nature machine intelligence

Article

https://doi.org/10.1038/s42256-024-00920-9

Efficient generation of protein pockets with PocketGen

Received: 19 February 2024

Accepted: 2 October 2024

Published online: 15 November 2024

Check for updates

Zaixi Zhang[1,2,3], Wan Xiang Shen[3], Qi Liu[1,2] ✉ & Marinka Zitnik[3,4,5,6] ✉

Designing protein-binding proteins is critical for drug discovery. However, artificial-intelligence-based design of such proteins is challenging due to the complexity of protein–ligand interactions, the flexibility of ligand molecules and amino acid side chains, and sequence–structure dependencies. We introduce PocketGen, a deep generative model that produces residue sequence and atomic structure of the protein regions in which ligand interactions occur. PocketGen promotes consistency between protein sequence and structure by using a graph transformer for structural encoding and a sequence refinement module based on a protein language model. The graph transformer captures interactions at multiple scales, including atom, residue and ligand levels. For sequence refinement, PocketGen integrates a structural adapter into the protein language model, ensuring that structure-based predictions align with sequence-based predictions. PocketGen can generate high-fidelity protein pockets with enhanced binding affinity and structural validity. It operates ten times faster than physics-based methods and achieves a 97% success rate, defined as the percentage of generated pockets with higher binding affinity than reference pockets. Additionally, it attains an amino acid recovery rate exceeding 63%.

Modulation of protein functions often involves modelling the interactions between proteins and small-molecule ligands[1–4]. These interactions are central to biological processes such as enzymatic catalysis, signal transduction and cellular regulation. Binding small molecules to specific protein sites can induce conformational changes, modulate protein activity and alter existing or produce new functional properties. This mechanism is invaluable for designing proteins with tailored small-molecule binders. Applications range from engineering enzymes and catalyse reactions in the absence of natural catalysts[5–8] to creating biosensors for detecting environmental compounds. Such biosensors are critical for environmental monitoring, clinical diagnostics, pathogen detection, drug delivery systems and food industry applications[9–12]. Typically, designs involve modifying existing ligand-binding pockets to enable more specific interactions with target ligands[13–15]. Nevertheless, challenges persist in computationally generating high-validity ligand-binding protein pockets due to the complexity of protein–ligand interactions, the flexibility of ligands and amino acid side chains, and the dependencies between sequence and structure[3,15,16].

Methods for pocket design have traditionally relied on physics-based modelling or template matching[10,11,13,17,18]. For example, PocketOptimizer[18–20] uses a pipeline that predicts mutations in protein pockets to enhance binding affinity, based on physics-based energy functions and search algorithms. Starting with a bound protein–ligand complex, PocketOptimizer explores possible side-chain structures and residue types, evaluating these mutations with energy functions and ranking them using integer linear programming techniques. Another widely used approach involves template matching and enumeration methods[11,13,14,17,21]. For instance, a two-step strategy[13] has been used for pocket design. First, they identify and assemble disconnected protein motifs (van der Mer structural units) around the target molecule to

[1]State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China, Hefei, China. [2]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, China. [3]Department of Biomedical Informatics, Harvard Medical School, Boston, MA, USA. [4]Kempner Institute for the Study of Natural and Artificial Intelligence, Harvard University, Cambridge, MA, USA. [5]Broad Institute of MIT and Harvard, Cambridge, MA, USA. [6]Harvard Data Science Initiative, Cambridge, MA, USA. ✉e-mail: qiliuql@ustc.edu.cn; marinka@hms.harvard.edu

https://doi.org/10.1038/s42256-024-00920-9

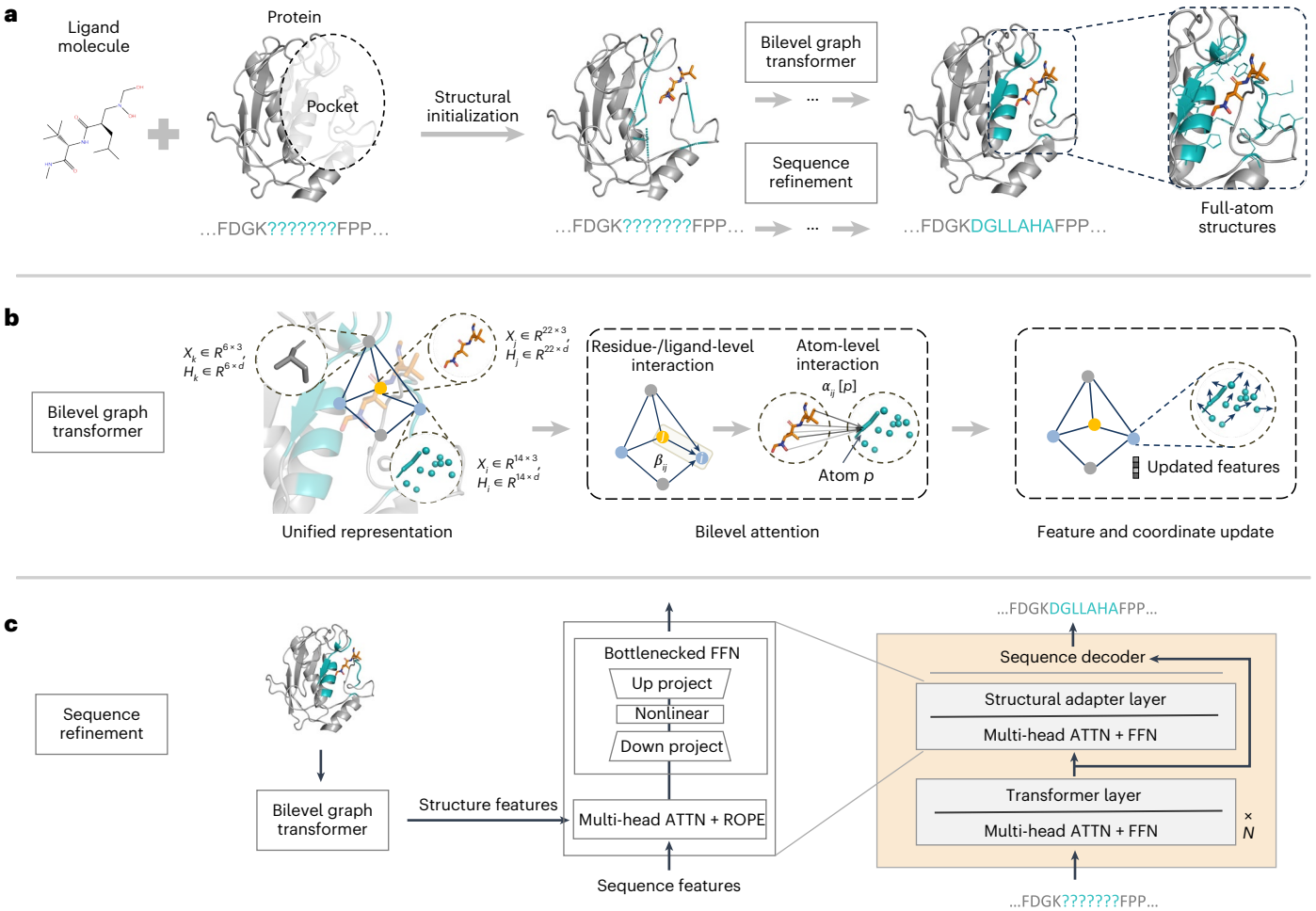


Fig. 1 | Overview of PocketGen generative model for the design of full-atom ligand-binding protein pockets. a, Conditioned on the binding ligand molecule and the remaining part of the protein except the pocket region (that is, scaffold), PocketGen aims to generate the full-atom pocket structure (backbone and side-chain atoms) and the residue-type sequence with iterative equivariant refinement. The ligand structure is also adjusted during protein pocket refinement. **b**, Bilevel graph transformer is leveraged in PocketGen for all-atom structural encoding and update. Bilevel attention captures both residue/ligand and atom-level interactions. Both protein pocket structure and ligand molecule structure are updated in the refinement. **c**, Sequence refinement module adds lightweight structural adapter layers into pLMs for sequence prediction. Only the adapter's parameters are fine-tuned during training, and the other layers are fixed. In the adapter, cross-attention between sequence and structure features is performed to achieve sequence–structure consistency.

form protein–ligand hydrogen bonds. Then, they graft these residues onto a protein scaffold and select the optimal protein–ligand pairs using scoring functions. This template-matching strategy enabled the de novo design of proteins binding the drug apixaban (APX)[22]. However, physics-based and template-matching methods can be time-consuming, often requiring several hours to design a single protein pocket. Furthermore, the focus on specific fold types, such as four-helix bundles[13] or NTF2 folds[14], can limit the broader applicability of these methods.

Recent advances in protein pocket design have been propelled by deep-learning-based approaches[3,8,16,23–25]. For instance, RFdiffusion[26] leverages denoising diffusion probabilistic models[27] alongside RoseTTAFold[28] for de novo protein structure generation. Although it can design pockets for specific ligands, RFdiffusion lacks precision in modelling protein–ligand interactions due to its auxiliary guiding potentials. To address this limitation, RFdiffusion All-Atom (RFAA)[16] extends the approach by enabling the direct generation of binding proteins around small molecules through iterative denoising. This is achieved through architectural modifications that simultaneously consider both protein structures and ligand molecules. However, in both RFdiffusion and RFAA, residue sequences are derived in post-processing using ProteinMPNN[29] or LigandMPNN[30], which can result in inconsistencies between the sequence and structure modalities. In contrast, full-atom iterative refinement (FAIR)[24] simultaneously designs the atomic pocket structure and the corresponding sequence using a two-stage refinement approach. FAIR employs a coarse-to-fine method, initially refining the backbone protein structure and subsequently refining the atomic structure, including the side chains. This iterative process continues until convergence is reached. However, the gap between these two refinement stages can introduce instability and limit performance, underscoring the need for an end-to-end generative approach to pocket design. Related research has explored the co-design of sequence and structure in complementarity-determining regions of antibodies[31–35]. Although these methods are effective for antibody design, they encounter difficulties when applied to pocket designs conditioned on the target ligand molecules.

Hybrid approaches that combine deep learning models with traditional methods are also being actively explored[3,8]. For example, a luciferase[8] was developed by integrating protein hallucination[36], trRosetta structure prediction neural network[37], hydrogen-bonding networks and RifDock[38]. This combination generated a range of idealized protein structures with diverse pocket shapes for subsequent

filtering. Although successful, this approach applies only to specific protein scaffolds and substrates and lacks a generalized solution. Similarly, deep learning was merged with physics-based methods[3] to design proteins featuring diverse and customizable pocket geometries. Their method utilizes backbone generation via trRosetta hallucination, sequence design through ProteinMPNN[29] and LigandMPNN[30], and filtering with AlphaFold[39]. Despite the advances made, pocket generation models continue to face challenges, such as achieving sequence–structure consistency and accurately modelling complex protein–ligand interactions.

Here we introduce PocketGen, a deep generative method designed for the efficient generation of protein pockets. PocketGen employs a co-design scheme (Fig. 1a) in which the model predicts both sequence and structure of the protein pocket based on the ligand molecule and the surrounding protein scaffold (excluding the pocket itself). The architecture of PocketGen is composed of two key modules: the bilevel graph transformer (Fig. 1b) and the sequence refinement module (Fig. 1c). PocketGen represents the protein–ligand complex as a geometric graph of blocks, allowing it to handle varying numbers of atoms across residues and ligands. Initialized pocket residues are assigned the maximum possible number of atoms (14 atoms) to accommodate this variability, and these atoms are mapped back to specific residue types during the generation process.

The graph transformer module uses a bilevel attention mechanism to capture interactions at multiple granularities—both at the atom and residue/ligand levels—and across various aspects, including intraprotein and protein–ligand interactions. To account for the redesigned pocket's influence on the ligand, the ligand structure is updated during the refinement process to reflect potential changes in the binding pose. To ensure consistency between the protein sequence and structure domains and to incorporate evolutionary information encoded in protein language models (pLMs)[40,41], PocketGen integrates a structural adapter into the sequence update process. This adapter enables cross-attention between the sequence and structure features, ensuring sequence–structure alignment. Only the adapter is fine-tuned during training, whereas the remaining layers of the pLM remain unchanged. PocketGen outperforms methods for protein pocket generation on two benchmarks. It achieves an average amino acid recovery rate (AAR) of 63.40% and a Vina score of −9.655 for the top-1-ranked generated protein pockets on the CrossDocked dataset. Comprehensive analyses show that PocketGen can generate diverse, high-affinity protein pockets for functional molecules, highlighting its potential for informing the design of small-molecule binders.

