A Protein-Ligand Interaction-focused 3D Molecular Generative Framework for Generalizable Structure-based Drug Design

Wonho Zhung[1], Hyeongwoo Kim[1], Woo Youn Kim[1,2,3*]

[1]Chemistry, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon, 34141, Republic of Korea.
[2]HITS Incorporation, 124 Teheran-ro, Gangnam-gu, Seoul, 06234, Republic of Korea.
[3]KI for Artificial Intelligence, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon, 34141, Republic of Korea.

*Corresponding author(s). E-mail(s): wooyoun@kaist.ac.kr;
Contributing authors: zpeter97@kaist.ac.kr; novainco98@kaist.ac.kr;

Abstract

Deep generative models have been the subject of immense interest in various fields of science. While seeking a molecule that favorably binds to a target is a long-established goal of drug design, various generative models have emerged to reach the goal. Here, we employ the concept of intermolecular interactions between a protein and a ligand in a 3D molecular generative model, empowering the generalizable structure-based drug design. Inspired by how the practitioners manage to improve the potency of a ligand toward a target protein, we devised a strategy where prior knowledge of appropriate interactions navigates the ligand generation. We thus propose an interaction-focused generative framework, which establishes a local interaction condition to capture the surrounding pocket environment. We demonstrate that the condition enables precise control of ligand generation, justifying its effectiveness in guiding a ligand design inside a binding pocket. Through this strategy, the generated ligands could stably bind to the target pocket by forming favorable interactions, regardless of pocket type. Furthermore, we highlight the broad applicability of our framework by leveraging the site-specific interaction condition suitable for designing ligands for various purposes.

Keywords: 3D molecular generative model, Prior knowledge, Protein-ligand interaction, Structure-based drug design

1 Introduction

Incorporating adequate prior knowledge is essential in applying deep learning to various fields of science.[1–3] By providing additional constraints that guide a deep learning model to learn more informative and representative features from the input data, numerous works have adopted the domain-specific prior knowledge to achieve noteworthy successes.[4–9] For instance, AlphaFold2[10] utilized co-evolutionary information to predict protein structure by conveying an initial guess of how residues might contact, reducing the possible conformational space of protein folding. As such, prior knowledge can enlighten deep learning models to be well generalized to a particular task or even improve the explainability of model output. The development of a reliable model for designing a hit molecule, a primary goal of computer-aided drug discovery, is one such well-known task where prior knowledge can play a crucial role.[11–13]

Recent advances in deep generative modeling led to the rapid development of the structure-based drug design paradigm, where generative models are employed to provide molecules that can strongly bind to a target protein.[14–18] However, the data-deficient nature often hinders the models' ability to generate promising molecules for unseen targets. Chan *et al.*[19] emphasized that poorly generalized generative models might prioritize the learned chemical space rather than exploring other regions where molecules with desired properties might exist. Regarding such a low generalization problem, there have been two claimed issues that generative models may encounter. First, designed molecules often omit the structure-activity relationships(SARs) and, therefore, unfavorably interact with the target.[17, 19, 20] This can result in low binding stability and drug potency. Next, designed molecules may lack structural diversity.[21–23] This limits the size of the chemical space that the model covers, losing the opportunity to identify a novel structure. These two problems act more severely when only a few ligand data are available for a protein-of-interest, for example, to target a newly found protein. Thus, in order to design potent and diverse ligand molecules, one should leverage chemical knowledge to encourage a model to extract generalizable rules from the protein-ligand binding data.

More recent works in 3D molecular generative models adopt a 3D structural context of a target binding pocket.[24–29] One such approach was accomplished by Ragoza *et al.*[26]; they represented the electron density of a ligand as voxels and trained their model to reconstruct the voxelized density from the input pocket structure. Designing a 3D structure of a ligand opened the possibility of designing a ligand directly inside a binding pocket. Luo *et al.*[27] first proposed a generative model employing the concept, which sequentially adds ligand atoms in a pocket-constrained space. This approach can explicitly utilize the geometric information of the surrounding protein atoms to avoid spatial occlusion and obtain a suitable ligand binding pose, saving much effort in conformer searching.

Although 3D contexts of both a ligand and a binding pocket can effectively regularize a ligand design process, a well-generalized model should comprehend the underlying rule of the local ligand structures' contribution to the interaction formation with the pocket.[30] Chemical knowledge about protein-ligand interactions has been actively considered in conventional structure-based drug design works. For example, practitioners elaborate on a known pharmacophore with a plausible potency to design a ligand

that forms favorable interactions with pocket residues, achieving higher potency.[31–33] Thus, the concepts of SARs and protein-ligand interactions can act as a suitable prior knowledge to improve the generalizability of the ligand design process while providing explainable reasoning. Nevertheless, current approaches overlook protein-ligand interactions during the generation processes, barely considering them.[34]

Here, we propose an interaction-focused 3D molecular generative framework that explicitly models protein-ligand interactions for a generalizable pocket-constrained ligand design. We first investigate and extract protein-ligand interaction patterns from reference binding structures to inform the model about SARs. Conceived from the idea that a particular non-covalent interaction would have a typical geometric profile, we model a distribution of types and positions of ligand atoms conditioned on a pocket structure and corresponding interaction patterns. Considering a local chemical environment confined by neighboring pocket atoms, our model determines how to design a ligand to fulfill the desired interaction. To our best knowledge, this is the first attempt to model the 3D binding structure of ligands regarding protein-ligand interactions and a local chemical environment of a binding pocket.

2 Results

2.1 Interaction-focused 3D Molecular Generative Framework

We demonstrate two molecular generation tasks - ligand elaboration and *de novo* ligand design - inside a target binding site. The former task aims to grow a known pharmacophore to improve its potency. The latter aims to design a ligand from scratch, providing diverse structures that can fit in a target protein. Both tasks are crucial in structure-based drug design, however challenging due to the enormous size of the chemical space and the complicated chemical environment of a binding site.

Our framework utilizes protein-ligand interaction information to guide ligand generation in a 3D pocket-constrained space. While sequentially adding atoms to a ligand, an atom's type and its position are conditioned by locally surrounding pocket atoms and a desired type of non-covalent interactions. The framework consists of two main stages as illustrated in Fig. 1; in the first stage, the framework investigates pocket atoms to designate possible interaction types from a known binding site. We use four types of non-covalent interactions - hydrogen bonds, salt bridges, hydrophobic interactions, and π-π stackings. Since we used the PDBbind 2020 dataset[35], which originated from the protein data bank(PDB)[36, 37], we only consider the four most dominant interaction types in the PDB.[38]

In the second stage, a ligand is sequentially built up based on the 3D context of a pocket and the interaction condition preset from the first stage. For this purpose, we devised a deep generative model named DeepICL(**Deep I**nteraction-**C**onditioned **L**igand generative model) for carrying out the generation tasks. For the ligand elaboration task, a binding structure of a pharmacophore is specified and used as an initial state. In the *de novo* ligand design task, one can manually select a point inside a pocket, which serves as a starting point. For convenience, we use a center-of-mass of each reference ligand in the following experiments. Since the ligand elaboration is a

truncation of a whole generation process, we formulate only the case of whole ligand generation in the next section.

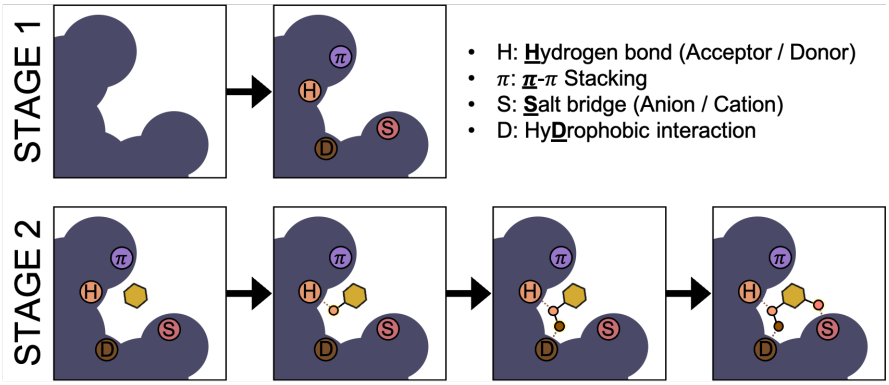

Fig. 1 An illustration of a proposed interaction-focused 3D ligand generative framework. The first stage profiles a protein pocket to designate an interaction pattern on each protein atom. The next stage then sequentially adds ligand atoms inside a protein pocket based on a predetermined interaction condition.

2.1.1 Factorizing a Probability Distribution of Ligand Structure

The goal of the second stage of our framework is to model a probability distribution of a ligand conditioned on a target protein and the interaction patterns. We represent a ligand and a protein as a set of atoms, $\mathbf{L} := \{L_i\}$ and $\mathbf{P} := \{P_j\}$, respectively. Each ligand atom, L_i, is defined as a tuple of an atom type, $\mathbf{X}_i^l \in \mathbb{R}^{F^l}$, and an atom position, $\mathbf{r}_i^l \in \mathbb{R}^3$. Similarly, each protein atom, P_j, is represented by an atom type, $\mathbf{X}_j^p \in \mathbb{R}^{F^p}$, and its position, $\mathbf{r}_j^p \in \mathbb{R}^3$. Note that F^l and F^p denote the dimension of atom features for a ligand and a protein, i and j correspond to a ligand atom and a protein atom index, respectively. Details of atom features are summarized in Appendix A.1. The main objective is to model a conditional probability distribution, $p(\mathbf{L}|\mathbf{P}, \mathbf{I})$, where $\mathbf{I}$ indicates the interaction pattern obtained from the first stage of the framework. We factorize the conditional distribution in an autoregressive manner similar to cG-SchNet[39], where the probability of the upcoming ligand atom depends on the existing atoms. By defining a protein-ligand complex at a time step t as $C_t := (\{L_i\}_{i=1}^t, \{P_j\})$, we can formulate the factorization as follows:

$$\begin{aligned} p(\mathbf{L}|\mathbf{P}, \mathbf{I}) &= \prod_{t=1}^{T} \left[p(L_t | \{L_i\}_{i=1}^{t-1}, \{P_j\}, \mathbf{I}) \right] \cdot p(\text{stop}|\mathbf{L}, \mathbf{P}, \mathbf{I}) \\ &= \prod_{t=1}^{T} \left[p(L_t | C_{t-1}, \mathbf{I}) \right] \cdot p(\text{stop}|\mathbf{L}, \mathbf{P}, \mathbf{I}), \end{aligned} \tag{1}$$

where T is the number of ligand atoms. $p(\text{stop}|\mathbf{L}, \mathbf{P}, \mathbf{I})$ is a probability of termination, which determines when to stop the generation. We further factorize the conditional probability of a ligand atom at a time step t as:

$$p(L_t|C_{t-1}, \mathbf{I}) = p(\mathbf{X}_t|C_{t-1}, \mathbf{I}) \cdot p(\mathbf{r}_t|\mathbf{X}_t, C_{t-1}, \mathbf{I}) \tag{2}$$

so that the position of the next ligand atom depends on its atom type. We regard both probabilities of the atom type and position as a joint distribution over each preceding atom in C_{t-1}:

$$p(\mathbf{X}_t|C_{t-1}, \mathbf{I}) \propto \prod_{i=1}^{t-1} p(\mathbf{X}_t|L_i, \mathbf{I}) \cdot \prod_j p(\mathbf{X}_t|P_j, \mathbf{I}), \tag{3}$$

$$p(\mathbf{r}_t|\mathbf{X}_t, C_{t-1}, \mathbf{I}) \propto \prod_{i=1}^{t-1} p(d_{t,i}^{ll}|\mathbf{X}_t, L_i, \mathbf{I}) \cdot \prod_j p(d_{t,j}^{lp}|\mathbf{X}_t, P_j, \mathbf{I}), \tag{4}$$

where $d_{t,i}^{ll}$ and $d_{t,j}^{lp}$ are Euclidean distances between corresponding pairs of atoms, respectively. We assume that the type and position of a ligand atom mostly depend on its proximal protein atoms since a non-covalent interaction between a protein and a ligand happens between closely contacting atom pairs. Hence, the probabilities conditioned on protein atoms can be approximated as follows:

$$\prod_j p(\mathbf{X}_t|P_j, \mathbf{I}) \simeq \prod_{j \in \mathcal{N}_k(t^*)} p(\mathbf{X}_t|P_j, \mathbf{I}), \tag{5}$$

$$\prod_j p(d_{t,j}^{lp}|\mathbf{X}_t, P_j, \mathbf{I}) \simeq \prod_{j \in \mathcal{N}_k(t^*)} p(d_{t,j}^{lp}|\mathbf{X}_t, P_j, \mathbf{I}), \tag{6}$$

where $\mathcal{N}_k(\cdot)$ yields k-nearest neighboring pocket atom indices of a given ligand atom index and t^* is an index of a ligand's atom-of-interest at a time step t which is sampled from an available set of ligand atoms. This approximation enables the atom addition to be rationally guided by a local pocket environment to enhance the possibility of constructing protein-ligand interactions.

