

State-specific protein-ligand complex structure prediction with a multi-scale deep generative model

Zhuoran Qiao^{1,*}, Weili Nie³, Arash Vahdat³, Thomas F. Miller III^{1,4}, and Animashree Anandkumar^{2,3}

¹Division of Chemistry and Chemical Engineering, California Institute of Technology, Pasadena, CA 91125

²Division of Engineering and Applied Science, California Institute of Technology, Pasadena, CA 91125

³Nvidia Corporation, Santa Clara, CA 95051

⁴Entos, Inc., Los Angeles, CA 90027

*Current e-mail address: zhuoran.qiao@entos.ai.

ABSTRACT

The binding complexes formed by proteins and small molecule ligands are ubiquitous and critical to life. Despite recent advancements in protein structure prediction, existing algorithms are so far unable to systematically predict the binding ligand structures along with their regulatory effects on protein folding. To address this discrepancy, we present NeuralPLexer, a computational approach that can directly predict protein-ligand complex structures solely using protein sequence and ligand molecular graph inputs. NeuralPLexer adopts a deep generative model to sample the three-dimensional (3D) structures of the binding complex and their conformational changes at an atomistic resolution. The generative model is based on a diffusion process that incorporates essential biophysical constraints and a multi-scale geometric deep learning system to iteratively sample residue-level contact maps and all heavy-atom coordinates in a hierarchical manner. NeuralPLexer achieves state-of-the-art performance compared to all existing methods on benchmarks for both protein-ligand blind docking and flexible binding site structure recovery. Moreover, owing to its specificity in sampling both ligand-free-state and ligand-bound-state ensembles, NeuralPLexer consistently outperforms AlphaFold2 in terms of global protein structure prediction accuracy on both representative structure pairs with large conformational changes (average TM-score=0.93) and recently determined ligand-binding proteins (average TM-score=0.89). Case studies reveal that the predicted conformational variations are consistent with structure determination experiments for important targets, including human KRAS^{G12C}, ketol-acid reductoisomerase, and purine GPCRs. Our study suggests that a data-driven approach can capture the structural cooperativity between proteins and small molecules, showing promise in accelerating the design of enzymes, drug molecules, and beyond.

Main

Deep learning has enabled great advancements in predicting protein structures from their one-dimensional amino acid sequences. State-of-the-art protein structure prediction networks such as AlphaFold2 [1] (AF2) adopt prediction pipelines based on evolutionary, physical, and geometric constraints of protein structures. Specifically, evolutionary constraints extracted from multiple sequence alignments (MSAs) [2–5] or protein language models (PLMs) [6–9] and specialized neural networks are systematically combined with the sequence-based information and geometrical representations to enable end-to-end 3D structure prediction. Despite remarkable success in predicting protein crystal structures, such a single-structure formulation of the protein folding problem provides incomplete information about protein function. It is also often found insufficient for structure-based drug design [10, 11]. As discussed by numerous studies [12–16], protein structures are dynamically modulated by their interactions with small molecules such as ligands and post-translational modifications. Such dynamical protein conformational changes occur on vastly varied kinetic timescales ranging from nanosecond-scale vibrations to millisecond-scale collective motions, and can trigger diverse downstream responses that are crucial to the regulation of biological functions. Proposing ligands that selectively target protein conformations has also become an increasingly important strategy in small molecule-based therapeutics [17–19].

Nevertheless, the computational modeling of protein-ligand complexes that are coupled to significant changes in receptor conformation is hampered by the prohibitive cost of simulating slow protein state transitions [20, 21]. In the past decades, several schemes have been proposed to remedy these issues; those include methods based on molecular dynamics simulation and enhanced sampling techniques [10, 22–24], molecular docking guided by template-based modeling [25–27] and iterative refinement protocols [28–31], as well as recently proposed modifications to structure prediction networks [32–35]. However, such methods often require case-specific expert interventions or constraints from experimental data, and are still not a unified

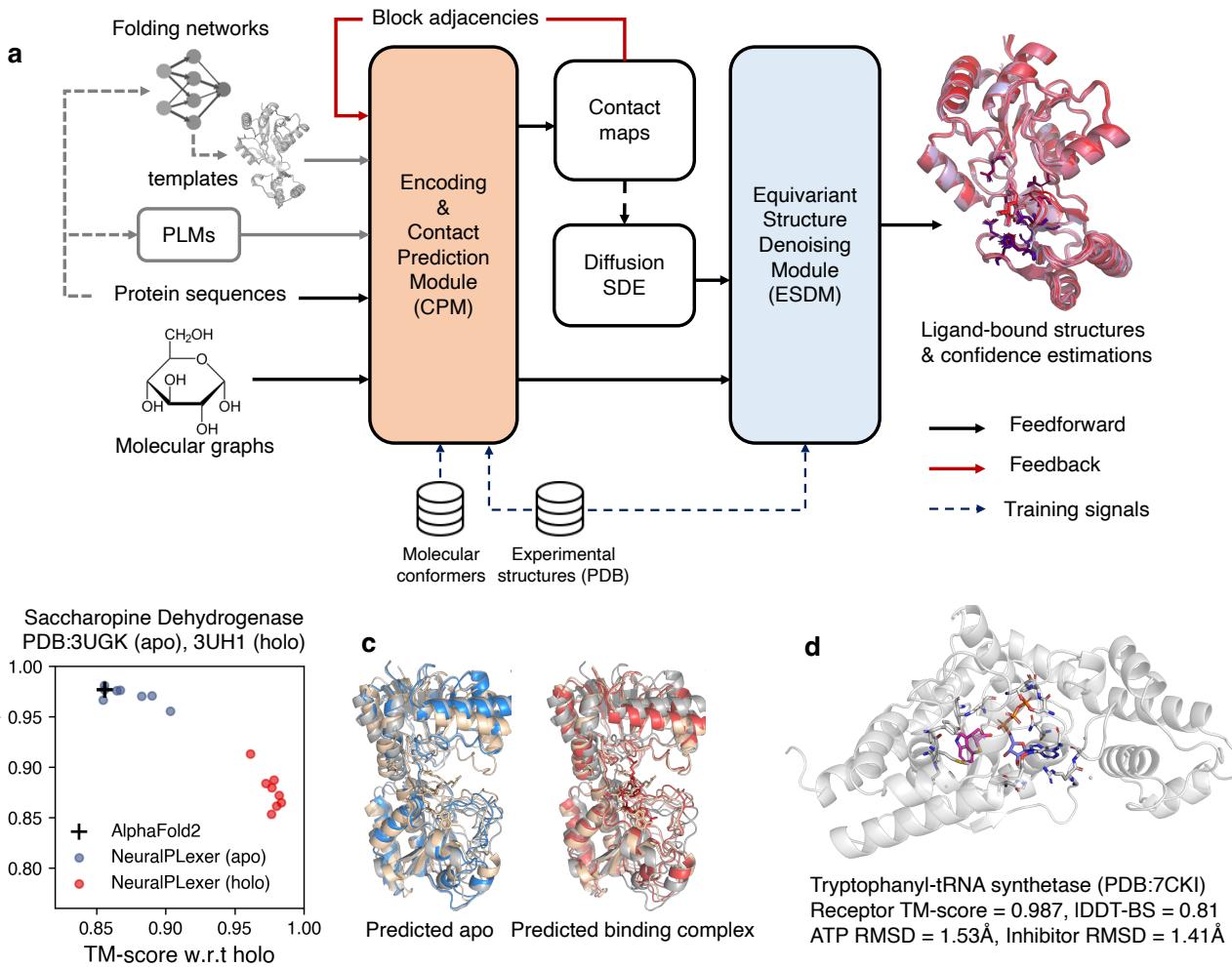


Figure 1. NeuralPLexer enables accurate prediction of protein-ligand complex structure and conformational changes. (a) Method overview. To perform predictions, the input protein sequence is first used to retrieve protein language model (PLM) features and structure templates; NeuralPLexer then combines the set of PLM and template features with molecular graph representations of the input ligands to directly sample an ensemble of binding complex structures via a multi-scale generative model. The main network of NeuralPLexer is comprised of a coarse-grained, auto-regressive contact prediction module (CPM) and an atomistic, diffusion-based equivariant structure denoising module (ESDM). SDE: stochastic differential equation. (b-c) Prediction example on a target with large-scale domain motions upon ligand binding (UniProt:P38998). (b) The structure similarities against experimental apo (i.e., ligand-free protein, PDB:3UGK) and holo (i.e., ligand-bound protein, PDB:3UH1) structures measured by TM-score are plotted for AlphaFold2 predictions (grey), ligand-free NeuralPLexer predictions (blue), and ligand-bound NeuralPLexer predictions (red). (c) Visualizations of representative NeuralPLexer-predicted structures (blue for apo, red for holo) are overlaid with the experimental apo structure (grey) and the holo structure (light yellow). (d) Visualization of a prediction example (PDB:7CKI, UniProt:P00953) for which NeuralPLexer achieves high atomic accuracy for both the ATP (blue) and an inhibitor bound to the tryptophan site (magenta) upon an induced-fit structure rearrangement.

framework to systematically predict binding complex structures at a proteome scale. Moreover, the applicability of existing deep-learning-based algorithms to ligand-binding proteins is limited by their single structure regression-based formulations, especially for non-endogenous small molecule binding [36] for which the ligand identity cannot be inferred from protein motifs.

Recent developments in generative deep learning have provided an alternative paradigm, and they have demonstrated substantial progress in understanding complex vision and language domains [37–40]. Two notable strategies for generative modeling include (a) auto-regressive models as widely adopted in transformer [41] networks for sequence data such as natural language and genomic [42, 43] sequences based on a sequential sampling process from learned token distributions; and (b) diffusion-based generative models that leverage a stochastic process to generate data by sampling from a prior distribution and using a neural network to progressively reverse the noising process. Several works have demonstrated that deep generative models are capable of producing de novo-designed proteins with experimentally validated functions, for example, some works use language models for protein sequence design (such as ESM [9, 44] and ProteinMPNN [45]) and others have applied diffusion models for protein backbone generation [46–49]. Despite so, these works have not accounted for small molecules when generating protein sequences or 3D structures. Other studies [50–56] have shown that diffusion models are effective in modeling molecular structures beyond protein backbones, especially in the context of molecular docking and structure-based drug design. However, no work has so far developed generative models capable of directly predicting binding complex structures at an atomistic resolution and with an accuracy comparable to structure determination experiments.

In this study, we conclusively tackle this challenge through a deep generative model informed by biophysical inductive biases. We demonstrate that the integration of these inductive biases into a deep neural network, which combines both auto-regressive and diffusion generative modeling, is key to accurately predicting protein-ligand complex structures at scale. This is aligned with two essential factors that dictate ligand binding: (a) the determination of the global contextual information related to ligand function, such as selectivity to orthosteric or allosteric sites, and (b) the process of resolving energetically favorable inter-atomic structures based on sub-nanometer-scale physical interactions.

Our framework, known as NeuralPLexer, is a computational system that predicts protein-ligand complex structures using an end-to-end generative modeling strategy. The presented method directly generates a structural ensemble of binding complexes given protein sequence and ligand molecular graphs inputs, conditioning on auxiliary features obtained from PLMs and template protein structures retrieved from experimentally resolved homologs or computational models. Both the prediction pipeline and the underlying neural network architecture are designed to mirror the multi-scale hierarchical organization of biomolecular complexes. In particular, NeuralPLexer comprises of (a) a graph-based network to encode the atomic scale chemical and geometrical features of individual small molecules and amino-acid graphs into tensor representations, enabled by a physics-inspired network architecture that is trained on million-scale molecular conformation and bio-activity databases; (b) a contact prediction module (CPM) to generate residue-scale inter-molecular distance distributions, coarse-grained contact maps, and associated pair representations using an attention-based network, motivated by recent vision-language models and folding prediction networks [1, 57]; and (c) an equivariant structure denoising module (ESDM) to generate the binding complex atomistic structure conditioned on outputs from the atomic-scale and residue-scale networks, using a structured denoising diffusion process that is equivariant and preserves chirality constraints of the protein and ligand molecules.

When benchmarked on blind protein-ligand docking, NeuralPLexer improves the ligand pose accuracy by up to 78% compared to the best-performing existing method on the PDBBind2020 dataset [58]. In the context of ligand binding site design, where to our knowledge no existing machine learning approaches are applicable, NeuralPLexer effectively recovers up to 46% of binding site structures by solely using truncated scaffolds generated from AF2. This represents an at least 59% improvement in success rate when compared to the physics-based method in Rosetta [59]. On two carefully-curated benchmark datasets of ligand-binding proteins with large structural plasticity, NeuralPLexer outperforms the state-of-the-art protein structure prediction algorithm AF2 as demonstrated by the highest TM-score [60] (0.906 on average) and by 11–13% improvements in terms of accuracy on domains that undergo substantial conformational changes upon ligand binding. The above results suggest that our strategy enjoys a systematic advantage over existing methods in selectively predicting protein structures that are subject to induced-fit binding or conformational selection. The versatile capability of NeuralPLexer to model ligand binding and protein structural variations can enable rapid characterization of conformational landscapes to facilitate a better understanding of molecular mechanisms governing protein functions, thus aiding the proteome-scale identification of unconventional targets for therapeutic interventions and protein engineering.

Results

NeuralPLexer

The NeuralPLexer network architecture and the information flows during model training and inference are illustrated in Figure 1a. The primary model inputs are a set of protein amino acid sequences $\{s\}$ (containing a single chain for monomers and multiple chains for protein complexes) and, when the binding ligands (one or multiple molecules) are present, a set of molecular graphs $\{G\}$ containing atomic numbers, bond types and stereochemistry labels (e.g., tetrahedral or E/Z isomerism [61]). To

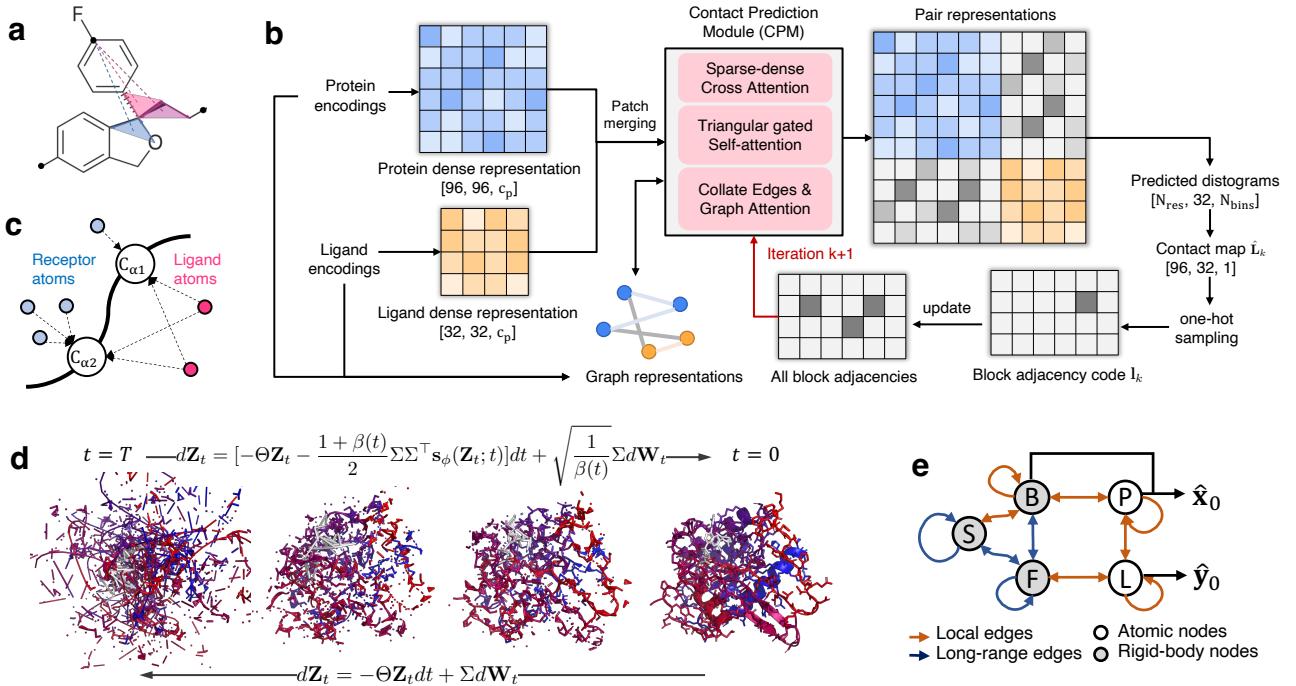


Figure 2. Architecture details. (a) Ligand molecules and amino acids are encoded as the collection of atoms, local coordinate frames (depicted as semi-transparent triangles), and stereochemistry-specific pairwise embeddings (depicted as dashed lines) representing their interactions. (b) Information flow in the contact prediction module (CPM) network. The CPM network samples block-wise adjacency matrices among protein and ligand nodes using an autoregressive decoding scheme, where the block adjacency matrices I_k sampled from last step are passed to the network to update the predicted histograms of pairwise distances ("distograms") and contact maps \hat{L} . (c) The forward-time SDE introduces structured drift and noise covariance terms among protein C α atoms, non-C α atoms, and ligand atoms. (d) Denoising diffusion process to generate the binding complex 3D atomistic structure. The protein (colored as red-blue from N- to C-terminus) and ligand (colored as grey) structures are jointly generated through a reverse-time, simulated annealing SDE starting from randomly initialized coordinate variables. (e) Information flow in the ESDM neural network. The ESDM network operates on a heterogeneous graph formed by protein atoms (P), ligand atoms (L), protein backbone frames for all residues (B), backbone frames of the selected patches (S), and ligand local frames (F) to predict the denoised atomic coordinates \hat{x}_0, \hat{y}_0 used in the reverse-time SDE. The heterogeneous graph comprises randomized local edges (orange arrows) and densely connected long-range edges (blue arrows), where the long-range edges and inter-residue local edges are initialized via the CPM embeddings.

perform structure prediction, NeuralPLexer jointly samples the 3D heavy-atom coordinates of the protein \mathbf{x} and those of the ligands \mathbf{y} from a generative model conditioned on the sequence and graph inputs $\{\mathbf{s}\}, \{G\}$. In addition to the primary sequence and graph inputs, we retrieved inputs from readily available transformer protein language models and templates from alternative experimental structures or protein structure prediction networks to provide extra conditioning signals to the generative model. In particular, we use protein sequence embeddings from the ESM-2 [9] language model and template structures generated from AlphaFold2 as auxiliary inputs in this study.

During inference, the input protein sequences, retrieved auxiliary protein embeddings, and molecule graphs are first encoded and passed into a contact prediction module (CPM) that auto-regressively generates the proximity distributions and associated pairwise embeddings. An equivariant structure denoising module (ESDM) then generates the 3D structures by denoising the atomic coordinates using a learned stochastic process (Figure 2d). Finally, for every sampled structure a predicted confidence score (pLDDT, Supplementary Information G) is assigned to each protein residue and ligand atom. A more technical summary of the model inference procedure is described in Supplementary Information, Algorithm S1.

Molecular representations To enable stereospecific molecular geometry representation and explicit reasoning about long-range geometrical correlations, NeuralPLexer hybridizes two types of elementary molecular representations (Figure 2a): (a) atom nodes and (b) rigid-body nodes representing coordinate frames formed by adjacent chemical bonds (hereby "frames"). We introduce a pre-trained encoder model termed Molecular Heat Transformer (MHT; Methods, molecular graph encoding) to transform the ligand and amino acid molecular graphs into a set of informative embeddings; once the MHT outputs are

gathered for all ligand molecular graphs, we uniformly select N_L frames and use the edge embeddings among the selected frames to create a dense tensor representation $\mathbf{F}_L \in \mathbb{R}^{N_L \times N_L \times c_p}$ of all input ligands, where c_p is the embedding dimension of the network. Parallel to this molecular graph branch, the input protein sequences and auxiliary features are encoded into a tensor representation $\mathbf{F}_P \in \mathbb{R}^{N_P \times N_P \times c_p}$ for N_P consecutive patches of the proteins (see Supplementary Information D for details). \mathbf{F}_P and \mathbf{F}_L are combined to form a dense representation of the protein-ligand complex $\mathbf{F} \in \mathbb{R}^{(N_P + N_L) \times (N_P + N_L) \times c_p}$ which is further processed by the contact prediction and denoising modules (Figure 2b,e).