Results

Benchmarking generated protein pockets

We benchmark PocketGen on two datasets. The CrossDocked dataset[42] consists of protein–molecule pairs generated through cross-docking and is divided into training, validation and test sets based on a 30% sequence identity threshold. The Binding MOAD dataset[43] contains experimentally determined protein–ligand complexes, which are split into training, validation and test sets according to the proteins' enzyme commission numbers[44]. In line with intermolecular distance scales relevant to protein–ligand interactions[45], our default experimental setup includes all the residues with atoms within 3.5 Å of any ligand-binding atoms, averaging about eight residues per pocket. We also explore PocketGen's ability to design larger pockets with a radius of 5.5 Å, incorporating more residues (Fig. 3c).

We use three groups of metrics to evaluate the quality of protein pockets generated by PocketGen. First, we assess the affinity between the generated pocket and the target ligand molecule using the AutoDock Vina score[46], MM-GBSA[47] and min-in-place GlideSP score[48]. Second, we evaluate the structural validity of the generated pockets using self-consistent root mean squared deviation (scRMSD), self-consistent template modelling score (scTM) and predicted local-distance difference test (pLDDT). The amino acid sequence for the protein pocket structure is derived using ProteinMPNN[29], and the pocket structure is predicted using ESMFold[49] or AlphaFold 2 (ref. 39). The scRMSD is calculated between the generated structure's backbone atoms and the predicted structure. Following an established strategy[50,51], eight sequences are predicted for each generated protein structure, and the sequence with the lowest scRMSD is used for reporting. Similarly, scTM is calculated by comparing the template modelling score[52] between the predicted and generated structures. Scores range from 0 to 1, with higher values indicating greater designability. We also report the ΔscTM score to assess whether the generated pocket improves or degrades the scTM score of the initial protein. The pLDDT score[39] reflects the confidence in structural predictions on a scale from 0 to 100, with higher scores indicating greater confidence. The average pLDDT score across pocket residues is reported. A generated protein pocket is defined as designable if the overall structure's scRMSD is less than 2 Å and the pocket's scRMSD is less than 1 Å (refs. 26,53,54). Supplementary Table 1 presents the percentage of designable generated pockets, and Supplementary Fig. 1 describes how these metrics are calculated. Finally, we report the AAR as the percentage of correctly predicted pocket residue types, which reflects the accuracy of the designed sequence. A higher AAR indicates better modelling of sequence–structure dependencies.

We compare PocketGen against six methods, including deep-learning-based approaches such as RFdiffusion[26], RFAA[16], FAIR[24] and dynamic multichannel equivariant graph network (dyMEAN)[25], as well as a template-matching method called Design Pocket as a Cluster based on Templates (DEPACT)[17] and a physics-based modelling method called PocketOpt[18] (Methods). In Fig. 2 and Supplementary Table 1, PocketGen and the other methods are tasked with generating 100 sequences and structures for each protein–ligand complex in the test sets of the CrossDocked and Binding MOAD datasets. PocketOpt is excluded from this comparison due to its focus on mutating existing pockets for optimization, making it too time-consuming to generate many protein pockets. Supplementary Table 1 presents the mean and standard deviation of the results across three independent runs with different random seeds. In Fig. 2, we apply bootstrapping to the generated results, illustrating the distributions to demonstrate the sensitivity of the results to the dataset composition[55]. As shown in Supplementary Table 1 and Supplementary Fig. 2, PocketGen outperforms all the baselines, including RFdiffusion and RFAA, in terms of designability (by 3% and 2% on CrossDocked, respectively) and Vina scores (by 0.199 and 0.123 on CrossDocked, respectively). This performance indicates PocketGen's effectiveness in generating structurally valid pockets with high binding affinities, a result attributed to PocketGen's ability to capture interactions at multiple granularities—both atom level and residue/ligand level—and across various aspects including intraprotein and protein–ligand interactions.

PocketGen substantially outperforms the best-performing alternative method, RFAA, with an average improvement of 13.95% in AAR, largely due to the inclusion of the pLM that captures evolutionary sequence information. In contrast, RFdiffusion and RFAA rely on post-processing to determine the amino acid types, which can lead to inconsistencies between sequence and structure and lower performance in AAR. In protein engineering, the common practice is to mutate several key residues to optimize properties and keeping most residues unchanged to preserve protein-folding stability[56,57]. The high AAR achieved by the generated protein pockets with PocketGen aligns well with this practice, supporting its utility for stable and effective protein design.

In Table 1, the top 1, 3, 5 and 10 protein pockets generated by PocketGen (ranked by Vina score) consistently show the lowest Vina scores, achieving an average reduction of 0.476 compared with RFAA. In addition to Vina scores, two other affinity metrics—MM-GBSA and GlideSP scores—further validate PocketGen's ability to generate higher-affinity

https://doi.org/10.1038/s42256-024-00920-9

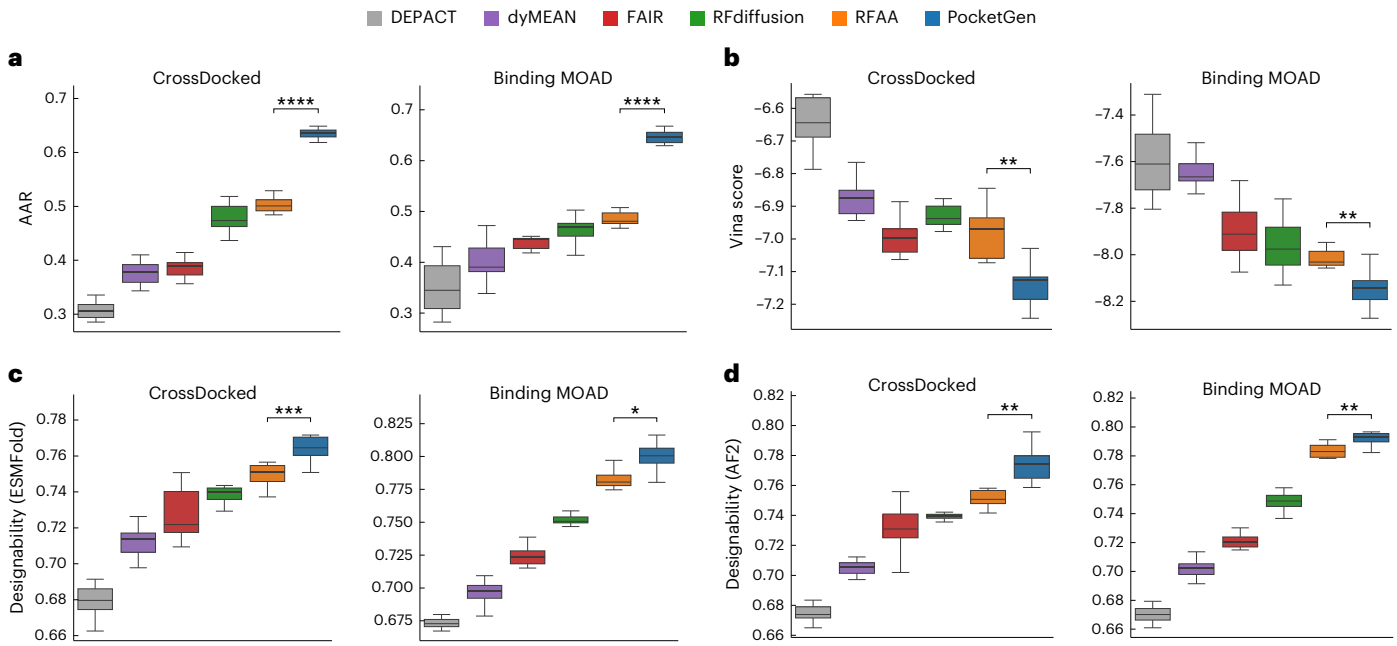


Fig. 2 | Benchmarking PocketGen on CrossDocked and Binding MOAD datasets. a, AAR (*P* values, 3.8×10^{-8} and 1.5×10^{-10}). **b**, Vina score performance (*P* values, 6.1×10^{-3} and 6.7×10^{-3}). **c**, Designability scores using ESMFold structure prediction method (*P* values, 6.0×10^{-4} and 2.5×10^{-2}). **d**, Designability scores using AlphaFold 2 structure prediction method (*P* values, 4.4×10^{-3} and 4.4×10^{-3}). Uncertainty is quantified via bootstrapping, two-sided Kolmogorov–Smirnov test is used to compare PocketGen with the best-performing existing model (RFAA). *P*-value annotation legend: $^{*}p \in [0, 0.1, 0.05]$, $^{**}p \in [0.001, 0.01]$, $^{***}p \in [0.0001, 0.001]$, $^{****}p \leq 0.0001$. The sample size in the plots is 10 for each model. In all the box plots, the minimum is the smallest value within the dataset, marked at the end of the lower whisker. The first quartile (Q1), or 25th percentile, forms the lower edge of the box. The median (50th percentile) is represented by a line inside the box, indicating the midpoint of the data. The third quartile (Q3), or 75th percentile, forms the upper edge of the box. The maximum is the largest value within the dataset, marked at the end of the upper whisker. The whiskers extend to the smallest and largest values within 1.5 times the interquartile range (IQR).

pockets, with reductions of 4.287 in MM-GBSA and 0.376 in GlideSP scores. PocketGen demonstrates competitive performance in pLDDT, scRMSD and ΔscTM scores, underscoring its capability to produce high-affinity pockets and maintain structural validity and sequence–structure consistency. With a 97% success rate in generating pockets with higher affinity than the reference cases (compared with a 93% success rate for the strongest baseline, RFAA) on the CrossDocked dataset, PocketGen proves its effectiveness and applicability across diverse ligand molecules.

To assess substructure validity and consistency with the reference datasets, we conduct a qualitative substructure analysis (Supplementary Table 4 and Supplementary Fig. 2). This analysis focuses on three covalent bonds in the residue backbone (C–N, C=O and C–C), three dihedral angles in the backbone (ϕ, ψ and ω) (ref. 58) and four dihedral angles in the side chains (χ_1, χ_2, χ_3 and χ_4) (http://www.mlb.co.jp/linux/science/garlic/doc/commands/dihedrals.html). Following prior research[59,60], we collect bond length and angle distributions from both generated pockets and test dataset and compute the Kullback–Leibler divergence to quantify the distance between these distributions. Lower Kullback–Leibler divergence scores for PocketGen indicate its effectiveness in accurately replicating the geometric features observed in the reference data.

Probing generative capabilities of PocketGen

Next, we explore PocketGen's generative capabilities. Beyond designing high-quality protein pockets, generative models need to be efficient and maximize the yield of biochemical experiments—rapidly producing high-fidelity pocket candidates with only a small number of designs necessary to find a hit. Figure 3a compares the average generation time across various methods. Physics-based modelling (PocketOpt) and template matching (DEPACT) can take over 1,000 s to generate 100 pockets. Advanced protein backbone generation models RFdiffusion and RFAA are computationally expensive due to their diffusion-based architectures, requiring 1,633.5 s and 2,210.1 s to design 100 pockets. Iterative refinement methods like PocketGen can substantially reduce generation time, with PocketGen taking just 44.2 s to generate 100 pockets.

Although recent methods for pocket generation focus on maximizing the binding affinity with target molecules, this strategy may not always align with practical needs for which pocket diversity is equally important. Examining a batch of designed pockets, rather than a single design, improves the success rate of pocket design. Therefore, we investigate the relationship between binding affinity and the diversity of the generated protein pockets (Fig. 3b). Diversity is quantified as (1 – average pairwise pocket residue sequence similarity) and can be adjusted by altering the sampling temperature τ (higher τ results in greater diversity). Figure 3b compares PocketGen with the most competitive baseline, RFAA[16] + LigandMPNN[30] and the latest version of ProteinMPNN[29]. We observe that there is a trade-off between binding affinity and diversity. PocketGen can generate protein pockets with higher affinity than RFAA at the same level of diversity.