2.1.2 Interaction Condition

In this work, we promote an interaction pattern between a protein and a ligand to guide a 3D generation of a ligand. Recently, Zhang *et al.*[34] built a conditional RNN-based molecular generative model which used interaction fingerprints(IFPs) to incorporate protein-ligand interaction information in SMILES generation. Likewise, we develop a protein atom-based local interaction conditioning strategy. Since k-nearest pocket atoms are captured at each atom addition, the corresponding local interaction condition is set instead of using the whole information at every step. This can mimic how the practitioners *think* to design a ligand molecule for a given target; they aim to build a ligand molecule to form a favorable interaction with the target by reflecting its proximal pocket environment.

Here, we define an *interaction condition* as a set of protein atoms' interaction classes, $\mathbf{I} := \{I_j\}$, which indicates whether the atom can be involved in a particular interaction and its role in the interaction. Protein atoms are categorized into one of 7 classes - anion, cation, hydrogen bond donor and acceptor, aromatic atoms for π-π stacking, hydrophobic atoms, and non-interacting atoms.

In the training phase, where a ground-truth structure of a protein-ligand complex is available, we run the protein-ligand interaction profiler(PLIP)[40] to extract an interaction condition. This software identifies non-covalent interactions between a protein and a ligand by analyzing their binding structure. Meanwhile, reference interaction information might not be available during the sampling phase. In case, we proceed with rule-based interaction typing of protein atoms using predefined criteria for each class. For instance, we render SMARTS patterns[41] to determine hydrogen bond acceptors and donors. More details of the rule-based interaction typing are described in Appendix D. From the practical perspective, one can manually designate a desired interaction condition from one's insight based on the knowledge of the target system.

2.2 Effect of Interaction Conditioning

We first demonstrate the effect of interaction conditioning on the ligand elaboration task. In drug design processes, it is crucial to construct specific protein-ligand interactions, as this can be directly related to the potency and selectivity of a drug. Supposing the "hot spots" of a binding pocket, sites where ligands can readily interact with, are known, then a generative framework should be capable of designing a ligand that can favorably interact with these sites. To establish a reasonable guess of hot spots, here, we extracted interaction patterns from original protein-ligand complexes. From this setting, we show that the local interaction conditioning strategy enables our model to satisfy the particular condition while elaborating on a pharmacophore.

Among the test complexes discussed in section 4.1, which belong to protein families different from those in the training data, we selected complexes that exhibit a wide range of protein-ligand interactions. From the original ligand, we extracted an interaction condition and removed several chains and functional groups to obtain a core structure. Fig. 2(a) illustrates several examples of interaction conditions and initial core structures with the surrounding protein binding pocket. Although interaction conditions are represented at an atom level, we illustrate them as patches for a better visual representation.

To analyze how well the resulting ligands satisfy the given condition, we measured *interaction similarities* between *interaction fingerprints* of the original ligand and the designed ones(for their definitions, see section 4.3). We elaborated each core structure to generate 1,000 ligands and sampled a ligand with the highest interaction similarity. The 3D structures of the original(green) and the sampled ligand(cyan), along with their interaction similarity values, are shown in Fig. 2(b). We confirmed that the sampled ligand showed a large portion of spatial overlaps with the original ligand and was well elaborated while avoiding collision with the pocket. For further analysis, we profiled the interaction between each protein-ligand complex with PLIP software. Fig. 2(c,d) depicts interactions for each ligand in 2D diagrams, where the circles indicate amino acid residues and the dashed lines indicate the interactions. Different colors are

used to distinguish interaction types, where circles with multiple colors correspond to the residues involved in more than one type of interaction. The core structures are highlighted in each ligand structure. We comprehensively investigated each case by contrasting the interaction of the sampled ligand(Fig. 2(d)) with that of the original ligand(Fig. 2(c)) to rationalize the high interaction similarities.

For the first case, which is displayed in the left column of Fig. 2, ligands were elaborated from azabicyclo[2.2.1]heptane to fit in the bone morphogenetic protein 1(BMP1, PDB ID: 6bto). The ligand sampled from our model, DeepICL, successfully constructed hydrogen bonds, π-π stacking, and salt bridges as the given interaction condition(**d-1**). Notably, the result shows that the model can generate the thiophene ring instead of the original benzene ring to construct π-π stacking with TYR68. It implies that the model learns the characteristic of aromatic motifs, which avails to form a π-π stacking. Although the model added the aliphatic carbons near the hydrophobic PHE157, the distance was slightly larger than the threshold to be profiled as a hydrophobic interaction.

Another example is an elaboration of 2-(oxan-3-yloxy)oxane, a skeletal structure of a disaccharide, to fit in the fibroblast growth factor-1(FGF1, PDB ID: 3ud9, the middle column of Fig. 2). Since the original ligand forms multiple hydrogen bonds with neighboring backbone atoms and the polar side chain of ASN9(**c-2**), the interaction condition extracted from the original complex is expected to induce the model to generate hydrogen bond acceptors on the core structure. Indeed, DeepICL successfully designed a ligand of an identical interaction pattern with the original complex by generating phosphate and carboxylate groups instead of a sulfate(**d-2**). Interestingly, LYS119 formed a hydrogen bond with an equatorial hydroxyl group of the generated ligand, where the original ligand possesses an axial methoxy group that is directed away from LYS119.

Lastly, we generated ligands from the benzene inside the pocket of the dihydrofolate reductase(DHFR, PDB ID: 1dis, the right column of Fig. 2). The original complex shows a sophisticated interaction pattern, including multiple salt bridges and π-π stackings(**c-3**). The sampled ligand contains a guanidine moiety similar to the original ligand so that it can form a salt bridge and hydrogen bonds with surrounding amino acids simultaneously(**d-3**). Additionally, the pyruvate-like group of the sampled ligand could form both a hydrogen bond and a salt bridge with ARG31. However, the sampled ligand could not interact with HIS28, which the original ligand forms a salt bridge with.

We further carried out an ablation study to validate the effect of the interaction conditioning. We masked the interaction condition **I** to neglect the information about both the interaction and non-interaction. We compared the distribution of interaction similarities from two different sets of generated ligands. Fig. 2(e) clearly shows that the ligand elaboration guided by the interaction information of the reference ligand achieves higher interaction similarities. In addition, the results show multi-modal distributions, which happened from the construction of particular motifs that can form multiple interactions existing in the original complex. For instance, whether the model constructs a hetero-aromatic ring at the position near PHE30 and ASP26 of DHFR

plays a critical role in the subsequent bimodal distribution(**e-3**). More examples from the ligand elaboration task are provided in Appendix G.

2.3 Binding Stability Analysis of Elaborated Ligands with Short MD Simulation

In section 2.2, we justified that our framework can design ligands that satisfy a specific interaction condition. Then the question follows: do the designed protein-ligand interactions from our framework actually contribute to the protein-ligand binding? We aim to validate the benefit of an interaction conditioning strategy in designing a ligand that can stably bind to a target protein. The ligand might change its binding pose dramatically or even be detached from the target if its interaction is unfavorable. Thus, we postulate that a well-designed ligand would undergo less conformational change while bound to a protein.

We adopt a short MD simulation to measure the conformational change of ligands while considering the solvation and protein flexibility. Albeit the large coverage of long MD simulation, short simulation($\sim$10 ns) has proved its ability to discriminate the correct binding poses in a virtual screening scheme.[43, 44] From the MD trajectories, the root-mean-square deviations(RMSDs) of ligand structures are calculated to evaluate their binding stabilities. Protein backbone structures from each frame are aligned to capture the ligand movement only.

We conducted a comparison between two sets of elaborated ligands, which were generated with and without interaction information, using the same test complexes as in section 2.2. Here, we used rule-based interaction typing, which we described in section 2.1.2, instead of using the reference information to enable diverse interactions. From both sets of elaborated ligands, we randomly sampled ten ligands in which the number of heavy atoms was the same as that of the original ligand. In this manner, we could strictly compare the quality of the ligand elaboration, which undergoes an identical number of atom addition.

We plotted ligand RMSDs during each simulation in Fig. 3. RMSD values of ten sampled ligands were averaged, and the 95% confidence interval was depicted. Blue curves indicate the elaborated ligands guided by rule-based interaction typing, while red curves indicate the ligands elaborated without the information. Grey curves illustrate the ligand RMSDs of original ligands, regarded as baselines of rational binding stabilities. In every case, a set of ligands elaborated with the help of interaction conditioning showed smaller RMSDs than the ones without. This tendency implies that the strategy is capable of designing a ligand with a stable binding pose. Especially, Fig. 3(a) shows a clear difference between the two sets of BMP1, where the ligands generated without the information showed extensive deviations in their binding poses. The ligands that employed the interaction information achieved comparable binding stability with the original, showing that the ligands designed from our strategy can favorably interact with a target.

Fig. 3(b) exhibits a large fluctuation in ligand RMSDs in the FGF1 case. We rationalize this result by recalling the structure of FGF1 in Fig. 2(b), where the original ligand is less surrounded by protein atoms and attached to the target only with multiple hydrogen bonds. Hence, the construction of hydrogen bonds was exceptionally

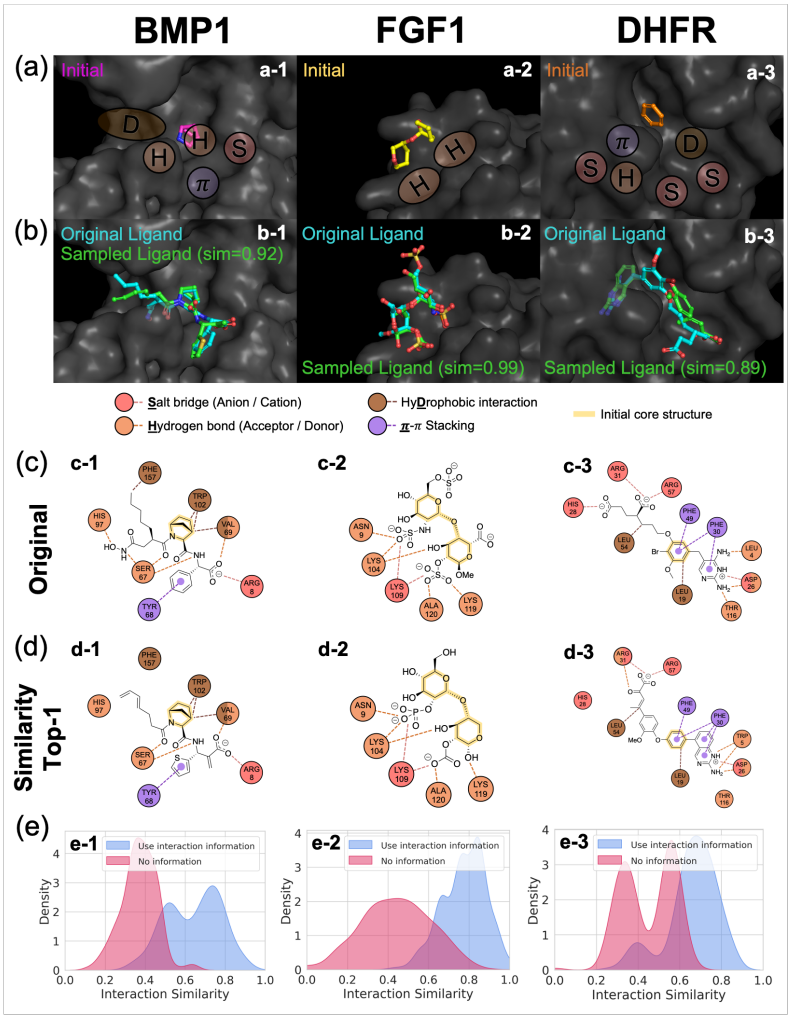

Fig. 2 Illustration of the effect of interaction conditioning. (a) Initial core structures and surfaces of binding pockets marked with interaction conditions, (b) original ligands and sampled ligands with top-1 interaction similarities, (c-d) diagrams of profiled interactions between the pocket and the original ligands or the sampled ligands, and (e) distributions of interaction similarities of ligands generated with and without interaction information of the reference ligand. **Left**: bone morphogenic protein 1(BMP1, PDB ID: 6bto), **middle**: fibroblast growth factor 1(FGF1, PDB ID: 3ud9), **right**: dihydrofolate reductase(DHFR, PDB ID: 1dis). We visualized the protein surface and the ligand structure in 3D via PyMOL software.[42]

crucial to stabilize the ligand binding in this case. Although the overall ligand RMSD values are high, the ligands elaborated with the interaction information exhibit relatively low ligand RMSDs. Additionally, Fig. 3(c) illustrates ligand binding stabilities of DHFR. The ligands of DHFR moved in a much narrower region than in other cases for both sets. The result suggests that the generation of a ligand directly inside a

well-defined binding pocket can be advantageous to achieve fair binding stability, even without the help of an interaction condition.

