Decipher Fundamental Atomic Interactions to Unify Generative Molecular Docking and Design

Xingang Peng[1,2], Ruihan Guo[3], Yan Xu[4], Jiaqi Guan[5], Yinjun Jia[6,7], Yanwen Huang[8], Muhan Zhang[1], Jian Peng[3], Jiayu Sun[9], Chuanhui Han[4,9*], Zihua Wang[10,11*], Jianzhu Ma[12,13,*]

[1]Institute for Artificial Intelligence, Peking University, Beijing 100871, China.

[2]School of Intelligence Science and Technology, Peking University, Beijing 100871, China.

[3]Helixon US Inc, USA.

[4]State Key Laboratory of Molecular Oncology, Frontiers Science Center for Cancer Integrative Omics, Beijing Key Laboratory of Carcinogenesis and Translational Research, Department of Lymphoma, Peking University Cancer Hospital and Institute, Peking University International Cancer Institute, Peking University Health Science Center, Beijing 100191, China.

[5]University of Illinois Urbana-Champaign, IL 61820, USA.

[6]School of Life Sciences, IDG/McGovern Institute for Brain Research, Tsinghua University, Beijing 100084, China.

[7]Tsinghua-Peking Center for Life Sciences, Beijing 100084, China.

[8]State Key Laboratory of Natural and Biomimetic Drugs, School of Pharmaceutical Sciences, Peking University, Beijing 100191, China.

[9]School of Basic Medical Sciences, Peking University Health Science Center, Beijing 100191, China.

[10]Fujian Provincial Key Laboratory of Brain Aging and Neurodegenerative Diseases, School of Basic Medical Sciences, Fujian Medical University, Fujian 350108, China.

[11]CAS Key Laboratory for Biomedical Effects of Nanomaterials and Nanosafety, CAS Center for Excellence in Nanoscience, National Center for Nanoscience and Technology, Beijing 100190, China.

[12]Department of Electronic Engineering, Tsinghua University, Beijing 100084, China.

[13]Institute for AI Industry Research, Tsinghua University, Beijing 100084, China.

*Corresponding author. Email: chuanhui.han@bjmu.edu.cn, wangzh@iccas.ac.cn, majianzhu@tsinghua.edu.cn

Atomic interactions are fundamental to molecular structures and functions. We constructed PocketXMol, an all-atom AI model, to learn these interactions for general pocket-interacting molecular generative tasks. PocketXMol unified distinct molecular tasks under a single computational framework without requiring fine-tuning. It was evaluated on 11 typical tasks, covering docking and design of small molecules and peptides, and compared against 49 baselines using 45 metrics. PocketXMol outperformed state-of-the-art methods in 9 tasks and was competitive in the remaining ones. We successfully adopted PocketXMol to design novel small molecules to inhibit caspase-9 with efficacy comparable to commercial pan-caspase inhibitors. We also adopted PocketXMol to design PD-L1-binding peptides and demonstrated its success rate 50,000 times higher than random library screening. Three peptides underwent rigorous cellular validation, exhibiting effective binding to the cell membrane. Their potential for molecular probing and therapy was further confirmed through injections in tumor mouse models and the ligand inhibition assay.

AI is revolutionizing the world by empowering both individuals and society with unprecedented capabilities in communication, analysis, and scientific discovery (*1–8*). The predominant feature of these new AI models lies in their capacity to accomplish a wide array of related tasks within a cohesive framework to capture the interactions among the basic units of the system. Such a unified learning scheme enables the models with powerful generalization ability to be extended to scenarios they have never encountered during training (*2, 9*). In the realm of structural molecular biology,

efforts have also been made toward constructing such unified AI models. AlphaFold 3 (*10*) and RosettaFold All-Atom (RFAA) (*11*) have been upgraded from structure prediction of proteins to all biomolecule types by learning the interactions among basic entities including atoms, amino acids, and nucleic acids. ESM 3 built a generative language model to learn the reliance of the sequence, structures, and functions among amino acid residues of proteins for a diversity of protein modeling tasks (*12*). Small molecule generation models have also explored the feasibility of learning similar molecular design tasks (such as structure-based drug design and lead optimization) within single models (*13–15*). Although they have made progress in combining molecular types or tasks, they were still restricted to a limited scope without exploiting the underlying transferability across tasks and data. From the perspective that the atomic interactions among molecules are universally applicable, it is natural to consider a unified model to integrate all molecular generative tasks related to atomic interactions and treat different molecular types exclusively at the atom level, to benefit from the shared fundamental principles and data.

However, there exist challenges to developing such unified models. Firstly, it is non-trivial to represent diverse and highly flexible molecular generative tasks in a precise and unified manner. In the domain of natural language processing, tasks are represented as text prompts, enabling their unification within language models (*9*). Although the text prompts can also be used to describe molecular tasks in the molecular models (*16–18*), they do not capture the essence of the tasks, i.e., they cannot properly express the relationship among tasks. Secondly, current generative frameworks are not suitable to accommodate diverse molecular generative tasks simultaneously. While the diffusion-based generative frameworks (*19, 20*) have demonstrated robust generative power and successful application to individual molecular generative tasks (*10, 21–24*), integrating all these tasks within a single diffusion model is challenging due to the distinct noise prior distributions and denoising trajectories that the generative process relies on for different tasks. Lastly, an intrinsic molecular representation is essential for learning the shared underlying principles among molecules. Currently, small molecules are represented as atoms with valence bonds, while proteins and peptides are generally represented as chains of amino acid residues. This difference in representations could potentially hinder the learning of unified knowledge across different molecules.

To tackle these challenges, we present an atom-level generative foundation model, named PocketXMol, to unify generative tasks related to the interactions among protein pockets, small

molecules, and peptides. Firstly, various molecular generative tasks, ranging from docking to design, are uniformly represented through a novel task prompt mechanism. Compared to the solutions relying on the text prompts (*17, 18, 25*), our task prompt directly defines the inputs and outputs of tasks at the atom level to encode the internal relationship among tasks. Users can precisely control the model to execute specific generative tasks through the prompt and, more importantly, can leverage the prompt to execute generative tasks that are difficult to express through text. Secondly, to perform generative tasks, PocketXMol takes the idea of denoising from the diffusion models but discards the complicated dependency between data and noise of the diffusion generative framework. Specifically, PocketXMol is built as a simple but effective universal denoiser, which facilitates accommodating distinct task noise distributions into a unified molecular noise space and enables flexibly integrating prior knowledge during the generation process. With the unified task prompt and the novel generative framework, all the relevant tasks and structural data are trained together, thereby enabling the model to learn the shared knowledge. PocketXMol is trained by restoring the 3D molecules from the noisy ones and does not require additional fine-tuning for individual tasks. Lastly, PocketXMol adopts the pure atom-level representation for all molecular types to learn the underlying atomic interactions. Notably, PocketXMol overcomes the conventional need to explicitly model amino acids for proteins and peptides, marking a pioneering achievement in generating amino acids through exclusive atom-level modeling. This results in a completely distinct treatment of peptides with all existing models and exploits the most fundamental transferability between small molecules and proteins.

We assembled a dataset of two million 3D structures, including small molecules, protein-peptide complexes, and protein-small-molecule complexes, and removed redundancy with test sets to train PocketXMol. We extensively evaluated its performance on 11 generative tasks (protein-small-molecule docking, protein-peptide docking, molecular conformation generation, structure-based drug design, 3D molecule generation, fragment linking, PROTAC drug design, fragment growing, molecule optimization, *de novo* peptide design, and peptide inverse folding) in non-redundant test sets and compared them against 49 different baseline methods using 45 different evaluation metrics. PocketXMol surpassed state-of-the-art methods on 9 out of 11 tasks and was competitive in the remaining ones. We also demonstrated the adeptness of PocketXMol in assimilating prior knowledge across a range of generative tasks, including protein-small-molecule docking, protein-

peptide docking, and shape-conditioned molecule generation. Next, we adopted PocketXMol to design small-molecule inhibitors targeting caspase-9 and successfully validated one molecule that could suppress the cleavage of caspase-3 and its downstream target PARP1 while treatment with a BCL-2 inhibitor ABT-737. In the second round of optimization for this molecule, four out of the fifteen molecules could specifically inhibit caspase-9 activation with efficacy comparable to the commercially available pan-caspase inhibitor with strong specificity. Finally, we leveraged PocketXMol to design peptides targeting PD-L1 and validated the efficacy of the designed binders. We validated 382 designed peptides, among which 76 peptides achieved a binding affinity of 10^{-7}M and 15 peptides reached 10^{-8}M. The success rate was 50,000 times higher than that of random library screening as a baseline. We selected three peptides to perform more rigorous validation at the cellular level, each showing effective binding to the cell membrane and distinct binding contrasts between PD-L1 positive and negative cells. Their effectiveness as molecular targeting probes was further proved by tail vein injections in lung tumor mouse models, and the potential for therapy was validated through the ligand inhibition assay.

Overview of PocketXMol

As a unified model, the denoiser abstracts any input molecule as a set of atoms and chemical bonds to facilitate learning fundamental atom-level interactions. All generative tasks are directly defined through a task prompt composed of groups of binary indicators to control the generation of the molecules. The backbone of PocketXMol is a versatile molecular denoiser built on geometric neural networks and can produce clean molecules from the input noisy one based on the designated task prompt (Fig. 1A). The generation process is repeatedly to add and remove noise on the molecule, as is inspired by the denoising mechanism of diffusion-based generative models (*19, 20*). However, PocketXMol has major differences from diffusion models (fig. S1). To incorporate distinct noise prior distributions of different tasks, our model eliminates the input requirement of the time step or noise scale but instead automatically determines the noise type and scale from the noisy molecule with the assistance of the task prompt. Besides, our model is designed to directly predict the clean molecule rather than the previous molecule at the generation trajectory as the diffusion models do. These modifications break the denoising Markov chain imposed by the diffusion framework and

thus lead to a flexible noise-adding process. Another extension is the detailed injection of noise down to the atom level to enable different molecular components to be injected with different noise, which has brought tremendous adaptability to support more generative tasks.

To specify a generative task, users can set the task prompt to precisely convey intentions. A task prompt is a group of binary variables indicating whether each atom or bond of the input molecule should be fixed or requires to be generated for a particular generative task (Fig. 1C). This task representation accurately captures the requirements of any molecular generative task in a more fundamental way than other representations such as natural language descriptions (*16, 18, 26*). As shown in Fig. 1B, with the unified molecule and task representation, all generative tasks were jointly trained on a comprehensive dataset covering all relevant structure data (redundancy with test sets was removed). Once trained, PocketXMol can be immediately applied to individual tasks without the requirement of fine-tuning, which is different from many other pre-training models (*5–7, 27, 28*). The training scheme of PocketXMol facilitates learning the shared commonality among tasks, as exampled by the distributions of the noisy molecules with different task-specific molecular noise (Fig. 1D). For instance, fragment linking, which generates partial molecules from the other parts, is located between the clean molecule and structure-based drug design (SBDD) that generates the complete molecule in the embedding space; Peptide design, which requires generating side-chain atoms and docking backbone atoms, is located between docking with Gaussian noise and the molecule design tasks (SBDD and fragment linking). Docking with flexible noise, which perturbs coordinates using random rigid translation and rotation of torsion angles, deviates from all other tasks where the Gaussian noise is applied to the coordinates (Fig. 1D). More details are outlined in the supplementary material.

Performance on molecular structure generation

Protein-small-molecule docking. Our evaluation began by assessing the performance of PocketXMol on small molecules docking using the PoseBusters benchmark set (*29*) that contains 428 protein-small-molecule complex structures. Following the PoseBusters benchmark setting, all baselines provided one docked pose per protein according to their default settings (*10, 11, 22, 29–36*). PocketXMol also provided one pose for each pocket, by selecting the top one pose from 100

generated docked poses sorted by the ranking scores (see supplementary material). The ranking scores were mainly based on the confidence scores produced together with the generated poses by PocketXMol (self ranking), and we also provided a confidence predictor tuned from the original PocketXMol to facilitate better ranking (tuned ranking). We calculated the atom RMSDs of the docked molecular poses and measured the ratio of RMSD < 2Å among 428 proteins. PocketXMol exhibited accurate docking performance, with 82.5% and 83.4% of poses achieving RMSD < 2Å with self ranking and tuned ranking, only second to the recent AlphaFold 3 with defined pockets (Fig. 2A and table S3). The RMSDs of PocketXMol exhibited correlations with the ranking scores (Fig. 2B and fig. S2A) and increasing the number of selected poses for each pocket also consistently enhanced the performance (fig. S2B). Moreover, if the best poses among the generated poses per protein could be precisely selected, i.e., with oracle ranking scores, the performance can be boosted to 96.5%, underscoring the robust generation capability and substantial room for improvement with more advanced ranking methodologies. We also observed no correlation between the RMSDs and molecular properties, confirming no bias toward specific molecule types (fig. S3).

For a more rigorous evaluation, we employed the PoseBusters test suite (*29*) as a docking validity checker. It employs a series of criteria to assess both the chemical and geometric coherence of 3D molecular structures, categorizing poses as PB-valid only when all the criteria are satisfied. We measured the ratio of poses satisfying both RMSDs < 2Å and being PB-valid. PocketXMol also achieved the best performance with 78.5% and 79.4% poses for self ranking and tuned ranking, respectively, among the 428 samples in the PoseBusters set (figs. S2C and S4). As AlphaFold 3 did not release code, PocketXMol was compared with it on the PoseBusters Version 2 set (containing 308 test samples) and achieved 84.7% and 84.4% for self ranking and ranker scores, respectively, comparable to 84.4% of AlphaFold 3 with defined pockets as reported in its paper (fig. S2D).