Figure 3c explores the effect of redesigned pocket size on PocketGen's performance. The redesign process targets all the residues with atoms within 3.5 Å, 4.5 Å and 5.5 Å of any binding ligand atoms. We observe a slight decline in the average AAR, root mean squared deviation (RMSD) and Vina scores as the size of the redesigned pocket increases. This trend is probably due to the increased complexity and reduced contextual information in the case of larger redesigned pocket areas. Larger pockets tend to enable the exploration of structures with potentially higher affinity, as indicated by the lowest Vina scores, which reach −17.5 kcal mol^{-1} for designs with a 5.5 Å radius. This can be

https://doi.org/10.1038/s42256-024-00920-9

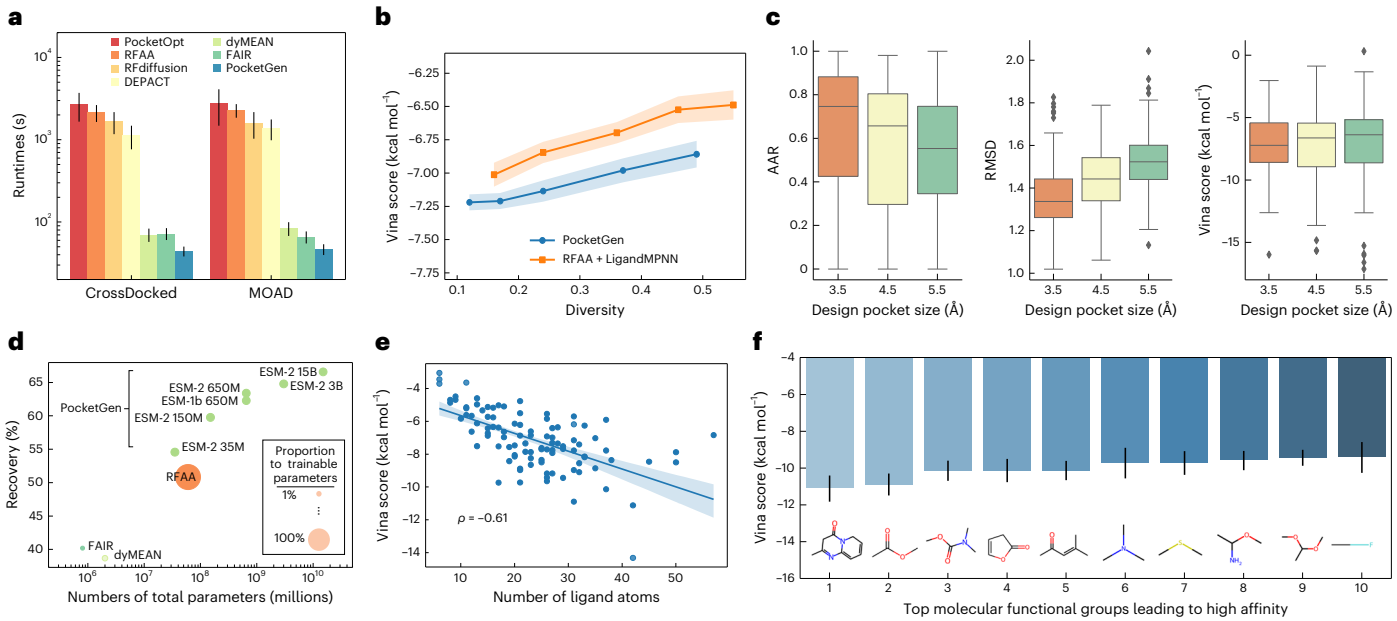


Fig. 3 | Exploring capabilities of PocketGen. a, Average runtime of different methods for generating 100 protein pockets for a ligand molecule on the two benchmarks. Data are presented as mean ± standard deviation. The sample size for each method is 100. **b**, Trade-off between quality (measured by Vina score) and diversity (1 − average pairwise sequence similarity) of PocketGen. We can balance the trade-off by tuning the temperature hyperparameter τ. We show the mean values with the standard deviations marked as shadows. **c**, Influence of the design pocket size on the metrics. We draw box plots and the sample size is 100. In the box plots, the minimum is the smallest value, excluding outliers, marked at the end of the lower whisker. The first quartile (Q1), or the 25th percentile, forms the lower edge of the box, whereas the median (50th percentile) is represented by a line within the box. The third quartile (Q3), or the 75th percentile, forms the upper edge of the box. The maximum is the largest value, excluding the outliers, marked at the end of the upper whisker. The whiskers extend to data points within 1.5 times the IQR, and any values beyond the whiskers are considered outliers. **d**, Performance with respect to model scales of pLMs using the ESM series on the CrossDocked dataset. The green dots represent the PocketGen models with different ESMs. The bubble size is proportional to the number of trainable parameters. **e**, PocketGen tends to generate pockets with higher affinity for larger ligand molecules (Pearson correlation, $\rho = -0.61$; bands indicate the 95% confidence interval). **f**, Top molecular functional groups leading to high affinity. The sample size is 100 and data are presented as mean ± standard deviation.

attributed to the enhanced structural complementarity in larger pocket designs. Extended Data Fig. 1a,b shows that PocketGen can generate full protein binders for two ligand molecules, with the generated protein binders achieving high scTM scores of 0.900 and 0.976.

A key feature that sets PocketGen apart from other pocket generation models is its integration of pLMs. In addition to using ESM (evolutionary scale modeling)-2 650M (ref. 49) throughout our experiments, we evaluated a broader family of ESM models, ranging in model size from 8M to 15B trainable parameters. As shown in Fig. 3d, PocketGen's performance improves with the scaling of pLMs. Specifically, the performance increases from 54.58% to 66.61% when transitioning from the ESM-2 35M to ESM-2 15B models. This follows a logarithmic scaling law, consistent with trends observed in large language models[61]. PocketGen efficiently trains large pLMs by fine-tuning only the adapter layers and maintains most pLM layers fixed. As a result, PocketGen requires substantially fewer trainable parameters than RFAA[16] (7.9M versus 82.9M trainable parameters).

The characteristics of the ligand molecule can affect the performance of PocketGen in generating binding pockets. Figure 3e shows the relationship between the average Vina score of the generated pockets and the number of ligand atoms, revealing that PocketGen tends to create pockets with higher affinity for larger ligand molecules. This trend may result from the increased surface area for interaction, the presence of additional functional groups and greater flexibility in the conformations of larger molecules[62,63]. Key functional groups in ligand molecules that contribute to high binding affinity were identified using IFG (identifying functional groups)[64]. Figure 3f highlights the top-10-ranked molecular functional groups, which include hydrogen-bond donors and acceptors (carbonyl groups), aromatic rings, sulfhydryl groups and halogens. These groups facilitate favourable interactions with protein pockets, thereby enhancing the binding affinity.

Since PocketGen also updates the ligand structures during pocket generation, we use PoseBusters[65] to evaluate the structural validity of the updated ligands. A detailed validity check (Extended Data Fig. 1e) shows that PocketGen achieves over 95% across all tests in PoseBusters. This is expected, as PocketGen makes only minor updates to ligand structures during pocket generation, successfully maintaining ligand structural integrity. In Extended Data Fig. 1c, we explore the relationship between binding affinity and RMSD to the crystal structure in PDBBind. Using a geometric interaction graph neural network[66–68] to predict affinity (log[K], where K is the equilibrium dissociation constant), we observe that generally, lower RMSD corresponds to higher affinity. Extended Data Fig. 1d demonstrates that PocketGen improves most protein–ligand complexes in PDBBind by redesigning the binding pockets.

We conducted ablation studies (Supplementary Table 5) and hyperparameter analysis (Supplementary Fig. 3) to assess the contribution of each module in PocketGen and the impact of hyperparameter choices on model performance. For comparison, we replaced the bilevel graph transformer in PocketGen with other popular encoders in structural biology, such as EGNN (E(n) equivariant graph neural network)[69], GVP (geometric vector perceptron)[70] and GMN (graph mechanics network)[71]. The results indicate that the bilevel graph transformer and the integration of pLM into PocketGen substantially enhance the performance. Furthermore, PocketGen demonstrates robustness to hyperparameter variations, consistently yielding competitive results.

https://doi.org/10.1038/s42256-024-00920-9

Table 1 | The top 1/3/5/10 generated protein pockets (ranked by Vina score) on the CrossDocked dataset

	PocketOpt	DEPACT	dyMEAN	FAIR	RFdiffusion	RFAA	PocketGen
Top-1-ranked generated protein pocket							
Vina score (↓)	−9.216±0.154	−8.527±0.061	−8.540±0.107	−8.792±0.122	−9.037±0.080	−9.216±0.091	**−9.655**±0.094
MM-GBSA (↓)	−58.754±1.220	−47.130±1.372	−48.248±0.816	−51.923±0.588	−54.817±1.091	−59.255±1.260	**−63.542**±0.717
GlideSP (↓)	−8.612±0.127	−7.495±0.053	−7.472±0.088	−7.584±0.094	−8.485±0.069	−8.540±0.065	**−8.916**±0.047
Success Rate (↑)	0.923±0.034	0.750±0.016	0.762±0.029	0.796±0.035	0.891±0.020	0.930±0.027	**0.974**±0.012
pLDDT (AF2) (↑)	–	82.164±0.241	83.053±0.397	83.285±0.240	84.432±0.152	86.571±0.178	**86.830**±0.145
scRMSD (AF2) (↓)	–	0.714±0.025	0.708±0.022	0.693±0.018	0.675±0.015	0.654±0.012	**0.645**±0.009
ΔscTM (AF2) (↑)	–	−0.008±0.003	−0.005±0.002	−0.011±0.005	0.022±0.006	0.020±0.003	**0.028**±0.002
ΔscTM (AF2+co) (↑)	–	−0.012±0.003	−0.025±0.004	−0.032±0.007	–	–	**0.008**±0.002
Top-3-ranked generated protein pockets							
Vina score (↓)	−8.878±0.112	−8.131±0.064	−8.196±0.090	−8.321±0.045	−8.876±0.107	−8.980±0.057	**−9.353**±0.063
MM-GBSA (↓)	−53.372±1.164	−43.790±1.029	−44.151±0.534	−46.050±0.809	−52.423±0.847	−53.593±0.722	**−60.770**±0.589
GlideSP (↓)	−8.360±0.094	−7.377±0.039	−7.325±0.078	−7.348±0.052	−8.219±0.049	−8.233±0.060	**−8.670**±0.056
pLDDT (AF2) (↑)	–	82.049±0.456	82.918±0.237	83.025±0.334	84.260±0.210	**86.289**±0.214	86.280±0.135
scRMSD (AF2) (↓)	–	0.713±0.017	0.722±0.011	0.692±0.016	0.685±0.007	**0.659**±0.014	0.660±0.012
ΔscTM (AF2) (↑)	–	−0.011±0.004	−0.006±0.002	−0.008±0.003	0.021±0.003	0.022±0.002	**0.026**±0.003
ΔscTM (AF2+co) (↑)	–	−0.016±0.005	−0.026±0.004	−0.034±0.003	–	–	**0.005**±0.001
Top-5-ranked generated protein pockets							
Vina score (↓)	−8.702±0.090	−7.786±0.052	−7.974±0.049	−7.943±0.035	−8.510±0.073	−8.689±0.044	**−9.239**±0.076
MM-GBSA (↓)	−52.080±1.071	−35.250±0.823	−37.924±0.340	−37.816±0.402	−46.847±0.700	−51.651±0.809	**−58.083**±0.561
GlideSP (↓)	−8.173±0.089	−7.126±0.035	−7.294±0.042	−7.289±0.041	−8.022±0.030	−8.093±0.048	**−8.417**±0.040
pLDDT (AF2) (↑)	–	82.445±0.307	82.763±0.102	83.748±0.271	84.505±0.288	85.617±0.105	**85.969**±0.080
scRMSD (AF2) (↓)	–	0.716±0.014	0.726±0.011	0.698±0.015	0.680±0.009	0.657±0.006	**0.655**±0.004
ΔscTM (AF2) (↑)	–	−0.009±0.003	−0.007±0.002	−0.012±0.004	0.019±0.003	0.020±0.001	**0.025**±0.001
ΔscTM (AF2+co) (↑)	–	−0.017±0.002	−0.025±0.006	−0.035±0.005	–	–	**0.006**±0.002
Top-10-ranked generated protein pockets							
Vina score (↓)	−8.556±0.104	−7.681±0.040	−7.690±0.054	−7.785±0.028	−8.352±0.061	−8.524±0.038	**−9.065**±0.057
MM-GBSA (↓)	−49.257±0.821	−32.534±0.680	−33.118±0.269	−33.670±0.440	−45.726±0.830	−47.325±0.540	**−54.800**±0.406
GlideSP (↓)	−7.935±0.082	−6.954±0.042	−7.022±0.034	−7.131±0.025	−7.806±0.022	−7.840±0.026	**−8.196**±0.027
pLDDT (AF2) (↑)	–	81.520±0.317	82.467±0.255	83.271±0.228	84.080±0.190	85.442±0.145	**85.945**±0.139
scRMSD (AF2) (↓)	–	0.712±0.013	0.733±0.014	0.706±0.013	0.688±0.009	0.680±0.010	**0.659**±0.007
ΔscTM (AF2) (↑)	–	−0.014±0.002	−0.006±0.001	−0.010±0.003	0.016±0.002	0.019±0.001	**0.023**±0.002
ΔscTM (AF2+co) (↑)	–	−0.018±0.004	−0.030±0.002	−0.033±0.002	–	–	**0.004**±0.002

The success rate measures the percentage of proteins for which the model generates binding pockets with higher affinity than those in the reference datasets. Besides the Vina score, we use MM-GBSA and min-in-space GlideSP scores to calculate the binding affinity. We report the average pLDDT of the predicted pocket, the scRMSD of the pocket backbone coordinates and the change in scTM scores of the whole protein. AF2 means the scores are calculated with AlphaFold 2 as the folding tool (Supplementary Table 2 lists the ESMFold results). Co indicates co-design, where co-design methods use the designed sequence for consistency calculation. The pLDDT, scRMSD and ΔscTM values for PocketOpt are not reported, as PocketOpt keeps protein backbone structures fixed. The results of affinity-related metrics, pocket-structure-related metrics and whole-protein-structure metrics are marked. We report the mean and standard deviation over three independent runs. Best-performing results are indicated in bold.