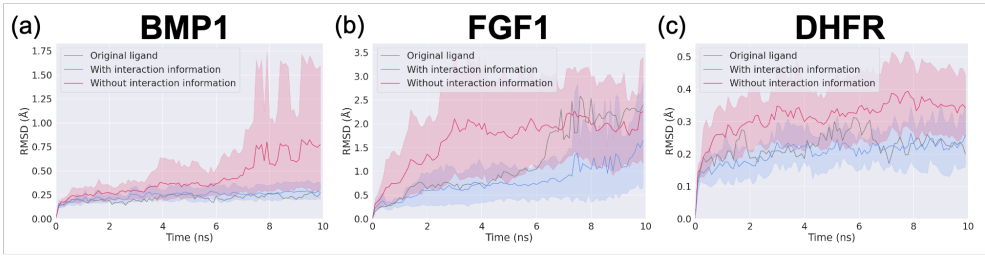

Fig. 3 Plots of ligand RMSDs during short MD simulations to evaluate the ligand binding stability toward target proteins: (a) BMP1, (b) FGF1, and (c) DHFR. Red and blue curves depict the averaged RMSDs of ten sampled ligands of each generated set with 95% confidence intervals. Grey curves show ligand RMSDs of the original ligands.

2.4 Exploring Modeled Intermolecular Geometry in *de novo* Ligand Design

We analyzed the overall structure of a target protein and corresponding generated ligands to reveal whether the intermolecular geometric distributions of each interaction type follow the observed distributions. For each type of protein-ligand interaction, the more stable structural pattern will be more populated, having a specific geometric distribution. Thus, how accurately the generative model reproduces the observed geometric distribution of each interaction type can serve as empirical evidence for the model's understanding of protein-ligand interactions. However, recent deep generative modeling approaches for structure-based ligand design have neglected the evaluation of protein-ligand interactions at a geometric level, instead focusing solely on the ligand's intramolecular geometry. In this section, we validate the performance of our framework by analyzing the interatomic distances between the atom-atom pairs involved in a particular protein-ligand interaction.

Here, we performed a *de novo* ligand design task; entire ligand atoms are generated instead of starting from a core structure. We used the interaction condition obtained from the original ligands. We first generated 100 ligands for each of the 100 test pockets and analyzed the interatomic distances between them. We plotted the distance distributions of each interaction type in Fig. 4. The blue histogram represents the training data, while the red histogram represents the generated data.

Fig. 4(a) shows a distance distribution of hydrophobic interactions, the most common type in the PDB.[45] Our DeepICL effectively captured the observed trend of density decaying as the distance decreases. As the distance of hydrophobic interaction is defined between two hydrophobic carbons, the plot shows that the model avoids spatial hindrance while adding a carbon atom. The distances are mostly populated at around 3.8 Å, much longer than hydrogen bonds or salt bridges, in accordance with the observed tendency.

Next, Fig. 4(b) shows a distribution of hydrogen bonding distance. It is known that heavy atoms involved in a hydrogen bond are separated at a median distance of around 3.0 Å.[45] The distribution from the generated data also shows a peak near 3.0 Å, overlapping with the tendency. Fig. 4(c) shows a distance distribution of salt bridges measured between two charge centers. The generated distribution shows a decaying pattern similar to the training data. Still, our model has room for improvement as the sharp peak near 3.5 Å does not appear on the generated distribution.

Finally, Fig. 4(d) demonstrates a distribution of π-π stacking distances, which is defined between two centers of aromatic rings. The distance distribution of π-π stackings exhibits bimodal behavior, each from the parallel and the perpendicular stackings. Generating a π-π stacking geometry poses a particular challenge for our model, as it requires adding multiple atoms in the same plane to construct an aromatic ring. Moreover, the lack of explicit information about an aromatic ring center during the generation process increases the difficulty of the task; these combinatorial requirements must be fulfilled to form a π-π stacking. Although the generated distribution deviates slightly from the training data, it apparently represents the two distinct modes.

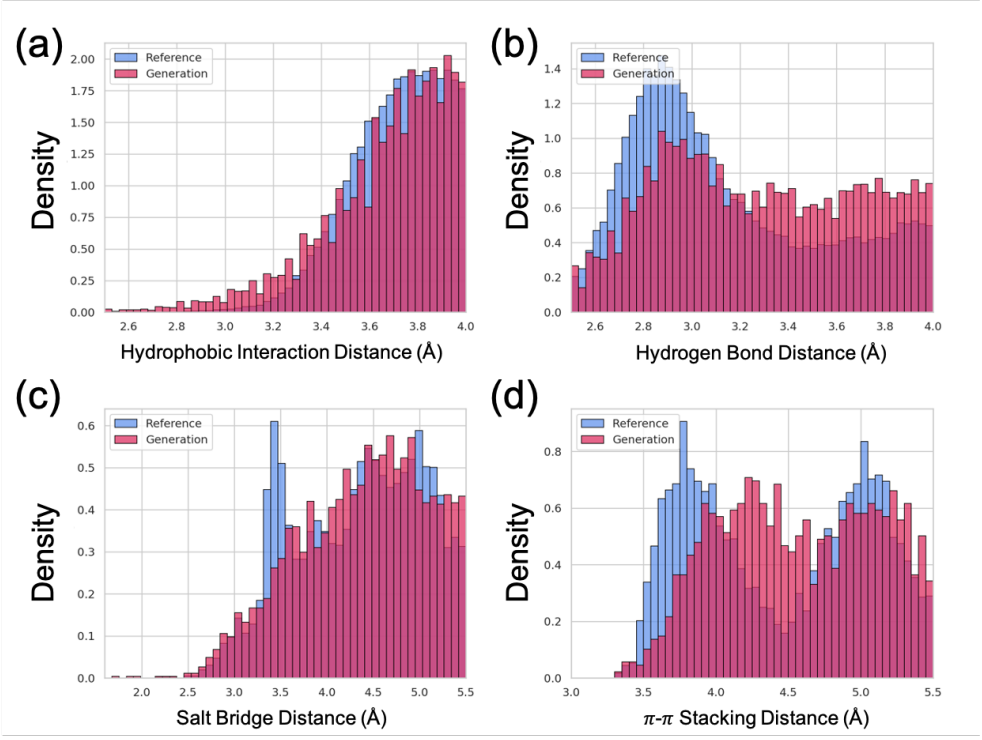

Fig. 4 Distributions of protein-ligand intermolecular distances for (a) hydrophobic interactions, (b) hydrogen bonds, (c) salt bridges, and (d) π-π stackings of the reference training data(blue) and the generated data(red).

2.5 Chemical Diversity and Novelty of Designed Ligands

Achieving high chemical diversity is another essential goal in structure-based ligand design, which can be assessed in terms of the diversity of core structures or scaffolds.[46–48] Although the overall structure of a molecule is novel, if it shares the same scaffold with existing drugs, it may be considered less innovative and, therefore, less likely to be accepted for drug development. Thus, we evaluated the diversity and novelty of generated ligands by enumerating Bemis-Murcko scaffolds[49] within the generated data. We compared them with the training data and the bioactive chemical database, ChEMBL[50, 51].

We first evaluated the chemical diversity in terms of the uniqueness of scaffolds among the 10,000 generated ligands in section 2.4. Out of the 9,679 extracted scaffolds, duplicates were removed to yield 5,667 unique scaffolds or a scaffold uniqueness of 58.6%. For comparison, we also assessed the scaffold uniqueness of the training data. It possesses 5,783 unique scaffolds out of 10,752, resulting in a scaffold uniqueness of 53.8%. Notably, our framework achieves greater diversity than the training data, despite using the interaction condition extracted from the reference ligands. We conducted further analysis on the ten most commonly appearing scaffolds among the generated ligands(Fig. H3(a)). Our model generates the benzene ring most frequently, which is also the most prevalent scaffold in the training data. Our model also generates molecules with scaffolds that are less frequently observed in the training data; diphenylmethane ranks 6th among the generated ligands, whereas it is in 17th place in the training ligands. Thus, our model does not solely follow the observed structural priority of the training data.

Then, we evaluated the structural novelty of the generated ligands at the scaffold level. We examined whether a generated ligand shares its scaffold with any compounds in the ChEMBL database. We used 1,568,892 compounds in the ChEMBL database, of which the molecular weight is under 500. Among the 10,000 generated ligands, 4,975 molecules exhibit novel scaffolds that are not present in the ChEMBL database. About half of the generated ligands comprise novel scaffolds, demonstrating that our model generates unique scaffolds instead of relying on those from the training data. A few examples are illustrated in Fig. H3(b).

2.6 Site-specific Interaction Conditioning

To demonstrate the broader applicability of our approach, we applied our generative framework to ligand design tasks specialized in particular target systems. One of the key advantages of our framework is the possibility to establish the interaction condition manually, guiding the ligand design based on one's insight. Hence, we set the specific interaction condition coinciding with the objective of each task. Here, we chose two well-known tasks where interactions at specific locations play a crucial role.

2.6.1 Selective ligand design of a double-mutated epidermal growth factor receptor(L858R/T790M EGFR)

The first task is to design ligands that can selectively bind to a mutant epidermal growth factor receptor(EGFR) while sparing the wild-type EGFR. A leucine-to-arginine mutation at residue 858(L858R) in a kinase domain of EGFR is one of the most frequently observed causes of non-small-cell lung cancer.[31, 52] Despite the early trials that have developed drugs to target the single-site mutated EGFR, patients often exhibit drug resistance due to the gate-keeper mutation of T790M.[53–55] Thus, developing a drug to selectively inhibit an L858R/T790M double-mutated EGFR while sparing the wild-type EGFR to prevent an off-target effect is a clinically important problem. Nevertheless, the task is exceptionally challenging due to the identical nature of the rest of the protein sequence, except for the two mutated residues, which leads to a remarkably conserved set of target structures.

We conceived from the idea that if a ligand strongly interacts with the mutated residues, the ligand will favorably bind to the mutated pocket more than the wild-type. Primarily, we retrieved complex structures of a wild-type and a double-mutated EGFR reported by Sogabe *et al.*(PDB ID: 3w2s and 3w2r, respectively)[53] Two complexes share the same ligand and have a similar pocket structure. Then, we underwent *de novo* ligand generation inside the double-mutated pocket(3w2r). To selectively target the two mutated residues, we manually designated the possible interaction types of the atoms of MET790 and ARG858 while sparing other atoms. We used this site-specific conditioning strategy, which informs the model about the explicit position and type of interactions, to gain selectivity.

After generating 1,000 ligands inside the double-mutated EGFR, we also placed them in the aligned pocket of the wild-type EGFR. We then carried out a local optimization followed by energy scoring via SMINA[56], a docking software based on AutoDock Vina[57], for each pocket. We selected a well-designed ligand with a visual inspection among the ones predicted to have a selectivity toward the mutated EGFR by forming desired interactions and provided its structure in Fig. 5(a). The ligand is forming hydrophobic interactions with a side chain of MET790 while forming a hydrogen bond with a backbone of ARG858. Although not all of the generated ligands exhibit strong interactions with the mutated residues, the utilization of site-specific conditioning allows for the identification of a desirable molecule by generating just 1,000 molecules. Additional statistics regarding predicted binding affinities of generated ligands on the wild-type and mutated EGFR are provided in Appendix I.