Generating contact maps and pair representations We define the contact map of a protein-ligand complex based on the pairwise proximities among all residues and selected frames (Methods, Distograms and contact maps). During model inference, the contact maps are modeled as the logits of a categorical posterior distribution; the CPM auto-regressively refines a single realization of the contact map \mathbf{L} using feedback from the previous iteration predictions. Specifically, the CPM samples a one-hot adjacency matrix from the last-iteration-predicted contact map and sums all sampled adjacency matrices $\{\mathbf{l}\}_{k=1}^{N_L}$ as an additional signal passed to the neural network, until $k = N_L$ when each ligand frame is assigned to a protein patch:

$$\hat{\mathbf{L}}_{k+1} = \text{CPM}\left(\sum_{r=1}^k \mathbf{l}_r, \mathbf{F}\right); \mathbf{l}_k = \text{OneHot}(i_k, j_k); (i_k, j_k) \sim \text{Categorical}_{N_P \times N_L}(\hat{\mathbf{L}}_k) \quad (1)$$

where $k \in \{0, \dots, N_L\}$. This auto-regressive sampling scheme accounts for the multi-modal nature of protein-ligand contact distributions and allows the model to be trained on diverse structural databases including complexes with multiple bound ligands (such as in substrate-cofactor interactions). Concurrently, the CPM maintains a sparsely connected graph representation of all protein residues and ligand atoms based on their chemical and spatial proximity; the sparse graph regularly communicates with the dense representation \mathbf{F} through a cross-attention mechanism (Methods, CPM). The last iteration contact map $\hat{\mathbf{L}}_{N_L}$ is passed back into the CPM network to produce the set of graph representations used by the ESDM for structure generation.

Equivariant diffusion for atomistic structure generation We leverage diffusion-based generative modeling [62, 63] to sample 3D molecular geometries from a learned statistical distribution. Diffusion models introduce a forward stochastic differential equation (SDE) that diffuses data into a known distribution and a neural-network-parameterized reverse-time SDE that generates data by reverting the noising process. Building upon this formulation, NeuralPLexer generates binding complex structures from a reverse-time process (Figure 2d):

$$d\mathbf{Z}_t = [-\Theta \mathbf{Z}_t - \Sigma \Sigma^\top \mathbf{s}_\phi(\mathbf{Z}_t; t)]dt + \Sigma d\mathbf{W}_t \quad (2)$$

where \mathbf{Z}_t denotes all 3D atomic coordinates of the binding complex structure at time t , and \mathbf{s}_ϕ is a network-predicted function called the *score function* [63] which drifts the geometry \mathbf{Z}_t towards the target data distribution. We adopt Θ and Σ matrices that involve anisotropic linear terms among chemically adjacent atoms (Figure 2c) with separated length-scale parameters such that global domain packing and local atomic structures are generated hierarchically. The diffusion processes are equivariant to translations and rotations, as detailed in Supplementary Information B.2. During inference, the atomic coordinates are first randomly initialized from a simple probability distribution that corresponds to the fully noised state of the forward-time SDE. At each integrator step along the reverse-time SDE sampling (2), the ESDM utilizes a stereochemistry-aware graph transformer neural network (Figure 2e) to predict a set of denoised coordinates $\hat{\mathbf{Z}}_0$ which is then transformed to \mathbf{s}_ϕ (Methods, ESDM). Moreover, we adopt a stochastic temperature-adjusted sampler (Langevin Simulated Annealing SDE, LSA-SDE) and introduce a correction scheme to improve consistency between integrator steps and reduce numerical discretization error (Methods, Temperature-annealed and continuity-corrected diffusion model sampling) which are found to improve accuracy.

Model training To train NeuralPLexer, we introduce a dataset named PL2019-74k (Methods, Datasets and training summary) compiled from a diverse set of experimentally determined apo protein structures and protein-ligand complex structures from the Protein Data Bank (PDB), cross-referenced to annotations in other public datasets to systematically filter experimental artifacts. The loss function used for model training comprises a cross-entropy term for CPM outputs, a translation-rotation-invariant structure denoising term evaluated for global and binding site structures, and regularization terms to improve local distance-geometry quality and reduce structural violations. During model training, an MSA bootstrapping technique [33] was applied to sample diverse template structures from AF2 as auxiliary model inputs complementary to alternative experimentally determined conformations of the same proteins retrieved from the PDB (Methods, Datasets and training summary). The training process in total took 16 days on 6 NVIDIA V100-32GB GPUs, which is overall substantially lower than the cost of training state-of-the-art protein structure prediction models (for example, 11 days on 128 TPU-v3 cores for training AF2 [1]).

Accurate ligand and binding pocket structure modeling

We evaluate NeuralPLexer on two benchmarking tasks that are simplified sub-problems for protein-ligand complex structure prediction, yet they are still considered challenging problems for conventional molecular docking workflows. Specifically,

| Methods | Top-1 Ligand RMSD | | | | | GPU time (s) | Top-5 Ligand RMSD | | | | |
|-------------------------|-------------------|------------|------------|-------------------|-------------|--------------|-------------------|------|------|-------------------|------|
| | Percentiles ↓ | | | below threshold ↑ | | | Percentiles ↓ | | | below threshold ↑ | |
| | 25th | 50th | 75th | 5 Å | 2 Å | | 25th | 50th | 75th | 5 Å | 2 Å |
| P2Rank+GNINA [54] | 1.7 | 5.5 | 15.9 | 47.8 | 28.8 | 127 | 1.7 | 5.5 | 15.9 | 47.8 | 28.8 |
| DiffDock [54] | 1.4 | 3.3 | 7.3 | 63.2 | 38.2 | 40 | 1.2 | 2.4 | 5.0 | 75.5 | 44.7 |
| PointSite+Uni-dock [64] | - | 5.5 | - | - | 32.1 | - | - | 2.5 | - | - | 46.1 |
| DiffDock+Uni-dock [64] | - | 4.1 | - | - | 38.9 | - | - | 1.9 | - | - | 51.1 |
| NeuralPLexer (ours) | 1.3 | 2.8 | 5.9 | 69.7 | 39.5 | 2.1 | 1.1 | 2.3 | 4.6 | 77.0 | 47.0 |

Table 1. Summary of performance statistics for rigid receptor blind protein-ligand docking on the PDBBind2020 benchmark. Statistics for the baseline methods are obtained from the sources referenced in their respective rows. Note that the top-5 statistics reported by baseline methods may depend on their pose selection schemes; the NeuralPLexer top-5 results are computed using 5 randomly sampled poses per protein-ligand pair, thus should be considered a success rate lower bound.

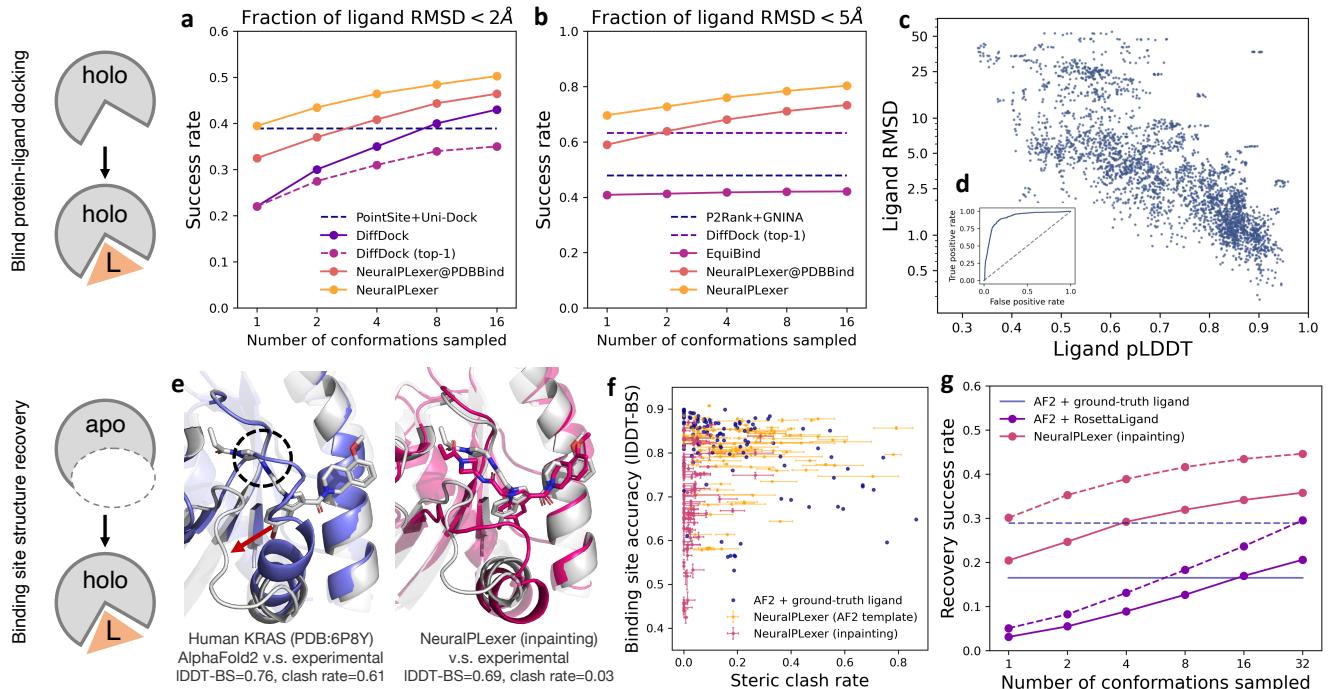


Figure 3. Model performance on benchmarking problems. (a-d) Rigid backbone blind protein-ligand docking. The cumulative fraction of predictions with ligand RMSD below (a) 2 Å and (b) 5 Å over the test dataset are plotted against the number of ligand poses sampled per protein-ligand pair. (c) Assessing the fidelity of model-assigned confidence estimations. The ligand RMSDs are plotted against the model-assigned pLDDT score averaged over all ligand atoms (ligand pLDDT). (d) The precision-recall curve is evaluated based on the sampled structures ranked by the ligand pLDDT score, with the binary value of ligand RMSD < 2 Å being treated as the class label. (e-g) Flexible binding site structure recovery. (e) Visualization of prediction results on a test set example (PDB:6P8Y) near the structure recovery accuracy cutoff considered in this work. NeuralPLexer generates the binding site protein-ligand structure consistent with experiments, while directly aligning the AF2 prediction to ground-truth complex results in steric clashes (dashed circle). The red arrow indicates the qualitative conformational change from AF2 to the experimental bound-state structure. (f) Summary of binding site accuracy (measured by the IDDT-BS score) and ligand clash rate over the test dataset. 32 conformations are sampled for each protein-ligand pair; dots indicate the median value and error bars indicate 25th and 75th percentiles. (g) The structure recovery accuracy is compared to baseline methods. Solid lines correspond to recovery rates evaluated based on strict cutoffs (IDDT-BS > 0.7, ligand RMSD < 2.0 Å, clash rate = 0.0), while the dashed lines correspond to relaxed cutoffs (IDDT-BS > 0.5, ligand RMSD < 2.5 Å, clash rate < 0.05) with clash rate cutoff matching the 95% percentile of experimental structure statistics.

we first benchmark the method on *blind protein-ligand docking* where the ground-truth receptor protein structure is given as input, and the ligand coordinates are predicted without any pre-specified binding site information. Model predictions are collected on the PDBBind2020 [58] dataset, a community-wide recognized benchmark for protein-ligand docking. We compare our results to alternative computational algorithms such as strategies combining binding site prediction models (such as PointSite [65] and P2Rank [66]) and conventional search-and-scoring-based docking algorithms (such as GNINA [67] and Uni-Dock [68]) for blind docking, as well as a recent generative-model-based method DiffDock [54] that is state-of-the-art on this benchmark. Using the dataset split in Ref. [69], we assess the results from a NeuralPlexer model that was directly trained on the PDBBind2020 training set (Figure 3, NeuralPlexer@PDBBind) from randomly initialized parameters as well as a model that is fine-tuned from the checkpoint trained for end-to-end structure prediction on PL2019-74k (Figure 3, NeuralPlexer). As shown in Table 1 and Figure 3a-c, NeuralPlexer-generated ligand poses achieve improved geometrical accuracy against reference methods, as consistently confirmed by the highest fraction of predictions with a ligand-heavy atom root-mean-square-deviation (RMSD) below the 2.0 Å and the 5.0 Å thresholds; pre-training the model on end-to-end structure prediction additionally improved success rate by around 20% on both the RMSD<2.0 Å and the RMSD<5.0 Å criteria, suggesting that learning on tasks related to protein folding prediction produces model representations that are better suited to identify functional ligand binding sites. To study how well the generated ligand structure distribution covers the ground truth pose, in Figure 3a-b we also plot the fraction of targets for which a successful prediction can be identified against the number of docking trials per target (x-axis). Remarkably, 39.5% of NeuralPlexer predictions are accurate to within an RMSD of 2.0 Å despite sampling only 1 ligand pose per target, representing a 78% improvement compared to the best competing method, DiffDock, at this limit. This result highlights the effectiveness of incorporating contextual information through the form of contact maps to localize the space of structural hypotheses, and implies the potential to accelerate virtual screening while easing the need for designing scoring functions. Last, we evaluate the relationship between the ligand pose prediction accuracy against the model-assigned confidence estimations; as shown in Figure 3c-d, the predicted pLDDT scores averaged over ligand atoms are well correlated with the true ligand RMSD, and 80% of the predicted structures with RMSD<2.0 Å can be identified by ranking the structures using ligand pLDDT, with a low false positive rate of 11.6%.

We then apply NeuralPlexer to perform *binding site structure recovery*, which can be interpreted as a principled approach to model highly flexible binding pockets. In particular, given a ligand-free receptor protein conformation and the binding site centroid coordinate, we adopt a fine-tuned NeuralPlexer model to jointly predict the structure for a cropped spherical region within 6.0 Å of any ligand atom by inpainting all the amino acid and ligand atomic coordinates conditioning on the uncropped parts of the receptor backbone. The binding pocket structure accuracy relative to the reference experimental structures is measured by the IDDT-BS, the all-atom Local Distance Difference Test [70] metric averaged for residues within 4.0 Å of any ligand atom consistent with CAMEO [71]. Input backbones are obtained using template-free AF2 predictions of 154 selected chains with TM-score>0.8 (that is, high backbone accuracy) but with IDDT-BS<0.9 out of the PDBBind2020 test set, a subset representing cases where AF2 correctly predicts the global protein folding but fails to reproduce the exact bound-state binding site structure. We first assessed the fidelity of these initial AF2 structures for ligand docking by computing the steric clash rate between the ligand and the receptor, defined as the fraction of ligand heavy atoms with a Lennard-Jones energy > 100 kcal/mol using UFF [72] parameters. The stringency of this criterion is reflected by the observation that 9% of the experimental structures are classified as clash-containing because of experimental errors and structures with cross-linking. Out of the 154 AF2-predicted receptor structures, we find that only 18% of structures contain no steric clash with the ligand when directly aligned to the reference experimental binding complex, which represents a success rate upper bound for template-based ligand modeling for such targets if using AF2 structures as references. In contrast, as shown in Figure 3f-g (NeuralPlexer-inpainting), our modeling strategy is able to accurately dock ligands to the specified sites while retaining a low clash rate and competitive binding pocket structure prediction accuracy which is evident from an improved successful recovery rate of up to 35% (and up to 46% on a relaxed criterion). The overall good all-atom binding site accuracy on all prediction targets as measured by IDDT-BS suggests that the improvement in ligand geometric accuracy and clash rate is not an effect of over-aggressively relaxing the binding site residues. As a reference, we perform inference using the entire AF2 structure as input template (Figure 3f, NeuralPlexer (AF2Template)) instead of only using non-binding-site residues; despite a similar ligand median RMSD of 3.34 Å and a higher average IDDT-BS score of 0.82 compared to NeuralPlexer-inpainting (RMSD=3.51 Å, IDDT-BS=0.71), NeuralPlexer (AF2Template) resulted in a substantially higher fraction of clash rate (0.22 on average) which implies the deficiency of these AF2 models for standard rigid-protein molecular docking.

Figure 3e shows a prediction example on human KRAS^{G12C} with a cysteine-reactive covalent inhibitor [73] (PDB:6P8Y) for which the opening of Switch-II pocket [18] harbors an unconventional druggable site. While the AF2 prediction for this target recapitulates the native-like closed pocket with a severe steric clash with the crystal structure ligand, the NeuralPlexer-inpainted binding site successfully predicted an open-like conformation with a ligand RMSD of 2.08 Å. Although NeuralPlexer qualitatively improved the binding site conformation model for this target, we still observed a slight drop in the measured IDDT-BS score from 0.76 of AF2 to 0.69. This observation suggests that the combined metric used in Figure 3g is a more

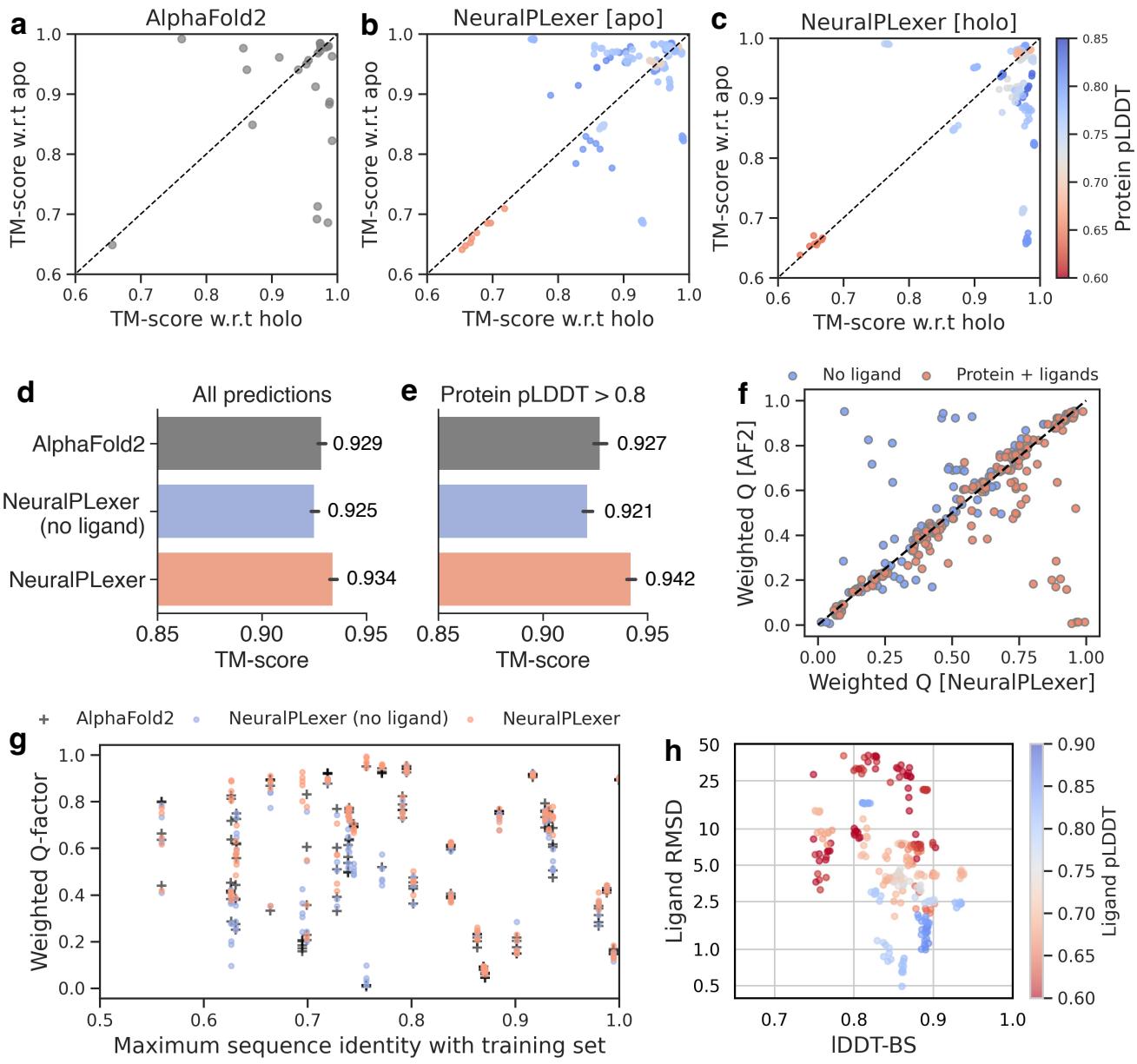


Figure 4. Model predictions for contrasting apo-holo pairs from the PocketMiner dataset. (a-c) The TM-score with respect to apo and holo experimental reference structures are plotted for (a) AlphaFold2 (AF2) structures (b) NeuralPlexer-sampled apo structures, and (c) NeuralPlexer-sampled holo structures. For NeuralPlexer-sampled structures, the model-assigned pLDDT scores averaged among all protein residues are indicated by the dot colors. (d-e) TM-scores for baseline methods and NeuralPlexer averaged against all samples and the subset of targets for which any NeuralPlexer-predicted structure is of protein pLDDT=0.8 or higher. Error bars indicate the standard error of the mean calculated from six sets of predictions using independent random seeds and different AF2 model checkpoints to obtain the initial template. (f) The weighted Q-factor metric for conformational change prediction accuracy (see Methods, Weighted Q-factor) is plotted between NeuralPlexer predictions and AF2 predictions on all ligand-bound holo structures. Grey dots correspond to NeuralPlexer predictions in the absence of ligand inputs. (g) The weighted Q-factors for predicted ligand-bound holo structures are plotted against the maximum sequence similarity of each target to samples in the training dataset. (h) Assessing the fidelity of NeuralPlexer internal confidence predictions. The ligand-binding-site accuracy as measured by IDDT-BS is plotted against the ligand RMSD for all NeuralPlexer predictions; the model-assigned pLDDT scores averaged among all ligand atoms are indicated by the dot colors.

appropriate criterion to assess binding site structure prediction fidelity, as the simultaneous satisfaction of low ligand RMSD and clash rate implies a correct pocket shape which may not be discerned by distance-based metrics such as IDDT. We have also compared our quantitative results against a baseline method RosettaLigand [59], an algorithm designed for ligand docking with fast energy-based flexible receptor relaxation (Figure 3g). We confirm that NeuralPLexer achieves consistently higher structure recovery rates without applying any posthoc filtering, even though RosettaLigand explicitly used the entire AF2 structure as the initial guess for relaxation while NeuralPLexer only used the binding-site-cropped AF2 scaffold as input. The capability of inpainting cropped structures without template information shows immediate applicability for de novo ligand-binding protein design [74], since the NeuralPLexer network can be viably adapted to jointly generate binding site backbones and sequences in the absence of protein sequence inputs.