Protein-small-molecule docking with prior knowledge. It is common to incorporate prior knowledge into the docking process during the drug design. For instance, when docking a new molecule extended from a fragment hit, it is usually assumed that the fragment pose remains unchanged in the new molecule (*37*), which necessitates fixing the fragment positions throughout the docking process. Our flexible framework can efficiently integrate prior knowledge by either editing the molecules before adding noise at each generation step or directly employing the task

prompt to express the constraints. Here we showcased docking with different prior knowledge, including known molecular center positions, bond lengths, and approximate or exact coordinates of anchor atoms. The findings highlighted the effectiveness of various forms of prior knowledge in decreasing the docking RMSD (Fig. 2C). Notably, utilizing the exact coordinates of anchor atoms yielded the most performance improvement, attributed to the substantial reduction in freedom of molecular poses (Fig. 2D).

Molecular conformation generation PocketXMol can also generate valid 3D conformations for small molecules without the consideration of protein pockets. Following the test settings of previous works (*28, 38, 39*), PocketXMol was benchmarked with software specifically designed for 3D conformation prediction and exhibited comparable performance to most of these tailored methods (table S4). Fig. 2G provides an example, demonstrating that PocketXMol can generate almost identical conformations to the low-energy ones.

Protein-peptide docking. Current peptide docking methods are primarily constructed at the amino acid level (*40, 41*). In contrast, PocketXMol leverages atom-level representation and does not distinguish between peptides and small molecules, regarding the peptide docking exactly the same as small-molecule docking. We evaluated PocketXMol's performance against peptide docking approaches including the AlphaFold series (*10, 42, 43*) and the energy-based method FlexPepDock (*40*) on 79 protein-peptide complexes compiled from a previous peptide docking benchmark set (*41, 44*). We also included two representative small-molecule docking tools Uni-Mol Docking V2 (*33*) and Vina (*34*) as baselines by treating peptides as small molecules. Similar to the small molecule docking, PocketXMol selected one docked peptide structure per protein pocket from 100 generated ones according to the ranking scores. PocketXMol achieved satisfying performance with an average DockQ (*45*) of 0.53 and 0.58 for self ranking and tuned ranking, respectively (Fig. 2E, table S5). The performance of small-molecule docking tools was much lower than PocketXMol (fig. S6), indicating that current docking tools did not have the transferability between small molecules and peptides like PocketXMol did. We observed that the accuracy of PocketXMol can also be consistently enhanced by selecting more structures from the generated ones (fig. S5A), and the average DockQ score could reach 0.73 if the oracle ranking scores were

available, highlighting considerable opportunities for enhancement through more advanced ranking strategies. When considering the peptide properties, PocketXMol can achieve much higher accuracy on peptides with shorter lengths or lower proportions of helix/sheet secondary structures (Fig. 2F), which is explained by the fact that these peptides exhibited more flexibility and were more similar to small molecules. Notably, peptides longer than 15 were not included in our training dataset but they could still be docked with modest accuracy, highlighting PocketXMol's extrapolation and generalization abilities. Additionally, we also benchmarked with AlphaFold-Multimer (*42*) and AlphaFold 3 (*10*) on another test set comprised of the latest protein-peptide complexes collected from the Q-BioLiP database (*46, 47*), and the conclusions were the same as the above ones (fig. S7).

Furthermore, owing to the atom-level representation, PocketXMol can naturally model peptides with modified amino acids. There are 13 peptides containing non-standard amino acids in the test set and all of them were successfully docked with good accuracy (DockQ≥0.49) and showed small RMSDs for those atoms in the non-standard amino acids (Fig. 2H). As a comparison, AlphaFold-Multimer used the closest standard amino acids to replace these non-standard ones and only six among the 13 peptides achieved DockQ≥0.49 (fig. S5B). PocketXMol offers a more expansive scope for real biological applications because modifications are commonly utilized in peptide drug design to enhance their stability and prolong the half-life (*48*).

Protein-peptide docking with prior knowledge. For peptide docking, there are also cases when the positions of certain atoms or residues of the peptides are known in advance (*49–51*), necessitating integrating such prior knowledge into the docking process. Here, we showcased docking peptides while fixing certain atoms at various positions, including 1) the atom that was the closest to the protein (named anchor atom), 2) atoms in the first residue (N-terminus), 3) atoms in the two terminal residues (N- and C-terminus), and 4) all backbone atoms. We observed that the performance on docking could be enhanced to different extents with prior knowledge (Fig. 2I). In particular, fixing only one anchor atom could yield an improvement of the median DockQ from 0.62 to 0.71. Fixing the atoms at the N-terminus and both N- and C-terminus could greatly enhance the performance to 0.78 and 0.93, respectively. These findings underscored the potential applicability of PocketXMol in protein loop modeling tasks, such as predicting the structures of antigen-binding antibody complementarity-determining regions (CDRs), which can be conceptualized as flexible peptides,

with terminal amino acids linked to a relatively rigid framework (*52*). A special case is fixing all the backbone atoms of the peptide and only determine the side-chain atoms, which is similar to side-chain packing. The RMSDs of the peptide side-chain atoms between the generated and the ground truth were small, with a median RMSD of 1.1 Å (Fig. 2I) for all side-chain atoms.

Performance on small molecule design

Structure-based drug design. Structure-based drug design (SBDD) with AI models, which involves designing novel small molecules based on the 3D protein pocket, has emerged as a successful approach in drug discovery as it directly models the interactions between amino acids and ligand atoms in the 3D space (*53*). One example is that Novartis has successfully designed three WDAC inhibitors using an SBDD model combined with their compound databases (*54, 55*). To evaluate the performance of PocketXMol on the SBDD task, we conducted a comprehensive comparison against multiple existing methods including a VAE-based method liGAN (*56*), autoregressive based methods 3DSBDD (*14*) and Pocket2Mol (*55*), and two diffusion-based methods TargetDiff (*23*) and DecompDiff (*57*). We designed 100 molecules for each of the 100 protein pockets in the benchmark set used in previous work (*14, 23, 55–58*). PocketXMol could employ two different generation strategies: a default refine-based strategy (PocketXMol) and an auto-regressive-like strategy (PocketXMol-AR) (supplementary text section 4.3.4). We first evaluated the designed molecules with 14 diverse metrics, spanning molecular properties, the coherence of molecular graph distribution, and the fidelity of the 3D structures of the generated molecules. PocketXMol ranked first in 11 out of 14 metrics with balanced overall performance (Fig. 3A and table S6). Following previous work (*23*), we utilized AutoDock Vina (*34*) to calculate ratios of molecules with higher affinities (lower Vina scores) than the corresponding molecules in the test set for each pocket to evaluate their binding. We also employed the PoseBusters 3D validity checker (*29*) as a filter due to the previous observation that some models tended to sacrifice the intrinsic rationality of the molecules in pursuit of stronger binding interactions with protein pockets (*59*). Results showed that the proportion of high affinity and 3D valid molecules generated by PocketXMol exceeded all baselines with clear margins (Fig. 3B and fig. S8). Notably, in the absence of 3D structural filtering, around half of the molecules generated by liGAN demonstrate higher affinities, however, almost all generated

molecules displayed a certain degree of irrationality, highlighting the importance of applying the 3D valid checker.

3D molecule generation. Compared to the SBDD task, the 3D molecule generation task operates without the constraints imposed by protein pockets. In drug discovery, it acts as an alternative *in silico* approach that first generates 3D molecules and then screens them to the target proteins. We evaluated the performance on this task and compared it with diffusion-based molecular generation models (*60–64*) by generating 1000 molecules with sizes akin to drug-like compounds. The molecules generated by PocketXMol exhibited a high ratio of validity and also matched the distributions of real drug-like molecules well (Fig. 3C and fig. S9).

In order to evaluate the capability of incorporating prior knowledge, we showcased generating 3D molecules under arbitrary volumetric shape specifications. Specifically, we generated 3D molecules conditioned on the shapes of the letters of our model's name, which is achieved by adjusting the atom coordinates to align with the point clouds of the shapes during the generation process (*65*) (supplementary text section 4.3.5). The resulting molecules not only exhibited the predefined shape but also featured reasonable topology and structures (Fig. 3D), demonstrating its designing capacity with prior knowledge conditions.

Fragment linking. Having explored PocketXMol's performance for *de novo* design, we next considered designing based on existing fragments or molecules, including fragment linking, fragment growing, and molecule optimization. These strategies are widely used in drug development to achieve desired properties by modifying or adding certain chemical groups of existing molecules (*37, 53, 66*). Fragment linking involves the strategic design of a linker to unite disconnected fragments into one complete and stable drug-like molecule (*67*). Such technology can be extended for PROTAC (proteolysis-targeting chimeras) drug development, where a molecular bridge needs to be designed to connect a protein-targeting ligand with an E3 ligase-recruiting ligand (*68*).

Due to the atom-level molecular representation, one of the advantages of PocketXMol is its multi-functionality in handling various settings of the fragment linking task, while other methods were proposed to cover only some specific settings (fig. S10A). In a typical fragment linking

task, molecular linkers are designed based on fixed fragments within the protein pocket. We conducted comparisons in this setting with models that support fixed-fragment linker design, including DeLinker (*67*), 3DLinker (*69*), and DiffLinker (*70*) on a linker design test set proposed by DiffLinker (*70*) from the Binding MOAD dataset (*71*). For each model to be compared, we generated 100 3D molecules for each fragment pair in the case of either knowing the connecting atoms of fragments or not. PocketXMol could generate higher proportions of molecules satisfying both 2D and 3D validity than the second-place model DiffLinker (Fig. 3E left). Here, 2D and 3D valid molecules indicate molecules with realistic chemical topology examined by RDKit (*72*) and valid 3D geometry examined by the PoseBusters validity checker, respectively. PocketXMol also recovered the 2D molecular graphs or the 3D poses of the ground truth 3D linkers for more fragment pairs than baselines (Fig. 3E right). Additionally, the molecules sampled by PocketXMol more closely resembled the ground truth molecules than those by baselines, based on the similarity score proposed by DeLinker (Fig. 3F).

PROTAC design. For PROTAC design, we considered two scenarios where the poses of the fragments were specified or not. We compared PocketXMol with DiffLinker (*70*), the state-of-the-art linker design model that was conditioned on known fragment poses, and LinkerNet (*73*) proposed specifically for unknown fragment poses. The test set was constructed by LinkerNet from the PROTAC-DB dataset (*74*) with 43 PROTAC molecules. In the scenario where fragment poses are provided, we generated 100 linkers for each pair of fragments (a pair of warhead and E3 ligand). PocketXMol produced 60% valid (both 2D and 3D) PROTAC molecules on average, much higher than 34% achieved by DiffLinker (Fig. 3G). PocketXMol also generated molecules exhibiting higher similarities with the real PROTAC molecules (fig. S10B). In the scenario where fragment poses were unknown, the models needed to generate the linker and determine the poses of fragments simultaneously. We thus perturbed the input fragment poses randomly at five distinct noise levels and generated 30 molecules for each pair of fragments and noise levels. In the comparison to LinkerNet, PocketXMol successfully designed more valid (both 2D and 3D) PROTAC molecules (Fig. 3H). The PROTACs designed by PocketXMol also demonstrated certain similarities with the real ones (fig. S10C).

Fragment growing. Fragment growing is another drug development strategy that starts from target-binding molecular fragments with relatively small sizes and applies chemical modifications to design complete molecules with improved binding affinity and other properties. To evaluate PocketXMol's capability for fragment growing, we constructed a test set using the data in the SBDD test set by segmenting molecules into multiple fragments using BRICS decomposition (*75*) and selected connecting fragments as the initial fragments with similar sizes. For each pair of the protein pocket and the fragment graph (without 3D poses), we invoked PocketXMol to generate 100 molecules with the same sizes as the original molecules in the test set. We compared PocketXMol with a baseline method DiffDec but had to first apply Vina to dock the fragments as the inputs for DiffDec because DiffDec cannot determine the fragment poses by itself. Results showed that PocketXMol generated molecules with better completeness and more similarities with the ground truth molecules than the baseline (fig. S11A,B). The molecules designed by PocketXMol showed better binding affinities in terms of the average of top 1, top 10, and all Vina scores of the generated ones for each test sample (Fig. 3I). Moreover, the molecules by PocketXMol with the top 1 or top 10 Vina scores displayed significantly improved binding affinity compared to the corresponding reference molecules containing the same initial fragments in the test set (fig. S11C).

Molecule optimization. The aim of molecule optimization is to modify the molecules to improve specific properties. PocketXMol achieved this by perturbing the given molecules with controlled levels of noise and subsequently removing the noise to generate new molecules, from which those satisfying the desired properties were selected as new starting points for repeated optimization rounds. Through several iterative rounds, PocketXMol generated molecules with specified properties while retaining similarity to the initial molecules (supplementary text 4.3.9). Here, we demonstrated the optimization of LogP values of the 100 molecules in the SBDD test set to around 1.8, a reasonable objective in some scenarios of drug design (*76*). Through three optimization rounds, the LogP values of molecules rapidly converged to the specified target, exhibiting significantly smaller errors to the objective than those before optimization (one-sided paired t-test, $P = 6.12 \times 10^{-20}$, $n = 100$, Fig. 3J and fig. S12).