2.6.2 Designing hinge binders of Rho-associated protein kinase 1(ROCK1)

The next is to design hinge binders of Rho-associated protein kinase 1(ROCK1). The ATP binding site of ROCK1 contains a hinge region that recognizes the adenine moiety through multiple hydrogen bonds.[58] Hence, ROCK1 inhibitors are often designed to target the hinge region, thus called hinge binders.[59, 60] We retrieved the ROCK1 structure reported from Li *et al.*(PDB ID: 3v8s)[59]. In the complex, the ligand forms hydrogen bonds with the carbonyl oxygen of GLU154 and the amide nitrogen of

MET156 within the hinge region. To design hinge binders that can form hydrogen bonds with both residues, we conditioned the carbonyl oxygen and the amide nitrogen as hydrogen bond acceptor and hydrogen bond donor, respectively.

We generated 100 ligands inside a pocket of ROCK1, and their structure ensemble is shown in Fig. 5(b). The yellow dashed lines indicate hydrogen bonds formed between hinge residues and the original ligand. Our framework successfully generated polar functional groups containing nitrogen or oxygen atoms near the hinge region, as highlighted with a white dashed circle. This implies that our generative framework is capable of designing hinge binders.

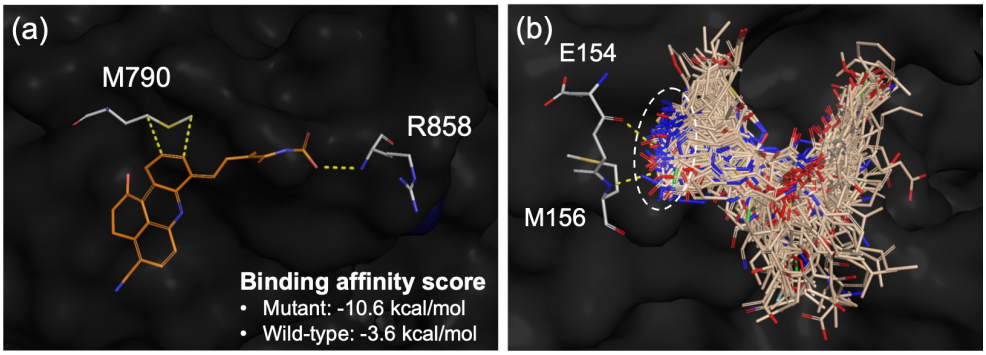

Fig. 5 (a) An example of a well-designed ligand that is expected to possess a selectivity toward double-mutated EGFR. Its interactions with mutated residues are shown in yellow dashed lines. (b) Designed ligands inside a ROCK1, where the hinge binding region is specified with a white dashed circle. Each pocket surface is colored grey.

3 Conclusions

In this work, we demonstrated the significance of incorporating prior knowledge for guiding a deep generative model to learn more informative features from the raw data. As a practical example, we employed the concept of protein-ligand interaction to generalize a deep generative model for structure-based ligand design. Conceiving from the idea that the patternization of protein-ligand binding structures can render a sophisticated contact surface more understandable, we focused on the local geometric patterns of non-covalent interactions. In this manner, we developed an interaction-focused generative framework that utilizes a local interaction conditioning strategy. Our main model, DeepICL, is trained to reconstruct the ligand structure with the interaction pattern profiled from the original protein-ligand complex. The results showed that the modeled probability distributions of atom-atom distances between a pocket and a ligand follow the observed distributions of each interaction type in experimental structures. Also, our model could reproduce the desired interaction condition with a different elaboration from the original core structure. Furthermore, ligands generated from the pocket-extracted interaction condition gained higher binding stabilities than

the ones generated without the interaction information. While chemical diversity and novelty can be a pitfall of a conditional molecular generative modeling scheme, ligands designed from the specific interaction condition included highly diverse and innovative scaffold structures.

Our proposed framework can enjoy further practical advantages when specific residues of a target binding site are known to play a crucial role in the pharmaceutical perspective. We demonstrated two such tasks - designing selective ligands toward a mutant pocket and designing hinge binders of a kinase. We introduced the site-specific conditioning scheme by manually setting up a condition to target the residue-of-interest. This scheme increases the chance of sampling a ligand satisfying the desired chemical properties by forming the interaction as expected. We, therefore, suggest that employing the proper prior knowledge can be a benign navigator for deep generative modeling in a variety of fields.

4 Materials and Methods

4.1 Training and Test data sets

We used the 2020 version of the PDBbind general set[35], consisting of X-ray crystal structures of 19,443 complexes. We split the data via protein sequence similarity with a 60% cutoff, which is calculated and clustered by CD-HIT software[61]. As a result of data processing, we used 11,284 structures for training our model and 2,109 structures for validation. We filtered out the rest of the data to leave 100 test complexes that satisfy the following three conditions: ligand's Tanimoto similarity is less than 0.6 with all the ligands in the training set, every data corresponds to distinct protein families, and the number of protein heavy-atoms is less than 300.

4.2 DeepICL

As briefly introduced in section 2.1, DeepICL is a deep generative model that designs a ligand suitable for a specific protein pocket, taking a given interaction condition into account. DeepICL utilizes information about the 3D structure of the binding pocket and its corresponding interaction pattern to produce a 3D binding structure of a newly designed ligand. We adopt a variational auto-encoder(VAE) architecture[62] consisting of two main modules, an encoder, and a decoder, as illustrated in Fig 6. The encoder module encodes the structure of a given protein-ligand complex, $\mathbf{L}$ and $\mathbf{P}$, into a latent vector z that follows a standard normal distribution. The decoder module then sequentially generates a ligand structure in an atom-wise manner from the latent vector z. The interaction condition is integrated into the latent vector z for placing a suitable ligand atom to form a desired interaction with the target. The encoder and decoder modules share the same embedding layers, which are composed of multiple layers of $E(3)$-invariant interaction network that propagates the messages between a protein and a ligand. More details about the model architecture can be found in Appendix A.

DeepICL employs two additional dummy atoms that only hold positional information, the center-of-mass and the atom-of-interest, to assist the ligand design process as

in the work of G-SchNet[63]. The center-of-mass of the original ligand roughly determines a global position of a ligand to be generated. The atom-of-interest confines a 3D space where the next ligand atom would be placed; only its neighboring protein atoms are concerned in the prediction of the next atom type and its position in each step. Consequently, DeepICL can learn the relationship between a local pocket environment and a structural preference of a ligand to fulfill the given interaction condition, leveraging the robustness of DeepICL in ligand design tasks for any protein. The above two dummy atoms are treated as individual ligand atoms in the training and sampling process. Then, they are removed when finalizing the ligand structure.

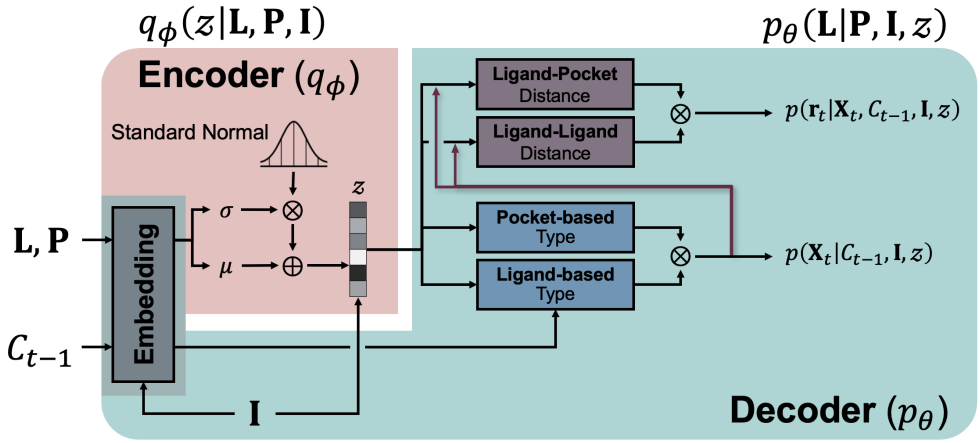

Fig. 6 Illustration of the model architecture of DeepICL. The encoder module(q_ϕ) is trained to encode a whole protein-ligand complex($\mathbf{L}, \mathbf{P}$) and corresponding interaction condition, $\mathbf{I}$, into a latent vector z which follows a prior distribution. The decoder module(p_θ) is trained to reconstruct the ligand structure from the given protein pocket and an interaction condition with an autoregressive process. Note that the decoder of the figure describes a single atom addition step, where a type and a position of t-th ligand atom is determined from the protein-ligand complex of step $t-1$.

4.2.1 Training DeepICL

The training objective of DeepICL is to predict the next atom, L_t, from a previous complex state, C_{t-1}, and the atom-of-interest, L_{t^*}. Since there is no canonical order in an atom-wise designing process, we randomly traverse a trajectory of placing atoms of a ligand in each training epoch. The next atom is always picked from the atoms covalently bonded to the atom-of-interest in the original ligand. DeepICL then learns the likelihood of a type of the next atom and its position.

For each step of an atom placement, DeepICL is trained to predict the next atom type, $\mathbf{X}_t$, and its position, $\mathbf{r}_t$, based on the previous complex state, C_{t-1}, the latent vector, z, and interaction condition, $\mathbf{I}$. DeepICL embeds the information of C_{t-1} into two sets of hidden vectors for the ligand and protein, $\mathbf{h}^l_{t-1} := \{h^l_{t-1,i}\}$ and $\mathbf{h}^p_{t-1} := \{h^p_{t-1,j}\}$, respectively. Again, i and j denote the atom index of a ligand

and a protein, respectively. We use two models for atom type prediction; one model, θ_l, predicts the likelihood from already placed ligand atoms, and the other, θ_p, predicts the likelihood from k-nearest neighboring protein pocket atoms. We minimize the Kullback-Leibler(KL) divergence between the predicted atom type distribution, p_t^{type}, and the ground-truth atom type distribution, q_t^{type}, which is a one-hot encoding of $\mathbf{X}_t$. Formally, we minimize the following type loss:

$$\ell_t^{type} = \underbrace{\frac{1}{t-1}\sum_{i=1}^{t-1} KL(p_{t,i}^{type}||q_t^{type})}_{\text{ligand-based type}} + \underbrace{\frac{1}{k}\sum_{j\in\mathcal{N}_k(t^*)} KL(p_{t,j}^{type}||q_t^{type})}_{\text{pocket-based type}}, \tag{7}$$

where $p_{t,i}^{type} = p_{\theta_l}(\mathbf{X}_t|h_{t-1,i}^l, \mathbf{I}, z)$ and $p_{t,j}^{type} = p_{\theta_p}(\mathbf{X}_t|h_{t-1,j}^p, \mathbf{I}, z)$. We also train the distance prediction model by minimizing the KL divergence loss for the distance distribution over the already placed ligand atoms and the proximal protein atoms,

$$\ell_t^{dist} = \underbrace{\frac{1}{t-1}\sum_{i=1}^{t-1} KL(p_{t,i}^{dist}||q_{t,i}^{dist})}_{\text{ligand-ligand distance}} + \underbrace{\frac{1}{k}\sum_{j\in\mathcal{N}_k(t^*)} KL(p_{t,j}^{dist}||q_{t,j}^{dist})}_{\text{ligand-pocket distance}}, \tag{8}$$

where $p_{t,i}^{dist} = p_{\theta_l}(d_{t,i}^{ll}|\mathbf{X}_\mathrm{t}, h_{t-1,i}^l, \mathbf{I}, z)$ and $p_{t,j}^{dist} = p_{\theta_p}(d_{t,j}^{lp}|\mathbf{X}_\mathrm{t}, h_{t-1,j}^p, \mathbf{I}, z)$. Here, q^{dist} is a Gaussian expansion of a ground-truth distance, whose detailed definition can be found in Appendix A.1.