End-to-end structure prediction for ligand-binding proteins

Having established a new mark on benchmarking problems, we then sought to evaluate the performance of NeuralPLexer on predicting the structures of ligand-binding proteins that exhibit large conformational variability. We first assessed NeuralPLexer-predicted structures on 33 contrasting apo-holo pair systems from the PocketMiner [75] dataset, which was held out from our training datasets based on their protein UniProt ID such that no structures with identical protein sequence is observed during training. To examine the overall prediction accuracy of such ligand-binding systems, we compared the NeuralPLexer predictions against the best-performing protein structure prediction algorithm AF2. Figure 4d and Table S6 summarizes the prediction accuracy statistics as evaluated by TM-score [60], on which NeuralPLexer exhibits the highest performance. We delineate the underlying factors that contributed to the improved prediction accuracy by performing pairwise structure comparisons against their corresponding experimental apo and holo reference structures; as shown in Figure 4b, the protein structure ensembles sampled from NeuralPLexer in the absence of ligand inputs are overall diverse and more similar to the experimental apo reference structures, while the protein parts of the binding complexes obtained with both protein and ligand graph inputs are well-aligned with the experimental holo structures. Instead, AF2 produces a mixture of structures for which no consistent trend of relative similarity to apo or holo states was observed, which leads to a decrease in TM-score when averaged against all apo and holo samples. In summary, we observe an average TM-score=0.934 for structures sampled from NeuralPLexer, representing a statistically-significant improvement of TM-score=0.929 from AF2 ($p=0.03$) and TM-score=0.925 from a ligand-free baseline model ($p=0.0004$) for which we ablate all ligand graph inputs from NeuralPLexer to only sample apo structures (Figure 4d, NeuralPLexer (no ligand)). The moderate decrease of structure accuracy from NeuralPLexer (no ligand) compared to AF2 additionally supports our hypothesis that the state selectivity between apo and holo structures is the key factor to the improvement in overall prediction accuracy, even for cases where the overall protein structure generation quality is lower than AF2 due to the lightweight nature of the network and training data.

Apart from standard C α -based metrics such as TM-score, to further elucidate the all-atom model prediction accuracy for regions that undergo major structural changes upon ligand binding, we introduce a weighted version of the Q-factor from Best and Hummer [76] based on the inter-residue native contacts that are not conserved between two distinct-state reference structures (Methods, Weighted Q factor). As shown in Figure 4f, we observe a stark differentiation between NeuralPLexer (average Q-factor=0.608) and the ligand-free baseline model (average Q-factor=0.501), while 16% of the predicted structures are found to qualitatively improve against AF2 models (average Q-factor=0.538) by 0.1 or more. These results clearly indicate that NeuralPLexer is able to selectively sample ligand-free and ligand-bound protein states on targets that are challenging to predict using state-of-the-art methods. The prediction accuracy as measured by Q-factor is also consistent among targets with different sequence similarities to the training dataset (Figure 4f), suggesting that the model is able to generalize beyond targets for which structures of close homologs are available. In addition to the overall prediction accuracy, we find that the NeuralPLexer-assigned pLDDT score can effectively identify more accurately predicted structures for such ligand-binding proteins, as manifested by the increase from TM-score=0.92 of AF2 to 0.95 of NeuralPLexer when only evaluated on the subset at which model is most confident (average protein residue pLDDT > 0.8).

To determine the accuracy of NeuralPLexer on novel protein-ligand systems, we then assess the predicted structures on 118 recently resolved protein and protein-ligand complexes from structures that are deposited to the PDB since January 2019 (Figure 5, Table S7). These targets are selected based on the criterion that a reference experimental structure of the same protein with sequence identity >98% and backbone RMSD>2.0 Å can be found from the PDB. Despite that 48 out of 98 holo structures in the dataset have homologs in the PDB with 100% sequence identity, the large conformational variability of these samples still renders it nontrivial to predict their structures using standard approaches. The prediction difficulty of these targets is evident from an overall poor Q-factor of 0.541 for AlphaFold2 predictions, since a contact map with random binary values would produce a Q-factor of 0.5. A similar result of average Q-factor=0.539 is observed for ligand-free NeuralPLexer predictions. In contrast, NeuralPLexer with ligand input improves the average Q-factor from 0.541 to 0.603. Out of the 50 holo targets for which identical-sequence homologs are not included within training (Figure 5d), the NeuralPLexer prediction on 13 targets shows a significant increase in Q-factor by 0.1 or more, as opposed to 7 targets for the ligand-free baseline. Overall, on this

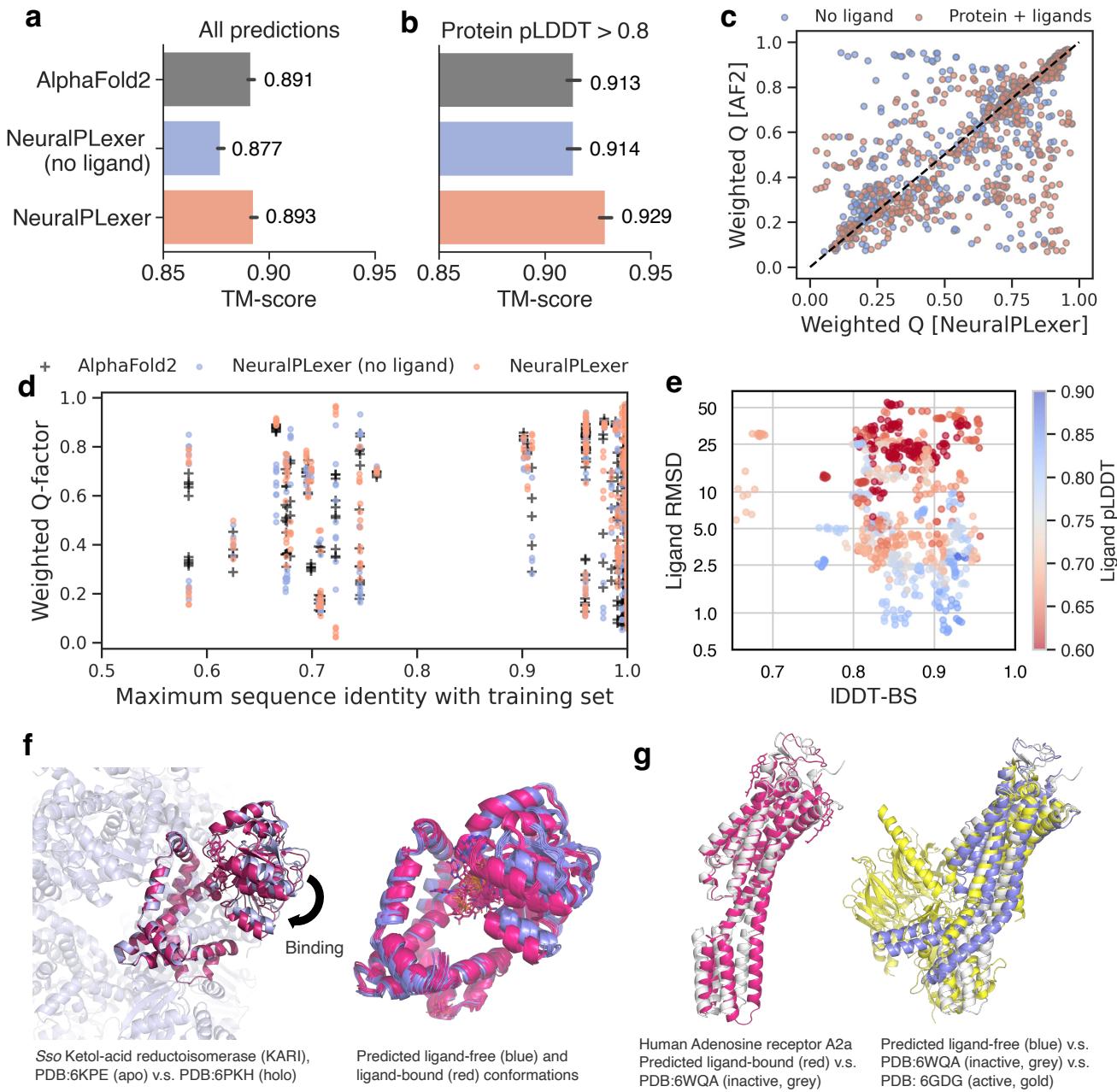


Figure 5. Model predictions for recently determined structures. (a-b) TM-scores for baseline methods and NeuralPLexer averaged against all samples and the subset of targets for which any NeuralPLexer-predicted structure is of protein pLDDT=0.8 or higher. Error bars indicate the standard error of the mean calculated from six sets of predictions using independent random seeds and different AF2 model checkpoints to obtain the initial template. (c) The weighted Q-factor metric for conformational change prediction accuracy (see Methods, Weighted Q-factor) is plotted between NeuralPLexer predictions and AF2 predictions on all ligand-bound holo structures. Grey dots correspond to NeuralPLexer predictions in the absence of ligand inputs. (d) The weighted Q-factors for predicted ligand-bound holo structures are plotted against the maximum sequence similarity of each target to samples in the training dataset. (e) Assessing the fidelity of NeuralPLexer internal confidence predictions. The ligand-binding-site accuracy as measured by IDDT-BS is plotted against the ligand RMSD for all NeuralPLexer predictions; the model-assigned pLDDT scores averaged among all ligand atoms are indicated by the dot colors. (f-g) Model predictions on representative targets (PDB:6KPE, PDB:6PKH, PDB:6WQA) suggest structural elements for protein self-assembly and plausible models for enzyme catalysis and the activation of membrane receptors.

dataset NeuralPLexer improves the average protein structure prediction accuracy by a TM-score increase from 0.877 that of NeuralPLexer (no ligand) to 0.893 which matches the accuracy of AF2 (average TM-score=0.891), and outperforms AF2 by an improvement from average TM-score=0.913 to TM-score=0.929 for the subset with protein pLDDT>0.8. On both test datasets, more than 70% predictions that are of ligand RMSD<2.5 Å and IDDT-BS>0.8 can be distinguished from others by a criterion of ligand pLDDT score > 0.8 (Figure 4e, Figure 5e), which confirms that the model-assigned confidence scores also provide a consistent metric to identify highly accurate predictions for both ligand and binding site residues.

Interpreting allostery and catalytic mechanisms from predicted structures

Furthermore, an examination of model predictions for recently determined structures reveals that the conformational variations among the predicted structures can be leveraged to gain insight into protein function and their interactions. On a ketol-acid reductoisomerase target (PDB:6KPE, Figure 5f) whose catalytic mechanism was recently characterized through cryo-EM experiment [77], by comparing the predicted ligand-bound and ligand-free ensembles we find that NeuralPLexer accurately captures the closure motion of the N-subdomain upon NADH and inhibitor binding. Apart from the structural domains with high conformational mobility, the protein was found to spontaneously form a dodecamer through the self-assembly of the C-subdomain which is also consistent with the low structural fluctuation near the observed protein-protein interface among the NeuralPLexer-sampled conformations. The predicted structures may also help the identification of key structural elements that are critical for protein activation and deactivation, especially for complex systems where experimental structure determination is challenging such as in G protein-coupled receptors (GPCRs). For example, Figure 5c shows representative predicted structures for the human Adenosine receptor A2a - Cytochrome b562 chimera (PDB:6WQA) on which the predicted antagonist-bound structure agrees well with the structure determined by XFEL experiments [78] with a TM-score=0.86. In contrast, the predicted ligand-free structures contain snapshots with significant bending near the terminal of helices TM5 and TM6; although no apo experimental structure has been determined for this target, structure alignment with the G-protein-bound active conformation [79] (PDB:6GDG) suggests that the sampled ligand-free structures are plausible intermediate states that may be attributed to constitutive activity in the absence of ligands [80]. The ability of NeuralPLexer to generate faithful structural ensembles may enable a powerful tool for generating hypotheses to help unravel various molecular mechanisms related to allosteric regulation and enzyme catalysis.

Discussions

Determining the structures and functional conformational changes of protein-ligand complexes at the proteome scale is a grand-challenge problem. We have demonstrated a key step towards solving this problem by showing that NeuralPLexer, a generative deep learning approach that systematically incorporates small molecule information and biophysical inductive bias, achieves consistent improvement against state-of-the-art molecular docking and protein structure prediction methods. NeuralPLexer is capable of generating structures within seconds on a standard GPU once all auxiliary features are gathered, thereby enabling the exploration of protein-ligand interactions on the vast space of sequence and chemical spaces by leveraging rapidly evolving databases of computationally modeled protein structures [9, 81] and chemical substances [82, 83].

NeuralPLexer is end-to-end differentiable and thus is also well-suited for applications related to ligand and protein design. A closed-loop combination of NeuralPLexer and recent advances in differentiable generative models for protein sequences [44, 84, 85], molecular graphs [86, 87], and structure-based bioactivity models holds the promise of accelerating the process of sequence and compound prioritization over conventional screening-based strategies. We also anticipate that NeuralPLexer can be directly applied to accelerate various physical simulation studies on protein-ligand interactions, such as guiding the design and optimization of collective variables in enhanced-sampling molecular dynamics simulations [23].

As a data-driven approach, NeuralPLexer is both generalizable and amenable to continuous improvement through the integration of better experimental and bioinformatic data. Improvements in the curation of training and benchmark datasets from a broader community could potentially enable a more systematic analysis of protein families with no experimentally determined homologs, as well as extending the method to more challenging systems such as post-translational modifications and multi-state large heteromeric protein complexes [88]. Moreover, refining the model on data such as high-resolution nuclear magnetic resonance (NMR) and molecular dynamics data can enable it to capture protein structure under physiological conditions beyond the distribution of crystal-like structures. A related promising direction to improve the methodology is to incorporate highly accessible auxiliary data related to protein-compound interactions such as binding affinities [89] and high-throughput mass spectrometry signals [90], given that the incorporation of large-scale protein sequencing data in the form of MSAs has already been demonstrated a crucial component to achieve transferable protein structure prediction [1, 2, 4]. Our study has provided a general computational framework for the exploration of these directions, paving the way toward the rapid and accurate protein-ligand complex structure prediction to facilitate structure biology, drug discovery, and protein engineering.

Methods

Neural network architecture overview

Molecular graph encoding. We introduce Molecular Heat Transformer (MHT), an attention-based neural network architecture for molecule inputs with learned edge encodings to infer inter-atomic geometrical correlations based on their graph-topological representations. The encoder is composed of 8 MHT blocks, where a single MHT block comprises a multi-head self-attention block with edge embeddings added to the computation of attention weights (MHAwithEdgeBias, Supplementary Information C), a pair update block, and a node update block. As a unique component of MHT, the pair update block updates the pair representations between the set of atomic nodes and frame nodes through discretized heat-kernel transformations [91], such that they are capable of incorporating the molecular graph chirality and are able to explicitly encode long-range dependencies beyond neighboring nodes on the molecular graph.

To train the MHT model in NeuralPlexer, we designed a specialized algorithm to align the learned molecular representations with both 3D molecular structure and bioactivity information. In particular, we introduced a hybridized loss function that comprises a mixture density network term to fit the 3D distance-angle distributions to the reference molecular geometries for all edges between the atom and frame nodes, a global denoising term to recover the molecular geometry from perturbed geometries sampled from a forward-time SDE, a supervised loss term to predict the bioactivity labels following a graph pooling operation, as well as a masked language modeling loss. The featurization details, MHT architecture, related datasets, and additional training details are described in Supplemental Information C.3.

Protein-ligand graph representation and the contact prediction module (CPM). As detailed in Supplementation Information D, we adopt a residue-scale graph representation combining sparse edges determined based on the input noisy protein geometry \mathbf{x}_t and densely connected edges associated with pair representations \mathbf{F} for a selected subset of protein and ligand nodes ("anchor nodes"). The protein anchor nodes are selected by first sequentially segmenting the input protein sequence into N_p equal-sequence-length patches, and then sampling one unique backbone frame for each protein patch; the N_L ligand anchor nodes are uniformly sampled from all N_{frame} frame nodes of the ligand molecular graphs. Protein node features are initialized from the PLM embeddings of the input sequence (or the concatenation of PLM embeddings of all sequences in $\{\mathbf{s}\}$, when the input is a protein complex); the intra-protein edge features are initialized via a geometrical encoding of the relative backbone frame distances and orientations between the source and destination nodes. When the template protein structure input is present, local encodings of side-chain internal coordinates and relative geometrical encodings of the template backbone frames are respectively added to node and edge features. The CPM (Figure 2b, Supplementation Information E) enables expressive and scalable processing of the residue-scale graph, featuring a linear scaling with respect to the number of nodes and a constant overhead. The neural network architecture comprises a graph attention layer to update all node and edge representations, a triangular attention layer to update the densely-connected edge representations while accounting for edge-edge interactions, and a cross-attention layer to synchronize information between the sparse and the dense components of the graph.

The equivariant structure denoising module (ESDM). The architecture of ESDM (Figure 2e, Supplementation Information F) is inspired by prior works on 3D graph and attentional neural networks for point clouds [92, 93], rigid-body simulations [94] and biopolymer representation learning [1, 95–97]. In ESDM, each node is associated with a stack of standard scalar features $\mathbf{f}_s \in \mathbb{R}^c$ and cartesian vector features $\mathbf{f}_v \in \mathbb{R}^{3 \times c}$ representing the displacements of a virtual point set relative to the node's coordinate $\mathbf{t} \in \mathbb{R}^3$. A rotation matrix $\mathbf{R} \in \text{SO}(3)$ is additionally attached to each rigid-body (i.e., frame) node. Geometry-aware messages are synchronously propagated among all nodes by encoding the pairwise distances among virtual point sets into graph transformer blocks (PointSetAttentionwithEdgeBias, Algorithm S8). Explicit non-linear transformation on vector features \mathbf{f}_v is solely performed on rigid-body nodes through a coordinate-frame-inversion mechanism (LocalUpdateUsingReferenceRotation, Algorithm S10), such that the node update blocks are sufficiently expressive without sacrificing equivariance or computational efficiency. On the contrary, 3D coordinates are solely updated for atomic nodes (PredictDrift, Algorithm S11) while the rigid-body frames (\mathbf{t}, \mathbf{R}) are passively reconstructed according to the updated atomic coordinates, circumventing numerical issues regarding fitting quaternion or axis-angle variables when manipulating rigid-body objects. The nontrivial actions of an inversion operation on rigid-body nodes ensure that ESDM captures the correct chiral-symmetry-breaking behavior that adheres to the molecular stereochemistry constraints.

Distograms and contact maps

Given the atomic coordinates of a binding complex, we define the ground-truth contact map as

$$L_{AJ} = \max\left(1 - \frac{\|\mathbf{x}_A - \mathbf{y}_J\|}{D_0}, \epsilon\right) \quad (3)$$

where \mathbf{x}_A stands for the centroid coordinate of amino acid residue A , and \mathbf{y}_J stands for the center-atom coordinates of a selected ligand frame J ; we use $D_0 = 8.0 \text{ \AA}$ and $\epsilon = 10^{-6}$ for all results reported in this study. At each auto-regressive contact

map refinement iteration, the CPM network outputs histograms of pairwise distances ("distograms") $\hat{P}(d_{AJ}|\{\mathbf{s}\}, \{G\}, \sum_{r=1}^k \mathbf{l}_r)$ between all protein residues $A \in \{1, 2, \dots, N_{\text{res}}\}$ and ligand frames $J \in \{1, 2, \dots, N_{\text{frame}}\}$. The distograms are then used to estimate the predicted contact map \hat{L}

$$(\hat{L})_{AJ} = \sum_{r=1}^{N_{\text{bins}}} \hat{P}(d_{AJ}|\{\mathbf{s}\}, \{G\}, \sum_{r=1}^k \mathbf{l}_r) \cdot \max(1 - \frac{d_{AJ}}{D_0}, \varepsilon), \quad d_{AJ} \in [d_{\min}, d_{\min} + \frac{d_{\max} - d_{\min}}{N_{\text{bins}} - 1}, \dots, d_{\max}] \quad (4)$$

we use $N_{\text{bins}} = 32$, $d_{\min} = 2.0 \text{ \AA}$, and $d_{\max} = 22.0 \text{ \AA}$ for the distograms. The contact map \hat{L} is further coarse-grained into the respective protein patches before sampling the block-adjacency matrices (Eq. (1)) to be recycled by the CPM network.