Caspase-9 inhibitor design. Mitochondrial permeabilization could release cytochrome c into the cytosol to form the apoptosome and initiate intrinsic apoptosis by activating caspase-9; caspase-9 thereafter cleaves and activates caspase-3/7, the central executioner of apoptosis, to further cleave downstream substrates, such as PARP1, DFFA, and lamin A, and promote cell death (*77*) (Fig. 4A). Caspase-9 mediated intrinsic apoptosis is a major barrier of tumor endogenous cyclic GMP–AMP synthase (cGAS)–stimulator of interferon genes (STING) pathway, and inhibition of caspase-9 could significantly provoke antitumor immunity to synergize with radiation and immunotherapy (*77*). However, the current first-in-class caspase inhibitor, emricasan, is a pan-caspases inhibitor that could inhibit caspase-9/3/7 medicated intrinsic apoptosis, as well as caspase-8 mediated extrinsic apoptosis and caspase-1 mediated inflammasome activation. Notably, the inflammasome pathway is also critical to antitumor immunity (*78, 79*); therefore, Emricasan might also target caspase-1 to suppress antitumor immunity. To investigate whether PocketXMol could help *de novo* design specific small molecule inhibitors, we applied it to design inhibitors targeting caspase-9 (supplementary text section 7.1) and validated them experimentally. We synthesized 16 designed molecules (fig. S13) and one of them (ID: 84663, Fig. 4B) could effectively inhibit ABT-737, a BCL-2 inhibitor triggering mitochondria-mediated intrinsic apoptosis, mediated caspase-9 and caspase-3 activation and cleavage of PARP1, a target of activated caspase-3 (Fig. 4C and fig. S14). This small molecule was completely novel and had low similarities with existing caspase inhibitors (fig. S15). There were no similar molecules found in the ZINC20 database (*80*) and the ChemDiv Library (`www.chemdiv.com`), and the most similar molecules in the ChEMBL (*81*) and PubChem (*82*) databases also had massive differences with the molecule 84663 (fig. S16). Based on the structure of the molecule, we then re-optimized it and synthesized 15 optimized molecules (fig. S17A), among which four (IDs: D12, D13, D18, and D19; Fig. 4D) showed remarkable effects. As shown in Fig. 4E, treatment with ABT-737 could enhance the cleavage of caspase-3 that was diminished after QVD (Q-OVD-oph, a powerful pan-caspase inhibitor) treatment; notably, the designed molecules could also suppress the activation of caspases-9 and caspases-3 with efficacy comparable to QVD. Using the caspase-3 function assay, we found that the four molecules could inhibit ABT-737-induced, caspase-9-mediated caspase-3 activation with EC_{50} values ranging from 0.85 to 8.56 μM (Fig. 4F). To further confirm the designed molecules function by targeting caspase-9 instead of caspase-3, we thereafter tested the effect of the molecules on the activity of

c-caspase-3 (cleavage caspase-3, an active caspase-3). As shown in Fig. 4G, QVD treatment, but not the molecules 84663 or D12, could abolish the enzyme activity of c-caspase-3, indicating that the designed molecules functioned to inhibit ABT-737 induced intrinsic apoptosis by targeting caspase-9.

Performance on peptide design

With the atom-level representation of PocketXMol, peptide design exhibits no inherent distinction from the small molecule design. The primary challenge lies in how to precisely control the model to generate chains of amino acids instead of arbitrary molecules. Based on the fact that the backbone atoms remained consistent across peptides with differences primarily stemming from their side-chain atoms, PocketXMol can regard peptide generation as a special case of fragment growing, wherein the side-chain atoms are generated based on the peptide backbone and the binding protein pocket. Specifically, given the length of the peptide, PocketXMol generates new atoms (and bonds) based on the backbone fragment and the binding pocket to form a complete molecule, and these newly generated atoms constitute the side chains, from which the amino acid types can be annotated (Fig. 5A). This turns the design of peptides' structures and sequences into the end-to-end design of molecular fragments, which distinguishes PocketXMol fundamentally from existing peptide design methods where they explicitly model the amino acid types during designing. Additionally, PocketXMol simultaneously generated all atom positions, eliminating the need for additional docking or side-chain packing using separated post-processing steps.

To validate this novel peptide design strategy, we built a peptide design test set by collecting 35 protein-peptide complexes from the peptide docking test set (*83*) and utilized PocketXMol to design peptides for each protein pocket (supplementary text section 4.3.10). Firstly, to ascertain whether PocketXMol learned that atoms could form amino acids, we retrieved the average embeddings of the side-chain atoms of the generated peptides and projected the amino acid embeddings into two-dimensional space using t-SNE (Fig. 5C). The amino acid clusters correspond precisely to the types of amino acids with an Adjusted Rand Score of 0.813. The learned embeddings reflect chemical similarities among side-chain atoms, positioning amino acids with similar side chains in closer proximity. Moreover, the learned embeddings contain the structured semantic information

of side chains. In the embedding space, the difference between glutamic acid (E) and glutamine (Q) is approximately equal to the difference between aspartic acid (D) and asparagine (N), both of which correspond to replacing an amine group -NH_2 with an oxhydryl group -OH. Additionally, the distances between the amino acid embeddings exhibited a strong correlation with the BLOSUM62 substitution scores, with a Pearson correlation of -0.82 (Fig. 5B). Since the amino acid types were inferred from the side chains, a potential concern is that the side-chain atoms may not always constitute standard amino acids. However, in practice, we observed that the ratios of generated peptides containing all standard amino acid types (referred to as valid peptides) are acceptable for various pockets and peptide lengths (Fig. 5D). For the subsequent experiments, we only considered and retained these valid peptides.

Next, we compared PocketXMol with the state-of-the-art protein/peptide design pipeline, which utilizes RFdiffusion (*21*) to generate backbones, followed by the sequence generation by ProteinMPNN (*84*) and side-chain packing by Rosetta (*85*) (referred to as RFdiffusion pipeline through this work). For both RFdiffusion and PocketXMol, we generated at most 100 valid peptides for each protein target in the test set and assessed the quality of the generated peptides. First, we analyzed the amino acid types and observed that the distribution of types generated by PocketXMol was much closer to the distribution of the test set than the RFdiffusion pipeline, with a Jensen–Shannon divergence of 0.16 for PocketXMol and 0.23 for the RFdiffusion pipeline (Fig. 5E). The average sequence recovery rate of PocketXMol reached 37.2%, better than 32.1% of RFdiffusion (Fig. 5F). For 3D structural consistency, we calculated the backbone RMSD of each generated peptide to the reference peptide in the test set. PocketXMol generated peptides with smaller backbone RMSDs than RFdiffusion (Fig. 5G). Moreover, we calculated the binding affinity ΔG between the proteins and peptides using Rosetta (*85*) and found that PocketXMol generated peptides with superior binding affinities compared to RFdiffusion, which can be attributed to directly modeling interactions at the atom level instead of the interactions between amino acids (Fig. 5H). Finally, we observed that the RFdiffusion pipeline generated peptides predominantly featuring helix structures, while PocketXMol produced mostly coils with a minority of helices, bends, and turns, which more closely resembled the distribution of peptides' secondary structures in the test set (Fig. 5I).

Peptide inverse folding. Peptide inverse folding is a generative task involving the design of peptide sequences based on the structures of peptide backbone and protein pocket (*84*). To assess PocketXMol's performance on this task, we removed the side chains of peptides in the peptide design test set while retaining the peptide backbone structures and protein pockets and employed PocketXMol and the baseline ProteinMPNN (*84*) to design 100 valid peptides for each pocket, respectively. PocketXMol demonstrated a significantly higher sequence recovery rate of 61.7% than 54.4% for ProteinMPNN (one-sided paired t-test, $P = 0.02, n = 35$), indicating the effectiveness of directly generating and placing the side-chain atoms within the pocket, as opposed to merely classifying amino acid types based on the backbone structures as ProteinMPNN did (Fig. 5J left). PocketXMol also exhibited significantly higher sequence diversity compared to ProteinMPNN (one-sided paired t-test, $P = 3.18 \times 10^{-24}, n = 35$), due to the fact that predicting side-chain atoms is less prone to overfitting compared to merely predicting amino acid types during training (Fig. 5J right). Moreover, the side-chain conformations were simultaneously determined by PocketXMol during the generation process. The atoms of the residues recovered by PocketXMol are consistent with the reference structure, indicating its robust ability to generate accurate all-atom 3D structures (Fig. 5K and fig. S18).

PD-L1 binder design. To rigorously validate PocketXMol's design capability, we employed PocketXMol to design peptides targeting PD-L1 and experimentally confirmed their binding affinities and specificities. Peptides were generated for three PD-L1 pocket structures (PDB ID: 3BIK, 4ZQK, and 5IUS), followed by thorough filtering and ranking using several scoring tools, including self-confidence, FoldX (*86*), Rosetta (*85*), and AlphaFold (*8*) (supplementary text section 7.2). Finally, 382 peptides were selected for validation of their binding with PD-L1 using SPRi experiments. The results revealed that 15 peptides achieved the dissociation constant (K_D) of 10^{-8}M and 76 peptides achieved K_D of 10^{-7}M (Fig. 6A). As a baseline, we constructed a random peptide library with a capacity of 10^7 to experimentally screen peptides targeting PD-L1, which only resulted in only 8 positive peptides with K_D values of or close to 10^{-8}M. PocketXMol achieved a success rate $50,000$ times that of the screening approach. For the 15 peptides with K_D of 10^{-8}M, we clustered the generated peptide structures (Fig. 6B), which demonstrated both the diversity and the consistency of the peptides. We presented three examples of the peptide binding curves and the

interaction profiles in Fig. 6C and fig. S19, which depicted the rich interactions between the PD-L1 pocket and the designed peptides, elucidating their strong binding affinities.

Next, we selected three peptides with high affinity and representative structures (P65, P73, and P282) and conjugated them with FITC (Fluorescein isothiocyanate) fluorescent dye, aiming to validate their affinity and specificity at the cellular level. After co-incubating the three fluorescent-labeled peptides with PD-L1 positive cells H1975, the peptides effectively bound to the cell membrane, emitting strong green fluorescence. Co-localization staining with PD-L1 antibodies further confirmed that the peptides were specifically bound to PD-L1 proteins on the cell membrane surface (Fig. 6D). In contrast, when these peptides were co-incubated with PD-L1 negative 293T cells, there was negligible green fluorescence observed on the cell membranes (fig. S20), which demonstrated the robust affinity and specificity of the designed peptides targeting PD-L1.

To further assess the in vivo targeting efficacy of the PD-L1 peptides, we conducted in vivo imaging experiments using indocyanine green (ICG), an FDA-approved dye commonly used for fluorescence imaging in vivo. The peptides conjugated with ICG were administered at a concentration of 50 μM via the tail vein in a PD-L1 positive tumor model with the non-small cell lung tumor. As depicted in Fig. 6E, P65, P73, and P282 exhibited significant accumulation at the tumor sites within 0.5h post-injection, with the fluorescence signal gradually increasing over time and peaking at 4h. In contrast, no signal was observed at the tumor site in the control group injected with free ICG, indicating that peptides modified with ICG can accurately target tumors expressing high levels of PD-L1. After 12 hours, we removed the heart, liver, spleen, lungs, kidneys, and tumor tissues from each group of mice for ex vivo imaging. As shown in Fig. 6F and fig. S21, there was still a strong signal at the tumor site 12 hours after injection in the peptide probe group, while there was almost no signal at the tumor site in the control group. The liver and kidneys of each group show higher non-specific accumulation due to peptide degradation and free ICG accumulation during circulation and metabolism through the liver and kidneys. As a comparison, we also conducted the same experiments for two positive peptides obtained through library screening (figs. S22 and S23) and found that they cannot maintain lasting in vivo targeting with tumors. These results demonstrate that the peptides designed by PocketXMol have excellent targeting and selectivity in vivo and can be used for targeted tumor imaging and diagnosis.

Finally, To determine whether the designed peptides can inhibit the interaction between PD-1

and PD-L1 and exert therapeutic effects, we conducted a ligand inhibition assay for the peptide P282 using flow cytometry analysis (Fig. 6G). In the absence of the peptide, PD-1 could effectively bind to PD-L1. However, in the presence of the peptide, the binding of PD-1 was reduced. As the concentration of the peptide increased, the PD-1-Cy5 fluorescence signal further diminished. These results indicate that the peptide can inhibit the interaction between PD-1 and PD-L1 to some extent and has the potential in cancer immunotherapy.

Discussion

To elucidate the fundamental atomic interactions and achieve the unification of molecular generative tasks, PocketXMol has introduced several methodological innovations with potentially profound impacts on the field. PocketXMol exclusively employs atom-level representation, eschewing advanced biological entities such as amino acids. This approach aligns with the core concept of molecular interactions, where all molecules are composed of atoms that interact through universally applicable mechanisms. Our experiments have demonstrated the feasibility of modeling molecules at the atomic level, showcasing the transferability among different molecular types. While some current molecular models can generate all-atom structures, they often treat different molecules using distinct entities. For example, AlphaFold 3 (*10*) represented proteins as amino acids, nuclear acids as nucleotides, and other molecules (e.g., modified amino acids and small molecules) as atoms and learned the interactions among these entities. However, a modified amino acid and its standard form may only differ in several atoms but are treated as entirely different representations in the model, which may hinder the model from learning the shared knowledge among different molecules. Another advantage of the atom-level representation is that any molecule can be directly modeled, avoiding the need for additional entity-specific representations, which is particularly beneficial for uncommon molecular types, such as macrocycles, peptoids, D-peptides, and modified peptides.

Secondly, we introduced a novel task representation that unifies diverse molecular tasks into a single computational framework, directly reflecting the relationships among tasks. The task representation bridges the gap among molecular tasks, especially for structure prediction and molecular designing. Recent advances in structure prediction models like AlphaFold 3 (*10*) and RFAA (*11*)

have made significant progress. Equipped with such a task representation, these successes can be directly extended to molecular designing without requiring additional fine-tuning or re-folding as current models do (*21, 87*). Finally, our new generative framework is capable of handling multiple tasks with different prior noise distributions. It allows the denoiser to automatically learn noise characteristics, which is equivalent to performing task identification during training. While initially proposed for molecular tasks, this generative framework can be applied to other fields, enabling mixed noise types and tasks within a single model.