Generating protein pockets for small molecule therapeutics

We demonstrate PocketGen's ability to redesign the pockets of antibodies, enzymes and biosensors for specific target ligands, building on previous research[3,10,16]. Specifically, we consider the following molecules. Cortisol (HCY)[72] is a primary stress hormone that raises glucose levels in the bloodstream and serves as a biomarker for stress and other conditions. We redesign the pocket of a cortisol-specific antibody (Protein Data Bank (PDB) ID: 8CBY), potentially aiding the development of immunoassays. APX[73] is an oral anticoagulant approved by the FDA in 2012 for patients with non-valvular atrial fibrillation to reduce the risk of stroke and blood clots[74]. APX target factor Xa (PDB ID: 2P16) is an enzyme in blood coagulation that converts prothrombin into thrombin to facilitate clot formation. Redesigning the pocket of factor Xa has therapeutic implications. Fentanyl (7V7)[75] is a widely abused opioid contributing to the opioid crisis. Computationally designing fentanyl-binding proteins (biosensors) can support detection and neutralization efforts[10]. In Fig. 4, a protein–ligand interaction profiler[76] illustrates the interactions between the redesigned protein pockets and ligands, comparing these predicted interactions with the original binding patterns.

To generate pockets for the aforementioned small molecules, we pretrained PocketGen on the Binding MOAD dataset, excluding protein–ligand complexes considered in this analysis. The pockets produced by PocketGen successfully replicate most non-bonded interactions observed in experimentally measured protein–ligand complexes (achieving a 13/15 match for HCY) and introduce additional physically

https://doi.org/10.1038/s42256-024-00920-9

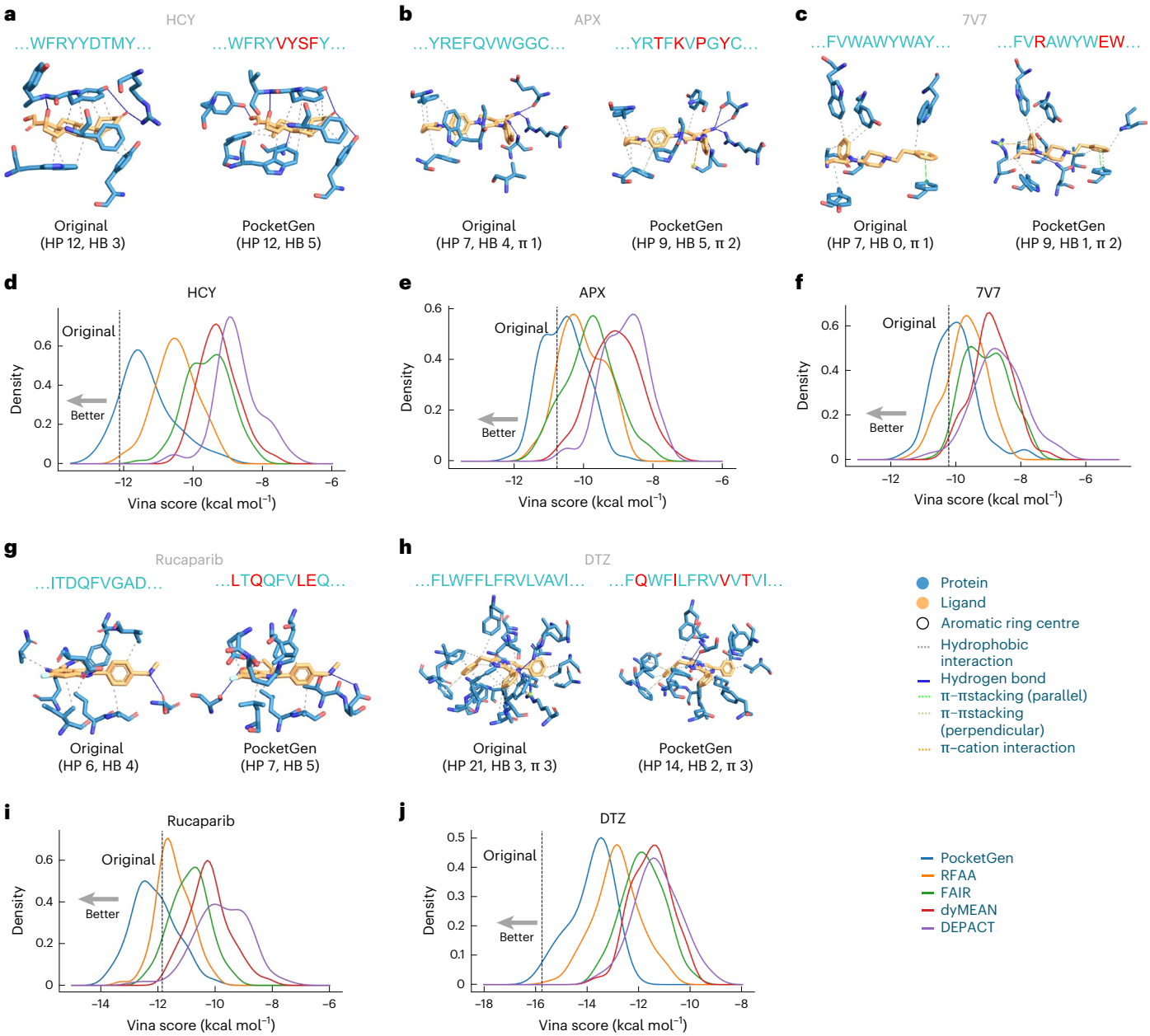


Fig. 4 | Using PocketGen to design protein pockets for binding with important ligands. a–c, Illustrations of protein–ligand interaction analysis for three target molecules (HCY (**a**), APX (**b**) and 7V7 (**c**)). PocketGen refers to the protein pocket designed by PocketGen, and Original denotes the original protein–ligand structure. HP indicates hydrophobic interactions, HB signifies hydrogen bonds and π denotes the π–π stacking/π–cation interactions. In the residue sequences, the red ones denote the designed residues that differ from the original pocket. **d–f**, Pocket binding affinity distributions of PocketGen and baseline methods for the three target molecules (HCY (**d**), APX (**e**) and 7V7 (**f**)). We mark the Vina Score of the original pocket with the vertical dotted lines. For each method, we sample 100 pockets for each target ligand. The ratio of the generated pockets by PocketGen with higher affinity than the corresponding reference pocket are 11%, 40% and 45%, respectively. **g**,**h**, Protein–ligand interaction analysis for unseen proteins in the training dataset (PiB[21] and luxsit[8]). The target molecules are rucaparib (**g**) and DTZ (**h**). **i**,**j**, Pocket binding affinity distributions of PocketGen and baselines for rucaparib (**i**) and DTZ (**j**).

plausible interaction patterns not present in the original complexes. For example, the generated pockets for HCY, APX and 7V7 molecules form 2, 3 and 4 extra interactions, respectively. Specifically for HCY, PocketGen preserves key interaction patterns such as hydrophobic interactions (TRP47, PHE50, TYR59 and TYR104) and hydrogen bonds (TYR59), as two new hydrogen-bond-mediated interactions are introduced within the pocket. For protein pockets designed to bind APX and 7V7 ligands, PocketGen maintains important interactions like hydrophobic contacts, hydrogen bonds and π–π stacking, as well as establishes additional interactions, for example, a π–cation interaction with LYS192 for APX and hydrogen bonds with ASN35 for 7V7, thereby enhancing the binding affinity with the target ligands. PocketGen effectively captures non-covalent interactions derived from protein–ligand structure data and introducing new, plausible interaction patterns to optimize the binding affinity.

With its ability to establish favourable protein–ligand interactions, PocketGen generates high-affinity pockets for these drug ligands. In Fig. 4d–f, we present the affinity distributions of pockets generated by PocketGen compared with alternative methods. The percentage of generated pockets with higher affinity than the reference is 11%, 40%

https://doi.org/10.1038/s42256-024-00920-9

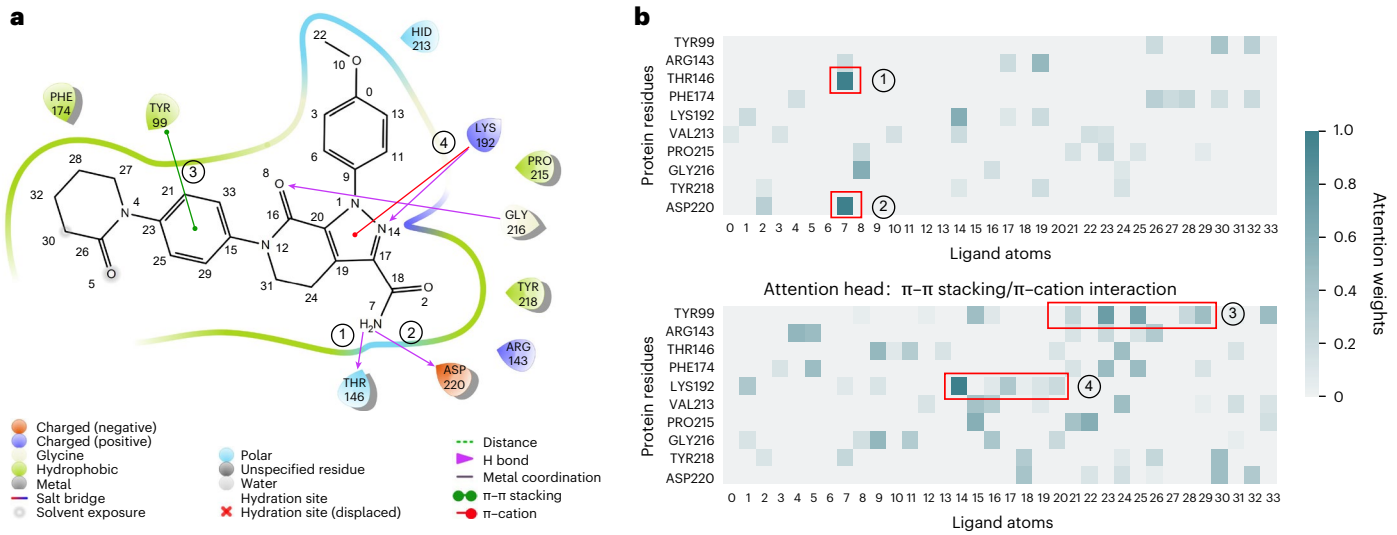


Fig. 5 | Attention maps in PocketGen capture interactions between atoms in protein and ligand molecules. a, Two-dimensional interaction plot of the designed pocket by PocketGen for APX. **b**, Heat map of attention matrices between residues and ligand atoms from the last layer of PocketGen. We show two selected attention heads with notable attention patterns marked with red rectangles. We notice that each head emphasizes different interactions. For example, PocketGen recognizes the hydrogen-bond interaction and assigns a strong attention weight between residues (1) THR146 and (2) ASP220 and ligand atom 7 in the first head. The π–π stacking and π–cation interactions of (3) TYR99 and (4) LYS192 are well captured in the second head. The values are normalized by the maximum value (v_{max}) and the minimum value (v_{min}) in each heat map (that is, $v' = \frac{v - v_{min}}{v_{max} - v_{min}}$).

and 45% for PocketGen. In contrast, the best runner-up method, RFAA, achieves only 0%, 10% and 18% across the same cases.

Protein stability is a critical factor in protein design, ensuring that the designed protein can fold into and maintain its three-dimensional structure[77]. Stability is quantified by the difference in Gibbs free energy ($\Delta\Delta G$) between the redesigned protein and the wild-type (original) protein, where $\Delta\Delta G = \Delta G_{orig} - \Delta G_{redesign}$. A positive $\Delta\Delta G$ value indicates increased stability, whereas a negative value suggests decreased stability. We used DDMut[78] to predict the change in stability for the pockets generated in Fig. 4, with $\Delta\Delta G$ values of 0.09 (HCY), 0.92 (APX), 0.13 (7V7), 0.27 (rucaparib) and 0.02 (DTZ). These results suggest that PocketGen can generate protein structures likely to remain sufficiently stable to bind the ligand molecules.