Diffusion SDE details

To motivate the design principles for our diffusion-based biomolecular structure generator, we first consider a general class of linear SDEs known as the multivariate Ornstein–Uhlenbeck process [98] for point cloud $\mathbf{Z} \in \mathbb{R}^{N \times 3}$:

$$d\mathbf{Z}_t = -\Theta \mathbf{Z}_t dt + \Sigma d\mathbf{W}_t \quad (5)$$

where $\Theta \in \mathbb{R}^{N \times N}$ is an invertible matrix of affine drift coefficients, $\Sigma \in \mathbb{R}^{N \times N}$ represents the factorized covariance matrix, and \mathbf{W}_t is a standard $3N$ -dimensional Wiener process. The forward noising SDEs used in standard diffusion models [99, 100] can be recovered by setting $\Theta = \theta \mathbf{I}$, converging to an isotropic Gaussian prior distribution at the $t \rightarrow \infty$ (often expressed as $t \rightarrow 1$ with reparameterized t [101]) limit. Instead, we introduce a multivariate SDE containing data-dependent drift and covariance matrix Θ of a factorized form;

$$d\mathbf{Z}_t = -(\mathbf{U} \boldsymbol{\lambda} \mathbf{U}^{-1}) \mathbf{Z}_t dt + (\sigma \mathbf{U}) d\mathbf{W}_t \quad (6)$$

where $\mathbf{Z} = [\mathbf{x}_{C\alpha}, \mathbf{x}_{\text{non}C\alpha}, \mathbf{y}]^\top$ according to the following symbolic convention

- $\mathbf{x}_{C\alpha} \in \mathbb{R}^{N_{\text{res}} \times 3}$ denotes the collection of alpha-carbon coordinates of the protein;
- $\mathbf{x}_{\text{non}C\alpha} \in \mathbb{R}^{(N_{\text{protatm}} - N_{\text{res}}) \times 3}$ denotes the set of coordinates for all non-alpha-carbon protein atoms (backbone N, C, O, and all side-chain heavy atoms);
- $\mathbf{y} \in \mathbb{R}^{N_{\text{ligatm}} \times 3}$ denotes all ligand heavy atom coordinates. Note that $N_{\text{ligatm}} := \sum_{k=1}^K N_{\text{ligatm},k}$ with $N_{\text{ligatm},k}$ being the number of heavy atoms in each ligand molecule G_k .

\mathbf{U} defines a linear transformation such that the forward diffusion process erases chemical-group-scale local details before it removes global information about protein domain packing and ligand binding interfaces. \mathbf{U} is defined through

$$\mathbf{z} = \mathbf{U}^{-1} \mathbf{Z} = [\mathbf{x}_{C\alpha}, \mathbf{x}_{\text{non}C\alpha} - \mathbf{x}_{C\alpha}, \mathbf{y} - \mathbf{c}^\top \mathbf{x}_{C\alpha}]^\top \quad (7)$$

where \mathbf{c} is the softmax-transformed contact map $c_{A,M} = \frac{\sum_{J \in \{M\}} \exp(L_{AJ})}{\sum_{A,J \in \{M\}} \exp(L_{AJ})}$, where $M \in \{1, \dots, N_{\text{ligands}}\}$ is a ligand index running over all molecular graphs in $\{G\}$; $\boldsymbol{\lambda}$ is a diagonal matrix representing the drift coefficients on latent coordinates \mathbf{z} . The diagonal entries of $\boldsymbol{\lambda}$ are set to 6.0 for alpha-carbon atoms $\mathbf{x}_{C\alpha}$ and 37.5 for all other internal coordinates $\mathbf{x}_{\text{non}C\alpha} - \mathbf{x}_{C\alpha}$, $\mathbf{y} - \mathbf{c}^\top \mathbf{x}_{\text{non}C\alpha}$. We set the diffusion coefficient parameter $\sigma = 12.25 \text{ \AA}$. For all SDE sampling, we use an exponential time schedule defined by

$$t = e^{-\log(t_0) \cdot \tau} \cdot t_0 \quad \text{where } t_0 = 10^{-3}, \tau \sim (0, 1.0] \quad (8)$$

analogous to common choices in variance-exploding [63] SDEs. The marginal densities of the latent coordinates \mathbf{z} sampled from the forward SDE are given by:

$$q(\mathbf{z}_t | \mathbf{z}_0) = \mathcal{N}(\sqrt{\boldsymbol{\alpha}_t} \cdot \mathbf{z}_0, (\mathbf{I} - \boldsymbol{\alpha}_t) \cdot \frac{\sigma^2}{2\boldsymbol{\lambda}}) \quad \text{where } \boldsymbol{\alpha}_t = \exp(-2\boldsymbol{\lambda}t) \quad (9)$$

Proofs regarding equivariance and chiral properties are stated in the Supplemental Information, B.2.

Temperature-annealed and continuity-corrected diffusion model sampling

Langevin simulated annealing SDE (LSA-SDE) To enable the generation of crystal-structure-like complexes corresponding to local minima of the model density distribution, we adopt an adjusted diffusion model sampling scheme where an auxiliary inverse temperature parameter $\beta(t)$ is smoothly varied from 1.0 (i.e., the data distribution temperature) to a target inverse temperature $\beta_{\max} = 10.0$ along the reverse-time SDE. Inspired by recent works on controllable diffusion-based generation for protein backbones [46], we introduce the following dynamics termed Langevin simulated annealing SDE (LSA-SDE):

$$d\mathbf{Z}_t = [-\Theta \mathbf{Z}_t - \frac{1 + \beta(t)}{2} \Sigma \Sigma^\top \nabla_{\mathbf{Z}_t} \log q_t(\mathbf{Z}_t)] dt + \sqrt{\frac{1}{\beta(t)}} \Sigma_t d\mathbf{W}_t \quad (10)$$

where we linearly varied the inverse temperature parameter $\beta(t)$ according to $\beta(t) = 1 + (\beta_{\max} - 1)(1 - \tau)$.

Kabsch alignment correction While marginal densities from the forward SDE depend on the global placement and orientation of the initial molecular structure, neural-network-based structure prediction is mainly found to achieve improved accuracy when trained with SE(3)-invariant losses [1]. However, structure prediction models trained with invariant loss can cause inconsistency between integration steps when used for diffusion model sampling due to the arbitrariness of predicted global rotation, which is found to harm the generation quality in related works [47]. To reconcile this discrepancy, at sampling time we introduce a rigid-body alignment operation between the $C\alpha$ traces of the current and previous-step network outputs using the Kabsch-Umeyama algorithm and then use the aligned structure to compute the score function s_ϕ :

$$s_\phi(\mathbf{Z}_t, t) = \mathbf{U} \frac{1}{1 - \boldsymbol{\alpha}_t} (\sqrt{\boldsymbol{\alpha}_t} \cdot \mathbf{U}^{-1} (\mathbf{R} \circ \hat{\mathbf{Z}}_0) - \mathbf{U}^{-1} \mathbf{Z}_t), \quad \mathbf{R} = \operatorname{argmin}_{\mathbf{r} \in \text{SE}(3)} \|\mathbf{r} \circ \hat{\mathbf{x}}_{C\alpha,0}(\mathbf{Z}_t) - \hat{\mathbf{x}}_{C\alpha,0}(\mathbf{Z}_{t+\Delta t})\|_2^2 \quad (11)$$

where we have assumed a time discretization interval of Δt for a single integrator step.

Semi-analytic integrator Furthermore, we introduce a semi-analytic integration scheme based on a harmonic approximation of the time-dependence score function s_ϕ to reduce discretization errors:

$$\mathbf{z}_t = \sqrt{\boldsymbol{\alpha}_t} \left[1 - \left(\frac{1/\boldsymbol{\alpha}_t - 1}{1/\boldsymbol{\alpha}_{t+\Delta t} - 1} \right)^{\frac{1+\beta(t)}{4\lambda} \sigma^2} \right] \mathbf{R} \circ \hat{\mathbf{z}}_0(\mathbf{Z}_{t+\Delta t}) + \left(\frac{1/\boldsymbol{\alpha}_t - 1}{1/\boldsymbol{\alpha}_{t+\Delta t} - 1} \right)^{\frac{1+\beta(t)}{4\lambda} \sigma^2} \sqrt{\frac{\boldsymbol{\alpha}_t}{\boldsymbol{\alpha}_{t+\Delta t}}} \mathbf{z}_{t+\Delta t} + \sigma \sqrt{\frac{\boldsymbol{\alpha}_t - \boldsymbol{\alpha}_{t+\Delta t}}{2\lambda \beta(t)}} \boldsymbol{\epsilon} \quad (12)$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$. Equation (12) can be derived by directly applying Ito's Lemma to a linearized form of the reverse-time SDE, as discussed in detail in Supplemental Information, Section B.

Datasets and training summary

The datasets used for training and testing end-to-end structure prediction were constructed from chains of all monomeric proteins and homomeric complexes in the Protein Data Bank accessed on April 2022. We filtered the retrieved structures by discarding structures with experimental resolution lower than 3.0 Å and chain coverage lower than 95%, and deleting ligands that contain more than 1000 heavy atoms and non-transition-metal single-atom ligands. The holo structures are obtained from the intersection between the filtered set and the annotated protein-ligand complex database BioLip [102] using their non-redundant index file as well as an extra set of chains from GPCRdb [103], and dropping samples with DNA and RNA ligands; the apo structures are obtained from the filtered set that contains no ligand or only ligands in the artifact list provided by Ref. [102]. After combining the apo and holo structures and removing duplicate chains, we obtained 85140 unique samples. The final PL2019-74k dataset is obtained by removing samples deposited after Jan 2019 and samples with UniProt ID in the PocketMiner dataset [75], resulting in 74477 samples for model training. The contrasting apo-holo pair test set is then obtained by taking the samples of the same PDB code and chain index in the original PocketMiner dataset; the recent receptors test set is obtained by taking the filtered samples deposited in Jan 2019 for which a reference experimental sample with sequence identity > 98%, backbone RMSD > 2 Å as reported by TMAlign [60], and distinct bound ligands can be found from the PDB.

During training, we perform a weighted subsampling on the PL2019-74k dataset at each epoch to balance the occurrence frequency of unique UniRef50 cluster IDs and maximum backbone RMSD to PDB structures. With a 50% probability at each training iteration, an experimental template structure or an AlphaFold2-predicted structure of pLLDT > 0.7 or higher is supplied to NeuralPlexer inputs as an auxiliary template structure. Those AlphaFold2 structures were obtained with ColabFold [104]; for each unique chain in PL2019-74k, we sample at most 20 structures using input MSAs provided by OpenFold [105] with reduced MSA sizes of [16, 32, 64, 128, 256] according to the bootstrapping technique described in Ref. [33] and without structure template inputs. Additional details related to loss functions, the training protocol, and hyperparameters can be found in Supplementary Information, Section H.

Weighted Q factor

The weighted Q factor metric between a predicted protein structure \mathbf{x}_{pred} and a reference protein structure \mathbf{x}_{alt} is defined based on comparing the overlap between the two contact maps, over residue pairs for which the contacts are not conserved between the reference structure and an alternative conformation of the protein \mathbf{x}_{alt} :

$$\text{WeightedQFactor}(\mathbf{x}_{\text{pred}}, \mathbf{x}_{\text{ref}}; \mathbf{x}_{\text{alt}}) = \frac{\sum_{A,B} (\mathbf{C}_{\text{pred},AB} \wedge \mathbf{C}_{\text{ref},AB}) \cdot [1 - (\mathbf{C}_{\text{ref},AB} \wedge \mathbf{C}_{\text{alt},AB})] \cdot \mathbb{1}_{|d_{\text{ref},AB} - d_{\text{alt},AB}| > 2.0 \text{ \AA}}}{\sum_{A,B} [1 - (\mathbf{C}_{\text{ref},AB} \wedge \mathbf{C}_{\text{alt},AB})] \cdot \mathbb{1}_{|d_{\text{ref},AB} - d_{\text{alt},AB}| > 2.0 \text{ \AA}}} \quad (13)$$

where $\mathbb{1}$ denotes a 0-1 indicator function, and \wedge denotes a logical and operation; the contact map \mathbf{C} is defined as a binary matrix: for two residues A, B , the matrix element $\mathbf{C}_{AB} = 1$ if the distance d_{AB} between the centroid coordinates of residue A and B is closer than 20 Å, and $\mathbf{C}_{AB} = 0$ otherwise. For all results reported in Figures 4-5, the alternative structure \mathbf{x}_{alt} is set to the most contrasting sample from the PL2019-74k dataset, that is, the structure with highest backbone RMSD as reported by TMAlign [60] among samples with sequence identity > 98% to the query structure.

Acknowledgements

Z.Q. acknowledges graduate research funding from Caltech and partial support from the Amazon–Caltech AI4Science fellowship. T.M. acknowledges partial support from the Caltech DeLogi fund, and A.A. acknowledges support from a Caltech Bren professorship. We thank Matthew Welborn, Chao Zhang, and Vignesh Bhethanabotla for helpful discussions on the work and for comments on the manuscript. We thank Artur Meller and Jonathan Borowsky for sharing the PocketMiner dataset.

References

- [1] John Jumper et al. “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596.7873 (2021), pp. 583–589.
- [2] Sergey Ovchinnikov et al. “Protein structure determination using metagenome sequence data”. In: *Science* 355.6322 (2017), pp. 294–298.
- [3] Jianyi Yang et al. “Improved protein structure prediction using predicted interresidue orientations”. In: *Proceedings of the National Academy of Sciences* 117.3 (2020), pp. 1496–1503.
- [4] Minkyung Baek et al. “Accurate prediction of protein structures and interactions using a three-track neural network”. In: *Science* 373.6557 (2021), pp. 871–876.
- [5] Minkyung Baek et al. “Accurate prediction of nucleic acid and protein-nucleic acid complexes using RoseTTAFoldNA”. In: *bioRxiv* (2022), pp. 2022–09.
- [6] Ratul Chowdhury et al. “Single-sequence protein structure prediction using a language model and deep learning”. In: *Nature Biotechnology* 40.11 (2022), pp. 1617–1623.
- [7] Wenkai Wang, Zhenling Peng, and Jianyi Yang. “Single-sequence protein structure prediction using supervised transformer protein language models”. In: *Nature Computational Science* 2.12 (2022), pp. 804–814.
- [8] Ruidong Wu et al. “High-resolution de novo structure prediction from primary sequence”. In: *BioRxiv* (2022), pp. 2022–07.
- [9] Zeming Lin et al. “Evolutionary-scale prediction of atomic-level protein structure with a language model”. In: *Science* 379.6637 (2023), pp. 1123–1130.
- [10] Yuqi Zhang et al. “Benchmarking Refined and Unrefined AlphaFold2 Structures for Hit Discovery”. In: (2022).
- [11] Felix Wong et al. “Benchmarking AlphaFold-enabled molecular docking predictions for antibiotic discovery”. In: *Molecular Systems Biology* 18.9 (2022), e11081.
- [12] Katherine Henzler-Wildman and Dorothee Kern. “Dynamic personalities of proteins”. In: *Nature* 450.7172 (2007), pp. 964–972.
- [13] Nataliya Popovych et al. “Dynamically driven protein allostery”. In: *Nature structural & molecular biology* 13.9 (2006), pp. 831–838.
- [14] Ruth Nussinov and Chung-Jung Tsai. “Allostery in disease and in drug discovery”. In: *Cell* 153.2 (2013), pp. 293–305.
- [15] Pelin Ayaz et al. “Structural mechanism of a drug-binding process involving a large conformational change of the protein target”. In: *Nature Communications* 14.1 (2023), p. 1885. DOI: [10.1038/s41467-023-36956-5](https://doi.org/10.1038/s41467-023-36956-5). URL: <https://doi.org/10.1038/s41467-023-36956-5>.
- [16] Thomas J Lane. “Protein structure prediction has reached the single-structure frontier”. In: *Nature Methods* (2023), pp. 1–4.
- [17] Alastair DG Lawson. “Antibody-enabled small-molecule drug discovery”. In: *Nature Reviews Drug Discovery* 11.7 (2012), pp. 519–525.
- [18] Amanda R Moore et al. “RAS-targeted therapies: is the undruggable drugged?” In: *Nature Reviews Drug Discovery* 19.8 (2020), pp. 533–552.
- [19] Christopher J Draper-Joyce et al. “Positive allosteric mechanisms of adenosine A1 receptor-mediated analgesia”. In: *Nature* 597.7877 (2021), pp. 571–576.
- [20] David E Shaw et al. “Atomic-level characterization of the structural dynamics of proteins”. In: *Science* 330.6002 (2010), pp. 341–346.
- [21] Yibing Shan et al. “How does a small molecule bind at a cryptic binding site?” In: *PLoS computational biology* 18.3 (2022), e1009817.

- [22] William Sinko, Steffen Lindert, and J Andrew McCammon. “Accounting for receptor flexibility and enhanced sampling methods in computer-aided drug design”. In: *Chemical biology & drug design* 81.1 (2013), pp. 41–49.
- [23] Qianqian Zhao et al. “Enhanced Sampling Approach to the Induced-Fit Docking Problem in Protein–Ligand Binding: The Case of Mono-ADP-Ribosylation Hydrolase Inhibitors”. In: *Journal of chemical theory and computation* 17.12 (2021), pp. 7899–7911.
- [24] Bodhi P Vani et al. “From sequence to Boltzmann weighted ensemble of structures with AlphaFold2-RAVE”. In: *bioRxiv* (2022), pp. 2022–05.
- [25] Ruben Abagyan, Maxim Totrov, and Dmitry Kuznetsov. “ICM—A new method for protein modeling and design: Applications to docking and structure prediction from the distorted native conformation”. In: *Journal of computational chemistry* 15.5 (1994), pp. 488–506.
- [26] Qi Wu et al. “COACH-D: improved protein–ligand binding sites prediction with refined ligand-binding poses through molecular docking”. In: *Nucleic acids research* 46.W1 (2018), W438–W442.
- [27] Maarten L Hekkelman et al. “AlphaFill: enriching AlphaFold models with ligands and cofactors”. In: *Nature Methods* 20.2 (2023), pp. 205–213.
- [28] Marcus Fischer et al. “Incorporation of protein flexibility and conformational energy penalties in docking screens to improve ligand discovery”. In: *Nature chemistry* 6.7 (2014), pp. 575–583.
- [29] Noah Ollikainen, René M de Jong, and Tanja Kortemme. “Coupling protein side-chain and backbone flexibility improves the re-design of protein-ligand specificity”. In: *PLoS computational biology* 11.9 (2015), e1004335.
- [30] Marta Amaral et al. “Protein conformational flexibility modulates kinetics and thermodynamics of drug binding”. In: *Nature communications* 8.1 (2017), pp. 1–14.
- [31] Jue Wang et al. “Scaffolding protein functional sites using deep learning”. In: *Science* 377.6604 (2022), pp. 387–394.
- [32] Richard A Stein and Hassane S Mchaourab. “Modeling alternate conformations with alphafold2 via modification of the multiple sequence alignment”. In: *bioRxiv* (2021).
- [33] Diego Del Alamo et al. “Sampling alternative conformational states of transporters and receptors with AlphaFold2”. In: *Elife* 11 (2022), e75751.
- [34] Lim Heo and Michael Feig. “Multi-State Modeling of G-protein Coupled Receptors at Experimental Accuracy”. In: *Proteins: Structure, Function, and Bioinformatics* (2022).
- [35] Davide Sala and Jens Meiler. “Biasing AlphaFold2 to predict GPCRs and Kinases with user-defined functional or structural properties”. In: *bioRxiv* (2022), pp. 2022–12.
- [36] Louis Sanford Goodman et al. *Goodman and Gilman's the pharmacological basis of therapeutics*. Vol. 1549. McGraw-Hill New York, 1996.
- [37] Prafulla Dhariwal and Alexander Nichol. “Diffusion models beat gans on image synthesis”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 8780–8794.
- [38] Tom Brown et al. “Language models are few-shot learners”. In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.
- [39] Aditya Ramesh et al. “Hierarchical text-conditional image generation with clip latents”. In: *arXiv preprint arXiv:2204.06125* (2022).
- [40] Chitwan Saharia et al. “Photorealistic text-to-image diffusion models with deep language understanding”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 36479–36494.
- [41] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).
- [42] Maxim Zvyagin et al. “GenSLMs: Genome-scale language models reveal SARS-CoV-2 evolutionary dynamics”. In: *bioRxiv* (2022). DOI: [10.1101/2022.10.10.511571](https://doi.org/10.1101/2022.10.10.511571). eprint: <https://www.biorxiv.org/content/early/2022/11/23/2022.10.10.511571.full.pdf>. URL: <https://www.biorxiv.org/content/early/2022/11/23/2022.10.10.511571>.
- [43] Žiga Avsec et al. “Effective gene expression prediction from sequence by integrating long-range interactions”. In: *Nature methods* 18.10 (2021), pp. 1196–1203.
- [44] Tristan Bepler and Bonnie Berger. “Learning the protein language: Evolution, structure, and function”. In: *Cell systems* 12.6 (2021), pp. 654–669.