These innovations enable PocketXMol to treat peptides in a fundamentally different manner from previous models, providing a new paradigm for the field. By considering peptides as a special type of small molecule, PocketXMol docks peptides in the same way as small molecules and designs peptides through fragment growing. However, this does not imply that existing small-molecule docking or fragment-growing tools can be easily extended to peptides. Our results showed that small-molecule docking tools like AutoDock Vina (*34*) and Uni-Mol Docking V2 (*33*) have limited performance on peptides. Designing peptides through fragment growing requires docking backbone fragments while generating side-chain atoms of standard amino acids, a capability that current fragment-growing tools lack. PocketXMol not only learns the commonalities between small molecules and peptides but also models their subtle differences.

It is also important to note that this work is exploratory, only demonstrating the feasibility of an atomic interaction foundational model. PocketXMol focuses on tasks related to pocket-interacting ligands and is not designed for more general molecular interactions, such as protein folding and protein docking, due to the scalability of hardware and experimental limitations. Larger molecular systems, such as entire proteins, contain significantly more atoms than the tasks considered in this work. The quadratic complexity of PocketXMol's architecture in terms of the number of atoms makes scaling to larger systems infeasible. Additionally, the inconsistency in the number of atoms across different molecules presents a challenge for the model in learning shared knowledge. Overall, PocketXMol is a step forward to a more general generative molecular model, but further research is needed to address these challenges.

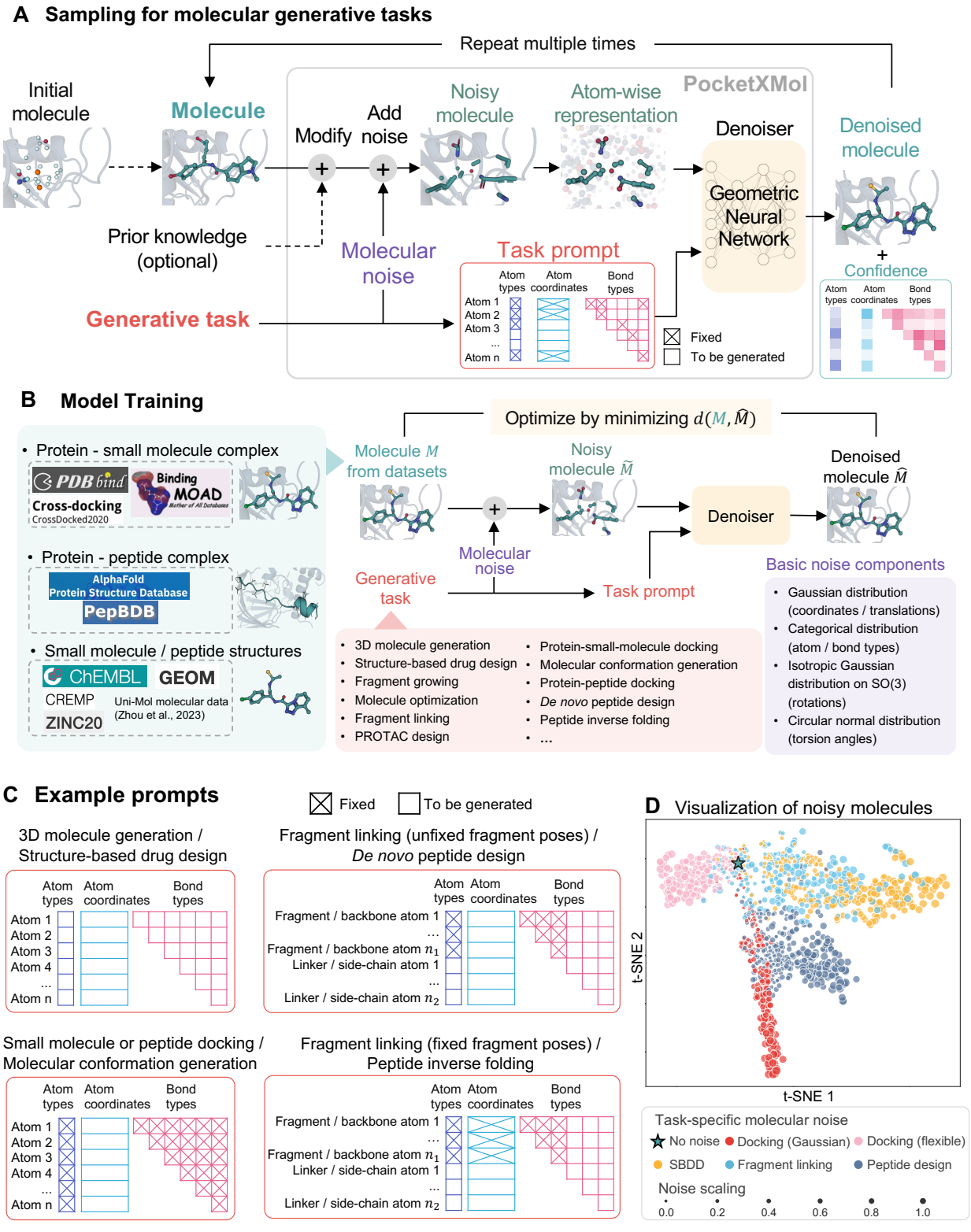
A Sampling for molecular generative tasks
Repeat multiple times
Initial molecule
Molecule
Modify
Add noise
Noisy molecule
Atom-wise representation
PocketXMol
Denoiser
Geometric Neural Network
Denoised molecule
+
Confidence
Prior knowledge (optional)
Molecular noise
Task prompt
Generative task
Atom types
Atom coordinates
Bond types
Atom 1
Atom 2
Atom 3
...
Atom n
Fixed
To be generated
B Model Training
Optimize by minimizing $d(M, \widehat{M})$
• Protein - small molecule complex
PDBbind
Binding MOAD
Cross-docking
CrossDocked2020
• Protein - peptide complex
AlphaFold Protein Structure Database
PepBDB
• Small molecule / peptide structures
ChEMBL
GEOM
CREMP
ZINC20
Uni-Mol molecular data (Zhou et al., 2023)
Molecule M from datasets
Noisy molecule $\widetilde{M}$
Denoised molecule $\widehat{M}$
Denoiser
Molecular noise
Generative task
Task prompt
• 3D molecule generation
• Structure-based drug design
• Fragment growing
• Molecule optimization
• Fragment linking
• PROTAC design
• Protein-small-molecule docking
• Molecular conformation generation
• Protein-peptide docking
• De novo peptide design
• Peptide inverse folding
• ...
Basic noise components
• Gaussian distribution (coordinates / translations)
• Categorical distribution (atom / bond types)
• Isotropic Gaussian distribution on SO(3) (rotations)
• Circular normal distribution (torsion angles)
C Example prompts
Fixed
To be generated
3D molecule generation / Structure-based drug design
Fragment linking (unfixed fragment poses) / De novo peptide design
Fragment / backbone atom 1
Fragment / backbone atom n_1
Linker / side-chain atom 1
Linker / side-chain atom n_2
Small molecule or peptide docking / Molecular conformation generation
Fragment linking (fixed fragment poses) / Peptide inverse folding
D Visualization of noisy molecules
t-SNE 1
t-SNE 2
Task-specific molecular noise
No noise
Docking (Gaussian)
Docking (flexible)
SBDD
Fragment linking
Peptide design
Noise scaling
0.0
0.2
0.4
0.6
0.8
1.0

Figure 1: **PocketXMol framework.** (**A**) Schematic representation of the sampling process of PocketXMol. PocketXMol perturbs the molecule using task-specific noise and prior-knowledge-guided modification and subsequently utilizes the denoiser neural network to generate the denoised molecule with the confidence scores. (**B**) The training process of PocketXMol. The denoiser was trained to recover the molecules from the noisy ones, with data assembled from multiple 3D molecular structure datasets. (**C**) Examples of how the task prompts are designed for different generative tasks. (**D**) t-SNE visualization of the noisy molecules with different task-specific molecular noise. Different colors represent noisy molecules after the injection of different types of noise. We selected a protein-peptide complex and added different task-specific noise with random noise scales to the peptide molecule. For each molecule, we calculated the probability distributions of atom pairwise distances and probability distributions of the atom and bond types as the molecular representation. We then applied t-SNE to visualize the representations.

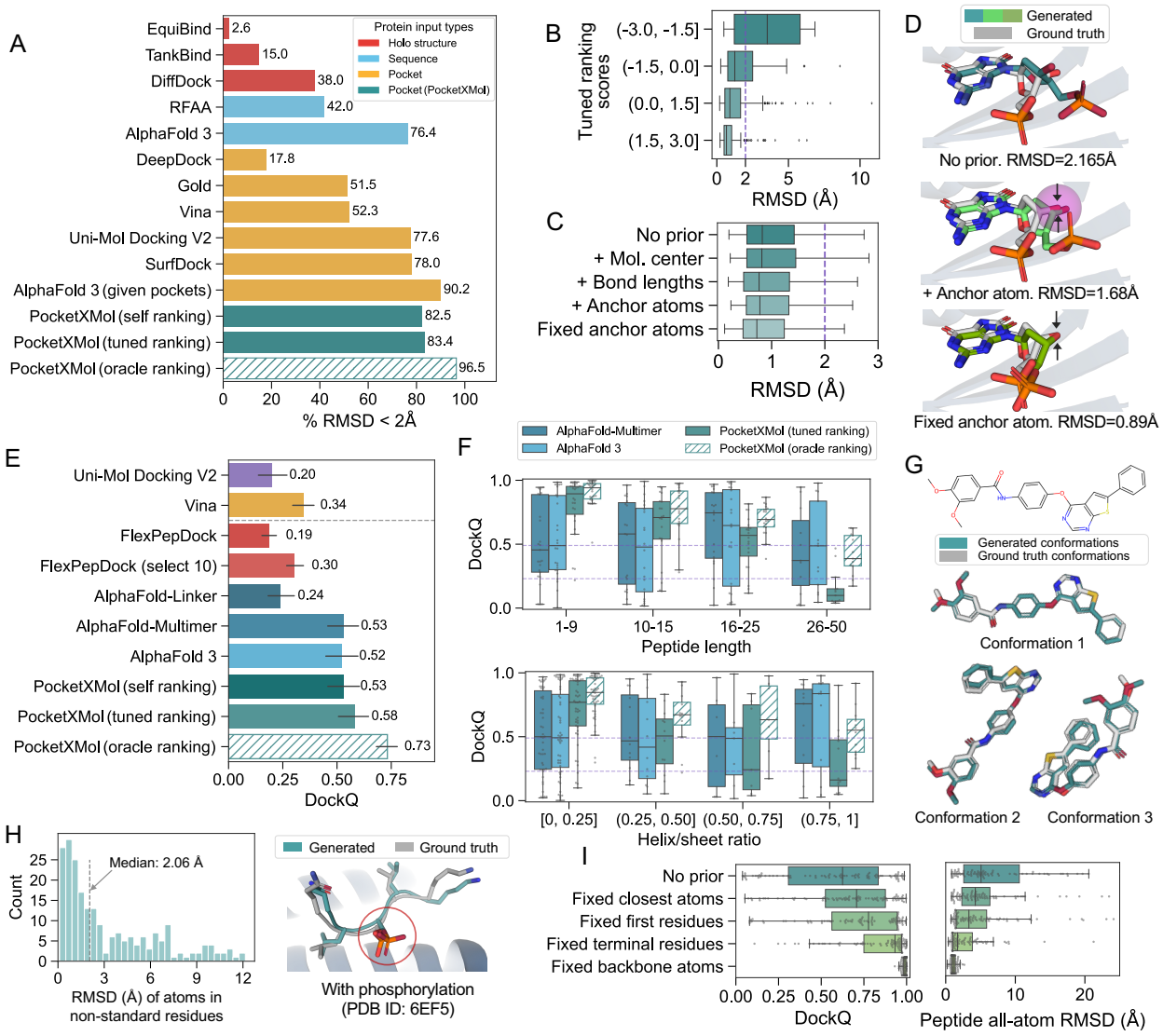
A
EquiBind 2.6
TankBind 15.0
DiffDock 38.0
RFAA 42.0
AlphaFold 3 76.4
DeepDock 17.8
Gold 51.5
Vina 52.3
Uni-Mol Docking V2 77.6
SurfDock 78.0
AlphaFold 3 (given pockets) 90.2
PocketXMol (self ranking) 82.5
PocketXMol (tuned ranking) 83.4
PocketXMol (oracle ranking) 96.5
Protein input types
Holo structure
Sequence
Pocket
Pocket (PocketXMol)
% RMSD < 2Å
B
Tuned ranking scores
(-3.0, -1.5]
(-1.5, 0.0]
(0.0, 1.5]
(1.5, 3.0]
RMSD (Å)
C
No prior
+ Mol. center
+ Bond lengths
+ Anchor atoms
Fixed anchor atoms
RMSD (Å)
D
Generated
Ground truth
No prior. RMSD=2.165Å
+ Anchor atom. RMSD=1.68Å
Fixed anchor atom. RMSD=0.89Å
E
Uni-Mol Docking V2 0.20
Vina 0.34
FlexPepDock 0.19
FlexPepDock (select 10) 0.30
AlphaFold-Linker 0.24
AlphaFold-Multimer 0.53
AlphaFold 3 0.52
PocketXMol (self ranking) 0.53
PocketXMol (tuned ranking) 0.58
PocketXMol (oracle ranking) 0.73
DockQ
F
AlphaFold-Multimer
AlphaFold 3
PocketXMol (tuned ranking)
PocketXMol (oracle ranking)
DockQ
1-9
10-15
16-25
26-50
Peptide length
[0, 0.25]
(0.25, 0.50]
(0.50, 0.75]
(0.75, 1]
Helix/sheet ratio
G
Generated conformations
Ground truth conformations
Conformation 1
Conformation 2
Conformation 3
H
Median: 2.06 Å
Count
RMSD (Å) of atoms in non-standard residues
Generated
Ground truth
With phosphorylation (PDB ID: 6EF5)
I
No prior
Fixed closest atoms
Fixed first residues
Fixed terminal residues
Fixed backbone atoms
DockQ
Peptide all-atom RMSD (Å)

Figure 2: **Performance on molecular structure generation.** (**A**) Ratio of docked poses with RMSD < 2Å by different models for protein-small-molecule docking. The numbers on the bars are the ratio values. (**B**) The relationship between PocketXMol's tuned ranking scores and the RMSDs for the docked poses. (**C**) The RMSD values by incorporating different types of prior knowledge using PocketXMol for protein-small-molecule docking. (**D**) An example showing that docking can be improved by incorporating prior knowledge. The top, middle, and bottom structures represent the generated docking poses with no prior information, with the approximated location of the anchor atom, and with the exact coordinate of the anchor atom, respectively. The arrows indicate the anchor atoms and the purple sphere represents the approximated locations of the anchor atom. (**E**) DockQ of protein-peptide docking by different models. Error bars represent 95% confidence intervals (n = 79). (**F**) Relationship between DockQ for peptide docking and the peptide length (left) or the ratio of secondary structures helix and sheet (right). The box plot outlines quartiles of the metrics with whiskers spanning up to 1.5 times the interquartile range (n = 79). Here we only displayed the best four methods ranked in panel E. (**G**) An example showing PocketXMol could generate high-quality conformations for a 2D molecular graph (ChEMBL ID: CHEMBL1307825). The three conformations here represent three low-energy conformations of this molecule in the dataset. (**H**) RMSD for atoms in non-standard amino acids (left), and an example of docked peptide with phosphorylation (right). (**I**) DockQ and RMSD of protein-peptide docking by PocketXMol with different types of prior knowledge. The box plot outlines quartiles of the metrics with whiskers spanning up to 1.5 times the interquartile range (n = 79).