To demonstrate the generalization capability of PocketGen, we tested it on unseen proteins from the training set, including PiB[21] and luxsit[8], with the binding ligands rucaparib and DTZ, respectively. Figure 4g,h shows the interaction analysis, whereas Fig. 4i,j presents the distribution of Vina scores. PocketGen consistently outperforms other methods in generating higher-affinity pockets. Generating pockets with higher affinity for DTZ proved more challenging, as the original pocket was designed using site-saturation mutagenesis[8] to achieve optimal design. In Extended Data Fig. 1f, we present case studies involving a pair of activity cliff ligand molecules (C19 and C52)[79] to further explore PocketGen's adaptability. The generated interactions vary across molecular fragments: for one fragment, hydrogen bonds and hydrophobic interactions are generated, whereas for another fragment, halogen bonds are produced. This suggests that PocketGen has learned key protein–ligand interaction rules, allowing it to design high-affinity binding pockets.

Interpreting PocketGen's protein–ligand interactions

We analyse attention maps learned by PocketGen using the generated pocket for the APX ligand. Figure 5a presents a two-dimensional interaction plot drawn with the Schrödinger Maestro tool (v.2018-1). To evaluate PocketGen's recognition of key protein–ligand interactions, we plot the heat map of attention weights produced by the final layer of its neural architecture. In Fig. 5b, two attention heads are shown, with each row and column representing a protein residue or a ligand atom, respectively. The attention heat maps are sparse, reflecting PocketGen's use of sparse attention (Methods). The attention heads exhibit diverse patterns, focusing on different aspects of the interactions. For example, the first attention head emphasizes hydrogen bonds, assigning high weights to interactions between residues THR146 and ASP220 and ligand atom 7. The second attention head captures π–π stacking and π–cation interactions, specifically between residue TYR99 and ligand atoms 15, 21, 23, 25, 29 and 33; and residue LYS192 and ligand atoms 1, 14, 17, 19 and 20. These findings suggest that, despite being data-driven, PocketGen has learned to recognize biochemical intermolecular interactions.

Discussion

Understanding how proteins bind to ligand molecules is critical for enzyme catalysis, immune recognition, cellular signal transduction, gene expression control and other biological processes. Recent developments include deep generative models designed to study protein–ligand binding, such as Lingo3DMol[80], ResGen[81] and PocketFlow[82], which generate de novo drug-like ligand molecules for fixed protein targets. NeuralPLexer[4] can create the structure of protein–ligand complexes given the protein sequence and ligand molecular graph. However, these models do not facilitate the de novo generation of protein pockets—the interfaces that bind with the ligand molecule for targeted ligand binding (critical in enzyme and biosensor engineering).

We developed PocketGen, a deep generative method capable of generating residue sequence and full-atom structure of the protein pocket region to maximize binding with the target ligand molecule. PocketGen includes two modules: a bilevel graph transformer for structural encoding, and a sequence refinement module that uses pLMs for sequence prediction. For structure prediction, the bilevel graph transformer directly updates coordinates of all atoms in the pocket region instead of separately predicting the backbone frame orientation and side-chain torsion angles. To achieve sequence–structure consistency and leverage evolutionary information encoded in pLMs, PocketGen integrates a structural adapter into a pLM for sequence updates. This adapter employs cross-attention between sequence and structure features to promote information flow and sequence–structure consistency. Experiments across benchmarks and case studies involving

therapeutic ligand molecules illustrate PocketGen's ability to generate high-fidelity pocket structures with high binding affinity and favourable interactions with target ligands. Analysis of PocketGen's performance across various settings reveals its proficiency in balancing diversity and affinity as well as generalizing across pocket sizes. Additionally, PocketGen offers computational efficiency, substantially reducing runtime compared with traditional physics-based methods, making it feasible to sample large quantities of pocket candidates. PocketGen surpasses existing methods in efficiently generating high-affinity protein pockets for target ligand molecules, finding important interactions between atoms on protein and ligand molecules, and attaining consistency in sequence and structure domains.

PocketGen creates several fruitful directions for future work. PocketGen could be expanded to design larger areas of the protein beyond pocket regions. Although PocketGen has been evaluated on larger pocket designs, modifications will be required to enhance scalability and robustness necessary to generate larger protein areas. Another fruitful future direction involves incorporating biochemical priors, subpockets[83] and interaction templates[17] to improve model generalizability. For instance, despite overall dissimilarity, two protein pockets might still bind the same fragment if they share similar subpockets[84]. Moreover, evaluation of new designs through wet laboratory experiments could further validate PocketGen's effectiveness. Approaches such as PocketGen have the potential to advance areas of machine learning and bioengineering and help with the design of small-molecule binders and enzymes.

Methods

Overview of PocketGen

Unlike previous methods focusing on protein sequence or structure generation, we aim to co-design both residue types (sequences) and three-dimensional (3D) structures of the protein pocket that can fit and bind with the target ligand molecules. Inspired by previous works on structure-based drug design[81,83] and protein generation[34,35], we formulate pocket generation in PocketGen as a conditional generation problem that generates the sequences and structures of pocket conditioned on the protein scaffold (other parts of the protein except the pocket region) and the binding ligand. To be specific, let $\mathcal{A} = \mathbf{a}_1 \ldots \mathbf{a}_{N_s}$ denote the whole protein sequence of residues, where N_s is the length of the sequence. The 3D structure of the protein can be described as a point cloud of protein atoms $\{\mathbf{a}_{i,j}\}_{1\le i\le N_s, 1\le j\le n_i}$ and let $x(\mathbf{a}_{i,j}) \in \mathbb{R}^3$ denote the 3D coordinate of protein atoms. n_i is the number of atoms in a residue determined by the residue types. The first four atoms in any residue correspond to its backbone atoms (C_α, N, C, O), and the rest are the side-chain atoms. The ligand molecule can also be represented as a 3D point cloud $\mathcal{M} = \{\mathbf{v}_k\}_{k=1}^{N_l}$, where $\mathbf{v}_k$ denotes the atom feature. Let $x(\mathbf{v}_k)$ denote the 3D coordinates of atom $\mathbf{v}_k$. Our work defines the protein pocket as a set of residues in the protein closest to the binding ligand molecule: $\mathcal{B} = \mathbf{b}_1 \ldots \mathbf{b}_m$. The pocket $\mathcal{B}$ can, thus, be represented as an amino acid subsequence of a protein: $\mathcal{B} = \mathbf{a}_{e_1} \ldots \mathbf{a}_{e_m}$, where $\mathbf{e} = \{e_1 \ldots e_m\}$ is the index of the pocket residues in the whole protein. The index $\mathbf{e}$ can be formally given as $\mathbf{e} = \{i \mid \min_{1\le j\le n_i, 1\le k\le N_l} \| x(\mathbf{a}_{i,j}) - x(\mathbf{v}_k)\|_2 \le \delta\}$, where $\|\cdot\|_2$ is the L2 distance norm and δ is the distance threshold. According to the distance range of pocket–ligand interactions[45], we set $\delta = 3.5$ Å in the default setting. With the above-defined notations, PocketGen aims to learn a conditional generative model formally defined as

$$P(\mathcal{B}|\mathcal{A} \setminus \mathcal{B}, \mathcal{M}), \tag{1}$$

where $\mathcal{A} \setminus \mathcal{B}$ denotes the other parts of the protein except the pocket region. We also adjust the structure ligand molecule $\mathcal{M}$ in PocketGen to encourage protein–ligand interactions and reduce steric clashes.

To effectively generate the structure and sequence of the protein pocket $\mathcal{B}$, the equivariant bilevel graph transformer and the sequence refinement module with pretrained pLMs and adapters are proposed, which are discussed below. The illustrative workflow is depicted in Fig. 1.

Equivariant bilevel graph transformer

It is critical to model the complex interactions in the protein pocket–ligand complexes for pocket generation. However, the multi-granularity (for example, atom level and residue level) and multi-aspect (intraprotein and protein–ligand) nature of interactions brings a lot of challenges. Inspired by recent works on hierarchical graph transformer[83] and generalist equivariant transformer[85], we propose an equivariant bilevel graph transformer to effectively model the multi-granularity and multi-aspect interactions. Each residue or ligand is represented as a block (that is, a set of atoms) for the conciseness of representation and ease of computation. Then, the protein–ligand complex can be abstracted as a geometric graph of sets $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{H_i, X_i | 1 \le i \le B\}$ denotes the blocks and $\mathcal{E} = \{e_{ij} | 1 \le i,j \le B\}$ includes all the edges between blocks (B is the total number of blocks). We added self-loops to the edges to capture interactions within the block (for example, the interactions between ligand atoms). Our model adaptively assigns different numbers of channels to H_i and X_i to accommodate different numbers of atoms in residues and ligands. For example, given a block with n_i atoms, the corresponding block has $H_i \in \mathbb{R}^{n_i \times d_h}$ indicating the atom features (d_h is the feature dimension size) and $X_i \in \mathbb{R}^{n_i \times 3}$ denoting the atom coordinates. Specifically, the pth row of H_i and X_i corresponds to the pth atom's trainable feature (that is, $H_i[p]$) and coordinates (that is, $X_i[p]$), respectively. The trainable feature $H_i[p]$ is first initialized with the concatenation of atom-type embedding, residue/ligand embeddings and the atom positional embeddings. To build $\mathcal{E}$, we connect the k nearest-neighbouring residues according to the pairwise C_α distances. To reflect the interactions between the protein pocket and ligand, we add edges between all the pocket residues and ligand blocks. We describe the modules in PocketGen's equivariant bilevel graph transformer, bilevel attention module and equivariant feed-forward networks (FFNs).

Bilevel attention module. Our model captures both atom-level and residue-/ligand-level interactions with the bilevel attention module. First, given two blocks i and j connected by an edge e_{ij}, we obtain the query, the key and the value matrices with the following transformations:

$$Q_i = H_i W_Q, \qquad K_j = H_j W_K, \qquad V_j = H_j W_V, \tag{2}$$

where W_Q, W_K and $W_V \in \mathbb{R}^{d_h \times d_r}$ are trainable parameters.

To calculate the atom-level attention across the ith and jth blocks, we denote $X_{ij} \in \mathbb{R}^{n_i \times n_j \times 3}$ and $D_{ij} \in \mathbb{R}^{n_i \times n_j}$ as the relative coordinates and distances between atom pairs in block i and j, namely, $X_{ij}[p,q] = X_i[p] - X_j[q]$, $D_{ij}[p,q] = \|X_{ij}[p,q]\|_2$, respectively. Then, we have

$$R_{ij} = \frac{1}{\sqrt{d_r}} \left(Q_i K_j^\top\right) + \sigma_D\left(\mathrm{RBF}(D_{ij})\right), \tag{3}$$

$$\boldsymbol{\alpha}_{ij} = \mathrm{Softmax}\left(R_{ij}\right), \tag{4}$$

where $\sigma_D(\cdot)$ is a multilayer perceptron (MLP) that adds distance bias to the attention calculation. RBF embeds the distance with radial basis functions. $\boldsymbol{\alpha}_{ij} \in \mathbb{R}^{n_i \times n_j}$ is the atom-level attention matrix obtained by applying row-wise Softmax on $R_{ij} \in \mathbb{R}^{n_i \times n_j}$. To encourage sparsity in the attention matrix, we keep the top-k' elements of each row in $\boldsymbol{\alpha}_{ij}$ and set the others as zeros.

The residue-/ligand-level attention from the jth block to the ith block is calculated as

$$r_{ij} = \frac{\mathbf{1}^\top R_{ij} \mathbf{1}}{n_i n_j}, \tag{5}$$

$$\beta_{ij} = \frac{\exp(r_{ij})}{\sum_{j \in \mathcal{N}(i)} \exp(r_{ij})}, \tag{6}$$

where $\mathbf{1}$ refers to the column vector with all the elements set as ones and $\mathcal{N}(i)$ denotes the neighbouring blocks of i. r_{ij} sums up all the values

in R_{ij} to represent the overall correlation between blocks i and j. Subsequently, β_{ij} denotes the attention across blocks at the block level.

We can update the representations and coordinates using the above atom-level and residue-/ligand-level attentions. PocketGen only updates the coordinates of the residues in the pocket and ligand molecule. The other protein residues are fixed. Specifically, for the pth atom in block i,

$$m_{ij,p} = \beta_{ij}\left(\boldsymbol{\alpha}_{ij}[p] \odot \phi_x(Q_i[p]||K_j||\mathrm{RBF}(D_{ij}[p]))\right), \tag{7}$$

$$H_i'[p]H_i[p] + \sum_{j\in\mathcal{N}(i)} \beta_{ij}\phi_h(\boldsymbol{\alpha}_{ij}[p]\cdot V_j), \tag{8}$$

$$X_i'[p] = X_i[p] + \begin{cases} \sum_{j\in\mathcal{N}(i)} m_{ij,p}\cdot X_{ij}[p], & \text{if } i \text{ belongs to ligand or pocket residues} \\ 0, & \text{if } i \text{ belongs to other protein residues} \end{cases} \tag{9}$$

where ϕ_h and ϕ_x are MLPs with concatenated representations as the input (concatenation along the second dimension and $Q_i[p]$ is repeated along rows). $\odot$ computes the element-wise multiplication. H_i' and X_i' denote the updated representation and coordinate matrices, respectively, and we can verify that the dimension size of H_i' and X_i' remains the same regardless of the neighbouring block size n_j. Furthermore, as the attention coefficients α_{ij} and β_{ij} are invariant under E(3) transformations, the modification of X_i' adheres to E(3) equivariance. Additionally, the permutation of atoms within each block does not affect this update process.