- [45] Justas Dauparas et al. “Robust deep learning–based protein sequence design using ProteinMPNN”. In: *Science* 378.6615 (2022), pp. 49–56.
- [46] John Ingraham et al. “Illuminating protein space with a programmable generative model”. In: *bioRxiv* (2022). DOI: [10.1101/2022.12.01.518682](https://doi.org/10.1101/2022.12.01.518682). eprint: <https://www.biorxiv.org/content/early/2022/12/02/2022.12.01.518682.full.pdf>. URL: <https://www.biorxiv.org/content/early/2022/12/02/2022.12.01.518682>.
- [47] Joseph L. Watson et al. “Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models”. In: *bioRxiv* (2022). DOI: [10.1101/2022.12.09.519842](https://doi.org/10.1101/2022.12.09.519842). eprint: <https://www.biorxiv.org/content/early/2022/12/14/2022.12.09.519842.full.pdf>. URL: <https://www.biorxiv.org/content/early/2022/12/14/2022.12.09.519842>.
- [48] Kevin E Wu et al. “Protein structure generation via folding diffusion”. In: *arXiv preprint arXiv:2209.15611* (2022).
- [49] Yeqing Lin and Mohammed AlQuraishi. “Generating Novel, Designable, and Diverse Protein Structures by Equivariantly Diffusing Oriented Residue Clouds”. In: *arXiv preprint arXiv:2301.12485* (2023).
- [50] Minkai Xu et al. “Geodiff: A geometric diffusion model for molecular conformation generation”. In: *arXiv preprint arXiv:2203.02923* (2022).
- [51] Bowen Jing et al. “Torsional Diffusion for Molecular Conformer Generation”. In: *arXiv preprint arXiv:2206.01729* (2022).
- [52] Wei Lu et al. “TANKBind: Trigonometry-Aware Neural NetworKs for Drug-Protein Binding Structure Prediction”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., 2022, pp. 7236–7249. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/2f89a23a19d1617e7fb16d4f7a049ce2-Paper-Conference.pdf.
- [53] Shuya Nakata, Yoshiharu Mori, and Shigenori Tanaka. “End-to-end protein-ligand complex structure generation with diffusion-based generative models”. In: *bioRxiv* (2022), pp. 2022–12.
- [54] Gabriele Corso et al. “Diffdock: Diffusion steps, twists, and turns for molecular docking”. In: *arXiv preprint arXiv:2210.01776* (2022).
- [55] Shuya Nakata, Yoshiharu Mori, and Shigenori Tanaka. “End-to-end protein-ligand complex structure generation with diffusion-based generative models”. In: *bioRxiv* (2022), pp. 2022–12.
- [56] Arne Schneuuing et al. “Structure-based drug design with equivariant diffusion models”. In: *arXiv preprint arXiv:2210.13695* (2022).
- [57] Jean-Baptiste Alayrac et al. “Flamingo: a visual language model for few-shot learning”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 23716–23736.
- [58] Renxiao Wang et al. “The PDBbind database: methodologies and updates”. In: *Journal of medicinal chemistry* 48.12 (2005), pp. 4111–4119.
- [59] Ian W Davis and David Baker. “RosettaLigand docking with full ligand and receptor flexibility”. In: *Journal of molecular biology* 385.2 (2009), pp. 381–392.
- [60] Yang Zhang and Jeffrey Skolnick. “TM-align: a protein structure alignment algorithm based on the TM-score”. In: *Nucleic acids research* 33.7 (2005), pp. 2302–2309.
- [61] Ernest L Eliel and Samuel H Wilen. *Stereochemistry of organic compounds*. John Wiley & Sons, 1994.
- [62] Jascha Sohl-Dickstein et al. “Deep unsupervised learning using nonequilibrium thermodynamics”. In: *International Conference on Machine Learning*. PMLR. 2015, pp. 2256–2265.
- [63] Yang Song et al. “Score-based generative modeling through stochastic differential equations”. In: *arXiv preprint arXiv:2011.13456* (2020).
- [64] Yuejiang Yu et al. “Do deep learning models really outperform traditional approaches in molecular docking?” In: *arXiv preprint arXiv:2302.07134* (2023).
- [65] Xu Yan et al. “PointSite: a point cloud segmentation tool for identification of protein ligand binding atoms”. In: *Journal of Chemical Information and Modeling* 62.11 (2022), pp. 2835–2845.
- [66] Radoslav Krivák and David Hoksza. “P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure”. In: *Journal of cheminformatics* 10 (2018), pp. 1–12.

- [67] Andrew T McNutt et al. “GNINA 1.0: molecular docking with deep learning”. In: *Journal of cheminformatics* 13.1 (2021), pp. 1–20.
- [68] Yuejiang Yu et al. “Uni-dock: A gpu-accelerated docking program enables ultra-large virtual screening”. In: (2022).
- [69] Hannes Stärk et al. “Equibind: Geometric deep learning for drug binding structure prediction”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 20503–20521.
- [70] Valerio Mariani et al. “IDDT: a local superposition-free score for comparing protein structures and models using distance difference tests”. In: *Bioinformatics* 29.21 (2013), pp. 2722–2728.
- [71] Xavier Robin et al. “Continuous Automated Model EvaluatiOn (CAMEO)—Perspectives on the future of fully automated evaluation of structure prediction methods”. In: *Proteins: Structure, Function, and Bioinformatics* 89.12 (2021), pp. 1977–1986.
- [72] Anthony K Rappé et al. “UFF, a full periodic table force field for molecular mechanics and molecular dynamics simulations”. In: *Journal of the American chemical society* 114.25 (1992), pp. 10024–10035.
- [73] Youngsook Shin et al. “Discovery of N-(1-Acryloylazetidin-3-yl)-2-(1 H-indol-1-yl) acetamides as Covalent Inhibitors of KRASG12C”. In: *ACS medicinal chemistry letters* 10.9 (2019), pp. 1302–1308.
- [74] Nicholas F Polizzi and William F DeGrado. “A defined structural unit enables de novo design of small-molecule–binding proteins”. In: *Science* 369.6508 (2020), pp. 1227–1233.
- [75] Artur Meller et al. “Predicting locations of cryptic pockets from single protein structures using the PocketMiner graph neural network”. In: *Nature Communications* 14.1 (2023), p. 1177. DOI: [10.1038/s41467-023-36699-3](https://doi.org/10.1038/s41467-023-36699-3). URL: <https://doi.org/10.1038/s41467-023-36699-3>.
- [76] Robert B Best, Gerhard Hummer, and William A Eaton. “Native contacts determine protein folding mechanisms in atomistic simulations”. In: *Proceedings of the National Academy of Sciences* 110.44 (2013), pp. 17874–17879.
- [77] Chin-Yu Chen et al. “Temperature-resolved cryo-EM uncovers structural bases of temperature-dependent enzyme functions”. In: *Journal of the American Chemical Society* 141.51 (2019), pp. 19983–19987.
- [78] M-Y Lee et al. “Harnessing the power of an X-ray laser for serial crystallography of membrane proteins crystallized in lipidic cubic phase”. In: *IUCrJ* 7.6 (2020), pp. 976–984.
- [79] Javier García-Nafriá et al. “Cryo-EM structure of the adenosine A2A receptor coupled to an engineered heterotrimeric G protein”. In: *elife* 7 (2018), e35946.
- [80] Nicolas Bertheleme et al. “Loss of constitutive activity is correlated with increased thermostability of the human adenosine A2A receptor”. In: *British journal of pharmacology* 169.5 (2013), pp. 988–998.
- [81] Kathryn Tunyasuvunakool et al. “Highly accurate protein structure prediction for the human proteome”. In: *Nature* 596.7873 (2021), pp. 590–596.
- [82] David S Wishart et al. “HMDB 5.0: the human metabolome database for 2022”. In: *Nucleic Acids Research* 50.D1 (2022), pp. D622–D631.
- [83] John J Irwin and Brian K Shoichet. “ZINC- a free database of commercially available compounds for virtual screening”. In: *Journal of chemical information and modeling* 45.1 (2005), pp. 177–182.
- [84] Alexander Rives et al. “Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences”. In: *Proceedings of the National Academy of Sciences* 118.15 (2021), e2016239118.
- [85] Ahmed Elnaggar et al. “ProtTrans: towards cracking the language of lifes code through self-supervised deep learning and high performance computing”. In: *IEEE transactions on pattern analysis and machine intelligence* (2021).
- [86] Chengxi Zang and Fei Wang. “MoFlow: an invertible flow model for generating molecular graphs”. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020, pp. 617–626.
- [87] Tianfan Fu et al. “Differentiable scaffolding tree for molecular optimization”. In: *arXiv preprint arXiv:2109.10469* (2021).
- [88] Andrew JR Plested. “Structural mechanisms of activation and desensitization in neurotransmitter-gated ion channels”. In: *Nature structural & molecular biology* 23.6 (2016), pp. 494–502.
- [89] Tiqing Liu et al. “BindingDB: a web-accessible database of experimentally determined protein–ligand binding affinities”. In: *Nucleic acids research* 35.suppl_1 (2007), pp. D198–D201.

- [90] Ilaria Piazza et al. “A map of protein-metabolite interactions reveals principles of chemical communication”. In: *Cell* 172.1-2 (2018), pp. 358–372.
- [91] Risi Imre Kondor and John Lafferty. “Diffusion kernels on graphs and other discrete structures”. In: *Proceedings of the 19th international conference on machine learning*. Vol. 2002. 2002, pp. 315–322.
- [92] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. “E (n) equivariant graph neural networks”. In: *International conference on machine learning*. PMLR. 2021, pp. 9323–9332.
- [93] Johannes Brandstetter et al. “Geometric and physical quantities improve e (3) equivariant message passing”. In: *arXiv preprint arXiv:2110.02905* (2021).
- [94] Yunzhu Li et al. “Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids”. In: *arXiv preprint arXiv:1810.01566* (2018).
- [95] Bowen Jing et al. “Learning from protein structure with geometric vector perceptrons”. In: *arXiv preprint arXiv:2009.01411* (2020).
- [96] Tao Shen et al. “E2Efold-3D: End-to-End Deep Learning Method for accurate de novo RNA 3D Structure Prediction”. In: *arXiv preprint arXiv:2207.01586* (2022).
- [97] Namrata Anand and Tudor Achim. “Protein Structure and Sequence Generation with Equivariant Denoising Diffusion Probabilistic Models”. In: *arXiv preprint arXiv:2205.15019* (2022).
- [98] Attilio Meucci. “Review of statistical arbitrage, cointegration, and multivariate Ornstein-Uhlenbeck”. In: *Cointegration, and Multivariate Ornstein-Uhlenbeck (May 14, 2009)* (2009).
- [99] Yang Song and Stefano Ermon. “Generative modeling by estimating gradients of the data distribution”. In: *Advances in Neural Information Processing Systems* 32 (2019).
- [100] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising diffusion probabilistic models”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6840–6851.
- [101] Tero Karras et al. “Elucidating the Design Space of Diffusion-Based Generative Models”. In: *arXiv preprint arXiv:2206.00364* (2022).
- [102] Jianyi Yang, Ambrish Roy, and Yang Zhang. “BioLiP: a semi-manually curated database for biologically relevant ligand–protein interactions”. In: *Nucleic acids research* 41.D1 (2012), pp. D1096–D1103.
- [103] Gáspár Pándy-Szekeres et al. “GPCRdb in 2023: state-specific structure models using AlphaFold2 and new ligand resources”. In: *Nucleic Acids Research* 51.D1 (2023), pp. D395–D402.
- [104] Milot Mirdita et al. “ColabFold: making protein folding accessible to all”. In: *Nature Methods* (2022), pp. 1–4.
- [105] Gustaf Ahdritz et al. “OpenFold: Retraining AlphaFold2 yields new insights into its learning mechanisms and capacity for generalization”. In: *bioRxiv* (2022), pp. 2022–11.

Supplementary Information

Contents

| | |
|-------------------------------------------------------------------|-----------|
| A Algorithm overview | 20 |
| B Diffusion SDEs | 20 |
| B.1 Preliminaries | 20 |
| B.2 Euclidean and chiral symmetries | 22 |
| B.3 Derivation of the semi-analytic integrator (Eq. 12) | 22 |
| C Molecular graph featurization and encoder details | 24 |
| C.1 Molecular representations | 24 |
| C.2 The MHT network architecture | 25 |
| C.3 MHT model pretraining | 26 |
| D Protein-ligand graph featurization | 28 |
| D.1 Protein residue sub-graph featurization | 29 |
| E The CPM network architecture | 29 |
| F The ESDM network architecture | 31 |
| G Confidence estimation heads | 32 |
| H Model training details | 33 |
| I Computational details | 35 |
| J Supplementary results | 35 |

A Algorithm overview

Given the set of query protein sequences, ligand molecular graphs, and optional template protein structure inputs, Algorithm S1 summarizes the process that NeuralPlexer uses to sample an ensemble of protein or protein-ligand complex structures.

Unless stated otherwise, $\text{LinearNoBias}(\cdot)$ denotes a standard linear transformation with a trainable weight matrix right-multiplied to the last input tensor dimension; $\text{MLP}(\cdot)$ denotes a standard 3-layer multilayer perceptron with GELU activation function [S2] and with layer normalization [S3] applied to the output activations.

B Diffusion SDEs

B.1 Preliminaries

For the temporally homogeneous forward-time SDE

$$d\mathbf{Z}_t = -(\mathbf{U}\boldsymbol{\lambda}\mathbf{U}^{-1})\mathbf{Z}_t dt + (\sigma\mathbf{U})d\mathbf{W}_t \quad (\text{S1})$$

or equivalently, in the space of latent coordinates

$$d\mathbf{z}_t = d(\mathbf{U}^{-1}\mathbf{Z}_t) = -\boldsymbol{\lambda} \cdot \mathbf{z}_t dt + \sigma d\mathbf{w}_t \quad (\text{S2})$$

the corresponding reverse-time SDE is given by

$$d\mathbf{Z}_t = [-(\mathbf{U}\boldsymbol{\lambda}\mathbf{U}^{-1})\mathbf{Z}_t - \sigma^2\mathbf{U}\mathbf{U}^\top\mathbf{s}_\phi(\mathbf{Z}_t; t)]dt + (\sigma\mathbf{U})d\mathbf{W}_t. \quad (\text{S3})$$

The initial coordinates at $t = T = 1.0$ are sampled from the prior distribution q_T

$$\mathbf{Z}_T \sim q_T := \mathcal{N}(\mathbf{0}, \mathbf{U}\frac{\sigma^2}{2\boldsymbol{\lambda}}\mathbf{U}^\top) \quad (\text{S4})$$

or equivalently,

$$\mathbf{Z}_T = \mathbf{U}\sqrt{\sigma/2}\boldsymbol{\lambda}^{-\frac{1}{2}}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad (\text{S5})$$

Algorithm S1 NeuralPLexer Inference

Require: $\{\mathbf{s}\}, \{G\}, N_{\text{conformations}}, N_{\text{steps}} = 40, use_template, compute_plddt$

1: $\{\mathbf{f}_{\text{PLM}}\} \leftarrow$ Compute ESM-2-650M features for all input chains in $\{\mathbf{s}\}$ Ref. [S1]

2: **if** $use_template$ **then**

3: $\{\mathbf{f}_{\text{template}}\} \leftarrow$ Retrieve template protein structure and compute template features

4: **end if**

5: **for** $i \in \{1, \dots, N_{\text{conformations}}\}$ **do**

6: $T = \text{DiffusionTimeSchedule}(\tau = 1.0)$ Equation (8)

7: Sample initial protein coordinates \mathbf{x}_T from the prior q_T Equation (S5)

8: $\text{residue_graph_0} \leftarrow$ Generate the residue-scale graph based on $(\mathbf{x}_T)_{C\alpha}$ Section D

9: $\mathbf{F}_P \leftarrow$ Sample N_P protein backbone frame nodes and gather embeddings

10: $\tilde{\mathbf{l}} \leftarrow \mathbf{0}$ $N_P = 96$

11: **# Generating contact maps and block-adjacency matrices (Equation (1))**

12: **if** $\{G\}$ contains more than 0 ligands **then** Algorithm S2

13: Compute MHT embeddings for all input ligand molecular graphs in $\{G\}$

14: $\mathbf{F}_L \leftarrow$ Sample N_L ligand frame nodes, gather and symmetrize MHT embeddings $N_L = 32$

15: **for** $k \in \{1, \dots, N_L\}$ **do** Section E

16: $\text{residue_graph_k} \leftarrow \text{CPMForward}(\text{residue_graph_0}, \tilde{\mathbf{l}}, \tau = 1.0)$

17: $\hat{P} \in \mathbb{R}^{N_{\text{res}} \times N_L \times N_{\text{bins}}}, N_{\text{bins}} = 32$

18: $\tilde{L}_k \leftarrow \text{HistogramToContactMap}(\hat{P})$ Equation (4)

19: $\hat{\mathbf{L}}_k \leftarrow \text{SumIntoPatches}(\tilde{L}_k) \odot [1 - \max_{\text{column-wise}}(\tilde{\mathbf{l}})]$

20: $\mathbf{l}_k = \text{OneHot}(i_k, j_k); (i_k, j_k) \sim \text{Categorical}_{N_P \times N_L}(\hat{\mathbf{L}}_k)$ $\mathbf{l}_k \in \{0, 1\}^{N_P \times N_L}$

21: $\tilde{\mathbf{l}} \leftarrow \tilde{\mathbf{l}} + \mathbf{l}_k$ $\tilde{\mathbf{l}} \in \{0, 1\}^{N_P \times N_L}$

22: **end for**

23: Compute \mathbf{c} and \mathbf{U} for ligand degrees of freedom using predicted $\hat{\mathbf{L}} := \hat{\mathbf{L}}_{N_L}$ Equation (7)

24: Sample initial ligand coordinates \mathbf{y}_T from the prior q_T Equation (S5)

25: $\mathbf{Z}_T \leftarrow \text{concat}(\mathbf{x}_T, \mathbf{y}_T)$

26: **else**

27: $\mathbf{Z}_T \leftarrow \mathbf{x}_T$

28: **end if**

29: **# Generating the 3D structure for all heavy atoms (Equation (2))**

30: Compute MHT embeddings for all amino acid molecular graphs Algorithm S2

31: **for** $\tau \in \{1, 1 - \Delta\tau, \dots, \Delta\tau\}$ **do** $\Delta\tau = 1/N_{\text{steps}}$

32: $t \leftarrow \text{DiffusionTimeSchedule}(\tau)$ Equation (8)

33: $\Delta t \leftarrow \text{DiffusionTimeSchedule}(\tau) - \text{DiffusionTimeSchedule}(\tau - \Delta\tau)$ Equation (8)

34: $\text{residue_graph, atomic_graph} \leftarrow$ Regenerate the residue-scale and atomic-scale graph based on \mathbf{Z}_t Section D

35: $\text{residue_graph} \leftarrow \text{CPMForward}(\text{residue_graph}, \tilde{\mathbf{l}}, \tau)$ Section E

36: $\text{graph_rep} \leftarrow \text{Collate}(\text{residue_graph}, \text{atomic_graph}, \text{cross_scale_graph})$ Table S4

37: $\hat{\mathbf{Z}}_0 \leftarrow \text{ESDMForward}(\mathbf{Z}_t; \text{graph_rep}, \tau)$ Section F

38: $\beta_{t-\Delta t} \leftarrow \text{InvTempSchedule}(\tau - \Delta\tau)$ See Methods, LSA-SDE

39: $\mathbf{Z}_{t-\Delta t} = \text{IntegratorStep}(\mathbf{Z}_t, \hat{\mathbf{Z}}_0, \beta_{t-\Delta t}, \Delta t)$ Equation (12)

40: **end for**

41: **# Assigning confidence estimations (per-residue and per-ligand-atom pLDDT)**

42: **if** $compute_plddt$ **then**

43: $\text{residue_graph, atomic_graph} \leftarrow$ Regenerate the residue-scale and atomic-scale graph based on \mathbf{Z}_t Section D