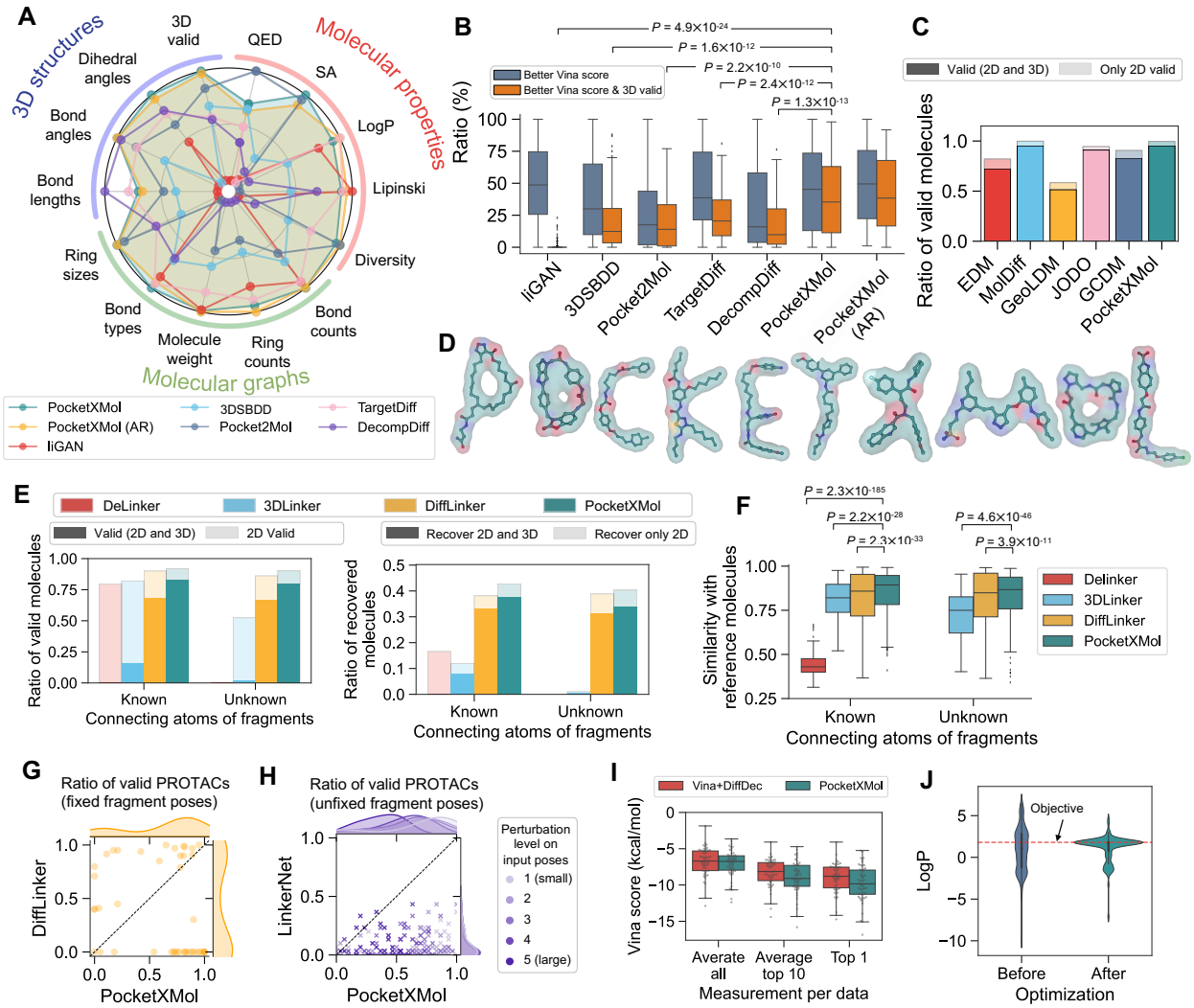
A
3D structures
Molecular properties
Molecular graphs
3D valid
QED
SA
LogP
Lipinski
Diversity
Bond counts
Ring counts
Molecule weight
Bond types
Ring sizes
Bond lengths
Bond angles
Dihedral angles
PocketXMol
PocketXMol (AR)
liGAN
3DSBDD
Pocket2Mol
TargetDiff
DecompDiff
B
Better Vina score
Better Vina score & 3D valid
Ratio (%)
liGAN
3DSBDD
Pocket2Mol
TargetDiff
DecompDiff
PocketXMol
PocketXMol (AR)
C
Valid (2D and 3D)
Only 2D valid
Ratio of valid molecules
EDM
MolDiff
GeoLDM
JODO
GCDM
PocketXMol
D
E
DeLinker
3DLinker
DiffLinker
PocketXMol
Valid (2D and 3D)
2D Valid
Recover 2D and 3D
Recover only 2D
Ratio of valid molecules
Ratio of recovered molecules
Known
Unknown
Connecting atoms of fragments
F
Similarity with reference molecules
Delinker
3DLinker
DiffLinker
PocketXMol
Known
Unknown
Connecting atoms of fragments
G
Ratio of valid PROTACs (fixed fragment poses)
DiffLinker
PocketXMol
H
Ratio of valid PROTACs (unfixed fragment poses)
LinkerNet
PocketXMol
Perturbation level on input poses
1 (small)
2
3
4
5 (large)
I
Vina+DiffDec
PocketXMol
Vina score (kcal/mol)
Averate all
Average top 10
Top 1
Measurement per data
J
Objective
LogP
Before
After
Optimization

Figure 3: **Performance on small molecule design.** (**A**) Performance on the SBDD task. Each metric on the radar plot was normalized to a scale from 0 to 1, with 1 representing the best method and 0 indicating the worst. The values and the definitions of the metrics can be found in table S6 and supplementary text section 6.1, respectively. (**B**) Ratios of the generated molecules exhibiting both high affinity and 3D validity for each pocket. High affinity is defined as those with Vina scores better than the corresponding reference molecule in the test set. The 3D validity was annotated by the PoseBusters validity checker. P values were computed based on one-sided paired t-test with $n = 100$. The box plot outlines quartiles of the metrics with whiskers spanning up to 1.5 times the interquartile range. (**C**) Ratio of valid molecules for pocket-free 3D molecule generation. Here, the 2D validity was defined as complete molecules by RDKit, and 3D validity was annotated by the PoseBusters. (**D**) Molecules generated by PocketXMol with shape specifications. (**E**) Ratios of valid molecules (left) and recovered molecules (right) for fragment linking, with known connecting atoms or not. (**F**) Similarities between the generated molecules and the reference molecules in the fragment linking test set. P value was calculated using a one-sided paired t-test. The sizes for each box from left to right are 282, 283, 413, 415, 255, 409, and 416. The differences in sizes were caused by the fact that no valid molecules were generated for some input pairs by some methods. (**G**) Comparison of the ratio of valid molecules between DiffLinker and PocketXMol for PROTAC design with fixed fragment poses. Here the validity included both 2D and 3D validity and the ratios were calculated for each fragment pair. (**H**) Comparison of the ratio of valid molecules between LinkerNet and PocketXMol for PROTAC design with unfixed fragment poses. The fragment poses were randomly perturbed with five different noise levels. Here the validity included both 2D and 3D validity and the ratios were calculated for each fragment pair and noise level. (**I**) AutoDock Vina scores of generated molecules for fragment growing. The x-axis represents the average of the top 1, top 10, and all Vina scores of the generated molecules for each test sample. (**J**) LogP distribution of 100 molecules in the SBDD test set before and after applying the molecule optimization by PocketXMol. The optimization was to modify the molecules to adjust their LogP values towards the objective of 1.8. The box within the violin plot outlined the interquartiles ($n = 100$).

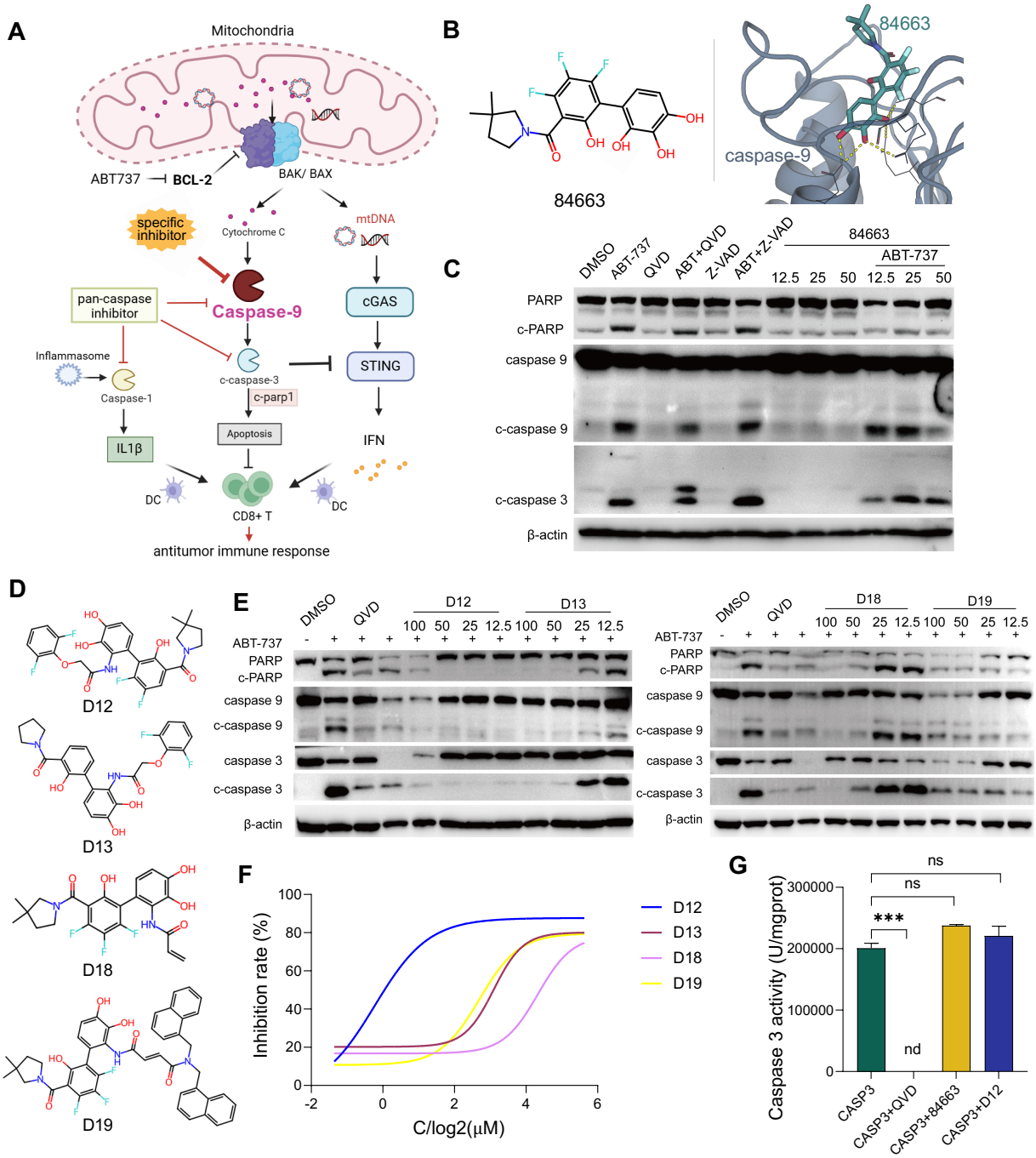
A
Mitochondria
ABT737
BCL-2
BAK/ BAX
specific inhibitor
Cytochrome C
mtDNA
pan-caspase inhibitor
Caspase-9
cGAS
Inflammasome
Caspase-1
c-caspase-3
STING
c-parp1
IL1β
Apoptosis
IFN
DC
CD8+ T
DC
antitumor immune response
B
84663
caspase-9
84663
C
DMSO
ABT-737
QVD
ABT+QVD
Z-VAD
ABT+Z-VAD
84663
ABT-737
12.5 25 50 12.5 25 50
PARP
c-PARP
caspase 9
c-caspase 9
c-caspase 3
β-actin
D
D12
D13
D18
D19
E
DMSO
QVD
D12
D13
100 50 25 12.5 100 50 25 12.5
ABT-737
PARP
c-PARP
caspase 9
c-caspase 9
caspase 3
c-caspase 3
β-actin
DMSO
QVD
D18
D19
100 50 25 12.5 100 50 25 12.5
ABT-737
PARP
c-PARP
caspase 9
c-caspase 9
caspase 3
c-caspase 3
β-actin
F
Inhibition rate (%)
C/log2(μM)
D12
D13
D18
D19
G
Caspase 3 activity (U/mgprot)
ns
ns
***
nd
CASP3
CASP3+QVD
CASP3+84663
CASP3+D12

Figure 4: **Caspase-9 inhibitor design.** (**A**) The diagram of the caspase-9 related pathway. (**B**) The designed molecule 84663 that was validated to bind with caspase-9 (left) and the generated binding pose by PocketXMol (right). The yellow dash is the polar contact identified by PyMOL. (**C**) Western blot analysis of the expression of PARP and caspase-9/3. MC38 cells were treated with QVD(10 μM), 84663 (12.5μM, 25 μM, and 50 μM) combined with ABT-737 (10 μM) or not. (**D**) The second-round molecules validated to bind with caspase-9 (D12, D13, D18, and D19). (**E**) Western blot analysis of the expression of PARP and caspase 9/3. MC38 cells were treated with different caspase inhibitors (50 μM, 12 μM) combined with ABT-737 or not. (**F**) Caspase-3 activity assay was done to determine the EC_{50} values of the molecules for suppressing the ABT-737 induced caspase-9 mediated caspase-3 activation. (**G**) Analysis of the activity of caspase-3 from the mix of caspase-3 protein (0.5ng/ml) and caspase inhibitors (0.0166 μM) by caspase-3 assay kit.