Equivariant FFN. We adapted the FFN module in the transformer model[86] to update H_i and X_i. Specifically, the representation and coordinates of atoms are updated to consider the block's feature/geometric centroids (means). The centroids are denoted as

$$h_c = \mathrm{centroid}(H_i), \qquad x_c = \mathrm{centroid}(X_i). \tag{10}$$

Then, we obtain the relative coordinate Δx_p and the relative distance representation $\boldsymbol{r}_p$ based on the L2 norm of Δx_p:

$$\Delta x_p = X_i[p] - x_c, \qquad r_p = \mathrm{RBF}(\|\Delta x_p\|_2). \tag{11}$$

The representation and coordinates of atoms are updated with MLPs σ_h and σ_x, respectively. The centroids are integrated to inform of the context of the block:

$$H'[p] = H[p] + \sigma_h(H_i[p], h_c, r_p), \tag{12}$$

$$X_i'[p] = X_i[p] + \Delta x_p\sigma_x(H_i[p], h_c, r_p). \tag{13}$$

To stabilize and accelerate training, layer normalization[87] is appended at each layer of the equivariant bilevel graph transformer to normalize H. The equivariant FFN satisfies E(3) equivariance. Owing to each module's E(3) equivariance, the whole proposed bilevel graph transformer has the desirable property of E(3) equivariance (Supplementary Theorem 1 provides the details). In PocketGen, we use an E(3) equivariant model for its simplicity similar to previous works[88,89], which is capable enough to achieve strong performance. We are aware that an SE(3) equivariant model architecture would be better for learning the chirality-related properties of the protein, which we left for future exploration.

Sequence refinement with pLMs and adapters

pLMs, such as the ESM family of models[40,41], have learned extensive evolutionary knowledge from the vast array of natural protein sequences, demonstrating a strong ability to design protein sequences. In PocketGen, we propose to leverage pLMs to help refine the designed protein pocket sequences. To infuse the pLMs with structural information, we implant lightweight structural adapters inspired by previous works[90,91]. Different from LM-Design[91], which focuses on protein sequence design given a fixed backbone structure, PocketGen co-designs both the amino acid sequence and the full-atom structure of the protein pocket. In our default setting, only one structural adapter was placed after the last layer of pLM. Only the adapter layers are fine-tuned during training, and the other layers of PLMs are frozen to save on computation costs. The structural adapter mainly has the following two parts.

Sequence–structure cross-attention. The structural representation of the ith residue h_i^{struct} is obtained by the mean pooling of H_i from the bilevel graph transformer. In the input to the pLMs, the pocket residue types to be designed are assigned with the mask, and we denote the ith residue representation from pLMs as h_i^{seq}. In the structural adapter, we perform cross-attention between the structural representations $H^{\mathrm{struct}} = \{h_1^{\mathrm{struct}}, h_2^{\mathrm{struct}} \ldots h_{N_s}^{\mathrm{struct}}\}$ and sequence representations $H^{\mathrm{seq}} = \{h_1^{\mathrm{seq}}, h_2^{\mathrm{seq}} \ldots h_{N_s}^{\mathrm{seq}}\}$. The respective query, key and value matrices are obtained as follows:

$$Q = H^{\mathrm{seq}}W_Q, \qquad K = H^{\mathrm{struct}}W_K, \qquad V = H^{\mathrm{struct}}W_V, \tag{14}$$

where W_Q, W_K and $W_V \in \mathbb{R}^{d_h\times d_r}$ are trainable weight matrices. Rotary positional encoding[92] is applied to the representations, and we omit it in the equations for simplicity. The output of the cross-attention is obtained as

$$\mathrm{CrossAttention}(Q,K,V) = \mathrm{Softmax}\left(\frac{QK^\top}{\sqrt{d_r}}\right)V. \tag{15}$$

Bottleneck FFN. A bottleneck FFN is appended after the cross-attention to impose nonlinearity and abstract representations, inspired by previous works[90]. The intermediate dimension of the bottleneck FFN is set to be half of the default representation dimension. Finally, the predicted pocket residue type p_i is obtained by using an MLP on the output residue representation.

Training protocol

Inspired by AlphaFold 2 (ref. 39), we use a recycling strategy for model training. Recycling facilitates the training of deeper networks without incurring extra memory costs by executing multiple forward passes and computing gradients solely for the final pass. The training loss of PocketGen is the weighted sum of the following three losses:

$$\mathcal{L}_{\mathrm{seq}} = \frac{1}{T}\sum_t\sum_i l_{\mathrm{ce}}\left(\hat{p}_i, p_i^t\right), \tag{16}$$

$$\mathcal{L}_{\mathrm{coord}} = \frac{1}{T}\sum_t\left[\sum_i l_{\mathrm{huber}}\left(\hat{X}_i, X_i^t\right) + \sum_j l_{\mathrm{huber}}\left(\hat{x}(\mathbf{v}_j), x^t(\mathbf{v}_j)\right)\right], \tag{17}$$

$$\mathcal{L}_{\mathrm{struct}} = \frac{1}{T}\sum_t\left[\sum_{b\in\mathcal{B}} l_{\mathrm{huber}}(\hat{b}, b^t) + \sum_{\theta\in\Theta} l_{\mathrm{huber}}(\cos\hat{\theta}, \cos\theta^t)\right], \tag{18}$$

$$\mathcal{L} = \mathcal{L}_{\mathrm{seq}} + \lambda_{\mathrm{coord}}\mathcal{L}_{\mathrm{coord}} + \lambda_{\mathrm{struct}}\mathcal{L}_{\mathrm{struct}}, \tag{19}$$

where T is the total refinement rounds. $\hat{p}_i, \hat{X}_i, \hat{x}(\mathbf{v}_j), \hat{b}$ and $\cos\hat{\theta}$ are the ground-truth residue types, residue coordinates and ligand coordinates; bond lengths; and bond/dihedral angles, respectively. $p_i^t, X_i^t, x^t(\mathbf{v}_j), b^t$ and $\cos\theta^t$ are the corresponding predicted ones at the tth round by PocketGen. The sequence loss $\mathcal{L}_{\mathrm{seq}}$ is the cross-entropy loss for pocket-residue-type prediction. The coordinate loss $\mathcal{L}_{\mathrm{coord}}$ uses

Huber loss[93] for the training stability. The structure loss $\mathcal{L}_{\text{struct}}$ is added to supervised bond lengths and bond/dihedral angles for realistic local geometry. $\mathcal{B}$ and Θ denote all the bonds and angles in the protein pocket (including side chains). λ_{coord} and λ_{struct} are hyperparameters balancing the three losses. We perform a grid search over {0.5, 1.0, 2.0, 3.0} and choose these hyperparameters based on the validation performance to select the specific parameter values. In the default setting, we set λ_{coord} to 1.0 and λ_{struct} to 2.0.

Generation protocol

In the generation procedure, PocketGen initializes the sequence with uniform distributions over 20 amino acid types and the coordinates based on linear interpolations and extrapolations. Specifically, we initialize the residue coordinates with linear interpolations and extrapolations based on the nearest residues with known structures in the protein. Denote the sequence of residues as $\mathcal{A} = \mathbf{a}_1 \cdots \mathbf{a}_{N_s}$, where N_s is the length of the sequence. Let $x(\mathbf{a}_{i,1}) \in \mathbb{R}^3$ denote the C_α coordinate of the ith residue. We take the following strategies to determine the C_α coordinate of the ith residue. (1) We use linear interpolation if there are residues with known coordinates at both sides of the ith residue. Specifically, assume p and q are the indexes of the nearest residues with known coordinates at each side of the ith residue ($p < i < q$), we have $x(\mathbf{a}_{i,1}) = \frac{1}{q-p}[(i-p)x(\mathbf{a}_{q,1}) + (q-i)x(\mathbf{a}_{p,1})]$. (2) We conduct linear extrapolation if the ith residue is at the ends of the chain, that is, no residues with known structures at one side of the ith residue. Specifically, let p and q denote the index of the nearest and second-nearest residue with known coordinates. The position of the ith residue can be initialized as $x(\mathbf{a}_{i,1}) = x(\mathbf{a}_{p,1}) + \frac{i-p}{p-q}(x(\mathbf{a}_{p,1}) - x(\mathbf{a}_{q,1}))$. Inspired by previous works[33,34], we initialize the other backbone atom coordinates according to their ideal local coordinates relative to the C_α coordinates. We initialize the side-chain atoms' coordinates with the coordinate of their corresponding C_α value, added with Gaussian noise. We initialize the ligand molecular structure with the reference ligand structure from the dataset. The ligand structure is updated during pocket generation and the updated ligand is used for Vina score calculation.

Since the number of pocket residue types and the number of side-chain atoms are unknown at the beginning of generation, each pocket residue is assigned 14 atoms, the maximum number of atoms for residues. After rounds of refinement by PocketGen, the pocket residue types are predicted, and the full-atom coordinates are determined by mapping the coordinates to the predicted residue types (taking the first n coordinates according to the residue type). In PocketGen, we directly predict the absolute atom coordinates, which reduces the model complexity and flexibly captures atom interactions. We also notice that PocketGen aligns with the recent trend of directly predicting full-atom coordinates. For example, the recent AlphaFold 3 (ref. 94) directly predicts the full-atom coordinates, replacing the AlphaFold 2 structure module that operated on amino-acid-specific frames and side-chain torsion angles, and achieves better performance on protein structure prediction. For generation efficiency, we set the number of refinement rounds to 3.

Experimental setting

Datasets. We consider two widely used datasets for benchmark evaluation. The CrossDocked dataset[42] contains 22.5M protein–molecule pairs generated through cross-docking. Following previous works[24,59,95], we filter out data points with binding pose RMSD greater than 1 Å, leading to a refined subset with around 180k data points. For data splitting, we use MMseqs2 (ref. 96) to cluster data at 30% sequence identity, and randomly draw 100k protein–ligand structure pairs for training and 100 pairs from the remaining clusters for testing and validation, respectively. The Binding MOAD dataset[43] contains around 41k experimentally determined protein–ligand complexes. Following previous work[97], we keep pockets with valid and moderately 'drug-like' ligands with a QED (quantitative estimate of drug-likeness) score of ≥0.3. We further filter the dataset to discard molecules containing atom types ∉ {C, N, O, S, B, Br, Cl, P, I, F} as well as binding pockets with non-standard amino acids. Then, we randomly sample and split the filtered dataset based on the Enzyme Commission number[44] to ensure different sets do not contain proteins from the same main class of the Enzyme Commission number. Finally, we have 40k protein–ligand pairs for training, 100 pairs for validation and 100 pairs for testing. For all the benchmark tasks in this paper, PocketGen and all the other baseline methods are trained with the same data split for a fair comparison. In real-world pocket generation and optimization case studies, the protein structures were downloaded from the PDB[98].

Implementation. Our PocketGen model is trained with the Adam[99] optimizer for 5k iterations, for which the learning rate is 0.0001 and the batch size is 64. We report the results corresponding to the checkpoint with the best validation loss. It takes around 48 h to finish training on one Tesla A100 GPU from scratch. In PocketGen, the number of attention heads is set as 4, the hidden dimension d is set as 128, k is set to 8 to connect the k nearest-neighbouring residues to build $\mathcal{E}$ and k' is set as 3 to encourage sparsity in the attention matrix. For all the benchmark tasks of pocket generation and optimization, PocketGen and all the other baseline methods are trained with the same data split for a fair comparison. We follow the implementation codes provided by the authors to obtain the results of the baseline methods. Supplementary Algorithms 1 and 2 show the pseudo-codes of the training and generation process of PocketGen.