44: $\text{residue_graph} \leftarrow \text{CPMForward}(\text{residue_graph}, \mathbf{0}, \tau = 0.0)$ Section E

45: $\text{graph_rep} \leftarrow \text{Collate}(\text{residue_graph}, \text{atomic_graph}, \text{cross_scale_graph})$ Table S4

46: $p\text{LDDT_gram} = \text{ConfidenceEstimationHead}(\mathbf{Z}_0; \text{graph_rep})$ Section G

47: $p\text{LDDT_prot}, p\text{LDDT_lig} = \text{compute_plDDT}(p\text{LDDT_gram})$ Equation S29

48: **yield** $\mathbf{Z}_0, (p\text{LDDT_gram}, p\text{LDDT_prot}, p\text{LDDT_lig})$

49: **else**

50: **yield** \mathbf{Z}_0

51: **end if**

52: **end for**

B.2 Euclidean and chiral symmetries

Given group G , a function $f : X \rightarrow Y$ is said to be equivariant if for all $g \in G$ and $x \in X$, $f(\varphi_X(g) \cdot x) = \varphi_Y(g) \cdot f(x)$, and f is said to be invariant if $f(\varphi_X(g) \cdot x) = f(x)$. We are interested in the special Euclidean group $G = \text{SE}(3)$ which consists of all global rigid translation and rotation operations $g \cdot \mathbf{Z} := \mathbf{t} + \mathbf{Z} \cdot \mathbf{R}$ where $\mathbf{t} \in \mathbb{R}^3$ and $\mathbf{R} \in \text{SO}(3)$. To adhere to the physical constraint that p_{data} is always $\text{SE}(3)$ -invariant, the transition kernels of the forward-time SDE need to satisfy $\text{SE}(3)$ -equivariance $q(\mathbf{Z}_{t+s} | \mathbf{Z}_t) = q(\varphi_{t+s}(g) \cdot \mathbf{Z}_{t+s} | \varphi_t(g) \cdot \mathbf{Z}_t)$ such that all marginals are invariant $q_t(\mathbf{Z}_t) = q_t(\varphi_t(g) \cdot \mathbf{Z}_t)$ for all diffusion times t . Since the forward SDE only involves terms that are isotropic in (x, y, z) components, the proof is straightforward:

$$\begin{aligned} & q(\sqrt{\alpha_s} \mathbf{t} + \mathbf{Z}_{t+s} \cdot \mathbf{R} | \mathbf{t} + \mathbf{Z}_t \cdot \mathbf{R}) \\ &= \mathcal{N}(\sqrt{\alpha_s} \mathbf{t} + \mathbf{Z}_{t+s} \cdot \mathbf{R}; \sqrt{\alpha_s} (\mathbf{t} + \mathbf{Z}_t \cdot \mathbf{R}), \mathbf{U} \frac{\sqrt{1 - \alpha_s} \sigma^2}{2\lambda} \mathbf{U}^T) \\ &= \mathcal{N}(\sqrt{\alpha_s} \mathbf{t} \mathbf{R}^\top + \mathbf{Z}_{t+s} \cdot \mathbf{R} \mathbf{R}^\top; \sqrt{\alpha_s} (\mathbf{t} \mathbf{R}^\top + \mathbf{Z}_t \cdot \mathbf{R} \mathbf{R}^\top), \mathbf{U} \frac{\sqrt{1 - \alpha_s} \sigma^2}{2\lambda} \mathbf{U}^T \otimes \mathbf{R} \mathbf{R}^\top) \\ &= \mathcal{N}(\mathbf{Z}_{t+s}; \mathbf{Z}_t, \mathbf{U} \frac{\sqrt{1 - \alpha_s} \sigma^2}{2\lambda} \mathbf{U}^T) \\ &= q(\mathbf{Z}_{t+s} | \mathbf{Z}_t). \end{aligned}$$

where the translation term after a time interval s is scaled by a factor $\sqrt{\alpha_s}$ due to the linear drift term in Eq. (S1). Because all transition kernels are $\text{SE}(3)$ -equivariant, it then follows that the score function $\nabla_{\mathbf{Z}} \log q_t(\mathbf{Z})$ is also equivariant:

$$\begin{aligned} & \nabla_{\mathbf{Z}'} \log q_t(\mathbf{Z}') \quad \text{where } \mathbf{Z}' = \sqrt{\alpha_t} \mathbf{t} + \mathbf{Z} \cdot \mathbf{R} \\ &= \nabla_{\mathbf{Z}'} \log [\mathbb{E}_{\mathbf{Z}_0 \sim q_{\text{data}}} q_t(\mathbf{Z}' | \mathbf{Z}_0)] \\ &= \nabla_{\mathbf{Z}'} \log [\mathbb{E}_{\mathbf{Z}_0 \sim q_{\text{data}}} q_t(\mathbf{Z}' | \mathbf{t} + \mathbf{Z}_0 \cdot \mathbf{R})] \\ &= \nabla_{\mathbf{Z}'} \log [\mathbb{E}_{\mathbf{Z}_0 \sim q_{\text{data}}} q_t(\mathbf{Z} | \mathbf{Z}_0)] \\ &= \frac{\partial \mathbf{Z}}{\partial \mathbf{Z}'} \nabla_{\mathbf{Z}} \log [\mathbb{E}_{\mathbf{Z}_0 \sim q_{\text{data}}} q_t(\mathbf{Z} | \mathbf{Z}_0)] \\ &= \nabla_{\mathbf{Z}} \log q_t(\mathbf{Z}) \cdot \mathbf{R}. \end{aligned}$$

While the forward SDE is $\text{E}(3)$ -equivariant as the noising process satisfies $q(-\mathbf{Z}(t+s) | -\mathbf{Z}(t)) = q(\mathbf{Z}(t+s) | \mathbf{Z}(t))$, it is worth noting that the reverse SDE is only $\text{SE}(3)$ -equivariant as parity-inversion transformations $i : \mathbf{Z} \mapsto -\mathbf{Z}$ on the data distribution p_{data} is physically forbidden and thus the score $\nabla_{\mathbf{Z}} \log q_t(\mathbf{Z})$ is of broken chiral symmetry; in general: there exists \mathbf{Z} such that $\nabla_{-\mathbf{Z}} \log q_t(-\mathbf{Z}) \neq -\nabla_{\mathbf{Z}} \log q_t(\mathbf{Z})$.

B.3 Derivation of the semi-analytic integrator (Eq. 12)

Without loss of generality, we consider a homogeneous, fixed-temperature analog of the temperature-adjusted reverse-time SDE (10) with $\mathbf{z}_t \in \mathbb{R}^3$, $t \in (0, \infty)$

$$d\mathbf{z}_t = [-\theta \mathbf{z}_t - \frac{1+\beta}{2} \sigma^2 \mathbf{s}_\phi(\mathbf{z}_t; t)] dt + \sqrt{\frac{1}{\beta}} \sigma d\mathbf{W}_t \quad (\text{S6})$$

where

$$\mathbf{s}_\phi(\mathbf{z}_t; t) = \frac{\sqrt{\alpha_t} \hat{\mathbf{z}}_0(\mathbf{z}_t) - \mathbf{z}_t}{1 - \alpha_t}; \quad \alpha_t = \exp(-2\theta t) \quad (\text{S7})$$

thus

$$d\mathbf{z}_t = [-\theta \mathbf{z}_t - \frac{1+\beta}{2} \sigma^2 \frac{\exp(-\theta t) \hat{\mathbf{z}}_0(\mathbf{z}_t) - \mathbf{z}_t}{1 - \exp(-2\theta t)}] dt + \sqrt{\frac{1}{\beta}} \sigma d\mathbf{W}_t \quad (\text{S8})$$

For an integration time interval from $s = t + \Delta t$ to t , we make the approximation that the score function is linear with respect to the attraction term $\hat{\mathbf{z}}_0(\mathbf{z}_s; \phi, s) - \frac{1}{\sqrt{\alpha_\tau}} \mathbf{z}_\tau$ for $\tau \in [t, s]$:

$$d\mathbf{z}_\tau = [-\theta \mathbf{z}_\tau - \frac{1+\beta}{2} \sigma^2 \frac{\exp(-\theta \tau) \hat{\mathbf{z}}_0(\mathbf{z}_s) - \mathbf{z}_\tau}{1 - \exp(-2\theta \tau)}] d\tau + \sqrt{\frac{1}{\beta}} \sigma d\mathbf{W}_\tau \quad (\text{S9})$$

or equivalently,

$$d\mathbf{z}_\tau = a(\tau) + b(\tau)\mathbf{z}_\tau d\tau + c d\mathbf{W}_\tau \quad (\text{S10})$$

where

$$a(\tau) = -\frac{1+\beta}{2}\sigma^2 \frac{\exp(-\theta\tau)\hat{\mathbf{z}}_0(\mathbf{z}_s)}{1-\exp(-2\theta\tau)}, \quad b(\tau) = [\frac{\frac{1+\beta}{2}\sigma^2}{1-\exp(-2\theta\tau)} - \theta]; \quad c = \sqrt{\frac{1}{\beta}}\sigma. \quad (\text{S11})$$

Defining $\Psi(t, s) := \exp(\int_s^t -b(\tau)d\tau)$, by applying Ito's Lemma we note that

$$d(\Psi(t, s)\mathbf{z}_t) = [\frac{d}{dt}\Psi(t, s)\mathbf{z}_t + a(t)\Psi(t, s) + b(t)\mathbf{z}_t\Psi(t, s)]dt + c\Psi(t, s)d\mathbf{W}_t \quad (\text{S12})$$

$$= a(t)\Psi(t, s)dt + c\Psi(t, s)d\mathbf{W}_t \quad (\text{S13})$$

which implies

$$\mathbb{E}(\mathbf{z}_t) = \int_s^t a(\tau) \frac{\Psi(\tau, s)}{\Psi(t, s)} d\tau + \frac{\mathbf{z}_s}{\Psi(t, s)}; \quad \text{Var}(\mathbf{z}_t) = \int_s^t \left(c \frac{\Psi(\tau, s)}{\Psi(t, s)} \right)^2 d\tau \quad (\text{S14})$$

Computing the integrals analytically yields

$$\Psi(t, s) = \left(\frac{e^{2\theta s} - 1}{e^{2\theta t} - 1} \right)^{\frac{1+\beta}{4\theta}\sigma^2} \cdot e^{\theta(t-s)} \quad (\text{S15})$$

$$\int_s^t a(\tau) \frac{\Psi(\tau, s)}{\Psi(t, s)} d\tau = \int_s^t -\frac{1+\beta}{2}\sigma^2 \frac{\exp(-\theta\tau)\hat{\mathbf{z}}_0(\mathbf{z}_s)}{1-\exp(-2\theta\tau)} \left(\frac{e^{2\theta t} - 1}{e^{2\theta\tau} - 1} \right)^{\frac{1+\beta}{4\theta}\sigma^2} \cdot e^{\theta(\tau-t)} d\tau \quad (\text{S16})$$

$$= \int_s^t -\frac{1+\beta}{2}\sigma^2 \frac{\hat{\mathbf{z}}_0(\mathbf{z}_s)}{1-\exp(-2\theta\tau)} \left(\frac{e^{2\theta t} - 1}{e^{2\theta\tau} - 1} \right)^{\frac{1+\beta}{4\theta}\sigma^2} \cdot e^{-\theta t} d\tau \quad (\text{S17})$$

$$= -\frac{1+\beta}{2}\sigma^2 \hat{\mathbf{z}}_0(\mathbf{z}_s) e^{-\theta t} \left(\frac{e^{2\theta t} - 1}{e^{2\theta\tau} - 1} \right)^{\frac{1+\beta}{4\theta}\sigma^2} \int_s^t \frac{e^{2\theta\tau}}{(e^{2\theta\tau} - 1)^{1+\frac{1+\beta}{4\theta}\sigma^2}} d\tau \quad (\text{S18})$$

$$= e^{-\theta t} \left[1 - \left(\frac{e^{2\theta t} - 1}{e^{2\theta s} - 1} \right)^{\frac{1+\beta}{4\theta}\sigma^2} \right] \hat{\mathbf{z}}_0(\mathbf{z}_s) \quad (\text{S19})$$

$$\mathbb{E}(\mathbf{z}_t) = e^{-\theta t} \left[1 - \left(\frac{e^{2\theta t} - 1}{e^{2\theta s} - 1} \right)^{\frac{1+\beta}{4\theta}\sigma^2} \right] \hat{\mathbf{z}}_0(\mathbf{z}_s) + \left(\frac{e^{2\theta t} - 1}{e^{2\theta s} - 1} \right)^{\frac{1+\beta}{4\theta}\sigma^2} \cdot e^{\theta(s-t)} \mathbf{z}_s \quad (\text{S20})$$

For simplicity, we adapt the variance term to that of the forward SDE: $\text{Var}(\mathbf{z}_t) \approx \frac{\sigma^2}{2\theta\beta}(e^{-2\theta t} - e^{-2\theta s})$. Matching the conditional expectations and variances to the Gaussian transition kernel $q_{t:t+\Delta t}(\cdot | \mathbf{z}_{t+\Delta t})$ and directly generalizing to the rotation-corrected multivariate setting, we recover Equation 12. Note that the DDIM [S4] integrator can be recovered by removing the noise term and setting $\beta \equiv 0, \sigma \equiv 1, \theta \equiv \frac{1}{2}$ which corresponds to the standard variance-preserving (VP)-SDE [S5].

C Molecular graph featurization and encoder details

C.1 Molecular representations

Given a set of molecular graphs $\{G\}$, the MHT network processes the following collection of embeddings:

- Atom representations $\mathbf{f}_{\text{atom}} \in \mathbb{R}^{N_{\text{atom}}} \times c$. The input atom representation is a concatenation of one-hot encodings of element group index and period index for the given atom, which is embedded by a linear projection layer $\mathbb{R}^{18+7} \rightarrow \mathbb{R}^c$;
- Frame representations $\mathbf{f}_{\text{frame}} \in \mathbb{R}^{N_{\text{frame}}} \times c$. For a given frame u , $(\mathbf{H}_{\text{frame}})_u$ is initialized by a 2-layer MLP $\mathbb{R}^{4*2+18+7} \rightarrow \mathbb{R}^c$ that embeds the bond type encodings (defined as [is_single, is_double, is_triple, is_aromatic]) of the "incoming" bond $(i(u), j(u))$, "outgoing" bond $(j(u), k(u))$, and the atom type encoding of the center atom $j(u)$;
- Stereochemistry edge encodings $\mathbf{S} \in \mathbb{R}^{N_{\text{frame}} \times N_{\text{frame}} \times c_s}$, as detailed in Table S1. \mathbf{S} is a sparse tensor where an element \mathbf{S}_{uv} is nonzero only if the pair of frames (u, v) is adjacent, i.e., frame u and frame v sharing a common chemical bond; only pairs with non-zero \mathbf{S}_{uv} are included as model inputs.
- 3-hop edge representations $\mathbf{f}_{\text{aa}} \in \mathbb{R}^{N_{\text{atom}} \times N_{\text{atom}} \times c_p}$. For each pair of atoms (i, j) , the element $(\mathbf{f}_{\text{aa}})_{ij}$ is initialized by a linear layer $\mathbb{R}^{4*4} \rightarrow \mathbb{R}^{c_p}$ that embeds the set of binary graph-distance encodings of whether a path of k ($k \in \{0, 1, 2, 3\}$) chemical bonds exists between atom i and j , as well as the bond type one-hot encoding in case a chemical bond exists between atom i and j ; only edges with non-zero \mathbf{f}_{aa} are included as model inputs.
- Pair representations $\mathbf{f}_{\text{fa}} \in \mathbb{R}^{N_{\text{frame}} \times N_{\text{atom}} \times c_p}$. For each frame-atom pair (u, l) , the element $(\mathbf{f}_{\text{fa}})_{ul}$ is initialized by a linear layer $\mathbb{R}^{3*4} \rightarrow \mathbb{R}^{c_p}$ that embeds the concatenation of graph-distance encodings $\{(\mathbf{f}_{\text{aa}})_{i(u)l}, (\mathbf{f}_{\text{aa}})_{j(u)l}, (\mathbf{f}_{\text{aa}})_{k(u)l}\}$.

We denote \mathbf{X}_s as the $N_{\text{frame}} \times N_{\text{frame}}$ binary adjacency matrix of edges among frame nodes, and \mathbf{X}_a as the $N_{\text{atom}} \times N_{\text{atom}}$ binary adjacency matrix of 3-hop edges among atomic nodes. We additionally denote \mathbf{H}_{fa} as the $N_{\text{frame}} \times N_{\text{atom}}$ incidence matrix between atomic nodes and frame nodes, i.e., $(\mathbf{H}_{\text{fa}})_{u,l} = 1$ if $l \in \{i(u), j(u), k(u)\}$, and otherwise zero.

Elements of the stereochemistry encoding tensor \mathbf{S} are determined based on the relative orientations among neighboring frames of the input molecular graph. $\text{is_above_plane}(u, v)$ is defined as one of the three atoms in frame v is above the plane formed by frame u with normal vector $\mathbf{v}_u = \frac{(\mathbf{r}_{j(u)} - \mathbf{r}_{i(u)}) \times (\mathbf{r}_{k(u)} - \mathbf{r}_{j(u)})}{\|\mathbf{r}_{j(u)} - \mathbf{r}_{i(u)}\| \|\mathbf{r}_{k(u)} - \mathbf{r}_{j(u)}\|}$; $\text{is_same_side}(u, v)$ is true iff the two bonds not shared between u, v are on the same side of the common bond, equivalent to $\mathbf{v}_u \cdot \mathbf{v}_v > 0$, or vice versa. Our current technical implementations for is_above_plane and is_same_side are based on computing the normal vectors and dot-products using the coordinates from an auxiliary conformer, but we note that in principle all stereochemistry encodings can be generated based on cheminformatic rules without explicitly generating all atomic coordinates.

Table S1. Stereochemistry encoding definitions.

| Feature | Definition |
|-------------------------------------------------------|------------------------------------------------------------------------------------------------|
| # Relative topological orientation between two frames | |
| $\mathbf{S}_{uv,0}$ | $\text{common_bond}(u, v) = \text{incoming_bond}(u)$ |
| $\mathbf{S}_{uv,1}$ | $\text{common_bond}(u, v) = \text{incoming_bond}(v)$ |
| $\mathbf{S}_{uv,2}$ | $\text{common_bond}(u, v) = \text{outgoing_bond}(u)$ |
| $\mathbf{S}_{uv,3}$ | $\text{common_bond}(u, v) = \text{outgoing_bond}(v)$ |
| # Detect small ring structures | |
| $\mathbf{S}_{uv,4}$ | $i(v) \in \{i(u), j(u), k(u)\}$ |
| $\mathbf{S}_{uv,5}$ | $j(v) \in \{i(u), j(u), k(u)\}$ |
| $\mathbf{S}_{uv,6}$ | $k(v) \in \{i(u), j(u), k(u)\}$ |
| # Polyhedral chiral center stereochemistry | |
| $\mathbf{S}_{uv,7}$ | $(j(u) = j(v)) \wedge \text{is_above_plane}(u, v)$ |
| $\mathbf{S}_{uv,8}$ | $(j(u) = j(v)) \wedge \text{is_below_plane}(u, v)$ |
| # Planar stereochemistry for double and π bonds | |
| $\mathbf{S}_{uv,9}$ | $\text{is_double_or_aromatic}(\text{common_bond}(u, v)) \vee \text{is_same_side}(u, v)$ |
| $\mathbf{S}_{uv,10}$ | $\text{is_double_or_aromatic}(\text{common_bond}(u, v)) \vee \text{not_same_side}(u, v)$ |

The notion of "frames" in a coordinate-free topological molecular graph is justified by the observation that most bending and stretching modes in molecular vibrations are of high frequency, i.e., most bond lengths and bond angles fall into a small range as predicted by valence bond theory, such that the local frames comprise a consistent molecular representation without prior knowledge of 3D coordinates.

C.2 The MHT network architecture

The forward pass of the Molecular Heat Transformer (MHT) propagates both the node and edges of a graph representation. After executing the MHT network for all ligands in $\{G\}$, we proceed by uniformly sampling N_L anchor nodes from all N_{frame} nodes for subsequent processing; we denote \mathbf{H}_L as the $N_L \times N_{\text{frame}}$ incidence matrix indicating whether a frame node is selected.