We note that the training losses on pocket atoms incorporate only k-nearest neighboring pocket atoms that are close to a ligand atom-of-interest so that the type of a newly added ligand atom is determined solely based on the surrounding local chemical environment. The atom type loss, ℓ_t^{type}, and distance loss, ℓ_t^{dist}, are minimized simultaneously to train the model to reconstruct a ligand structure. Thus, the reconstruction loss can be written as follows:

$$\ell_{recon} = \sum_t \left[\ell_t^{type} + \ell_t^{dist}\right]. \tag{9}$$

The VAE architecture of DeepICL also requires a minimization of the following additional loss known as the regularization loss:

$$\ell_{reg} = KL(q_\phi(z|\mathbf{L}, \mathbf{P}, \mathbf{I})||p(z)), \tag{10}$$

where $p(z)$ is a standard normal distribution.

4.2.2 Designing ligands with DeepICL

DeepICL produces a ligand in three stages, 1) initialization, 2) sequential addition of atoms, and 3) termination of the process.

In the initialization stage, two additional dummy atoms, the center-of-mass and atom-of-interest, are combined into C_0 to guide the overall sampling process. The center-of-mass remains unmoved throughout the entire sampling process, whereas the atom-of-interest moves its position to one of the already placed ligand atoms in each addition step. Although one can manually select an arbitrary point as a starting point, in this work, we choose the center-of-mass of a reference ligand for convenience. In order to increase the diversity of generated ligands and decrease the dependency on the center-of-mass of the original ligand, we introduce a roto-translational Gaussian noise during a sampling phase. For the ligand elaboration task, where the generation starts from a pre-defined core structure, the initial structure is noised without the change in internal coordinates. More details about the Gaussian noise can be found in Appendix C.1.2.

In the second stage, DeepICL designs a ligand by sequentially adding new atoms. Based on the initialized state, DeepICL predicts the following atom type and its position autoregressively. Each likelihood of type and position is composed of the likelihoods obtained from the ligand and protein sides, respectively. Thus, we integrate them as follows:

$$\log p(\mathbf{X}_t|C_{t-1},\mathbf{I}) \propto \sum_{i=1}^{t-1} \log p_{t,i}^{type} + \lambda \sum_{j\in\mathcal{N}_k(t^*)} \log p_{t,j}^{type}, \tag{11}$$

$$\log p(\mathbf{r}_t|\mathbf{X}_t,C_{t-1},\mathbf{I}) \propto \sum_{i=1}^{t-1} \log p_{t,i}^{dist} + \lambda \sum_{j\in\mathcal{N}_k(t^*)} \log p_{t,j}^{dist}. \tag{12}$$

Here, λ is a pocket coefficient that tunes a contribution of a pocket in determining the next atom. The value of λ is determined depending on how far an upcoming ligand atom is apart from pocket atoms. We tempt to decrease the contribution of the pocket if the ligand atom is placed away from the pocket since the protein-ligand interaction occurs at a short range. The detailed approach is explained in Appendix C.2.

If DeepICL predicts the STOP sign for the next atom type, the current atom-of-interest t^* is marked as *unavailable* and no longer selected as an atom-of-interest. Then, the next atom-of-interest, $(t+1)^*$, is sampled from a currently available set of ligand atoms and used for the next step. The sampling process terminates when every placed ligand atom is marked as unavailable. As a result, DeepICL yields a ligand structure designed inside a target pocket.

4.3 Interaction Fingerprint and Interaction Similarity

We define *interaction fingerprint* and *interaction similarity* to evaluate how well the sampled ligands satisfy the given interaction condition. The interaction fingerprint describes the pattern of a protein's interaction with a specific ligand at an atom level. Each protein atom falls into one of four classes depending on the type of interaction it is involved in - hydrogen bond, hydrophobic interaction, salt bridge, and π-π stacking - to represent a one-hot vector. Unlike the interaction condition we introduced in section 2.1.2, the non-interaction class is neglected to build an interaction fingerprint. We then

concatenate all the atom-wise one-hot vectors to obtain an interaction fingerprint as a single vector while preserving the atomic order in the protein. This ensures that the resulting interaction fingerprints can be compared across different ligands bound to a single target.

Next, we define interaction similarity as a cosine similarity between the interaction fingerprints of two ligands for a single target. To measure how well the ligand satisfies the given condition, we use the interaction fingerprint obtained from the original ligand as a reference to compare with those of the generated ligands. High interaction similarity indicates that the sampled ligand possesses an interaction pattern similar to that of the original ligand. Hence, it follows the given interaction condition. With this interaction similarity metric, we can quantitatively evaluate the performance of our local interaction conditioning strategy to control the ligand design process. We note that a low interaction similarity does not necessarily imply a low binding affinity for the sampled ligand. Still, the ligand may have the potential to form a better binding with a target by adopting a different interaction pattern compared to the original one.

Author Contributions

W.Z. conceptualized the work and developed the model. W.Z. trained the model and carried out the experiments. W.Z. and H.K. designed the experiments and analyzed the results. All authors contributed to the manuscript writing. The whole work is supervised by W.Y.K.

Funding

This work was supported by Basic Science Research Programs through the National Research Foundation of Korea(NRF), grant funded by the Ministry of Science and ICT(NRF-2018R1A5A1025208, NRF-2023R1A2C2004376).

Competing Interests

There are no competing interests to declare.

Data Availability

The code for processing data from the PDBbind dataset and the information on data splitting is available at Github: https://github.com/ACE-KAIST/DeepICL.git.

Code Availability

The code for our whole framework, including the training and sampling of DeepICL and evaluating generated ligands, is available at Github: https://github.com/ACE-KAIST/DeepICL.git.

Appendix A Model architectures of DeepICL

A.1 Node and edge features

The input node features of ligand and protein atoms, $\mathbf{X}_i^l \in \mathbb{R}^{F^l}$ and $\mathbf{X}_j^p \in \mathbb{R}^{F^p}$, are used as described in the Table A1. Each of the features is represented as a one-hot vector of a corresponding category. Only an atom type is used for a ligand atom feature, while more informative features are used for a protein atom feature by concatenating every one-hot vector. The resulting feature dimension of a ligand and a protein atom are $F^l = 9$ and $F^p = 51$, respectively. Initial node features of a ligand and a protein are then embedded into a hidden dimension F^h with a single linear layer.

Ligand atom feature, $\mathbf{X}_i^l$	Available list
Atom type	C, N, O, F, P, S, Cl, Br, stop (one-hot)
Protein atom feature, $\mathbf{X}_j^p$	Available list
Atom type	C, N, O, F, P, S, Cl, Br, *else* (one-hot)
Atom degree	0, 1, 2, 3, 4, *else* (one-hot)
Hybridization	s, sp, sp^2, sp^3, sp^3d, sp^3d^2, *else* (one-hot)
Formal charge	-2, -1, 0, 1, 2, 3, *else* (one-hot)
Amino acid type	G, A, V, L, I, C, M, F, Y, W, P, S, T, Q, N, D, E, H, R, K, *else* (one-hot)
Aromaticity	0 or 1

Table A1 The table of used node features of ligand and protein atoms, and each available item list for a one-hot vector

We used the Gaussian expansion of a distance between the i-th and j-th nodes as an edge feature, $\mathbf{e}_{ij} := \{e_n(d_{ij})\}_{n=1}^N$. Each Gaussian distribution is located at each center value of a distance bin, where a total distance is divided into N bins with a spacing of $\Delta\mu$. The smoothness of the expansion is controlled by γ. Formally,

$$e_n(d_{ij}) = \frac{\mathrm{e}^{-\gamma(d_{ij}-n\Delta\mu)^2}}{\sum_{n'=0}^{N-1}\mathrm{e}^{-\gamma(d_{ij}-n'\Delta\mu)^2}}. \tag{A1}$$

Specific values of each hyper-parameter will be summarized in Appendix E.

A.2 E(3)-Invariant interaction network architecture

While updating the node features of a protein and a ligand, we devise an $E(3)$-invariant interaction network that can propagate the inter- and intra-molecular messages between a protein and a ligand. A single layer of the network consists of three steps: inter- and intra-molecular message calculation and aggregation(equation A2 to A5), gate coefficient calculation(equation A6), and node feature update(equation A7). Formally,

$$\mathbf{m}_{k\to i} = \phi_{intra}(h_k^l,\ h_i^l,\ \mathbf{e}_{ki}),\ \mathbf{m}_{l\to j} = \phi_{intra}(h_l^p,\ h_j^p,\ \mathbf{e}_{lj}), \tag{A2}$$

$$\boldsymbol{\mu}_{l\to i} = \phi_{inter}(h_l^p,\ h_i^l,\ \mathbf{e}_{li}),\ \boldsymbol{\mu}_{k\to j} = \phi_{inter}(h_k^l,\ h_j^p,\ \mathbf{e}_{kj}), \tag{A3}$$

$$\mathbf{m}_i = \sum_k \mathbf{m}_{k\to i},\ \mathbf{m}_j = \sum_l \mathbf{m}_{l\to j}, \tag{A4}$$

$$\boldsymbol{\mu}_i = \sum_l \boldsymbol{\mu}_{l\to i},\ \boldsymbol{\mu}_j = \sum_k \boldsymbol{\mu}_{k\to j}, \tag{A5}$$

$$z_i = \psi_{gate}(\mathbf{m}_i,\ \boldsymbol{\mu}_i),\ z_j = \psi_{gate}(\mathbf{m}_j,\ \boldsymbol{\mu}_j), \tag{A6}$$

$$h_i^{l\prime} = \chi^l\big(h_i^l,\ z_i \cdot \mathbf{m}_i + (1 - z_i) \cdot \boldsymbol{\mu}_i\big),\ h_j^{p\prime} = \chi^p\big(h_j^p,\ z_j \cdot \mathbf{m}_j + (1 - z_j) \cdot \boldsymbol{\mu}_j\big). \tag{A7}$$

For convenience, subscript i, k denotes ligand atom indices, and j, l denotes protein atom indices. ϕ_{intra}, ϕ_{inter}, and ψ_{gate} are learnable models shared on both a ligand and a protein. ϕ_{intra} and ϕ_{inter} are multi-layer perceptrons (MLPs) activated by sigmoid linear units (SiLUs), while ψ_{gate} is a single linear layer followed by a sigmoid function. Intramolecular message $\mathbf{m}$ and intermolecular message $\boldsymbol{\mu}$ are linearly interpolated by a gate coefficient z, then used to update the current node state. The atom feature updates for a ligand and a protein are done by χ^l and χ^p, respectively, which are gated recurrent units (GRUs)[64].

A.3 Atom type and position prediction model

After node feature updates through multiple layers of $E(3)$-invariant interaction networks, the features are joined with a latent vector, z. After that, only protein node features are additionally joined with an interaction condition, $\mathbf{I}$. The resulting features then undergo fully-connected layers into dimensions of type and distance distributions individually. We used SiLU as an activation function, where final prediction outputs go through a softmax function for normalization.

Appendix B Training details

B.1 KL divergence loss annealing

A decoder that generates an output in an autoregressive fashion can be susceptible to the KL-vanishing problem, which might cause undesired model behaviors such as mode collapse[65, 66]. To mitigate this KL-vanishing problem, one can employ an annealing schedule for the KL divergence term. Various strategies for the annealing schedules have been proposed[65–68]. In our study, we adopted the simplest monotonic KL annealing, gradually increasing the weight of the KL divergence term up to a predefined value during training. Formally, the weight $\beta(t)$ is scheduled at t-th epoch as:

$$\beta(t) = \beta_f + (\beta_i - \beta_f) \cdot \eta^t, \tag{B8}$$

where β_i is the initial weight, β_f is the final wight, and η is the weight annealing factor. Their specific values used in this work can be found in Appendix E. Thus, our final loss function can be written as follows:

$$\ell_{total}(t) = \ell_{recon} + \beta(t) \cdot \ell_{reg}. \tag{B9}$$

Appendix C Sampling details

C.1 Controlling the randomness

In conditional deep generative models, ensuring diversity and novelty in the sampled outputs is an important concern. One simple way to increase them is by introducing some randomness into the sampling process. In our study, we control the randomness of the proposed ligands by employing a temperature factor and a roto-translational noise to the sampling process using DeepICL.