Baseline methods. PocketGen is compared with five state-of-the-art representative baseline methods. PocketOptimizer[18] is a physics-based method that optimizes energies such as packing- and binding-related energies for ligand-binding protein design. Following the suggestion of the paper, we fixed the backbone structures. DEPACT[17] is a template-matching method that follows a two-step strategy[100] for pocket design. It first searches the protein–ligand complexes in the database with similar ligand fragments. It then grafts the associated residues into the protein scaffold to output the complete protein structure with PACMatch[17]. Both backbone and side-chain structures are changed in DEPACT. RFdiffusion[26], RFAA[16], FAIR[24] and dyMEAN[25] are deep-learning-based models for protein generation. RFdiffusion does not explicitly model protein–ligand interactions and is not directly applicable to small-molecule binding protein generation. Following the suggestions in RFdiffusion[26] and RFAA[16], we use a heuristic attractive–repulsive potential to encourage the formation of pockets with shape complementarity to a target molecule. The residue sequence for the generated protein by RFdiffusion is derived with ProteinMPNN, and the side-chain conformation is decided with Rosetta[101] side-chain packing. RFAA is the latest version of RFdiffusion, which can directly generate protein structures surrounding small molecules by combining the residue-based representation of amino acids with the atomic representation of small molecules. For RFdiffusion and RFAA, we let them paint the pocket area to obtain a consistent setting with other methods for comparison. We also note that RFdiffusion and RFAA do not provide the training/fine-tuning scripts; therefore, we use the provided pretrained checkpoints for all the related experiments in our paper. FAIR[24] was specially designed for full-atom protein pocket design via iterative refinement. dyMEAN[25] was originally proposed for full-atom antibody design, and we adapted it to our pocket design task with proper modifications. Detailed information on baselines is included in Supplementary Notes 1–4. The setting of the key hyperparameters is summarized in Supplementary Table 6. All the baselines are run on the same Tesla A100 GPU for a fair comparison with our PocketGen data.

Data availability

This study's training and test data are available via Zenodo at https://doi.org/10.5281/zenodo.10125312 (ref. 102). The project website for

PocketGen is https://zitniklab.hms.harvard.edu/projects/PocketGen. Source data are provided with this paper.

Code availability

The source code for this study is freely available via GitHub at https://github.com/zaixizhang/PocketGen and via Zenodo at https://doi.org/10.5281/zenodo.13762085 (ref. 103).

References

1. Tinberg, C. E. et al. Computational design of ligand-binding proteins with high affinity and selectivity. *Nature* **501**, 212–216 (2013).
2. Kroll, A., Ranjan, S., Engqvist, M. K. & Lercher, M. J. A general model to predict small molecule substrates of enzymes based on machine and deep learning. *Nat. Commun.* **14**, 2787 (2023).
3. Lee, G. R. et al. Small-molecule binding and sensing with a designed protein family. Preprint at *bioRxiv* https://doi.org/10.1101/2023.11.01.565201 (2023).
4. Qiao, Z., Nie, W., Vahdat, A., Miller III, T. F. & Anandkumar, A. State-specific protein–ligand complex structure prediction with a multiscale deep generative model. *Nat. Mach. Intell.* **6**, 195–208 (2024).
5. Jiang, L. et al. De novo computational design of retro-aldol enzymes. *Science* **319**, 1387–1391 (2008).
6. Röthlisberger, D. et al. Kemp elimination catalysts by computational enzyme design. *Nature* **453**, 190–195 (2008).
7. Dou, J. et al. De novo design of a fluorescence-activating β-barrel. *Nature* **561**, 485–491 (2018).
8. Yeh, A. H.-W. et al. De novo design of luciferases using deep learning. *Nature* **614**, 774–780 (2023).
9. Beltrán, J. et al. Rapid biosensor development using plant hormone receptors as reprogrammable scaffolds. *Nat. Biotechnol.* **40**, 1855–1861 (2022).
10. Bick, M. J. et al. Computational design of environmental sensors for the potent opioid fentanyl. *eLife* **6**, e28909 (2017).
11. Glasgow, A. A. et al. Computational design of a modular protein sense-response system. *Science* **366**, 1024–1028 (2019).
12. Herud-Sikimić, O. et al. A biosensor for the direct visualization of auxin. *Nature* **592**, 768–772 (2021).
13. Polizzi, N. F. & DeGrado, W. F. A defined structural unit enables de novo design of small-molecule–binding proteins. *Science* **369**, 1227–1233 (2020).
14. Basanta, B. et al. An enumerative algorithm for de novo design of proteins with diverse pocket structures. *Proc. Natl Acad. Sci. USA* **117**, 22135–22145 (2020).
15. Dou, J. et al. Sampling and energy evaluation challenges in ligand binding protein design. *Protein Sci.* **26**, 2426–2437 (2017).
16. Krishna, R. et al. Generalized biomolecular modeling and design with RoseTTAFold All-Atom. *Science* **384**, eadl2528 (2024).
17. Chen, Y., Chen, Q. & Liu, H. DEPACT and PACMatch: a workflow of designing de novo protein pockets to bind small molecules. *J. Chem. Inf. Model.* **62**, 971–985 (2022).
18. Noske, J., Kynast, J. P., Lemm, D., Schmidt, S. & Höcker, B. PocketOptimizer 2.0: a modular framework for computer-aided ligand-binding design. *Protein Sci.* **32**, e4516 (2023).
19. Malisi, C. et al. Binding pocket optimization by computational protein design. *PLoS ONE* **7**, e52505 (2012).
20. Stiel, A. C., Nellen, M. & Höcker, B. PocketOptimizer and the design of ligand binding sites. In *Computational Design of Ligand Binding Proteins. Methods in Molecular Biology* Vol. 1414 (Humana Press, 2016).
21. Lu, L. et al. De novo design of drug-binding proteins with predictable binding energy and specificity. *Science* **384**, 106–112 (2024).
22. Byon, W., Garonzik, S., Boyd, R. A. & Frost, C. E. Apixaban: a clinical pharmacokinetic and pharmacodynamic review. *Clin. Pharmacokinet.* **58**, 1265–1279 (2019).
23. Stark, H., Jing, B., Barzilay, R. & Jaakkola, T. Harmonic prior self-conditioned flow matching for multi-ligand docking and binding site design. In *NeurIPS 2023 AI for Science Workshop* (Curran Associates, 2023).
24. Zhang, Z., Lu, Z., Hao, Z., Zitnik, M. & Liu, Q. Full-atom protein pocket design via iterative refinement. In *Thirty-Seventh Conference on Neural Information Processing Systems* (Curran Associates, 2023).
25. Kong, X., Huang, W. & Liu, Y. End-to-end full-atom antibody design. In *Proc. 40th International Conference on Machine Learning* 718 (JMLR.org, 2023).
26. Watson, J. L. et al. De novo design of protein structure and function with RFdiffusion. *Nature* **620**, 1089–1100 (2023).
27. Ho, J., Jain, A. & Abbeel, P. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.* **33**, 6840–6851 (2020).
28. Baek, M. et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **373**, 871–876 (2021).
29. Dauparas, J. et al. Robust deep learning–based protein sequence design using ProteinMPNN. *Science* **378**, 49–56 (2022).
30. Dauparas, J. et al. Atomic context-conditioned protein sequence design using LigandMPNN. Preprint at *bioRxiv* https://doi.org/10.1101/2023.12.22.573103 (2023).
31. Jin, W., Wohlwend, J., Barzilay, R. & Jaakkola, T. Iterative refinement graph neural network for antibody sequence-structure co-design. In *International Conference on Learning Representations* (ICLR, 2022).
32. Jin, W., Barzilay, R. & Jaakkola, T. Antibody-antigen docking and design via hierarchical structure refinement. In *Proc. 39th International Conference on Machine Learning* 10217–10227 (PMLR, 2022).
33. Luo, S. et al. Antigen-specific antibody design and optimization with diffusion-based generative models. In *Advances in Neural Information Processing Systems* 9754–9767 (Curran Associates, 2022).
34. Kong, X., Huang, W. & Liu, Y. Conditional antibody design as 3D equivariant graph translation. In *The Eleventh International Conference on Learning Representations* (ICLR, 2023).
35. Shi, C., Wang, C., Lu, J., Zhong, B. & Tang, J. Protein sequence and structure co-design with equivariant translation. In *The Eleventh International Conference on Learning Representations* (ICLR, 2023).
36. Anishchenko, I. et al. De novo protein design by deep network hallucination. *Nature* **600**, 547–552 (2021).
37. Yang, J. et al. Improved protein structure prediction using predicted interresidue orientations. *Proc. Natl Acad. Sci. USA* **117**, 1496–1503 (2020).
38. Cao, L. et al. Design of protein-binding proteins from the target structure alone. *Nature* **605**, 551–560 (2022).
39. Jumper, J. et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).
40. Rives, A. et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl Acad. Sci. USA* **118**, e2016239118 (2019).
41. Lin, Z. et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. Preprint at *bioRxiv* https://doi.org/10.1101/2022.07.20.500902 (2022).
42. Francoeur, P. G. et al. Three-dimensional convolutional neural networks and a cross-docked data set for structure-based drug design. *J. Chem. Inf. Model.* **60**, 4200–4215 (2020).
43. Hu, L., Benson, M. L., Smith, R. D., Lerner, M. G. & Carlson, H. A. Binding MOAD (mother of all databases). *Proteins* **60**, 333–340 (2005).
44. Bairoch, A. The ENZYME database in 2000. *Nucleic Acids Res.* **28**, 304–305 (2000).
45. Marcou, G. & Rognan, D. Optimizing fragment and scaffold docking by use of molecular interaction fingerprints. *J. Chem. Inf. Model.* **47**, 195–207 (2007).

