adv. Mater.*, 2017, **18**, 972–976.

10 M. Liu, K. Yan, B. Oztekin and S. Ji, Graphebm: Molecular graph generation with energy-based models, *arXiv*, 2021, preprint, arXiv:2102.00546, DOI: **10.48550/arXiv.2102.00546**.

11 W. Jin, R. Barzilay and T. Jaakkola, Hierarchical generation of molecular graphs using structural motifs, in *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research (PMLR), ed. H. Daume III and A. Singh, 2020, vol. 119, pp. 4839–4848.

12 C. Zang and F. Wang, MoFlow: an invertible flow model for generating molecular graphs, in *ACM SIGKDD*, 2020, pp. 617–626.

13 R. J. Townshend, M. Vögele, P. Suriana, A. Derry, A. Powers, Y. Laloudakis, S. Balachandar, B. Jing, B. Anderson and S. Eismann, Atom3d: Tasks on molecules in three dimensions, *arXiv*, 2020, preprint, arXiv:2012.04035, DOI: **10.48550/arXiv.2012.04035**.

14 Y. Du, T. Fu, J. Sun and S. Liu, Molgensurvey: A systematic survey in machine learning models for molecule design, *arXiv*, 2022, preprint, arXiv:2203.14500, DOI: **10.48550/arXiv.2203.14500**.

15 R. Evans, M. O'Neill, A. Pritzel, N. Antropova, A. Senior, T. Green, A. Žídek, R. Bates, S. Blackwell and J. Yim, Protein complex prediction with AlphaFold-Multimer, *bioRxiv*, 2021, 2021.2010. 2004.463034.

16 J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek and A. Potapenko, Highly accurate protein structure prediction with AlphaFold, *Nature*, 2021, **596**, 583–589.

17 M. Baek, F. DiMaio, I. Anishchenko, J. Dauparas, S. Ovchinnikov, G. R. Lee, J. Wang, Q. Cong, L. N. Kinch and R. D. Schaeffer, Accurate prediction of protein structures and interactions using a three-track neural network, *Science*, 2021, **373**, 871–876.

18 I. R. Humphreys, J. Pei, M. Baek, A. Krishnakumar, I. Anishchenko, S. Ovchinnikov, J. Zhang, T. J. Ness, S. Banjade and S. R. Bagde, Computed structures of core eukaryotic protein complexes, *Science*, 2021, **374**, eabm4805.

19 Y. Huang, L. Wu, H. Lin, J. Zheng, G. Wang and S. Z. Li, Data-Efficient Protein 3D Geometric Pretraining *via* Refinement of Diffused Protein Structure Decoy, *arXiv*, 2023, preprint, arXiv:2302.10888, DOI: **10.48550/arXiv.2302.10888**.

20 I. D. Kuntz, Structure-based strategies for drug design and discovery, *Science*, 1992, **257**, 1078–1082.

21 J. Drews, Drug discovery: a historical perspective, *Science*, 2000, **287**, 1960–1964.

22 P. G. Francoeur, T. Masuda, J. Sunseri, A. Jia, R. B. Iovanisci, I. Snyder and D. R. Koes, Three-dimensional convolutional neural networks and a cross-docked data set for structure-based drug design, *J. Chem. Inf. Model.*, 2020, **60**, 4200–4215.

23 T. Masuda, M. Ragoza and D. R. Koes, Generating 3d molecular structures conditional on a receptor binding site with deep generative models, *arXiv*, 2020, preprint, arXiv:2010.14442, DOI: **10.48550/arXiv.2010.14442**.

24 S. Luo, J. Guan, J. Ma and J. Peng, A 3D generative model for structure-based drug design, *Adv. Neural Inf. Process. Syst.*, 2021, **34**, 6229–6239.

25 X. Peng, S. Luo, J. Guan, Q. Xie, J. Peng and J. Ma, Pocket2mol: efficient molecular sampling based on 3d protein pockets, in *International Conference on Machine Learning*, 2022.

26 M. Liu, Y. Luo, K. Uchino, K. Maruhashi and S. Ji, Generating 3D Molecules for Target Protein Binding, *arXiv*, 2022, preprint, arXiv:2204.09410, DOI: **10.48550/arXiv.2204.09410**.

27 A. Morehead and J. Cheng, Geometry-complete diffusion for 3D molecule generation and optimization, *Commun. Chem.*, 2024, **7**, 150.

28 E. Hoogeboom, V. G. Satorras, C. Vignac and M. Welling, Equivariant diffusion for molecule generation in 3d, *Proceedings of the 39th International Conference on Machine Learning*, PMLR 162, Baltimore, Maryland, USA, 2022.