C.1.1 Temperature factor

As in the work of G-SchNet[63], we use an additional temperature factor that allows for controlling the randomness of the generation. We reformulate equation 11 and equation 12, which is to define the likelihood of the next atom type and its position in the sampling process, to introduce temperature factors τ_{type} and τ_{pos}:

$$\tilde{p}(\mathbf{X}_t|C_{t-1},\mathbf{I},z) = \frac{1}{a}\exp\left(\frac{\log p(\mathbf{X}_t|C_{t-1},\mathbf{I},z)}{\tau_{type}}\right), \tag{C10}$$

$$\tilde{p}(\mathbf{r}_t|\mathbf{X}_t,C_{t-1},\mathbf{I},z) = \frac{1}{b}\exp\left(\frac{\log p(\mathbf{r}_t|\mathbf{X}_t,C_{t-1},\mathbf{I},z)}{\tau_{pos}}\right), \tag{C11}$$

where a and b are normalization constants. Increasing τ_{type} and τ_{pos} will smoothen the predicted probability distributions, adding more randomness to the next atom prediction, whereas small values will produce sharper distributions, leading to less randomness.

C.1.2 Adding roto-translational noise

We adopt an additional method that introduces a random noise to the initial state of the sampling process. We use a different random noise depending on the ligand design task being considered; in the case of *de novo* ligand design, a translational noise is added to the given center-of-mass, whereas a roto-translational noise is added to the initial ligand core structure for a ligand elaboration task. A translational noise is simply a vector sampled from a Gaussian distribution in 3D space centered at the origin with variance σ_t, which is a hyper-parameter to control the randomness of the translational noise. Then, the center-of-mass is moved by the vector.

For applying extra rotational noise, the rotation axis is sampled from a uniform distribution, and its rotational angle is sampled from a Gaussian distribution centered at the origin with variance σ_r, which is a hyper-parameter to control the randomness of the rotational noise. By increasing σ_t and σ_r, the initial state will be scattered with a greater variance, appending more randomness to the ligand sampling process.

C.2 Determining pocket coefficient

We introduce a pocket coefficient λ in equation 11 and 12 of section 4.2.2, which controls the weight of a pocket-side prediction during the atom sampling. When the space where the next atom is going to be added is located far from the pocket, the dependence of ligand atom types and positions on the pocket diminishes. To reflect this, we allow the distance between a ligand and a pocket determines the pocket coefficient. We average the distances between the ligand atom-of-interest and its k-nearest pocket atoms as described in equation C12. Then, the pocket coefficient λ is determined by a function defined in equation C13. The function decays as the average distance, $\bar{d}$, increases in the region where $\bar{d}$ is larger than 2.5 Å. We set its decaying coefficient so that $\lambda \simeq 1$ at $\bar{d} = 5.0$ Å.

$$\bar{d}(t^*) = \frac{1}{k} \sum_{j \in \mathcal{N}_k(t^*)} d_{t^*,j}, \tag{C12}$$

$$\lambda = \begin{cases} 10 & \text{if } 0 \leq \bar{d} \leq 2.5 \text{ Å} \\ 10 \cdot e^{-0.91(\bar{d}-2.5)} & \text{otherwise.} \end{cases} \tag{C13}$$

Appendix D Rule-based interaction typing details

We used a rule-based interaction typing for a given protein pocket in the case when no ligand information is available. Atoms involved in salt bridge anions and cations, hydrogen bond acceptors, and donors are selected by using SMARTS descriptors summarized in Table D2. Since only particular motifs that appear on amino acid chains are known to have cations or anions, we set SMARTS patterns to fully cover them, even their resonance structures. Halogen atoms or carbons, which are surrounded only by carbon or hydrogen atoms, are classified as hydrophobic atoms, and atoms within aromatic rings are classified as aromatic atoms.

Anion	[O;$([OH0-,OH][CX3](=[OX1])),$([OX1]=[CX3]([OH0-,OH]))]
Cation (Lysine)	[N;$([NX3H2,NX4H3+;!$(NC=[!#6]);!$(NC#[!#6])][#6])]
Cation (Arginine)	[#7;$([NH2X3][CH0X3](=[NH2X3+,NHX2+0]) [NHX3]),$([NH2X3+,NHX2+0]=[CH0X3]([NH2X3])[NHX3])]
Cation (Histidine)	[#7;$([$([#7X3H+,#7X2H0+0]:[#6X3H]:[#7X3H]), $([#7X3H])]1:[#6X3H]:[$([#7X3H+,#7X2H0+0]: [#6X3H]:[#7X3H]),$([#7X3H])]:[#6X3H]:[#6X3]1), $([$([#7X3H+,#7X2H0+0]:[#6X3H]:[#7X3H]), $([#7X3H])]1:[#6X3H]:[$([#7X3H+,#7X2H0+0]: [#6X3H]:[#7X3H]),$([#7X3H])]:[#6X3]: [#6X3H]1)]
Hydrogen bond acceptor	[$([!#6;+0);!$([F,Cl,Br,I]); !$([o,s,nX3]);!$([Nv5,Pv5,Sv4,Sv6])]
Hydrogen bond donor	[!#6;!H0]

Table D2 SMARTS descriptors for anion, cation, hydrogen bond acceptor, and donor

Appendix E Hyper-parameter settings

We summarized hyper-parameter settings used during the model training and sampling in Table E3. We used `ReduceLROnPlateau` module implemented in PyTorch[69], which reduces a learning rate if there is no improvement in the validation loss for a fixed number of training epochs until the learning rate reaches a minimum threshold value.

Hyper-parameters	Comments	Values
k	number of nearest neighbors	8
F^h	hidden dimension	128
num_interaction_layers	number of interaction layers	6
num_fc_layers	number of FC layers	3
N	number of bins for Gaussian expansion	25, 300
$\Delta\mu$	spacing between distance bins	0.4 Å, 0.05 Å
γ	smoothness of Gaussian expansion	10, 50
τ_{type}, τ_{pos}	temperature factor	0.1, 0.1
σ_t, σ_r	variance of roto-translational noise	0.2 Å, 2°
β_i, β_f	KL-divergence annealing initial and final	0.0, 1.0
η	KL-divergence annealing factor	0.2
lr	initial learning rate	10^{-3}
lr_min	minimum learning rate	10^{-6}
lr_decay	lr decaying factor	0.8
lr_tol	lr decay tolerance epochs	4

Table E3 Descriptions of the notations used in this chapter

Appendix F Molecular dynamics simulation details

In this work, topology and parameter files for the ligands were generated using the GAFF-2.11 force field[70] via the OpenMM Toolkit[71]. Protein-ligand structures were solvated in a cubic box using TIP3P water models[72], extending 10 Å from the protein to provide padding. The systems were neutralized by adding Na^+ and Cl^- ions. Periodic boundary conditions were applied to the systems in the NPT ensemble using the Langevin thermostat[73]. To simulate the interactions, we employed the Amber FF14SB force field[74]. Equilibration and production runs were performed using the OpenMM toolkit. Initially, the systems underwent energy minimization followed by 1 ns of equilibration. Production runs were conducted at 303.5 K and 1 bar, using a 2 fs integration time step. Each protein-ligand complex underwent 10 ns production runs, initiated with the same initial structure but differently randomized initial velocities. To assess the stability of ligand binding to the receptor protein, we calculated the ligand RMSD by aligning the protein structures and measuring the RMSD of the ligand's translation and rotation during the simulation. The proteins were superimposed based on their heavy atom coordinates using MDtraj software[75]. For each protein-ligand complex, the ligand RMSD was averaged over the entire 10 ns simulation time, yielding the averaged ligand RMSD value and its variance.

Appendix G More examples of an interaction conditioned ligand elaboration task

Here, we provide a few more examples of interaction-conditioned ligand elaboration results, continued from section 2.2. Five more data points were selected from the test set, which is composed of human sialidase(PDB ID: 2f0z), beta-lactamase Mox-1(PDB ID: 4wbg), ubiquitin ligase(PDB ID: 6do4), abscisic acid receptor(PDB ID: 6nwc), and MAP kinase-activated protein kinase 2(PDB ID: 3fpm). With DeepICL, we sampled 100 molecules starting from the core structures of the original ligands by using the interaction condition extracted from the original complexes. In Fig. G1, the first column depicts the binding conformation of the original ligands, and their core structures are highlighted in orange. The second column shows the structures of generated ligands that achieved the highest interaction similarities, and the values are also denoted. Finally, the last column shows shifts in the distribution of interaction similarities as we use the interaction information rather than using a blank condition.

Notably, similarity distributions have dramatically diverged in the case of human sialidase(PDB ID: 2f0z). However, one might concern about the chemical diversity of the generated ligands since the structure of the top-1 ligand is so much similar to the original ligand(the first row of Fig. G1). Thus, we further provide more molecular structures of ligands that were generated to target the human sialidase in Fig. G2. While sharing the common core structure, which is oxane, generated ligands are highly diverse.

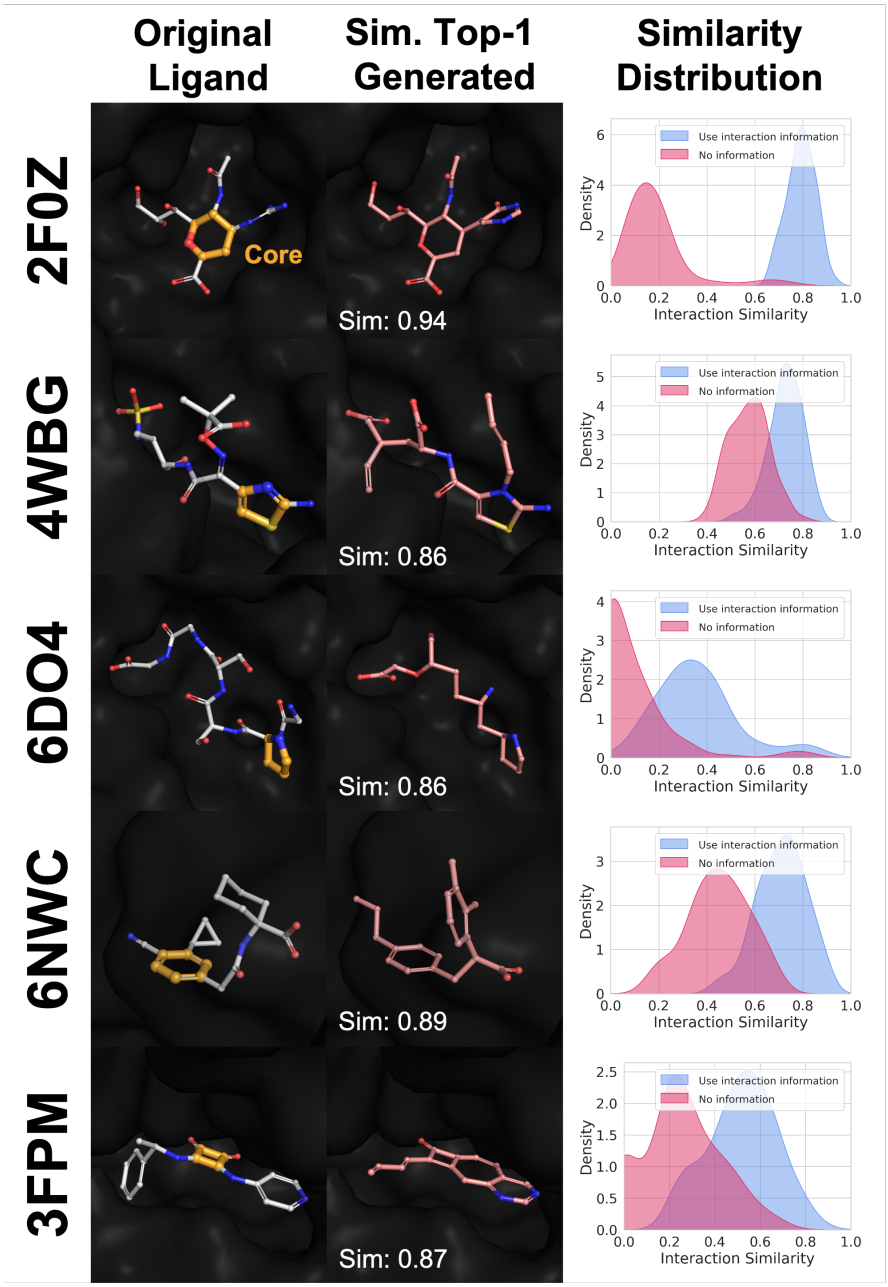

Fig. G1 More examples of an interaction-conditioned ligand elaboration task. Left column: structures of original ligands(white) and their core structures(orange), where the surface of surrounding pockets are shown in grey. Middle column: structures of the most similar ligand molecules elaborated from each core structure in terms of interaction similarity, and their values are denoted in the bottom left corner. Right column: distribution of interaction similarities of generated ligands, where the set of ligands that employs interaction condition clearly shows higher similarities in every case.