Algorithm S2 Molecular Heat Transformer (MHT) Inference

```

def MHT( $\mathbf{f}_{\text{atom}}, \mathbf{f}_{\text{frame}}, \mathbf{f}_{\text{aa}}, \mathbf{f}_{\text{fa}}, \mathbf{S}, N_{\text{blocks}} = 8, K = 8, c_p = 512, c_p = 64$ )
1: for  $i = 1$  to  $N_{\text{blocks}}$  do
2:   # Pair update block (Line 3-7)
3:    $\mathbf{f}_K, \mathbf{f}_Q, \mathbf{b} = \text{LinearNoBias}(\mathbf{f}_{\text{frame}}), \text{LinearNoBias}(\mathbf{f}_{\text{frame}}), \text{LinearNoBias}(\mathbf{S})$ 
   # Computing a normalized affinity matrix for nearest-neighbor frames
4:    $\mathbf{U}_l = \text{Softmax}_{\text{row-wise}}\left(\frac{1}{\sqrt{c_p}}(\mathbf{f}_K \cdot \mathbf{f}_Q^\top) + \mathbf{b} - \text{Inf} \cdot (1 - \mathbf{X}_s)\right)$ 
   # Propagating to all frame-frame pairs via heat kernel expansion
   # Approximating the matrix exponential  $\exp(\mathbf{U}) \approx (\mathbf{1} + \frac{1}{K}\mathbf{U})^K$ 
5:    $\tilde{\mathbf{U}} = (\mathbf{I} + \frac{1}{K}\mathbf{U}_l)^K$ 
   # Updating frame-atom pairs using broadcasted kernel matrices
6:    $\mathbf{g} = \tilde{\mathbf{U}} \cdot \text{LinearNoBias}(\mathbf{f}_{\text{fa}})$ 
7:    $\mathbf{f}_{\text{fa}} \leftarrow \text{MLP}(\text{concat}(\mathbf{g}, \mathbf{f}_{\text{fa}})) + \mathbf{f}_{\text{fa}}$ 
   # Graph attention block (Line 8-12)
8:    $\mathbf{f}_{\text{node}} = \text{concat}_{\text{column-wise}}(\mathbf{f}_{\text{atom}}, \mathbf{f}_{\text{frame}})$ 
9:    $\mathbf{f}_{\text{edge}} = \text{concat}_{\text{column-wise}}(\text{concat}_{\text{row-wise}}(\mathbf{f}_{\text{aa}}, \mathbf{f}_{\text{fa}}^\top), \text{concat}_{\text{row-wise}}(\mathbf{f}_{\text{fa}}, \mathbf{S}))$ 
10:   $\mathbf{X} = \text{concat}_{\text{column-wise}}(\text{concat}_{\text{row-wise}}(\mathbf{X}_a, \mathbf{1}), \text{concat}_{\text{row-wise}}(\mathbf{1}, \mathbf{X}_s))$ 
11:   $\mathbf{f}_{\text{out}, \dots, \mathbf{z}_{\text{out}}} = \text{MHAwithEdgeBias}(\mathbf{f}_{\text{node}}, \mathbf{f}_{\text{node}}, \mathbf{f}_{\text{edge}}, \mathbf{X}, n_{\text{heads}} = 8, c_{\text{head}} = 8)$ 
12:   $\mathbf{f}_{\text{edge}} \leftarrow \text{MLP}(\text{LinearNoBias}(\mathbf{z}_{\text{out}}) + \mathbf{f}_{\text{edge}}) + \mathbf{f}_{\text{edge}}$ 
   # Note update block
13:   $\mathbf{f}_{\text{node}} \leftarrow \text{MLP}_{\text{hidden\_dim}=2048}(\mathbf{f}_{\text{out}} + \mathbf{f}_{\text{node}}) + \mathbf{f}_{\text{node}}$ 
14:  Update  $\mathbf{f}_{\text{frame}}, \mathbf{f}_{\text{fa}}$  from merged embeddings  $\mathbf{f}_{\text{node}}, \mathbf{f}_{\text{edge}}$ 
15: end for
16: if Subsample ligand pair representations then
17:    $\mathbf{F}_L \leftarrow \frac{1}{2}(\mathbf{H}_L \cdot \mathbf{f}_{\text{fa}} + (\mathbf{H}_L \cdot \mathbf{f}_{\text{fa}})^\top)$ 
18: end if
19: return  $\mathbf{f}_{\text{atom}}, \mathbf{f}_{\text{frame}}, \mathbf{f}_{\text{fa}}, \mathbf{f}_{\text{aa}}, \mathbf{F}_L$ 

```

$$\mathbf{F}_L \in \mathbb{R}^{N_L \times N_L \times c_p}$$

MHAwithEdgeBias (Algorithm S3) denotes a multi-head cross-attention layer between source node embeddings and destination node embeddings, with edge embeddings entering attention computation as a relative positional encoding term.

Algorithm S3 Multi-head Graph Attention with Edge Bias. \mathbf{X} denotes the adjacency matrix of all edges on the graph.

```

def MHAwithEdgeBias( $\mathbf{f}_{\text{src}}, \mathbf{f}_{\text{dst}}, \mathbf{f}_{\text{edge}}, \mathbf{X}, c = 512, c_p = 64, n_{\text{head}}, c_{\text{head}}$ )  $\mathbf{f}_{\text{src}} \in \mathbb{R}^{N_{\text{src}} \times c}, \mathbf{f}_{\text{dst}} \in \mathbb{R}^{N_{\text{dst}} \times c}, \mathbf{f}_{\text{edge}} \in \mathbb{R}^{N_{\text{edges}} \times c_p}$ 
1:  $\mathbf{f}_K, \mathbf{f}_{V_s} = \text{LinearNoBias}(\mathbf{f}_{\text{src}})$   $\mathbf{f}_K, \mathbf{f}_{V_s} \in \mathbb{R}^{N_{\text{src}} \times n_{\text{heads}} \times c_{\text{heads}}}$ 
2:  $\mathbf{f}_Q, \mathbf{f}_{V_d} = \text{LinearNoBias}(\mathbf{f}_{\text{dst}})$   $\mathbf{f}_Q, \mathbf{f}_{V_d} \in \mathbb{R}^{N_{\text{dst}} \times n_{\text{heads}} \times c_{\text{heads}}}$ 
3:  $\mathbf{b} = \text{LinearNoBias}(\mathbf{f}_{\text{edge}})$   $\mathbf{b} \in \mathbb{R}^{N_{\text{edges}} \times n_{\text{heads}} \times 1}$ 
4:  $\mathbf{z}_{ij} = \frac{1}{\sqrt{c_{\text{head}}}}(\mathbf{f}_{K,i} \cdot \mathbf{f}_{Q,j}^\top) + \mathbf{b}_{ij}$   $\mathbf{z} \in \mathbb{R}^{N_{\text{edges}} \times n_{\text{heads}} \times 1}$ 
5:  $\mathbf{a}_{ij} = \text{Softmax}_j^{X_{ij}=1} \mathbf{z}_{ij}$   $\mathbf{a} \in \mathbb{R}^{N_{\text{edges}} \times n_{\text{heads}} \times 1}$ 
6:  $\mathbf{f}_{os,i} \leftarrow \text{LinearNoBias}(\sum_j^{X_{ij}=1} \mathbf{a}_{ij} \odot \mathbf{f}_{V_d,j})$   $\mathbf{f}_{os} \in \mathbb{R}^{N_{\text{src}} \times c}$ 
7:  $\mathbf{f}_{od,j} \leftarrow \text{LinearNoBias}(\sum_i^{X_{ij}=1} \mathbf{a}_{ij} \odot \mathbf{f}_{V_s,i})$   $\mathbf{f}_{od} \in \mathbb{R}^{N_{\text{dst}} \times c}$ 
8: return  $\mathbf{f}_{os}, \mathbf{f}_{od}, \mathbf{z}$ 

```

Algorithm S4 rigidFrom3Points (adapted from Ref. [S6], Alg. 21)

```

def rigidFrom3Points( $\mathbf{x}_1 \in \mathbb{R}^3, \mathbf{x}_2 \in \mathbb{R}^3, \mathbf{x}_3 \in \mathbb{R}^3, \epsilon = 0.01 \text{ \AA}$ )
1:  $\mathbf{v}_1 \leftarrow \mathbf{x}_3 - \mathbf{x}_2$ 
2:  $\mathbf{v}_2 \leftarrow \mathbf{x}_1 - \mathbf{x}_2$ 
3:  $\mathbf{e}_1 \leftarrow \frac{\mathbf{v}_1}{\sqrt{\|\mathbf{v}_1\|_2^2 + \epsilon^2}}$ 
4:  $\mathbf{u}_2 \leftarrow \mathbf{v}_2 - (\mathbf{e}_1^\top \mathbf{v}_2) \mathbf{e}_1$ 
5:  $\mathbf{e}_2 \leftarrow \frac{\mathbf{u}_2}{\sqrt{\|\mathbf{u}_2\|_2^2 + \epsilon^2}}$ 
6:  $\mathbf{e}_3 \leftarrow \mathbf{e}_1 \times \mathbf{e}_2$ 
7:  $\mathbf{R} \leftarrow \text{vstack}(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$   $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ 
8:  $\mathbf{t} \leftarrow \mathbf{x}_2$ 
9: return  $\mathbf{t}, \mathbf{R}$ 

```

C.3 MHT model pretraining

Table S2. Composition of the dataset used for pretraining the MHT chemical encoder.

| Data source | Num. samples collected | Sampling weight | \mathcal{L}_{3D} | \mathcal{L}_{REG} | \mathcal{L}_{MLM} |
|--------------------------------------------------|------------------------|-----------------|--------------------|---------------------|---------------------|
| BioLip [S7] ligands (deposited date<2019.1.1) | 160k | 1.0 | + | - | + |
| GEOM [S8] | 450k * 5 | 0.25 | + | - | + |
| PEPCONF [S9] | 3775 | 5.0 | + | - | + |
| PubChemQC [S10, S11] | 3.4M | 0.25 | + | - | + |
| Chemical Checker [S12] | 800k | 1.0 | - | + | + |

In Table S2 we summarize the small molecule datasets used for training the MHT encoder used in the reported NeuralPlexer model. The loss function used in MHT pretraining is the following:

$$\mathcal{L}_{\text{lig-pretraining}} = \mathcal{L}_{3D-\text{marginal}} + \mathcal{L}_{3D-\text{DSM}} + \mathcal{L}_{CC-\text{regression}} + 0.01 \cdot \mathcal{L}_{CC-\text{ismask}} + 0.1 \cdot \mathcal{L}_{MLM} \quad (\text{S21})$$

We use a mixture density network head to encourage alignment between the learned last-layer pair representations \mathbf{G} and the intra-molecular 3D coordinate marginals. For a single training sample with 3D coordinate observation \mathbf{y} :

$$\mathcal{L}_{3D-\text{marginal}} = \sum_u^{N_{\text{frame}}} \sum_i^{N_{\text{atom}}} \log \left[\sum_l^{N_{\text{modes}}} \frac{\exp(w_{iul}) \cdot q_{3D}(T_u^{-1} \circ \mathbf{y}_i | \mathbf{m}_{iul})}{\sum_l^{N_{\text{modes}}} \exp(w_{iul})} \right] \quad (\text{S22})$$

where $T_u := (\mathbf{R}_u, \mathbf{t}_u)$, $T_u^{-1} \circ \mathbf{y}_i := (\mathbf{y}_i - \mathbf{t}_u) \cdot \mathbf{R}_u^\top$. $\mathbf{t}_u \in \mathbb{R}^3$ and $\mathbf{R}_u \in \text{SO}(3)$ are given by

$$(\mathbf{R}_u, \mathbf{t}_u) = \text{rigidFrom3Points}(\mathbf{y}_{i(u)}, \mathbf{y}_{j(u)}, \mathbf{y}_{k(u)}) \quad (\text{S23})$$

where rigidFrom3Points (Alg. S4) is the Gram–Schmidt-based frame construction operation originally described in Ref. [S6]; we additionally add a numerical stability factor of 0.01 \AA to the vector-norm calculations to handle edge cases when computing the rotation matrices from perturbed coordinates. Each component the 3D distance-angle distribution q^{3D} is parameterized by

$$q_{3D}(\mathbf{t} | \mu, \sigma, \mathbf{v}) = \text{Gaussian}(\|\mathbf{t}\|_2 | \mu, \sigma) \times \text{PowerSpherical}\left(\frac{\mathbf{t}}{\|\mathbf{t}\|_2} | \mathbf{v}, d=3\right) \quad (\text{S24})$$

where PowerSpherical is a power spherical distribution introduced in [S13]; $\mathbf{m}_{iul} := (\mu, \sigma, \mathbf{v})_{iul}$, and

$$[\mathbf{w}_{iu}, \mathbf{m}_{iu}] = \text{3DMixtureDensityHead}(\mathbf{G}_{l_{\max}})_{iu}. \quad (\text{S25})$$

where 3DMixtureDensityHead is a 3-layer MLP.

Using an equivariant graph transformer similar to ESDM (see Sec. F) but with all receptor nodes dropped, we construct a geometry prediction head to perform global molecular 3D structure denoising. We sample noised coordinates $\mathbf{y}(t)$ from

a VPSDE [S5] and introduce a SE(3)-invariant denoising score matching loss based on the Frame Aligned Point Error (FAPE) [S6]:

$$\mathcal{L}_{\text{3D-DSM}} = \mathbb{E}_{t \sim (0,1], \mathbf{y}_t \sim q_{0:t}(\cdot|\mathbf{y})} [\text{mean}_{u,i} \min(\|T_u^{-1} \circ \mathbf{y}_i - \hat{T}_u^{-1} \circ \hat{\mathbf{y}}_i\|_2, 10 \text{ \AA}) \cdot \sqrt{\alpha_t}] \quad (\text{S26})$$

where

$$\hat{\mathbf{y}} = \text{GeometryPredictionHead}(\mathbf{y}_t; \mathbf{H}_{l_{\max}}, \mathbf{F}_{l_{\max}}, \mathbf{G}_{l_{\max}}) \quad (\text{S27})$$

$\mathcal{L}_{\text{CC-regression}}$ is a SmoothL1 loss for fitting the "level 1" chemical checker (CC) [S12] embeddings which represents harmonized and integrated bioactivity data, and $\mathcal{L}_{\text{CC-ismask}}$ is an auxiliary binary cross entropy loss for classifying whether a specific CC entry is available for any molecule in the chemical checker dataset. \mathcal{L}_{MLM} is a standard cross-entropy loss for predicting the masked tokens. The MHT model is trained with a 50% masking ratio for all the atom, bond, edge, and stereochemistry encodings, and with dropout=0.1; we trained the model with batch size = 32 for 1.5×10^6 iterations using a cosine annealing schedule, taking 184 hours on a single NVIDIA-Tesla-V100-SXM2-32GB GPU.

D Protein-ligand graph featurization

Table S3. Notations for all node types in the protein-ligand graph (also see Figure 2e).

| Abbr. | Count | Meaning | Initial features |
|-------|----------------------|------------------------------|-------------------------------------------------------|
| B | N_{res} | Residue-wise backbone frames | PLM + template node features (see Sec. D.1) |
| S | N_P | Patch-wise anchor frames | subset of above (B) |
| F | N_L | Ligand anchor frames | MHT frame node embeddings $\mathbf{f}_{\text{frame}}$ |
| P | N_{protatm} | All protein atoms | MHT atomic node embeddings \mathbf{f}_{atom} |
| L | N_{ligatm} | All ligand atoms | MHT atomic node embeddings \mathbf{f}_{atom} |

Table S4. All edge types in the protein-ligand graph (also see Figure 2e). All edges comprised of two distinct node types are bidirectional; for conciseness, only one of the two directions is explicitly shown below. $\text{RBF}(\cdot)$ denotes a damped random Fourier basis function layer as defined in Eq. (S28). For the exponential-kNN scheme, see Algorithm S5.

| Abbr. | Edge type | src type | dst type | Connectivity | Initial features |
|---------------------------------------------------|------------|----------|----------|-----------------------------------------------------------------|---------------------------------------------------------------------------------------|
| # Residue-scale subgraph | | | | | |
| BB | local | B | B | Exponential-kNN($\mathbf{x}_{C\alpha}$), $k_{\text{NN}} = 32$ | See Section D.1 |
| BS | local | B | S | Intra-patch, dense | See Section D.1 |
| BF | long-range | B | F | Dense | Node embedding outer sum |
| SS | long-range | S | S | Dense | See Section D.1 |
| SF | long-range | S | F | Dense | Zero tensors |
| FF | long-range | F | F | Dense | Symmetrized MHT embeddings \mathbf{F}_L |
| # Atomic-scale subgraph | | | | | |
| As | local | P+L | P+L | Exponential-kNN(\mathbf{Z}), $k_{\text{NN}} = 8$ | $\text{RBF}_{32}(\ \mathbf{Z}_{\text{src}} - \mathbf{Z}_{\text{dst}}\ /10\text{\AA})$ |
| Ac | local | P+L | P+L | Molecular graph n-hop, n=3 | Gathered MHT embeddings \mathbf{f}_{aa} |
| Ab | local | P+L | P+L | Inter-molecular covalent bonds | Trainable random embedding |
| # Cross-scale edges connecting atoms and residues | | | | | |
| BP | local | B | P | Intra-residue, dense | Gathered MHT embeddings \mathbf{f}_{fa} |
| FL | local | F | L | Intra-ligand, dense | Gathered MHT embeddings \mathbf{f}_{fa} |

Given the primary model inputs and a noisy geometry, the schemes for constructing the residue-scale and atomic-scale graph representations are summarized in Table S3-S4. The protein anchor frame nodes (Table S3 S) are selected by first sequentially segmenting the input protein sequence into $N_P = 96$ patches of the same sequence length (with the last patch potentially truncated), and then sampling one unique backbone frame index for each protein patch. The intra-patch edges (Table S4 BS) then connect each protein residue node to the anchor node within the same patch.

The geometry-dependent local edges (Table S4 BB, As) are generated using a randomized k-nearest-neighbor (kNN) scheme with an exponentially-decaying attachment probability $p(\text{add_edge}(i, j)) \propto \exp(-\|\mathbf{Z}_i - \mathbf{Z}_j\|/5.0\text{\AA})$ with respect to the distance matrix among nodes, implemented via a Gumbel-Topk trick (note that a similar scheme is adopted in Ref. [S14]):

Algorithm S5 Exponential-kNN scheme for generating local edges

Require: All node Euclidean coordinates \mathbf{Z} , node degree k_{NN} , decay scale $D = 5.0\text{\AA}$

```

1: for  $i \in \{1, \dots, N_{\text{nodes}}\}$  do
2:   for  $j \in \{1, \dots, N_{\text{nodes}}\}$  do
3:      $s_{ij} \leftarrow -\|\mathbf{Z}_i - \mathbf{Z}_j\|/D$ 
4:      $u_{ij} \sim \text{Uniform}(0, 1)$ 
5:      $\tilde{s}_{ij} \leftarrow s_{ij} - \log(-\log(u_{ij}))$                                 Gumbel distribution over logits  $s_{ij}$ 
6:   end for
7:    $\{\text{src\_idx}\}_i, \{\text{dst\_idx}\}_i \leftarrow \text{ArgTopK}_j(\{\tilde{s}_{ij}\}_i), i$           Row-wise top-k node indices, k=kNN
8: end for
9: return  $\{\text{src\_idx}\}, \{\text{dst\_idx}\}$ 

```

$\text{RBF}(\cdot) : \mathbb{R} \rightarrow \mathbb{R}^{2 \times N_{\text{basis}}}$ denotes a damped random Fourier Feature encoding layer with sine and cosine frequencies $\kappa \in \mathbb{R}^{N_{\text{basis}}}$ initialized from a univariate Gaussian distribution:

$$\text{RBF}_{N_{\text{basis}}}(r) := [\sin(\kappa \cdot r), \cos(\kappa \cdot r)]^\top / (1 + r) \quad (\text{S28})$$

The inter-molecular covalent edges (Table S4 Ab, such as those in post-translational modifications and polysaccharides for which monomers are deposited as individual ligands) are determined based on the reference complex structure; an atom pair (i, j) is considered a covalent bond iff $d_{ij} < 1.2\sigma_{ij}$ where $\sigma_{ij} = \frac{1}{2}(\sigma_i + \sigma_j)$ is the average Van der Waals (VdW) radius.

D.1 Protein residue sub-graph featurization

The initial node features of all protein residue nodes are a concatenation of (a) the one-hot amino-acid types (20 standard residues + 1 "unknown" token) and (b) the concatenation of ESM-2-650M embeddings [S1] computed for all input protein sequences, and, when available, (c) internal coordinates of all atoms in the template protein structure $\mathbf{x}_{\text{template}}$ in the corresponding backbone coordinate frame, padded into the fixed-length atom37 format of PDB amino acid atom types. The initial node features are then embedded by a standard 3-layer MLP: $\mathbb{R}^{N_{\text{res}} \times (1280+21+37*3)} \rightarrow \mathbb{R}^{N_{\text{res}} \times c}$.

As detailed in Algorithm S6, the initial edge features of all protein residue-residue edges (Table S4 BB, BS, SS) are a combination of (a) an outer-sum of source and destination node features, (b) relative positional encodings of residue indices in the protein sequences, (c) relative geometrical encodings of residue backbones in the input noisy protein structure \mathbf{x}_t , and, when available, (d) relative geometrical encodings of residue backbones in the template protein structure $\mathbf{x}_{\text{template}}$.