46. Trott, O. & Olson, A. J. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J. Comput. Chem.* **31**, 455–461 (2010).
47. Yang, M. et al. Uni-GBSA: an open-source and web-based automatic workflow to perform MM/GB(PB)SA calculations for virtual screening. *Brief. Bioinform.* **24**, bbad218 (2023).
48. Friesner, R. A. et al. Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. *J. Med. Chem.* **47**, 1739–1749 (2004).
49. Lin, Z. et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379**, 1123–1130 (2023).
50. Trippe, B. L. et al. Diffusion probabilistic modeling of protein backbones in 3D for the motif-scaffolding problem. In *The Eleventh International Conference on Learning Representations* (ICLR, 2023).
51. Lin, Y. & Alquraishi, M. Generating novel, designable, and diverse protein structures by equivariantly diffusing oriented residue clouds. In *Proc. 40th International Conference on Machine Learning* 20978–21002 (PMLR, 2023).
52. Zhang, Y. & Skolnick, J. Scoring function for automated assessment of protein structure template quality. *Proteins* **57**, 702–710 (2004).
53. Yim, J. et al. Improved motif-scaffolding with SE(3) flow matching. *Transactions on Machine Learning Research* (2024).
54. Yim, J. et al. SE(3) diffusion model with application to protein backbone generation. In *Proc. 40th International Conference on Machine Learning* 40001–40039 (PMLR, 2023).
55. Tibshirani, R. J. & Efron, B. *An Introduction to the Bootstrap* Vol. 57 (Chapman & Hall, 1993).
56. Yoo, Y. J., Feng, Y., Kim, Y.-H. & Yagonia, C. F. J. *Fundamentals of Enzyme Engineering* (Springer, 2017).
57. Traut, T. W. Protein engineering: principles and practice. *Am. Sci.* **85**, 571–573 (1997).
58. Spencer, R. K. et al. Stereochemistry of polypeptoid chain configurations. *Biopolymers* **110**, e23266 (2019).
59. Peng, X. et al. Pocket2Mol: efficient molecular sampling based on 3D protein pockets. In *Proc. 39th International Conference on Machine Learning* 17644–17655 (PMLR, 2022).
60. Zhang, Z., Liu, Q., Lee, C.-K., Hsieh, C.-Y. & Chen, E. An equivariant generative framework for molecular graph-structure co-design. *Chem. Sci.* **14**, 8380–8392 (2023).
61. Kaplan, J. et al. Scaling laws for neural language models. Preprint at https://arxiv.org/abs/2001.08361 (2020).
62. Alberts, B. *Molecular Biology of the Cell* (Garland Science, 2017).
63. Shoichet, B. K. Virtual screening of chemical libraries. *Nature* **432**, 862–865 (2004).
64. Ertl, P. An algorithm to identify functional groups in organic molecules. *J. Cheminform.* **9**, 36 (2017).
65. Buttenschoen, M., Morris, G. M. & Deane, C. M. PoseBusters: AI-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chem. Sci.* **15**, 3130–3139 (2024).
66. Yang, Z., Zhong, W., Lv, Q., Dong, T. & Yu-Chian Chen, C. Geometric interaction graph neural network for predicting protein–ligand binding affinities from 3D structures (GIGN). *J. Phys. Chem. Lett.* **14**, 2020–2033 (2023).
67. Maier, J. A. et al. ff14SB: improving the accuracy of protein side chain and backbone parameters from ff99SB. *J. Chem. Theory Comput.* **11**, 3696–3713 (2015).
68. Shapovalov, M. V. & Dunbrack Jr, R. L. A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions. *Structure* **19**, 844–858 (2011).
69. Satorras, V. G., Hoogeboom, E., Fuchs, F. B., Posner, I. & Welling, M. E(*n*) equivariant normalizing flows. In *35th Conference on Neural Information Processing Systems* (2021).
70. Jing, B., Eismann, S., Suriana, P., Townshend, R. J. & Dror, R. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations* (ICLR, 2021).
71. Huang, W. et al. Equivariant graph mechanics networks with constraints. In *International Conference on Learning Representations* (ICLR, 2022).
72. Eronen, V. et al. Structural insight to elucidate the binding specificity of the anti-cortisol Fab fragment with glucocorticoids. *J. Struct. Biol.* **215**, 107966 (2023).
73. Pinto, D. J. et al. Discovery of 1-(4-methoxyphenyl)-7-oxo-6-(4-(2-oxopiperidin-1-yl)phenyl)-4,5,6,7-tetrahydro-1Hpyrazolo[3,4-c] pyridine-3-carboxamide (apixaban, BMS-562247), a highly potent, selective, efficacious, and orally bioavailable inhibitor of blood coagulation factor Xa. *J. Med. Chem.* **50**, 5339–5356 (2007).
74. Hernandez, I., Zhang, Y. & Saba, S. Comparison of the effectiveness and safety of apixaban, dabigatran, rivaroxaban, and warfarin in newly diagnosed atrial fibrillation. *Am. J. Cardiol.* **120**, 1813–1819 (2017).
75. Stanley, T. H. The fentanyl story. *J. Pain* **15**, 1215–1226 (2014).
76. Salentin, S., Schreiber, S., Haupt, V. J., Adasme, M. F. & Schroeder, M. PLIP: fully automated protein–ligand interaction profiler. *Nucleic Acids Res.* **43**, W443–W447 (2015).
77. Yang, J., Li, F.-Z. & Arnold, F. H. Opportunities and challenges for machine learning-assisted enzyme engineering. *ACS Cent. Sci.* **10**, 226–241 (2024).
78. Zhou, Y., Pan, Q., Pires, D. E., Rodrigues, C. H. & Ascher, D. B. DDMut: predicting effects of mutations on protein stability using deep learning. *Nucleic Acids Res.* **51**, W122–W128 (2023).
79. Hu, E. et al. Discovery of aryl aminoquinazoline pyridones as potent, selective, and orally efficacious inhibitors of receptor tyrosine kinase c-Kit. *J. Med. Chem.* **51**, 3065–3068 (2008).
80. Feng, W. et al. Generation of 3D molecules in pockets via a language model. *Nat. Mach. Intell.* **6**, 62–73 (2024).
81. Zhang, O. et al. ResGen is a pocket-aware 3D molecular generation model based on parallel multiscale modelling. *Nat. Mach. Intell.* **5**, 1020–1030 (2023).
82. Jiang, Y. et al. PocketFlow is a data-and-knowledge-driven structure-based molecular generative model. *Nat. Mach. Intell.* **6**, 326–337 (2024).
83. Zhang, Z. & Liu, Q. Learning subpocket prototypes for generalizable structure-based drug design. In *Proc. 40th International Conference on Machine Learning* 41382–41398 (PMLR, 2023).
84. Kalliokoski, T., Olsson, T. S. & Vulpetti, A. Subpocket analysis method for fragment-based drug discovery. *J. Chem. Inf. Model.* **53**, 131–141 (2013).
85. Kong, X. et al. Generalist equivariant transformer towards 3D molecular interaction learning. In *Proc. 41st International Conference on Machine Learning* 25149–25175 (2024).
86. Vaswani, A. et al. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30** (2017).
87. Ba, J. L., Kiros, J. R. & Hinton, G. E. Layer normalization. Preprint at https://arxiv.org/abs/1607.06450 (2016).
88. Igashov, I. et al. Equivariant 3D-conditional diffusion model for molecular linker design. *Nat. Mach. Intell.* **6**, 417–427 (2024).
89. Batzner, S. et al. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nat. Commun.* **13**, 2453 (2022).
90. Houlsby, N. et al. Parameter-efficient transfer learning for NLP. In *Proc. 36th International Conference on Machine Learning* 2790–2799 (PMLR, 2019).

91. Zheng, Z. et al. Structure-informed language models are protein designers. In *International Conference on Machine Learning* 42317–42338 (PMLR, 2023).
92. Su, J. et al. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing* **568**, 127063 (2024).
93. Huber, P. J. Robust estimation of a location parameter. in *Breakthroughs in Statistics: Methodology and Distribution* 492–518 (Springer, 1992).
94. Abramson, J. et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature* **630**, 493–500 (2024).
95. Luo, S., Guan, J., Ma, J. & Peng, J. A 3D generative model for structure-based drug design. *NeurIPS* **34**, 6229–6239 (2021).
96. Steinegger, M. & Söding, J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat. Biotechnol.* **35**, 1026–1028 (2017).
97. Schneuing, A. et al. Structure-based drug design with equivariant diffusion models. Preprint at https://arxiv.org/abs/2210.13695 (2022).
98. Sussman, J. L. et al. Protein Data Bank (PDB): database of three-dimensional structural information of biological macromolecules. *Acta Cryst.* **D54**, 1078–1084 (1998).
99. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. Preprint at https://arxiv.org/abs/1412.6980 (2014).
100. Zanghellini, A. et al. New algorithms and an in silico benchmark for computational enzyme design. *Protein Sci.* **15**, 2785–2794 (2006).
101. Alford, R. F. et al. The Rosetta all-atom energy function for macromolecular modeling and design. *J. Chem. Theory Comput.* **13**, 3031–3048 (2017).
102. Zhang, Z. PocketGen datasets. *Zenodo* https://doi.org/10.5281/zenodo.10125312 (2024).
103. Zhang, Z. PocketGen. *Zenodo* https://doi.org/10.5281/zenodo.13762085 (2024).

Acknowledgements

Z.Z. gratefully acknowledges grants from the National Natural Science Foundation of China (no. 623B2095) and the Excellent PhD Students Overseas Study Program of the University of Science and Technology of China. Q.L. gratefully acknowledges grants from the National Natural Science Foundation of China (no. 62337001) and the Fundamental Research Funds for the Central Universities. W.X.S. and M.Z. gratefully acknowledge support from NIH grant no. R01-HD108794, NSF CAREER grant no. 2339524, US DoD grant no. FA8702-15-D-0001, and awards from Harvard Data Science Initiative, Amazon Faculty Research, Google Research Scholar Program, AstraZeneca Research, Roche Alliance with Distinguished Scientists, Sanofi iDEA-iTECH Award, Pfizer Research, Chan Zuckerberg Initiative, John and Virginia Kaneb Fellowship award at Harvard Medical School, Biswas Computational Biology Initiative in partnership with the Milken Institute, and Kempner Institute for the Study of Natural and Artificial Intelligence at Harvard University. We thank E. Chen, Y. Chen and H. Liu from the University of Science and Technology of China for their constructive discussions on implementing and evaluating baseline methods, which greatly helped this research.

Author contributions

Z.Z., Q.L. and M.Z. designed the research. Z.Z. conducted the experiments. Z.Z., W.X.S., Q.L. and M.Z. analysed the results. All authors wrote and reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

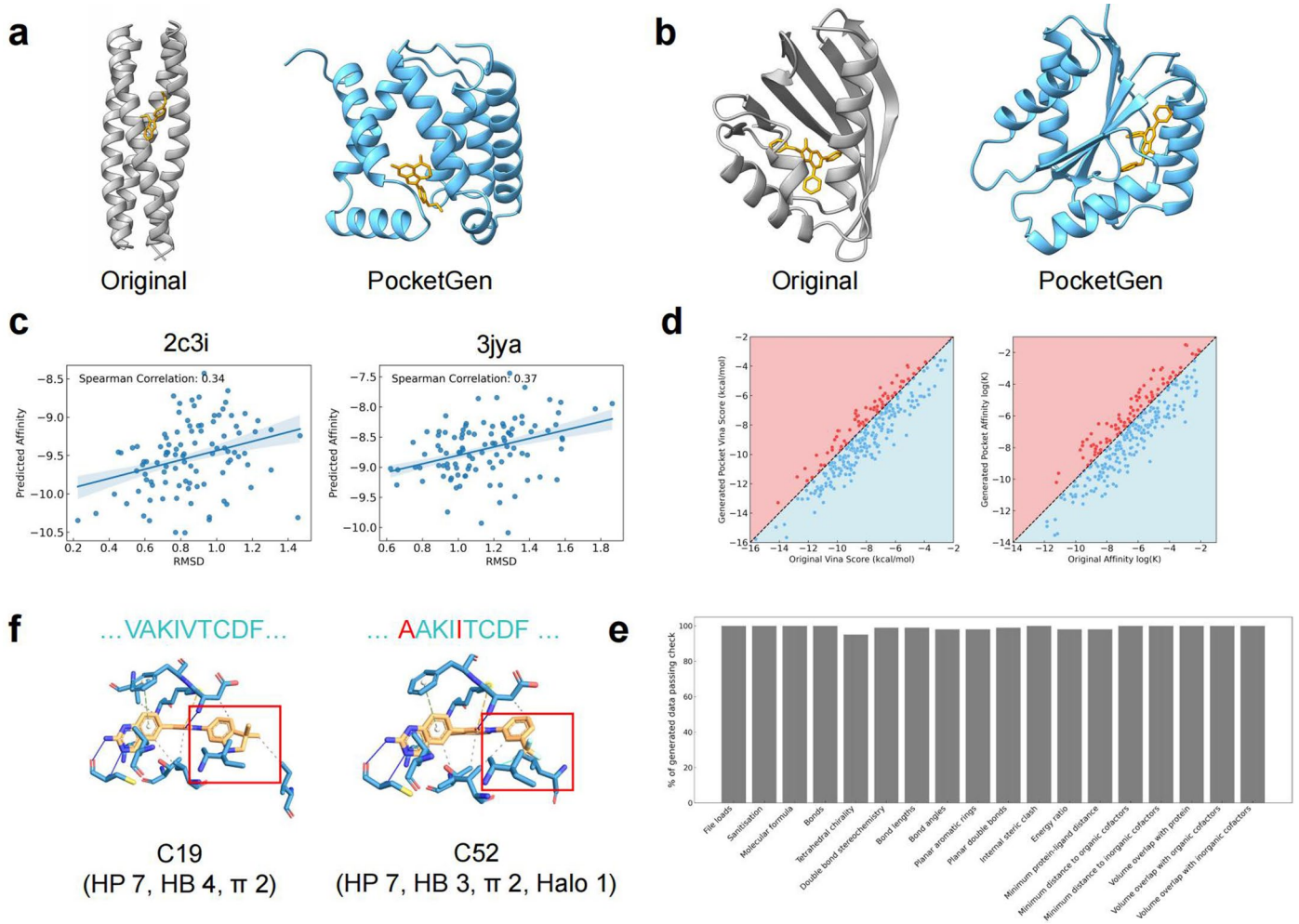
Extended data is available for this paper at https://doi.org/10.1038/s42256-024-00920-9.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-024-00920-9.

Correspondence and requests for materials should be addressed to Qi Liu or Marinka Zitnik.

Peer review information *Nature Machine Intelligence* thanks Fergus Boyles, Shuiwang Ji and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

https://doi.org/10.1038/s42256-024-00920-9



Extended Data Fig. 1 | More case studies and evaluations of PocketGen. a, The originally designed protein binder for Rucaparib[21] (left panel) and the generated protein binder by PocketGen (right panel). **b**, The originally designed protein binder for DTZ[8] (left panel) and the generated protein binder by PocketGen (right panel). Note that PocketGen generates the whole protein instead of the pocket region in **a**&**b**. The generated protein binder has high scTM scores (0.900 and 0.976). **c**, The predicted affinity (log K) by GIGN[66] of the generated pockets by PocketGen with respect to RMSD. We randomly select two protein-ligand complexes from PDBBind (PDB id 2c3i and 3jya). **d**, The Vina score/binding affinity (log K) of the generated pockets by PocketGen and the original pockets from PDBBind. The black region/dots indicate the generated pockets have higher affinities than the original pockets while the red region/dots indicate lower affinities. **f**, The generated interactions by PocketGen with respect to a pair of activity cliff ligand molecules, that is, C19 and C52[79]. As marked with red rectangles, PocketGen adaptively generates different interactions for different molecular fragments (hydrogen bonds+hydrophobic interactions and halogen bonds respectively). 'HP' indicates hydrophobic interactions, 'HB' signifies hydrogen bonds, 'π' denotes the π-stacking/cation interactions, and 'Halo' indicates the Halogen bonds. **e**, Detailed validity check with PoseBusters on CrossDocked and Binding MOAD.