29 J. Ho, A. Jain and P. Abbeel, Denoising diffusion probabilistic models, *Adv. Neural Inf. Process. Syst.*, 2020, **33**, 6840–6851.
30 Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon and B. Poole, Score-based generative modeling through stochastic differential equations, *arXiv*, 2020, preprint, arXiv:2011.13456, DOI: **10.48550/arXiv.2011.13456**.
31 H. Cao, C. Tan, Z. Gao, Y. Xu, G. Chen, P.-A. Heng and S. Z. Li, A survey on generative diffusion model, *arXiv*, 2022, preprint, arXiv:2209.02646, DOI: **10.48550/arXiv.2209.02646**.
32 E. Hoogeboom, V. G. Satorras, C. Vignac and M. Welling, E(n) equivariant normalizing flows, *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021.
33 B. Jing, S. Eismann, P. N. Soni and R. O. Dror, Equivariant graph neural networks for 3d macromolecular structure, *International Conference on Machine Learning, Workshop on Computational Biology*, 2021.
34 M. Ragoza, J. Hochuli, E. Idrobo, J. Sunseri and D. R. Koes, Protein–ligand scoring with convolutional neural networks, *J. Chem. Inf. Model.*, 2017, **57**, 942–957.
35 A. T. McNutt, P. Francoeur, R. Aggarwal, T. Masuda, R. Meli, M. Ragoza, J. Sunseri and D. R. Koes, GNINA 1.0: molecular docking with deep learning, *J. Cheminf.*, 2021, **13**, 1–20.
36 M. Wang, C.-Y. Hsieh, J. Wang, D. Wang, G. Weng, C. Shen, X. Yao, Z. Bing, H. Li and D. Cao, Relation: A deep generative model for structure-based *de novo* drug design, *J. Med. Chem.*, 2022, **65**, 9478–9492.
37 G. Schneider and D. E. Clark, Automated *de novo* drug design: are we nearly there yet?, *Angew. Chem., Int. Ed.*, 2019, **58**, 10792–10803.
38 S. Axelrod and R. Gomez-Bombarelli, GEOM, energy-annotated molecular conformations for property prediction and molecular generation, *Sci. Data*, 2022, **9**, 185.
39 S. Luo, Y. Su, X. Peng, S. Wang, J. Peng and J. Ma, Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures, *Adv. Neural Inf. Process. Syst.*, 2022, **35**, 9754–9767.
40 O. Trott and A. J. Olson, AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading, *J. Comput. Chem.*, 2010, **31**, 455–461.
41 N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch and G. R. Hutchison, Open Babel: An open chemical toolbox, *J. Cheminf.*, 2011, **3**, 1–14.
42 W. L. DeLano, Pymol: An open-source molecular graphics tool, *CCP4 Newsletter on Protein Crystallography*, 2002, vol. 40, pp. 82–92.
43 J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan and S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, in *International Conference on Machine Learning*, PMLR, 2015, pp. 2256–2265.
44 Y. Song and S. Ermon, Generative modeling by estimating gradients of the data distribution, *Adv. Neural Inf. Process. Syst.*, 2019, **32**, 11895–11907.
45 D. Kingma, T. Salimans, B. Poole and J. Ho, Variational diffusion models, *Adv. Neural Inf. Process. Syst.*, 2021, **34**, 21696–21707.
46 J. Austin, D. D. Johnson, J. Ho, D. Tarlow and R. Van Den Berg, Structured denoising diffusion models in discrete state-spaces, *Adv. Neural Inf. Process. Syst.*, 2021, **34**, 17981–17993.
47 S. Bond-Taylor, P. Hessey, H. Sasaki, T. P. Breckon and C. G. Willcocks, Unleashing transformers: parallel token prediction with discrete absorbing diffusion for fast high-resolution image generation from vector-quantized codes, in *European Conference on Computer Vision (ECCV)*, 2022.
48 J. Köhler, L. Klein and F. Noé, Equivariant flows: exact likelihood generative learning for symmetric densities, *Proceedings of the 37th International Conference on Machine Learning*, Article No.: 497, 2020, pp. 5361–5370.
49 M. Xu, L. Yu, Y. Song, C. Shi, S. Ermon and J. Tang, Geodiff: A geometric diffusion model for molecular conformation generation, *International Conference on Learning Representation*, 2022.
50 B. Jing, S. Eismann, P. Suriana, R. J. Townshend and R. Dror, learning from protein structure with geometric vector perceptrons, *International Conference on Learning Representation*, 2021.
51 O.-E. Ganea, X. Huang, C. Bunne, Y. Bian, R. Barzilay, T. Jaakkola and A. Krause, Independent se (3)-equivariant models for end-to-end rigid protein docking, *International Conference on Learning Representation*, 2023.
52 F. Sverrisson, J. Feydy, B. E. Correia and M. M. Bronstein, Fast end-to-end learning on protein surfaces, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 15267–15276.
53 V. Venkatraman, Y. D. Yang, L. Sael and D. Kihara, Protein-protein docking using region-based 3D Zernike descriptors, *BMC Bioinf.*, 2009, **10**, 1–21.
54 G. Landrum, RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling, *Greg Landrum*, 2013, vol. 8, p. 31.
55 A. K. Rappé, C. J. Casewit, K. Colwell, W. A. Goddard III and W. M. Skiff, UFF, a full periodic table force field for molecular mechanics and molecular dynamics simulations, *J. Am. Chem. Soc.*, 1992, **114**, 10024–10035.