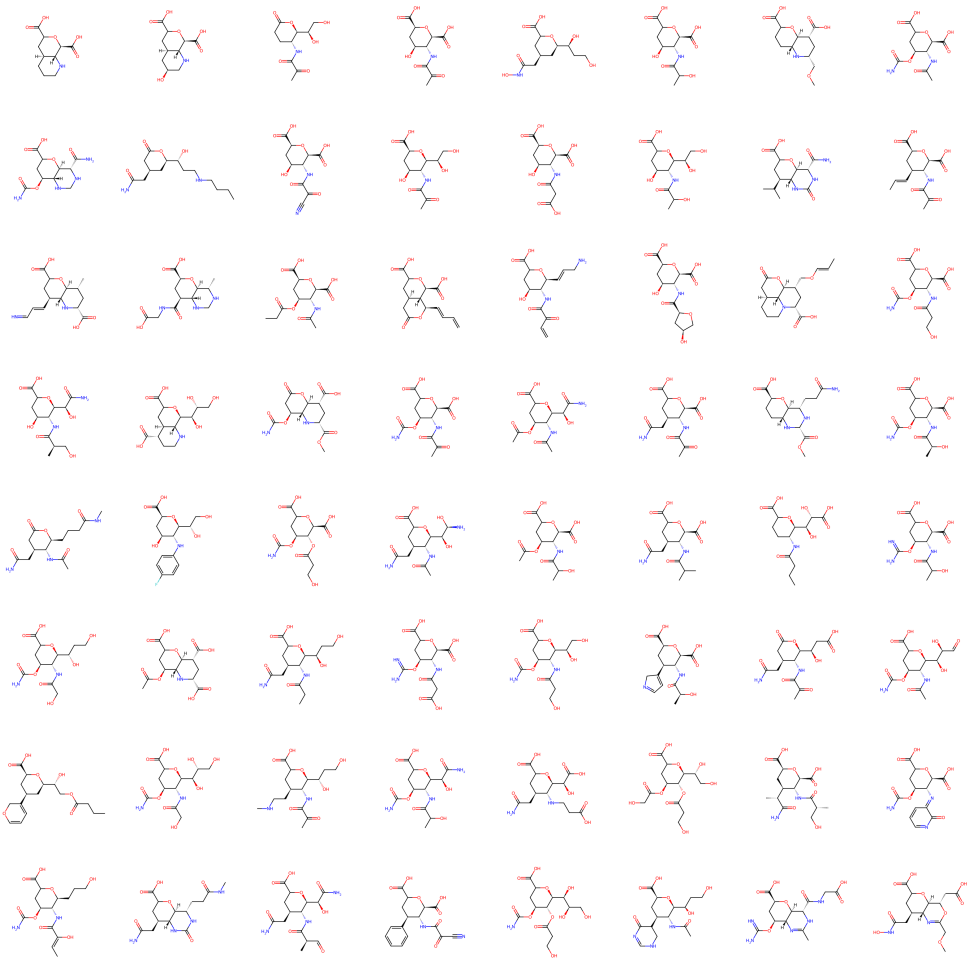

Fig. G2 Additional 64 ligands elaborated from the core structure of the original ligand binding to human sialidase(PDB ID: 2f0z) are shown. The structures are diverse, although they achieve high interaction similarities.

Appendix H Scaffold diversity and novelty of ligands provided by DeepICL

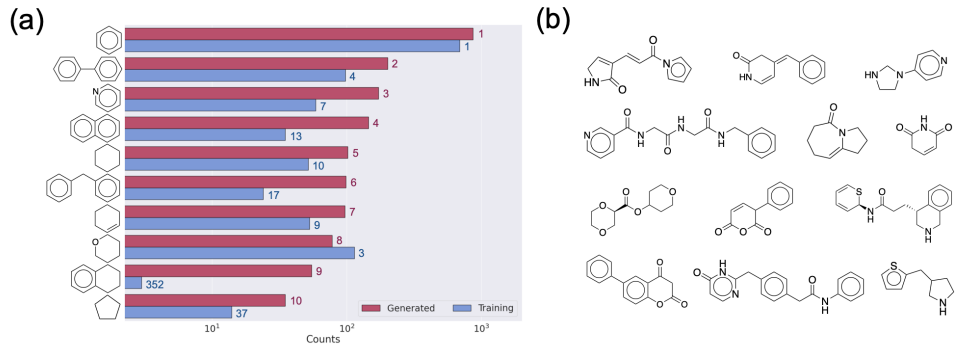

Fig. H3 (a) Bar graphs illustrating the frequency of the 10 most commonly appearing scaffolds in the generated set. For each scaffold, the graph displays the log-scaled frequencies in the generated set(red) and the training set(blue). The labeled numbers on each bar indicate rankings of their appearing frequencies in each set, respectively. (b) Few examples of novel scaffolds in generated ligands.

Appendix I Predicted binding affinities for selectively designed ligands of EGFR

Continued from section 2.6.1, the predicted binding affinities toward the wild-type and the mutated EGFR are illustrated in Fig. I4. Points above the solid line have lower scores on the mutated pocket than on the wild-type pocket. As a lower score indicates a stronger binding affinity, this tendency implies that many of the generated ligands possess the selectivity toward the mutated EGFR. Red points above the dashed line are predicted to have 100 times lower inhibitory concentration on the mutated EGFR than on the wild-type. Since the energy lower by 1.36 kcal/mol corresponds to a 10 times lower inhibitory concentration, we set the difference cutoff as 2.72 kcal/mol.

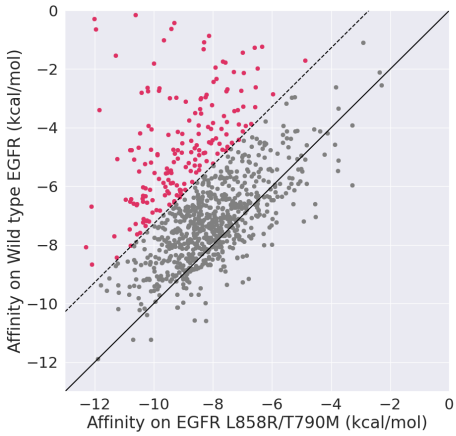

Fig. I4 The predicted binding affinity of locally optimized ligands toward both a wild-type EGFR and a double-mutated EGFR. Red points show 2.72 kcal/mol lower binding affinity(100 times less inhibitory concentration) for the mutated EGFR.

References

[1] Muralidhar, N., Islam, M.R., Marwah, M., Karpatne, A., Ramakrishnan, N.: Incorporating prior domain knowledge into deep neural networks. In: 2018 IEEE International Conference on Big Data (big Data), pp. 36–45 (2018). IEEE

[2] Dash, T., Chitlangia, S., Ahuja, A., Srinivasan, A.: A review of some techniques for inclusion of domain-knowledge into deep neural networks. Scientific Reports **12**(1), 1040 (2022)

[3] Yu, Y., Wang, C., Fu, Q., Kou, R., Huang, F., Yang, B., Yang, T., Gao, M.: Techniques and challenges of image segmentation: A review. Electronics **12**(5), 1199 (2023)

[4] Culos, A., Tsai, A.S., Stanley, N., Becker, M., Ghaemi, M.S., McIlwain, D.R., Fallahzadeh, R., Tanada, A., Nassar, H., Espinosa, C., *et al.*: Integration of mechanistic immunological knowledge into a machine learning pipeline improves predictions. Nature machine intelligence **2**(10), 619–628 (2020)

[5] Kirkpatrick, J., McMorrow, B., Turban, D.H., Gaunt, A.L., Spencer, J.S., Matthews, A.G., Obika, A., Thiry, L., Fortunato, M., Pfau, D., *et al.*: Pushing the frontiers of density functionals by solving the fractional electron problem. Science **374**(6573), 1385–1389 (2021)

[6] Li, L., Hoyer, S., Pederson, R., Sun, R., Cubuk, E.D., Riley, P., Burke, K., *et al.*: Kohn-sham equations as regularizer: Building prior knowledge into machine-learned physics. Physical review letters **126**(3), 036401 (2021)

[7] Guo, R., Xue, S., Hu, J., Sari, H., Mingels, C., Zeimpekis, K., Prenosil, G., Wang, Y., Zhang, Y., Viscione, M., *et al.*: Using domain knowledge for robust and generalizable deep learning-based ct-free pet attenuation and scatter correction. Nature Communications **13**(1), 5882 (2022)

[8] Qiao, C., Li, D., Liu, Y., Zhang, S., Liu, K., Liu, C., Guo, Y., Jiang, T., Fang, C., Li, N., *et al.*: Rationalized deep learning super-resolution microscopy for sustained live imaging of rapid subcellular processes. Nature biotechnology **41**(3), 367–377 (2023)

[9] Cornelio, C., Dash, S., Austel, V., Josephson, T.R., Goncalves, J., Clarkson, K.L., Megiddo, N., El Khadir, B., Horesh, L.: Combining data and theory for derivable scientific discovery with ai-descartes. Nature Communications **14**(1), 1777 (2023)

[10] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., *et al.*: Highly accurate protein structure prediction with alphafold. Nature **596**(7873), 583–589 (2021)

[11] Anderson, A.C.: The process of structure-based drug design. Chemistry & biology

10(9), 787–797 (2003)

[12] Keserű, G.M., Makara, G.M.: Hit discovery and hit-to-lead approaches. Drug discovery today **11**(15-16), 741–748 (2006)

[13] Jhoti, H., Leach, A.R.: Structure-based Drug Discovery vol. 1. Springer, ??? (2007)

[14] Lim, J., Hwang, S.-Y., Moon, S., Kim, S., Kim, W.Y.: Scaffold-based molecular design with a graph generative model. Chemical science **11**(4), 1153–1164 (2020)

[15] Li, Y., Hu, J., Wang, Y., Zhou, J., Zhang, L., Liu, Z.: Deepscaffold: a comprehensive tool for scaffold-based de novo drug discovery using deep learning. Journal of chemical information and modeling **60**(1), 77–91 (2019)

[16] Imrie, F., Hadfield, T.E., Bradley, A.R., Deane, C.M.: Deep generative design with 3d pharmacophoric constraints. Chemical science **12**(43), 14577–14589 (2021)

[17] Green, H., Koes, D.R., Durrant, J.D.: Deepfrag: a deep convolutional neural network for fragment-based lead optimization. Chemical Science **12**(23), 8036–8047 (2021)

[18] Zhavoronkov, A., Ivanenkov, Y.A., Aliper, A., Veselov, M.S., Aladinskiy, V.A., Aladinskaya, A.V., Terentiev, V.A., Polykovskiy, D.A., Kuznetsov, M.D., Asadulaev, A., *et al.*: Deep learning enables rapid identification of potent ddr1 kinase inhibitors. Nature biotechnology **37**(9), 1038–1040 (2019)

[19] Chan, L., Kumar, R., Verdonk, M., Poelking, C.: A multilevel generative framework with hierarchical self-contrasting for bias control and transparency in structure-based ligand design. Nature Machine Intelligence, 1–13 (2022)

[20] Guha, R.: On exploring structure–activity relationships. In silico models for drug discovery, 81–94 (2013)

[21] Lavecchia, A.: Deep learning in drug discovery: opportunities, challenges and future prospects. Drug discovery today **24**(10), 2017–2032 (2019)

[22] Martinelli, D.: Generative machine learning for de novo drug discovery: A systematic review. Computers in Biology and Medicine, 105403 (2022)

[23] Seo, S., Lim, J., Kim, W.Y.: Molecular generative model via retrosynthetically prepared chemical building block assembly. Advanced Science, 2206674 (2023)

[24] Baillif, B., Cole, J., McCabe, P., Bender, A.: Deep generative models for 3d molecular structure. Current Opinion in Structural Biology **80**, 102566 (2023)