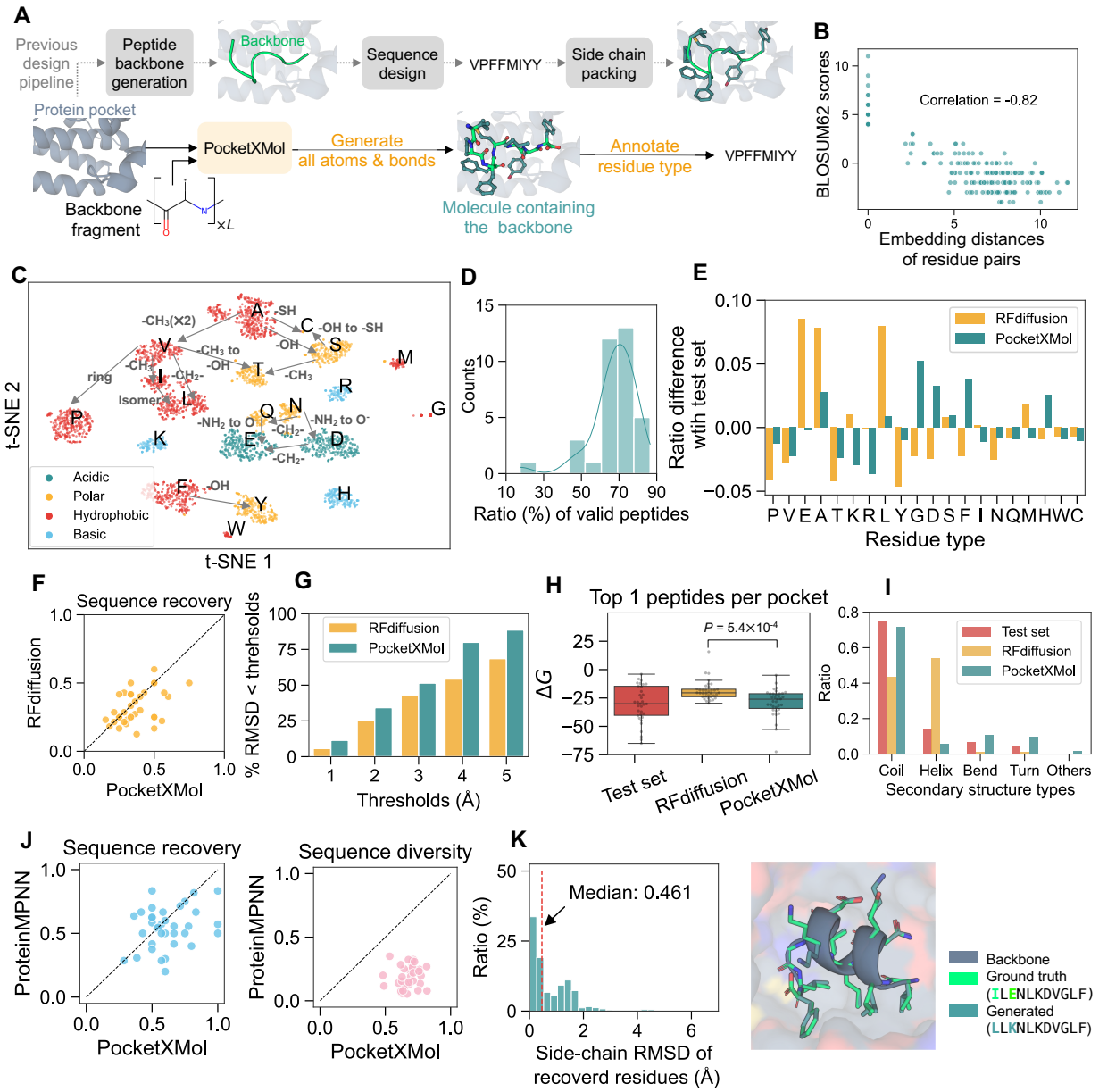
A
Previous design pipeline
Peptide backbone generation
Backbone
Sequence design
VPFFMIYY
Side chain packing
Protein pocket
PocketXMol
Generate all atoms & bonds
Annotate residue type
VPFFMIYY
Backbone fragment
×L
Molecule containing the backbone
B
BLOSUM62 scores
Correlation = -0.82
Embedding distances of residue pairs
C
t-SNE 2
t-SNE 1
Acidic
Polar
Hydrophobic
Basic
D
Counts
Ratio (%) of valid peptides
E
Ratio difference wtih test set
RFdiffusion
PocketXMol
P V E A T K R L Y G D S F I N Q M H W C
Residue type
F
Sequence recovery
RFdiffusion
PocketXMol
G
% RMSD < threhsolds
RFdiffusion
PocketXMol
Thresholds (Å)
H
Top 1 peptides per pocket
P = 5.4×10-4
ΔG
Test set
RFdiffusion
PocketXMol
I
Ratio
Test set
RFdiffusion
PocketXMol
Coil
Helix
Bend
Turn
Others
Secondary structure types
J
Sequence recovery
ProteinMPNN
PocketXMol
Sequence diversity
ProteinMPNN
PocketXMol
K
Ratio (%)
Median: 0.461
Side-chain RMSD of recoverd residues (Å)
Backbone
Ground truth
(ILENLKDVGLF)
Generated
(LLKNLKDVGLF)

Figure 5: **Performance on peptide design.** (**A**) Differences between PocketXMol and the previous peptide design pipeline. (**B**) Relationship between BLOSUM62 substitution scores and embedding distances for each pair of amino acids. (**C**) t-SNE visualization of the amino acid embeddings. The embeddings were calculated as the average of the side-chain atoms' embeddings at the last layer of the denoiser neural network (embeddings of glycine were set as all-zero vectors since their side chains do not contain heavy atoms). The differences of the side chains between similar amino acids were annotated. (**D**) Ratio distribution of valid peptides (i.e., peptides containing only standard amino acid types) generated by PocketXMol for each pocket in the test set. (**E**) Ratio differences of amino acid types between the generated peptides and the test set. (**F**) Comparison of the sequence recovery rates between the RFdiffusion pipeline and PocketXMol. For both methods, the peptides with the highest sequence recovery rates were selected for comparison among the 100 generated samples for each pocket. (**G**) Ratios of peptides with backbone atom RMSD less than thresholds for RFdiffusion and PocketXMol. The minimum backbone RMSD among 100 generated peptides for each pocket was selected for comparison. (**H**) Distributions of the binding affinity for the test set, RFdiffusion, and PocketXMol. The binding affinity was calculated using Rosetta, and the peptides with the lowest Rosetta energies for each pocket were selected for comparison (one-sided paired t-test with $n = 35$). (**I**) Ratio of secondary structures in the peptides from the test set, RFdiffusion and PocketXMol. (**J**) Sequence recovery rate and sequence diversity for ProteinMPNN and PocketXMol on the inverse folding task. For both methods, the peptides with the highest sequence recovery among the 100 generated samples per pocket were selected for comparison. The sequence diversity was calculated as *1-average sequence similarity* among the generated peptides for each pocket. (**K**) RMSD distribution of the side-chain atoms whose amino acid types were recovered by PocketXMol in inverse folding (left), and an example showing the structure and the sequence of the generated peptide (right).

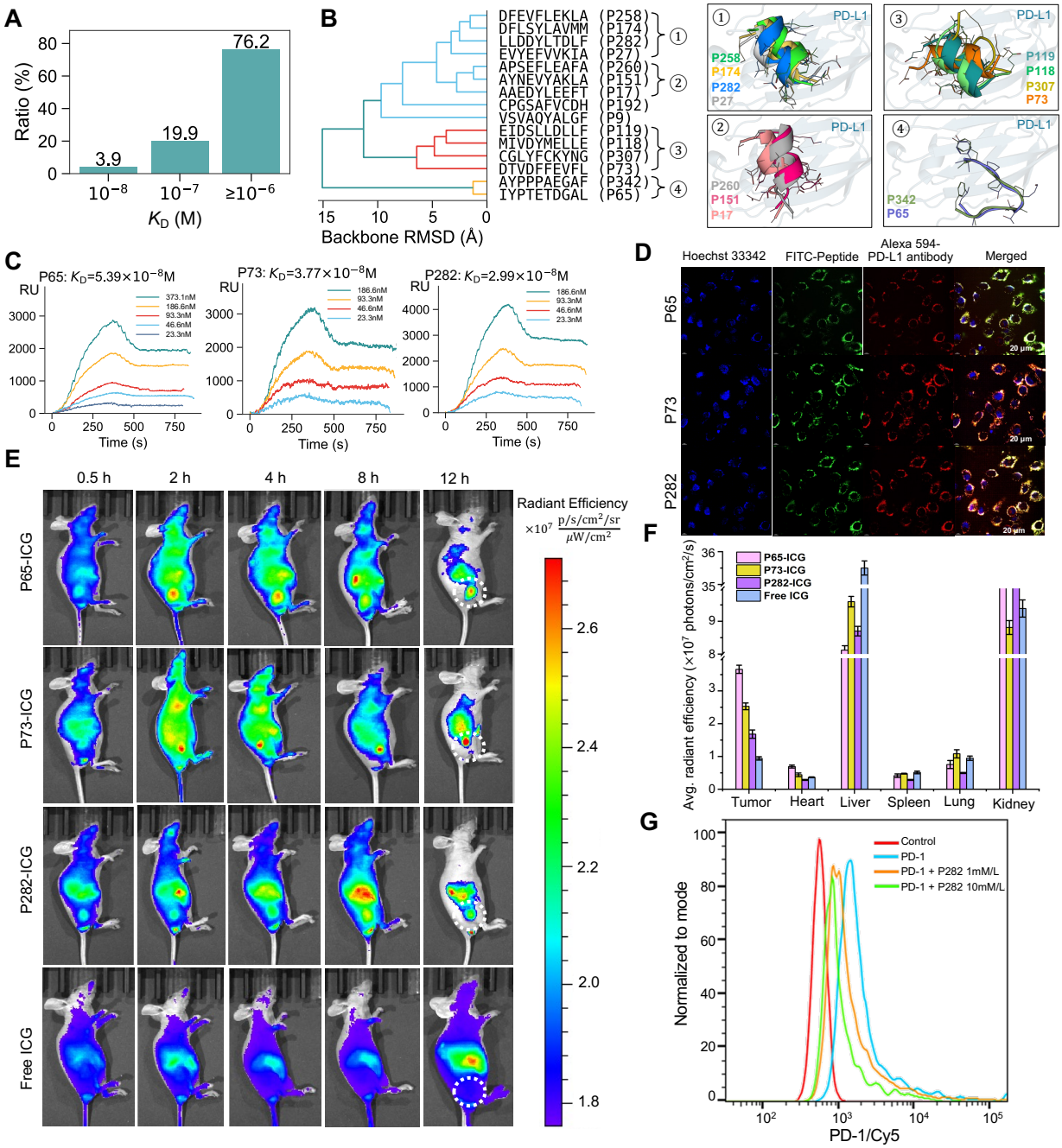
A
Ratio (%)
3.9
19.9
76.2
10^{-8}
10^{-7}
≥10^{-6}
K_D (M)
B
DFEVFLEKLA (P258)
DFLSYLAVMM (P174)
LLDDYLTDLF (P282)
EVYEFVVKIA (P27)
APSEFLEAFA (P260)
AYNEVYAKLA (P151)
AAEDYLEEFT (P17)
CPGSAFVCDH (P192)
VSVAQYALGF (P9)
EIDSLLDLLF (P119)
MIVDYMELLE (P118)
CGLYFCKYNG (P307)
DTVDFFEVFL (P73)
AYPPPAEGAF (P342)
IYPTETDGAL (P65)
Backbone RMSD (Å)
PD-L1
P258
P174
P282
P27
P119
P118
P307
P73
P260
P151
P17
P342
P65
C
P65: K_D=5.39×10^{-8}M
P73: K_D=3.77×10^{-8}M
P282: K_D=2.99×10^{-8}M
RU
Time (s)
D
Hoechst 33342
FITC-Peptide
Alexa 594-PD-L1 antibody
Merged
P65
P73
P282
20 μm
E
0.5 h
2 h
4 h
8 h
12 h
P65-ICG
P73-ICG
P282-ICG
Free ICG
Radiant Efficiency
$\times 10^7$ $\frac{p/s/cm^2/sr}{\mu W/cm^2}$
F
P65-ICG
P73-ICG
P282-ICG
Free ICG
Avg. radiant efficiency ($\times 10^7$ photons/cm²/s)
Tumor
Heart
Liver
Spleen
Lung
Kidney
G
Control
PD-1
PD-1 + P282 1mM/L
PD-1 + P282 10mM/L
Normalized to mode
PD-1/Cy5

Figure 6: **PD-L1 binding peptide design.** (**A**) Ratio of designed peptides with different disassociation constant K_D ranges with PD-L1. The disassociation constants were obtained using the SPRi experiments. (**B**) Structure clusters of the 15 generated peptides with K_D of 10^{-8}M, with the structures of peptides in four major clusters. The hierarchical clusters were constructed based on the backbone RMSDs among peptides and used the maximum distances between two sets as the linkage criterion. (**C**) The SPRi binding curves of the generated peptides selected for further experiments (P65, P73, and P282) with PD-L1. (**D**) Confocal image of FITC labeled candidate peptides binding toward the positive cell H1975. The negative cells 293T are shown in fig. S20. Scale bar was 20 μm. (**E**) In vivo fluorescence imaging was conducted on H1975 xenograft model mice at various time points after intravenous injection of P65-ICG, P73-ICG, P282-ICG, or free ICG, each administered at a dose of 50 μM. The tumors are indicated by white circles. (**F**) The average fluorescence signal intensity of isolated organs at 12 hours post-injection. (**G**) Flow cytometry analysis to investigate the PD-1/PD-L1 interaction using different concentrations of P282 peptide.