[25] Isert, C., Atz, K., Schneider, G.: Structure-based drug design with geometric deep learning. Current Opinion in Structural Biology **79**, 102548 (2023)

[26] Ragoza, M., Masuda, T., Koes, D.R.: Generating 3d molecules conditional on receptor binding sites with deep generative models. Chemical science **13**(9), 2701–2713 (2022)

[27] Luo, S., Guan, J., Ma, J., Peng, J.: A 3d generative model for structure-based drug design. Advances in Neural Information Processing Systems **34**, 6229–6239 (2021)

[28] Peng, X., Luo, S., Guan, J., Xie, Q., Peng, J., Ma, J.: Pocket2mol: Efficient molecular sampling based on 3d protein pockets. In: International Conference on Machine Learning, pp. 17644–17655 (2022). PMLR

[29] Wang, L., Bai, R., Shi, X., Zhang, W., Cui, Y., Wang, X., Wang, C., Chang, H., Zhang, Y., Zhou, J., *et al.*: A pocket-based 3d molecule generative model fueled by experimental electron density. Scientific reports **12**(1), 15100 (2022)

[30] Moon, S., Zhung, W., Yang, S., Lim, J., Kim, W.Y.: Pignet: a physics-informed deep learning model toward generalized drug–target interaction predictions. Chemical Science **13**(13), 3661–3673 (2022)

[31] Elsocht, M., Giron, P., Maes, L., Versées, W., Gutierrez, G.J., De Grève, J., Ballet, S.: Structure–activity relationship (sar) study of spautin-1 to entail the discovery of novel nek4 inhibitors. International Journal of Molecular Sciences **22**(2), 635 (2021)

[32] Wangtrakuldee, P., Byrd, M.S., Campos, C.G., Henderson, M.W., Zhang, Z., Clare, M., Masoudi, A., Myler, P.J., Horn, J.R., Cotter, P.A., *et al.*: Discovery of inhibitors of burkholderia pseudomallei methionine aminopeptidase with antibacterial activity. ACS medicinal chemistry letters **4**(8), 699–703 (2013)

[33] Güzel, Ö., Innocenti, A., Scozzafava, A., Salman, A., Supuran, C.T.: Carbonic anhydrase inhibitors. phenacetyl-, pyridylacetyl-and thienylacetyl-substituted aromatic sulfonamides act as potent and selective isoform vii inhibitors. Bioorganic & medicinal chemistry letters **19**(12), 3170–3173 (2009)

[34] Zhang, J., Chen, H.: De novo molecule design using molecular generative models constrained by ligand–protein interactions. Journal of Chemical Information and Modeling **62**(14), 3291–3306 (2022)

[35] Liu, Z., Su, M., Han, L., Liu, J., Yang, Q., Li, Y., Wang, R.: Forging the basis for developing protein–ligand interaction scoring functions. Accounts of chemical research **50**(2), 302–309 (2017)

[36] Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E.: The protein data bank. Nucleic acids research **28**(1), 235–242 (2000)

[37] Burley, S.K., Bhikadiya, C., Bi, C., Bittrich, S., Chao, H., Chen, L., Craig, P.A., Crichlow, G.V., Dalenberg, K., Duarte, J.M., *et al.*: Rcsb protein data bank (rcsb.org): delivery of experimentally-determined pdb structures alongside one million computed structure models of proteins from artificial intelligence/machine learning. Nucleic Acids Research **51**(D1), 488–508 (2023)

[38] Freitas, R.F., Schapira, M.: A systematic analysis of atomic protein–ligand interactions in the pdb. Medchemcomm **8**(10), 1970–1981 (2017)

[39] Gebauer, N.W., Gastegger, M., Hessmann, S.S., Müller, K.-R., Schütt, K.T.: Inverse design of 3d molecular structures with conditional generative neural networks. Nature communications **13**(1), 973 (2022)

[40] Salentin, S., Schreiber, S., Haupt, V.J., Adasme, M.F., Schroeder, M.: Plip: fully automated protein–ligand interaction profiler. Nucleic acids research **43**(W1), 443–447 (2015)

[41] Wunderlich, R.E., Wenisch, T.F., Falsafi, B., Hoe, J.C.: Smarts: Accelerating microarchitecture simulation via rigorous statistical sampling. In: Proceedings of the 30th Annual International Symposium on Computer Architecture, pp. 84–97 (2003)

[42] DeLano, W.L., *et al.*: Pymol: An open-source molecular graphics tool. CCP4 Newsl. Protein Crystallogr **40**(1), 82–92 (2002)

[43] Radom, F., Plückthun, A., Paci, E.: Assessment of ab initio models of protein complexes by molecular dynamics. PLoS computational biology **14**(6), 1006182 (2018)

[44] Guterres, H., Im, W.: Improving protein-ligand docking results with high-throughput molecular dynamics simulations. Journal of Chemical Information and Modeling **60**(4), 2189–2198 (2020)

[45] Ferreira, L.G., Dos Santos, R.N., Oliva, G., Andricopulo, A.D.: Molecular docking and structure-based drug design strategies. Molecules **20**(7), 13384–13421 (2015)

[46] Dimova, D., Bajorath, J.: Assessing scaffold diversity of kinase inhibitors using alternative scaffold concepts and estimating the scaffold hopping potential for different kinases. Molecules **22**(5), 730 (2017)

[47] Wills, T.J., Lipkus, A.H.: Structural approach to assessing the innovativeness of new drugs finds accelerating rate of innovation. ACS Medicinal Chemistry Letters **11**(11), 2114–2119 (2020)

[48] Polykovskiy, D., Zhebrak, A., Sanchez-Lengeling, B., Golovanov, S., Tatanov, O., Belyaev, S., Kurbanov, R., Artamonov, A., Aladinskiy, V., Veselov, M., *et al.*: Molecular sets (moses): a benchmarking platform for molecular generation

models. Frontiers in pharmacology **11**, 565644 (2020)

[49] Bemis, G.W., Murcko, M.A.: The properties of known drugs. 1. molecular frameworks. Journal of medicinal chemistry **39**(15), 2887–2893 (1996)

[50] Gaulton, A., Bellis, L.J., Bento, A.P., Chambers, J., Davies, M., Hersey, A., Light, Y., McGlinchey, S., Michalovich, D., Al-Lazikani, B., *et al.*: Chembl: a large-scale bioactivity database for drug discovery. Nucleic acids research **40**(D1), 1100–1107 (2012)

[51] Gaulton, A., Hersey, A., Nowotka, M., Bento, A.P., Chambers, J., Mendez, D., Mutowo, P., Atkinson, F., Bellis, L.J., Cibrián-Uhalte, E., *et al.*: The chembl database in 2017. Nucleic acids research **45**(D1), 945–954 (2017)

[52] Zhang, T., Qu, R., Chan, S., Lai, M., Tong, L., Feng, F., Chen, H., Song, T., Song, P., Bai, G., *et al.*: Discovery of a novel third-generation egfr inhibitor and identification of a potential combination strategy to overcome resistance. Molecular cancer **19**, 1–15 (2020)

[53] Sogabe, S., Kawakita, Y., Igaki, S., Iwata, H., Miki, H., Cary, D.R., Takagi, T., Takagi, S., Ohta, Y., Ishikawa, T.: Structure-based approach for the discovery of pyrrolo [3, 2-d] pyrimidine-based egfr t790m/l858r mutant inhibitors. ACS medicinal chemistry letters **4**(2), 201–205 (2013)

[54] Yan, X.-E., Zhu, S.-J., Liang, L., Zhao, P., Choi, H.G., Yun, C.-H.: Structural basis of mutant-selectivity and drug-resistance related to co-1686. Oncotarget **8**(32), 53508 (2017)

[55] Bollinger, M.K., Agnew, A.S., Mascara, G.P.: Osimertinib: A third-generation tyrosine kinase inhibitor for treatment of epidermal growth factor receptor-mutated non-small cell lung cancer with the acquired thr790met mutation. Journal of Oncology Pharmacy Practice **24**(5), 379–388 (2018)

[56] Koes, D.R., Baumgartner, M.P., Camacho, C.J.: Lessons learned in empirical scoring with smina from the csar 2011 benchmarking exercise. Journal of chemical information and modeling **53**(8), 1893–1904 (2013)

[57] Trott, O., Olson, A.J.: Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. Journal of computational chemistry **31**(2), 455–461 (2010)

[58] Urich, R., Wishart, G., Kiczun, M., Richters, A., Tidten-Luksch, N., Rauh, D., Sherborne, B., Wyatt, P.G., Brenk, R.: De novo design of protein kinase inhibitors by in silico identification of hinge region-binding fragments. ACS chemical biology **8**(5), 1044–1052 (2013)

[59] Li, R., Martin, M.P., Liu, Y., Wang, B., Patel, R.A., Zhu, J.-Y., Sun, N., Pireddu,

R., Lawrence, N.J., Li, J., *et al.*: Fragment-based and structure-guided discovery and optimization of rho kinase inhibitors. Journal of medicinal chemistry **55**(5), 2474–2478 (2012)

[60] Beroza, P., Crawford, J.J., Ganichkin, O., Gendelev, L., Harris, S.F., Klein, R., Miu, A., Steinbacher, S., Klingler, F.-M., Lemmen, C.: Chemical space docking enables large-scale structure-based virtual screening to discover rock1 kinase inhibitors. Nature Communications **13**(1), 6447 (2022)

[61] Fu, L., Niu, B., Zhu, Z., Wu, S., Li, W.: Cd-hit: accelerated for clustering the next-generation sequencing data. Bioinformatics **28**(23), 3150–3152 (2012)

[62] Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)

[63] Gebauer, N., Gastegger, M., Schütt, K.: Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. Advances in neural information processing systems **32** (2019)

[64] Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. Preprint at arXiv:1412.3555 (2014) arXiv:1412.3555

[65] Bowman, S.R., Vilnis, L., Vinyals, O., Dai, A.M., Jozefowicz, R., Bengio, S.: Generating sentences from a continuous space. arXiv preprint arXiv:1511.06349 (2015)

[66] Sønderby, C.K., Raiko, T., Maaløe, L., Sønderby, S.K., Winther, O.: Ladder variational autoencoders. Advances in neural information processing systems **29** (2016)

[67] Kim, Y., Wiseman, S., Miller, A., Sontag, D., Rush, A.: Semi-amortized variational autoencoders. In: International Conference on Machine Learning, pp. 2678–2687 (2018). PMLR

[68] Fu, H., Li, C., Liu, X., Gao, J., Celikyilmaz, A., Carin, L.: Cyclical annealing schedule: A simple approach to mitigating kl vanishing. arXiv preprint arXiv:1903.10145 (2019)

[69] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)

[70] Wang, J., Wolf, R.M., Caldwell, J.W., Kollman, P.A., Case, D.A.: Development and testing of a general amber force field. Journal of computational chemistry **25**(9), 1157–1174 (2004)

[71] Eastman, P., Swails, J., Chodera, J.D., McGibbon, R.T., Zhao, Y., Beauchamp,

K.A., Wang, L.-P., Simmonett, A.C., Harrigan, M.P., Stern, C.D., *et al.*: Openmm 7: Rapid development of high performance algorithms for molecular dynamics. PLoS computational biology **13**(7), 1005659 (2017)

[72] Jorgensen, W.L., Chandrasekhar, J., Madura, J.D., Impey, R.W., Klein, M.L.: Comparison of simple potential functions for simulating liquid water. The Journal of chemical physics **79**(2), 926–935 (1983)

[73] Allen, M.P., Tildesley, D.J.: Computer Simulation of Liquids. Oxford university press, ??? (2017)

[74] Maier, J.A., Martinez, C., Kasavajhala, K., Wickstrom, L., Hauser, K.E., Simmerling, C.: ff14sb: improving the accuracy of protein side chain and backbone parameters from ff99sb. Journal of chemical theory and computation **11**(8), 3696–3713 (2015)

[75] McGibbon, R.T., Beauchamp, K.A., Harrigan, M.P., Klein, C., Swails, J.M., Hernández, C.X., Schwantes, C.R., Wang, L.-P., Lane, T.J., Pande, V.S.: Mdtraj: a modern open library for the analysis of molecular dynamics trajectories. Biophysical journal **109**(8), 1528–1532 (2015)