References and Notes

1. OpenAI, *et al.*, GPT-4 Technical Report (2023).

2. L. Ouyang, *et al.*, Training language models to follow instructions with human feedback (2022), doi:10.48550/ARXIV.2203.02155.

3. A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, M. Chen, Hierarchical Text-Conditional Image Generation with CLIP Latents (2022), doi:10.48550/ARXIV.2204.06125.

4. R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-Resolution Image Synthesis with Latent Diffusion Models (2021), doi:10.48550/ARXIV.2112.10752.

5. C. V. Theodoris, *et al.*, Transfer learning enables predictions in network biology. *Nature* **618** (7965), 616–624 (2023), doi:10.1038/s41586-023-06139-9.

6. M. Hao, *et al.*, Large Scale Foundation Model on Single-cell Transcriptomics. *bioRxiv* (2023), doi:10.1101/2023.05.29.542705, `https://www.biorxiv.org/content/early/2023/06/02/2023.05.29.542705`.

7. H. Cui, *et al.*, scGPT: toward building a foundation model for single-cell multi-omics using generative AI. *Nature Methods* (2024), doi:10.1038/s41592-024-02201-0, `http://dx.doi.org/10.1038/s41592-024-02201-0`.

8. J. Jumper, *et al.*, Highly accurate protein structure prediction with AlphaFold. *Nature* **596** (7873), 583–589 (2021), doi:10.1038/s41586-021-03819-2.

9. A. Radford, K. Narasimhan, Improving Language Understanding by Generative Pre-Training (2018), `https://api.semanticscholar.org/CorpusID:49313245`.

10. J. Abramson, *et al.*, Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature* (2024), doi:10.1038/s41586-024-07487-w, `http://dx.doi.org/10.1038/s41586-024-07487-w`.

11. R. Krishna, *et al.*, Generalized biomolecular modeling and design with RoseTTAFold All-Atom. *Science* **384** (6693) (2024), doi:10.1126/science.adl2528, `http://dx.doi.org/10.1126/science.adl2528`.

12. T. Hayes, *et al.*, Simulating 500 million years of evolution with a language model. *bioRxiv* (2024), doi:10.1101/2024.07.01.600583, `https://www.biorxiv.org/content/early/2024/07/02/2024.07.01.600583`.

13. H. Zhang, *et al.*, Delete: Deep Lead Optimization Enveloped in Protein Pocket through Unified Deleting Strategies and a Structure-aware Network (2023).

14. S. Luo, J. Guan, J. Ma, J. Peng, A 3D Generative Model for Structure-Based Drug Design, in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan, Eds. (Curran Associates, Inc.), vol. 34 (2021), pp. 6229–6239.

15. A. Schneuing, *et al.*, Structure-based Drug Design with Equivariant Diffusion Models. *arXiv preprint arXiv:2210.13695* (2022).

16. A. Madani, *et al.*, Large language models generate functional protein sequences across diverse families. *Nature Biotechnology* **41** (8), 1099–1106 (2023), doi:10.1038/s41587-022-01618-2, `http://dx.doi.org/10.1038/s41587-022-01618-2`.

17. N. Ferruz, S. Schmidt, B. Höcker, ProtGPT2 is a deep unsupervised language model for protein design. *Nature Communications* **13** (1) (2022), doi:10.1038/s41467-022-32007-7, `http://dx.doi.org/10.1038/s41467-022-32007-7`.

18. S. Liu, *et al.*, Chatgpt-powered conversational drug editing using retrieval and domain feedback. *arXiv preprint arXiv:2305.18090* (2023).

19. J. Ho, A. Jain, P. Abbeel, Denoising Diffusion Probabilistic Models, in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20 (Curran Associates Inc., Red Hook, NY, USA) (2020).

20. Y. Song, *et al.*, Score-Based Generative Modeling through Stochastic Differential Equations, in *International Conference on Learning Representations* (2021).

21. J. L. Watson, *et al.*, De novo design of protein structure and function with RFdiffusion. *Nature* **620** (7976), 1089–1100 (2023), doi:10.1038/s41586-023-06415-8.

22. G. Corso, H. Stärk, B. Jing, R. Barzilay, T. S. Jaakkola, DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking, in *The Eleventh International Conference on Learning Representations* (2023).

23. J. Guan, *et al.*, 3D Equivariant Diffusion for Target-Aware Molecule Generation and Affinity Prediction, in *The Eleventh International Conference on Learning Representations* (2023).

24. A. E. Chu, *et al.*, An all-atom protein generative model. *Proceedings of the National Academy of Sciences* **121** (27) (2024), doi:10.1073/pnas.2311500121, `http://dx.doi.org/10.1073/pnas.2311500121`.

25. H. Liu, *et al.*, One for all: Towards training one graph model for all classification tasks. *arXiv preprint arXiv:2310.00149* (2023).

26. G. Ye, *et al.*, Drugassist: A large language model for molecule optimization. *arXiv preprint arXiv:2401.10334* (2023).

27. Z. Lin, *et al.*, Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv* **2022**, 500902 (2022).

28. G. Zhou, *et al.*, Uni-Mol: A Universal 3D Molecular Representation Learning Framework, in *The Eleventh International Conference on Learning Representations* (2023).

29. M. Buttenschoen, G. M. Morris, C. M. Deane, PoseBusters: AI-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science* (2024), doi:10.1039/d3sc04185a.

30. O. Méndez-Lucio, M. Ahmad, E. A. del Rio-Chanona, J. K. Wegner, A geometric deep learning approach to predict binding conformations of bioactive molecules. *Nature Machine Intelligence* **3** (12), 1033–1039 (2021), doi:10.1038/s42256-021-00409-9, `http://dx.doi.org/10.1038/s42256-021-00409-9`.

31. H. Stärk, O. Ganea, L. Pattanaik, R. Barzilay, T. Jaakkola, Equibind: Geometric deep learning for drug binding structure prediction, in *International Conference on Machine Learning* (PMLR) (2022), pp. 20503–20521.

32. W. Lu, *et al.*, TANKBind: Trigonometry-Aware Neural NetworKs for Drug-Protein Binding Structure Prediction. *bioRxiv* (2022), doi:10.1101/2022.06.06.495043.

33. E. Alcaide, *et al.*, Uni-Mol Docking V2: Towards Realistic and Accurate Binding Pose Prediction. *arXiv preprint arXiv:2405.11769* (2024).

34. J. Eberhardt, D. Santos-Martins, A. F. Tillack, S. Forli, AutoDock Vina 1.2. 0: New docking methods, expanded force field, and python bindings. *Journal of Chemical Information and Modeling* **61** (8), 3891–3898 (2021).

35. M. L. Verdonk, J. C. Cole, M. J. Hartshorn, C. W. Murray, R. D. Taylor, Improved protein–ligand docking using GOLD. *Proteins: Structure, Function, and Bioinformatics* **52** (4), 609–623 (2003), doi:10.1002/prot.10465, `http://dx.doi.org/10.1002/prot.10465`.

36. D. Cao, *et al.*, SurfDock is a Surface-Informed Diffusion Generative Model for Reliable and Accurate Protein-ligand Complex Prediction. *bioRxiv* (2023), doi:10.1101/2023.12.13.571408, `https://www.biorxiv.org/content/early/2023/12/14/2023.12.13.571408`.

37. M. P. Ferla, *et al.*, Fragmenstein: predicting protein-ligand structures of compounds derived from known crystallographic fragment hits using a strict conserved-binding–based methodology. *ChemRxiv* (2024).

38. S. Luo, C. Shi, M. Xu, J. Tang, Predicting Molecular Conformation via Dynamic Graph Score Matching, in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan, Eds. (Curran Associates, Inc.), vol. 34 (2021), pp. 19784–19795.

39. C. Shi, S. Luo, M. Xu, J. Tang, Learning Gradient Fields for Molecular Conformation Generation, in *International Conference on Machine Learning* (2021).

40. Z. Qiao, W. Nie, A. Vahdat, T. F. Miller III, A. Anandkumar, State-specific protein–ligand complex structure prediction with a multiscale deep generative model. *Nature Machine Intelligence* pp. 1–14 (2024).

41. I. Johansson-Åkhe, B. Wallner, Improving peptide-protein docking with AlphaFold-Multimer using forced sampling. *Frontiers in Bioinformatics* **2** (2022), doi:10.3389/fbinf.2022.959160.

42. R. Evans, *et al.*, Protein complex prediction with AlphaFold-Multimer. *bioRxiv* (2021), doi:10.1101/2021.10.04.463034, `https://www.biorxiv.org/content/early/2021/10/04/2021.10.04.463034`.

43. T. Tsaban, *et al.*, Harnessing protein folding neural networks for peptide–protein docking. *Nature communications* **13** (1), 176 (2022).

44. Y. Lei, *et al.*, A deep-learning framework for multi-level peptide–protein interaction prediction. *Nature Communications* **12** (1) (2021), doi:10.1038/s41467-021-25772-4, `http://dx.doi.org/10.1038/s41467-021-25772-4`.

45. S. Basu, B. Wallner, DockQ: A Quality Measure for Protein-Protein Docking Models. *PLOS ONE* **11** (8), e0161879 (2016), doi:10.1371/journal.pone.0161879.

46. C. Zhang, X. Zhang, P. L. Freddolino, Y. Zhang, BioLiP2: an updated structure database for biologically relevant ligand–protein interactions. *Nucleic Acids Research* **52** (D1), D404–D412 (2023), doi:10.1093/nar/gkad630, `http://dx.doi.org/10.1093/nar/gkad630`.

47. H. Wei, W. Wang, Z. Peng, J. Yang, Q-BioLiP: A Comprehensive Resource for Quaternary Structure-based Protein–ligand Interactions. *Genomics, Proteomics and Bioinformatics* **22** (1) (2024), doi:10.1093/gpbjnl/qzae001, `http://dx.doi.org/10.1093/gpbjnl/qzae001`.

48. L. Wang, *et al.*, Therapeutic peptides: current applications and future directions. *Signal Transduction and Targeted Therapy* **7** (1) (2022), doi:10.1038/s41392-022-00904-4, `http://dx.doi.org/10.1038/s41392-022-00904-4`.

49. A. Ben-Shimon, M. Niv, AnchorDock: Blind and Flexible Anchor-Driven Peptide Docking. *Structure* **23** (5), 929–940 (2015), doi:10.1016/j.str.2015.03.010, `http://dx.doi.org/10.1016/j.str.2015.03.010`.

50. A. Ben-Shimon, M. Eisenstein, Computational Mapping of Anchoring Spots on Protein Surfaces. *Journal of Molecular Biology* **402** (1), 259–277 (2010), doi:10.1016/j.jmb.2010.07.021, `http://dx.doi.org/10.1016/j.jmb.2010.07.021`.

51. A. Ben-Shimon, M. Y. Niv, Deciphering the Arginine-Binding Preferences at the Substrate-Binding Groove of Ser/Thr Kinases by Computational Surface Mapping. *PLoS Computational Biology* **7** (11), e1002288 (2011), doi:10.1371/journal.pcbi.1002288, `http://dx.doi.org/10.1371/journal.pcbi.1002288`.

52. B. Abanades, *et al.*, ImmuneBuilder: Deep-Learning models for predicting the structures of immune proteins. *Communications Biology* **6** (1) (2023), doi:10.1038/s42003-023-04927-7, `http://dx.doi.org/10.1038/s42003-023-04927-7`.

53. Z. Zhang, J. Yan, Q. Liu, E. Chen, M. Zitnik, A Systematic Survey in Geometric Deep Learning for Structure-based Drug Design (2023), doi:10.48550/ARXIV.2306.11768.

54. L. Shen, *et al.*, Pocket Crafter: a 3D generative modeling based workflow for the rapid generation of hit molecules in drug discovery. *Journal of Cheminformatics* **16** (1) (2024), doi:10.1186/s13321-024-00829-w, `http://dx.doi.org/10.1186/s13321-024-00829-w`.

55. X. Peng, *et al.*, Pocket2Mol: Efficient Molecular Sampling Based on 3D Protein Pockets, in *International Conference on Machine Learning* (2022).

56. T. Masuda, M. Ragoza, D. R. Koes, Generating 3d molecular structures conditional on a receptor binding site with deep generative models. *arXiv preprint arXiv:2010.14442* (2020).

57. J. Guan, *et al.*, DecompDiff: Diffusion Models with Decomposed Priors for Structure-Based Drug Design, in *Proceedings of the 40th International Conference on Machine Learning*, A. Krause, *et al.*, Eds. (PMLR), vol. 202 of *Proceedings of Machine Learning Research* (2023), pp. 11827–11846.

58. P. G. Francoeur, *et al.*, Three-dimensional convolutional neural networks and a cross-docked data set for structure-based drug design. *Journal of Chemical Information and Modeling* **60** (9), 4200–4215 (2020).

59. C. Harris, *et al.*, Benchmarking Generated Poses: How Rational is Structure-based Drug Design with Generative Models? (2023), doi:10.48550/ARXIV.2308.07413, `https://arxiv.org/abs/2308.07413`.

60. E. Hoogeboom, V. G. Satorras, C. Vignac, M. Welling, Equivariant Diffusion for Molecule Generation in 3D (2022), doi:10.48550/ARXIV.2203.17003.

61. X. Peng, J. Guan, Q. Liu, J. Ma, MolDiff: Addressing the Atom-Bond Inconsistency Problem in 3D Molecule Diffusion Generation, in *Proceedings of the 40th International Conference on Machine Learning*, A. Krause, *et al.*, Eds. (PMLR), vol. 202 of *Proceedings of Machine Learning Research* (2023), pp. 27611–27629.

62. M. Xu, A. S. Powers, R. O. Dror, S. Ermon, J. Leskovec, Geometric latent diffusion models for 3d molecule generation, in *International Conference on Machine Learning* (PMLR) (2023), pp. 38592–38610.

63. H. Huang, L. Sun, B. Du, W. Lv, Learning Joint 2-D and 3-D Graph Diffusion Models for Complete Molecule Generation. *IEEE Transactions on Neural Networks and Learning Systems* **35** (9), 11857–11871 (2024), doi:10.1109/tnnls.2024.3416328, `http://dx.doi.org/10.1109/TNNLS.2024.3416328`.

64. A. Morehead, J. Cheng, Geometry-complete diffusion for 3D molecule generation and optimization. *Communications Chemistry* **7** (1) (2024), doi:10.1038/s42004-024-01233-z, `http://dx.doi.org/10.1038/s42004-024-01233-z`.

65. J. B. Ingraham, *et al.*, Illuminating protein space with a programmable generative model. *Nature* **623** (7989), 1070–1078 (2023), doi:10.1038/s41586-023-06728-8.

66. P. Kirsch, *et al.*, Fragment-Based Discovery of a Qualified Hit Targeting the Latency-Associated Nuclear Antigen of the Oncogenic Kaposi's Sarcoma-Associated Herpesvirus/Human Herpesvirus 8. *Journal of Medicinal Chemistry* **62** (8), 3924–3939 (2019), doi:10.1021/acs.jmedchem.8b01827, `http://dx.doi.org/10.1021/acs.jmedchem.8b01827`.

67. F. Imrie, A. R. Bradley, M. van der Schaar, C. M. Deane, Deep Generative Models for 3D Linker Design. *Journal of Chemical Information and Modeling* **60** (4), 1983–1995 (2020), doi:10.1021/acs.jcim.9b01120.

68. M. Békés, D. R. Langley, C. M. Crews, PROTAC targeted protein degraders: the past is prologue. *Nature Reviews Drug Discovery* **21** (3), 181–200 (2022), doi:10.1038/s41573-021-00371-6, http://dx.doi.org/10.1038/s41573-021-00371-6.

69. Y. Huang, X. Peng, J. Ma, M. Zhang, 3DLinker: An E(3) Equivariant Variational Autoencoder for Molecular Linker Design (2022), doi:10.48550/ARXIV.2205.07309.

70. I. Igashov, *et al.*, Equivariant 3d-conditional diffusion models for molecular linker design. *arXiv preprint arXiv:2210.05274* (2022).

71. M. L. Benson, *et al.*, Binding MOAD, a high-quality protein ligand database. *Nucleic Acids Research* **36** (Database), D674–D678 (2007), doi:10.1093/nar/gkm911, http://dx.doi.org/10.1093/nar/gkm911.

72. G. Landrum, *et al.*, rdkit/rdkit: 2024_03_1 (Q1 2024) Release (2024), doi:10.5281/ZENODO.11102446, https://zenodo.org/doi/10.5281/zenodo.11102446.

73. J. Guan, *et al.*, LinkerNet: Fragment Poses and Linker Co-Design with 3D Equivariant Diffusion, in *Thirty-seventh Conference on Neural Information Processing Systems* (2023).

74. G. Weng, *et al.*, PROTAC-DB 2.0: an updated database of PROTACs. *Nucleic Acids Research* **51** (D1), D1367–D1372 (2022), doi:10.1093/nar/gkac946, http://dx.doi.org/10.1093/nar/gkac946.

75. J. Degen, C. Wegscheid-Gerlach, A. Zaliani, M. Rarey, On the Art of Compiling and Using "Drug-Like" Chemical Fragment Spaces. *ChemMedChem* **3** (10), 1503–1507 (2008), doi:10.1002/cmdc.200800178.

76. A. A. Petrauskas, E. A. Kolovanov, ACD/Log P method description. *Perspectives in drug discovery and design* **19**, 99–116 (2000).

77. C. Han, *et al.*, Tumor cells suppress radiation-induced immunity by hijacking caspase 9 signaling. *Nature Immunology* **21** (5), 546–554 (2020), doi:10.1038/s41590-020-0641-5, http://dx.doi.org/10.1038/s41590-020-0641-5.

78. C. Han, *et al.*, The AIM2 and NLRP3 inflammasomes trigger IL-1–mediated antitumor effects during radiation. *Science Immunology* **6** (59) (2021), doi:10.1126/sciimmunol.abc6998, http://dx.doi.org/10.1126/sciimmunol.abc6998.

79. M. Segovia, *et al.*, Targeting TMEM176B Enhances Antitumor Immunity and Augments the Efficacy of Immune Checkpoint Blockers by Unleashing Inflammasome Activation. *Cancer Cell* **35** (5), 767–781.e6 (2019), doi:10.1016/j.ccell.2019.04.003, http://dx.doi.org/10.1016/j.ccell.2019.04.003.

80. J. J. Irwin, *et al.*, ZINC20—A Free Ultralarge-Scale Chemical Database for Ligand Discovery. *Journal of Chemical Information and Modeling* **60** (12), 6065–6073 (2020), doi:10.1021/acs.jcim.0c00675, http://dx.doi.org/10.1021/acs.jcim.0c00675.

81. A. Gaulton, *et al.*, ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic acids research* **40** (D1), D1100–D1107 (2012).

82. S. Kim, *et al.*, PubChem 2019 update: improved access to chemical data. *Nucleic acids research* **47** (D1), D1102–D1109 (2019).

83. Z. Wen, J. He, H. Tao, S.-Y. Huang, PepBDB: a comprehensive structural database of biological peptide–protein interactions. *Bioinformatics* **35** (1), 175–177 (2018), doi:10.1093/bioinformatics/bty579, http://dx.doi.org/10.1093/bioinformatics/bty579.

84. J. Dauparas, *et al.*, Robust deep learning–based protein sequence design using ProteinMPNN. *Science* **378** (6615), 49–56 (2022), doi:10.1126/science.add2187.

85. R. F. Alford, *et al.*, The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. *Journal of Chemical Theory and Computation* **13** (6), 3031–3048 (2017), doi:10.1021/acs.jctc.7b00125, http://dx.doi.org/10.1021/acs.jctc.7b00125.

86. J. Schymkowitz, *et al.*, The FoldX web server: an online force field. *Nucleic Acids Research* **33** (Web Server), W382–W388 (2005), doi:10.1093/nar/gki387, http://dx.doi.org/10.1093/nar/gki387.

87. M. Jendrusch, J. O. Korbel, S. K. Sadiq, AlphaDesign: A de novo protein design framework based on AlphaFold. *bioRxiv* (2021), doi:10.1101/2021.10.11.463937, `https://www.biorxiv.org/content/early/2021/10/12/2021.10.11.463937`.

88. Z. Liu, *et al.*, PDB-wide collection of binding data: current status of the PDBbind database. *Bioinformatics* **31** (3), 405–412 (2015).

89. M. Varadi, *et al.*, AlphaFold Protein Structure Database in 2024: providing structure coverage for over 214 million protein sequences. *Nucleic Acids Research* **52** (D1), D368–D375 (2023), doi:10.1093/nar/gkad1011, `http://dx.doi.org/10.1093/nar/gkad1011`.

90. S. Axelrod, R. Gómez-Bombarelli, GEOM, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data* **9** (1), 185 (2022), doi:10.1038/s41597-022-01288-4.

91. C. A. Grambow, H. Weir, C. N. Cunningham, T. Biancalani, K. V. Chuang, CREMP: Conformer-Rotamer Ensembles of Macrocyclic Peptides for Machine Learning. *arXiv preprint arXiv:2305.08057* (2023).

92. A. Leach, S. M. Schmon, M. T. Degiacomi, C. G. Willcocks, Denoising Diffusion Probabilistic Models on SO(3) for Rotational Alignment, in *ICLR 2022 Workshop on Geometrical and Topological Representation Learning* (2022).

93. I. Loshchilov, F. Hutter, Decoupled Weight Decay Regularization, in *International Conference on Learning Representations* (2019), `https://openreview.net/forum?id=Bkg6RiCqY7`.

94. Z. Wang, *et al.*, Microarray based screening of peptide nano probes for HER2 positive tumor. *Analytical chemistry* **87** (16), 8367–8372 (2015).

95. S. Luo, *et al.*, Antigen-Specific Antibody Design and Optimization with Diffusion-Based Generative Models. *bioRxiv* (2022), doi:10.1101/2022.07.10.499510.

96. K. V. Mardia, P. E. Jupp, *Directional statistics* (John Wiley & Sons) (2009).

97. N. M. O'Boyle, *et al.*, Open Babel: An open chemical toolbox. *Journal of cheminformatics* **3** (1), 1–14 (2011).

98. A. G. Dossetter, E. J. Griffen, A. G. Leach, Matched molecular pair analysis in drug discovery. *Drug Discovery Today* **18** (15-16), 724–731 (2013).

99. M. Baek, *et al.*, Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **373** (6557), 871–876 (2021), doi:10.1126/science.abj8754.

100. M. Mirdita, M. Steinegger, J. Söding, MMseqs2 desktop and local web server app for fast, interactive sequence searches. *Bioinformatics* **35** (16), 2856–2858 (2019), doi:10.1093/bioinformatics/bty1057, `http://dx.doi.org/10.1093/bioinformatics/bty1057`.

101. P. J. A. Cock, *et al.*, Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **25** (11), 1422–1423 (2009), doi:10.1093/bioinformatics/btp163, `http://dx.doi.org/10.1093/bioinformatics/btp163`.

102. H. Berman, K. Henrick, H. Nakamura, Announcing the worldwide protein data bank. *Nature structural & molecular biology* **10** (12), 980–980 (2003).

103. P. Eastman, *et al.*, OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLOS Computational Biology* **13** (7), e1005659 (2017), doi:10.1371/journal.pcbi.1005659, `http://dx.doi.org/10.1371/journal.pcbi.1005659`.

104. B. Raveh, N. London, O. Schueler-Furman, Sub-angstrom modeling of complexes between flexible peptides and globular proteins. *Proteins: Structure, Function, and Bioinformatics* **78** (9), 2029–2040 (2010).

105. P. B. Stranges, B. Kuhlman, A comparison of successful and failed protein interface designs highlights the challenges of designing buried hydrogen bonds. *Protein Science* **22** (1), 74–82 (2013).

106. S. Salentin, S. Schreiber, V. J. Haupt, M. F. Adasme, M. Schroeder, PLIP: fully automated protein–ligand interaction profiler. *Nucleic Acids Research* **43** (W1), W443–W447 (2015), doi:10.1093/nar/gkv315, `http://dx.doi.org/10.1093/nar/gkv315`.

Acknowledgments

The authors express gratitude to Shitong Luo, Xingchao Liu, and Qiang Liu for their insightful discussions, and to Fanxu Meng for inspiring this work.

Author contributions: X.P. and J.M. conceived and designed the generative model with contributions from R.G. and J.G. X.P., R.G., J.G., and Y.J. processed the data. X.P., R.G., and J.G. conducted computational experiments. Y.X., Y.H, J.S, C.H, and Z.W designed and conducted the wet-lab experiments. X.P., Y.X, Y.H, M.Z., C.H, Z.W., C.H., and J.M. analyzed the results. X.P., C.H., Z.W., and J.M. wrote the manuscript with feedback and contributions from all authors.

Data and materials availability: The raw data are available in public databases. To train the model, we compiled a large dataset containing 85,134 3D structures of protein-small-molecule complexes, 39,741 protein-peptide complexes, 11,947,602 small molecules, and 35,698 short peptides. Protein-small-molecule structure datasets include PDBBind (*88*), Binding MOAD (*71*), and CrossDocked2020 (*58*) datasets. Protein-peptide 3D complex structures were collected from the PepBDB database (*83*). We also constructed another in-house synthetic dataset by searching the protein-binding loop structures from protein complexes in AlphaFold DB (*89*). For small molecule or peptide structures, we leveraged the GEOM small molecule dataset (*90*), CREMP macrocyclic peptide dataset (*91*), and large molecular data used by Uni-Mol (*28*), which contained small molecules from ZINC (*80*), ChEMBL (*81*) and other purchasable molecular databases. Based on the raw databases, we filtered data and avoided data leakage as described in supplementary text section 6.2.

Supplementary materials

Materials and Methods

Supplementary Text

Figs. S1 to S22

Tables S1 to S6