Algorithm S6 Computing the initial feature for a single residue-residue edge

Require: src_idx, dst_idx , residue node features \mathbf{f}_B , input protein coordinates \mathbf{x}_t , input template coordinates $\mathbf{x}_{\text{template}}$ (optional)

- 1: $\mathbf{f}_{\text{osum}} = (\mathbf{f}_B)_{src_idx} + (\mathbf{f}_B)_{dst_idx}$ Eq. S28
 - 2: $\mathbf{f}_{\text{seq}} = \text{RBF}_{16}(residue_idx(src_idx) - residue_idx(dst_idx)) \cdot \text{IsSameChain}(src_idx, dst_idx)$
 - 3: $\mathbf{f}_{\text{geom}} = \text{RelGeomEnc}(src_idx, dst_idx, \mathbf{x}_t, N_{\text{basis}} = 15)$ Alg. S7
 - 4: $\mathbf{f}_{\text{edge}} \leftarrow \text{MLP}(\text{concat}(\mathbf{f}_{\text{osum}}, \mathbf{f}_{\text{seq}}, \mathbf{f}_{\text{geom}}))$
 - 5: **if** $\mathbf{x}_{\text{template}}$ is not None **then**
 - 6: $\mathbf{f}_{\text{edge}} \leftarrow \mathbf{f}_{\text{edge}} + \text{LinearNoBias}(\text{RelGeomEnc}(src_idx, dst_idx, \mathbf{x}_{\text{template}}, N_{\text{basis}} = 15))$ Alg. S7
 - 7: **end if**
 - 8: **return** \mathbf{f}_{edge}
-

Algorithm S7 Computing the relative geometrical encodings for a single residue-residue edge

```
def RelGeomEnc(src_idx, dst_idx, x, Nbasis, D0 = 10.0 Å)
1:  $\{\mathbf{x}_N, \mathbf{x}_{C\alpha}, \mathbf{x}_C\} \leftarrow$  Get protein backbone (N, C $\alpha$ , C) coordinates from  $\mathbf{x}$ 
2:  $\mathbf{t}_{\text{src}}, \mathbf{R}_{\text{src}} = \text{rigidFrom3Points}((\mathbf{x}_N)_{src\_idx}, (\mathbf{x}_{C\alpha})_{src\_idx}, (\mathbf{x}_C)_{src\_idx})$  Alg. S4
3:  $\mathbf{t}_{\text{dst}}, \mathbf{R}_{\text{dst}} = \text{rigidFrom3Points}((\mathbf{x}_N)_{dst\_idx}, (\mathbf{x}_{C\alpha})_{dst\_idx}, (\mathbf{x}_C)_{dst\_idx})$  Alg. S4
4:  $d = \|\mathbf{t}_{\text{src}} - \mathbf{t}_{\text{dst}}\|$ 
5:  $\mathbf{f}_{\text{dist}} = \text{RBF}_{N_{\text{basis}}}(d/D_0)$ 
6:  $\mathbf{f}_{\text{dirs}} = \mathbf{R}_{\text{src}}^\top \cdot (\mathbf{t}_{\text{dst}} - \mathbf{t}_{\text{src}})/(d + 1.0 \text{ Å})$ 
7:  $\mathbf{f}_{\text{dird}} = \mathbf{R}_{\text{dst}}^\top \cdot (\mathbf{t}_{\text{src}} - \mathbf{t}_{\text{dst}})/(d + 1.0 \text{ Å})$ 
8:  $\mathbf{f}_{\text{ori}} = \text{flatten}(\mathbf{R}_{\text{src}}^\top \cdot \mathbf{R}_{\text{dst}})$ 
9:  $\mathbf{f}_{\text{geom}} = \text{concat}(\mathbf{f}_{\text{dist}}, \mathbf{f}_{\text{dirs}}, \mathbf{f}_{\text{dird}}, \mathbf{f}_{\text{ori}})$   $\mathbf{f}_{\text{geom}} \in \mathbb{R}^{N_{\text{basis}} + 15}$ 
10: return  $\mathbf{f}_{\text{geom}}$ 
```

E The CPM network architecture

The neural network architecture of a single contact prediction module (CPM) block is summarized in Figure S1; for all models reported in this work, we use a stack of $N_{\text{CPM}} = 6$ blocks to construct the network (i.e., CPMForward in Algorithm S1).

Before the CPM forward pass, all frame node representations are concatenated with a 64-bit random Fourier encoding of the diffusion time $\text{RBF}(\tau)$ which is embedded by a standard MLP. The last step sampled block-adjacency matrix $\tilde{\mathbf{I}}$ is encoded by a LinearNoBias layer; the block-adjacency encodings are then added to the edge representations between patch-wise protein nodes and selected ligand nodes $\mathbf{f}_{\text{SF}} \in \mathbb{R}^{N_p * N_l \times c_p}$.

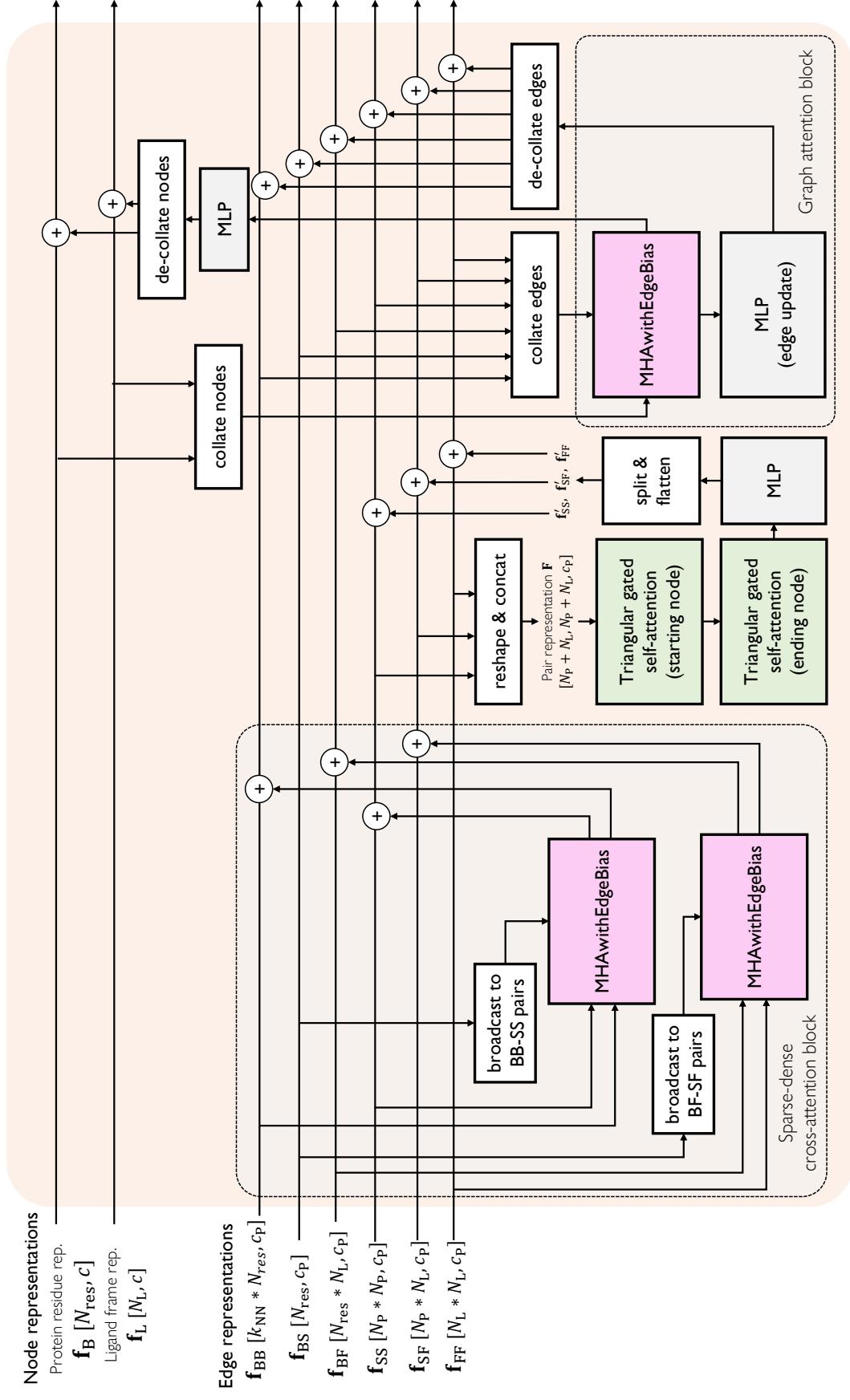


Figure S1. Network architecture of a single contact prediction module (CPM) block. Arrows indicate information flow directions, and "+" indicates an element-wise tensor summation. The Triangular gated self-attention (around starting/ending node) blocks refer to Algorithm 13-14 of Ref.[S6]. k_{NN} denotes the neighbor size of local residue-residue edges (see Section D), and we use $k_{NN} = 32$ for all models reported in this work. Note that $\mathbf{f}_{SS} \equiv \mathbf{F}_P$ and $\mathbf{f}_{FF} \equiv \mathbf{F}_L$ up to a tensor reshaping operation.

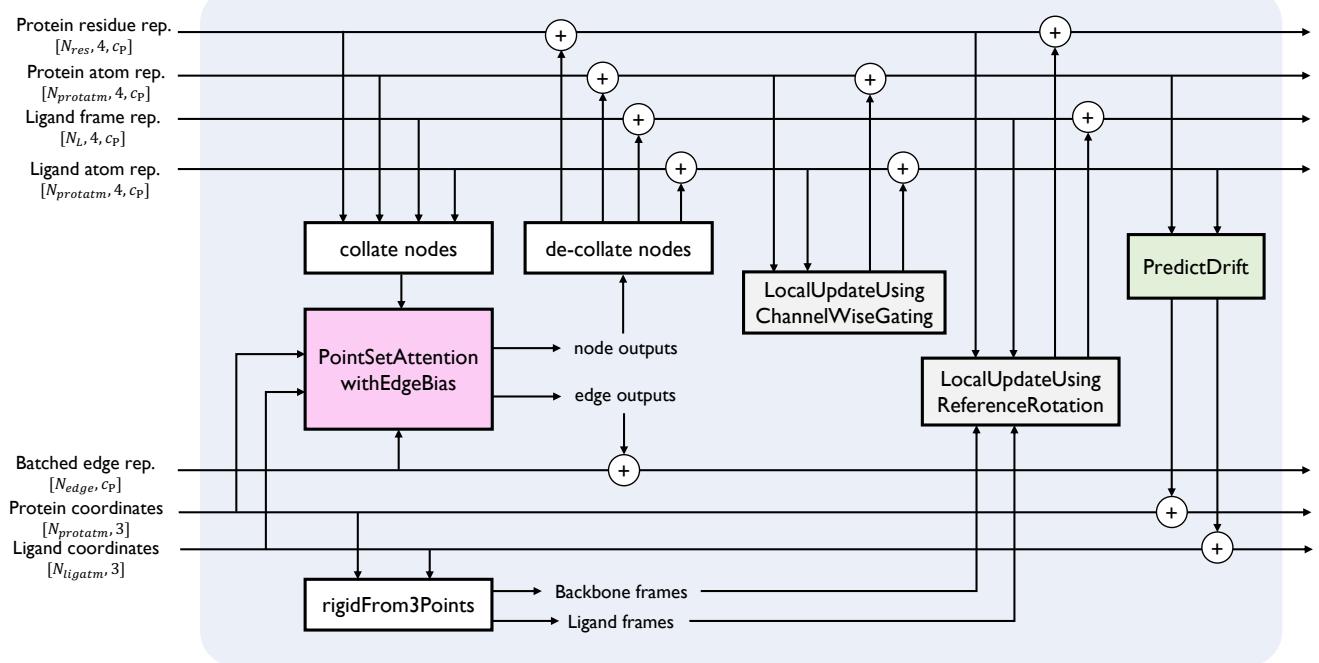


Figure S2. Network architecture of a single block in the equivariant structure diffusion module (ESDM). Arrows indicate information flow directions, and "+" indicates an element-wise tensor summation.

The first 2 CPM blocks are executed on the protein sub-graph only (i.e., BB, BS, and SS edges as defined in Table S4), and the remaining 4 CPM blocks are executed on the entire residue-scale graph (i.e., BB, BS, BF, SS, SF, and FF edges as defined in Table S4).

The asymptotic computational complexity of CPMForward is $\mathcal{O}(N_{CPM} \cdot (N_{res} \cdot (k_{NN} + N_L) + (N_P + N_L)^3))$.

F The ESDM network architecture

The Equivariant Structure Denoising Module (ESDM) of the NeuralPlexer network predicts denoised three-dimensional structures $\hat{\mathbf{Z}}_0$ using the noise input coordinates \mathbf{Z}_t and graph representations of the binding complex. The neural network architecture of a single ESDM block is summarized in Figure S2; for all models reported in this work, we use a stack of $N_{ESDM} = 4$ blocks to construct the network (i.e., ESDMForward in Algorithm S1).

Before the ESDM forward pass, all atomic node representations are concatenated with a 64-bit random Fourier encoding of the diffusion time $RBF(\tau)$ which is embedded by a standard MLP. The forward pass expressions of trainable modules PointSetAttentionwithEdgeBias, LocalUpdateUsingChannelWiseGating, LocalUpdateUsingReferenceRotation, PredictDrift are defined as:

Algorithm S8 PointSetAttentionwithEdgeBias

def PointSetAttentionwithEdgeBias($\mathbf{f}_s, \mathbf{f}_v, \mathbf{f}_e, \mathbf{t}, c = 64$)

$\mathbf{f}_s \in \mathbb{R}^{N_{nodes} \times c}, \mathbf{f}_v \in \mathbb{R}^{N_{nodes} \times 3 \times c}, \mathbf{f}_e \in \mathbb{R}^{N_{edges} \times c}, \mathbf{t} \in \mathbb{R}^{N_{nodes} \times 3}$

- 1: $\mathbf{f}_Q, \mathbf{f}_K, \mathbf{f}_V = \text{LinearNoBias}(\mathbf{f}_s)$
 - 2: $\mathbf{t}_Q, \mathbf{t}_K, \mathbf{t}_V = \mathbf{t}/D_{\text{points}} + \text{LinearNoBias}(\mathbf{f}_v)$
 - 3: $\mathbf{b} = \text{LinearNoBias}(\mathbf{f}_e)$
 - # Computing attention weights on all edges of the graph
 - 4: $\mathbf{z}_{ij} = \frac{1}{\sqrt{c_{\text{head}}}} (\mathbf{f}_{Q,i}^\top \cdot \mathbf{f}_{K,j}) + \mathbf{b}_{ij} - \frac{\mathbf{w}_{ij}}{\sqrt{18c_{\text{head}}}} \|\mathbf{t}_{Q,i} - \mathbf{t}_{K,j}\|_2^2$
 - 5: $\boldsymbol{\alpha}_{ij} = \text{Softmax}_{j \in \{i\}}(\mathbf{z}_{ij})$
 - 6: $\mathbf{f}'_s \leftarrow \sum_{j \in \{i\}} \boldsymbol{\alpha}_{ij} \odot \mathbf{f}_V$
 - 7: $\mathbf{f}'_v \leftarrow (\sum_{j \in \{i\}} \boldsymbol{\alpha}_{ij} \odot \mathbf{t}_V) - \mathbf{t}/D_{\text{points}}$
 - 8: $\mathbf{f}'_e \leftarrow \text{MLP}(\mathbf{z}_{ij}) + \mathbf{f}_e$
 - 9: **return** $\mathbf{f}'_s, \mathbf{f}'_v, \mathbf{f}'_e$
-

$\mathbf{f}_Q, \mathbf{f}_K, \mathbf{f}_V \in \mathbb{R}^{N_{nodes} \times n_{\text{heads}} \times c_{\text{head}}}$
 $\mathbf{t}_Q, \mathbf{t}_K, \mathbf{t}_V \in \mathbb{R}^{N_{nodes} \times n_{\text{heads}} \times c_{\text{point}} \times 3}, D_{\text{points}} = 10 \text{ \AA}$
 $\mathbf{b} \in \mathbb{R}^{N_{edges} \times c_{\text{heads}}}$

$\mathbf{z} \in \mathbb{R}^{N_{edges} \times n_{\text{heads}}}$

The expression for computing attention weights \mathbf{z} is adapted from Invariant Point Attention (IPA) [S6].

Algorithm S9 LocalUpdateUsingChannelWiseGating

def LocalUpdateUsingChannelWiseGating($\mathbf{f}_s, \mathbf{f}_v$)

$$\mathbf{f}_s \in \mathbb{R}^{N_{\text{nodes}} \times c}, \mathbf{f}_v \in \mathbb{R}^{N_{\text{nodes}} \times 3 \times c}$$

- 1: $\mathbf{f}_{\text{loc}} = \text{concat}(\mathbf{f}_s, \|\mathbf{f}_v\|_2)$ $\mathbf{f}_{\text{loc}} \in \mathbb{R}^{N_{\text{nodes}} \times 2c}$
 - 2: $\mathbf{f}'_s, \mathbf{f}_{\text{gate}} \leftarrow \text{MLP}(\mathbf{f}_{\text{loc}})$ $\mathbf{f}_{\text{gate}} \in \mathbb{R}^{N_{\text{nodes}} \times 1 \times c}$
 - 3: $\mathbf{f}'_v \leftarrow \text{LinearNoBias}(\mathbf{f}_v) \odot \mathbf{f}_{\text{gate}}$ Channel-wise product (broadcasting along xyz)
 - 4: **return** $\mathbf{f}'_s, \mathbf{f}'_v$
-

As only linear layers and vector scaling operations are used to update the vector representations \mathbf{f}_v , (S9) is E(3)-equivariant.

Algorithm S10 LocalUpdateUsingReferenceRotation

def LocalUpdateUsingReferenceRotation($\mathbf{f}_s, \mathbf{f}_v, \mathbf{R}$)

$$\mathbf{f}_s \in \mathbb{R}^{N_{\text{nodes}} \times c}, \mathbf{f}_v \in \mathbb{R}^{N_{\text{nodes}} \times 3 \times c}, \mathbf{R} \in \text{SO}(3)^{N_{\text{nodes}}}$$

- 1: $(\mathbf{f}_{\text{vloc}})_i = (\mathbf{R}_i)^T \cdot (\mathbf{f}_v)_i$ $i \in \{1, \dots, N_{\text{nodes}}\}$
 - 2: $\mathbf{f}_{\text{loc}} = \text{concat}(\mathbf{f}_s, \mathbf{f}_{\text{vloc}}, \|\mathbf{f}_v\|_2)$ $\mathbf{f}_{\text{loc}} \in \mathbb{R}^{N_{\text{nodes}} \times 5c}$
 - 3: $\mathbf{f}'_s, \mathbf{f}_{\text{vloc}} \leftarrow \text{MLP}(\mathbf{f}_{\text{loc}})$
 - 4: $(\mathbf{f}'_v)_i \leftarrow \mathbf{R}_i \cdot (\mathbf{f}_{\text{vloc}})_i$ $i \in \{1, \dots, N_{\text{nodes}}\}$
 - 5: **return** $\mathbf{f}'_s, \mathbf{f}'_v$
-

Since the third row of \mathbf{R} is a pseudovector as described in rigidFrom3Points, the determinant of the rotation matrix \mathbf{R} is unchanged under parity inversion transformations $i : \mathbf{x} \mapsto -\mathbf{x}$ and thus the intermediate quantity \mathbf{f}_{vloc} is SE(3)-invariant but in general **not invariant** under parity inversion i . This property ensures that ESDM-predicted coordinates can capture the correct chiral symmetry-breaking behaviors in molecular 3D conformation distributions.

Algorithm S11 PredictDrift

def PredictDrift($\mathbf{f}_s, \mathbf{f}_v$)

$$\mathbf{f}_s \in \mathbb{R}^{N_{\text{nodes}} \times c}, \mathbf{f}_v \in \mathbb{R}^{N_{\text{nodes}} \times 3 \times c}$$

- 1: $\mathbf{o}_{\text{scale}} \leftarrow \text{Softplus}(\text{MLP}(\mathbf{f}_s))$ $\mathbf{o}_{\text{scale}} \in \mathbb{R}^{N_{\text{nodes}} \times 1}$
 - 2: $\Delta \mathbf{t} \leftarrow \text{LinearNoBias}(\mathbf{f}_v) \odot \mathbf{o}_{\text{scale}}$ $\Delta \mathbf{t} \in \mathbb{R}^{N_{\text{nodes}} \times 3}$
 - 3: **return** $\Delta \mathbf{t}$
-

The predicted drift vectors $\Delta \mathbf{t}$ are added to the atomic coordinates from the last ESDM block; after N_{ESDM} blocks, the final coordinate outputs are taken as the predicted denoised structure $\hat{\mathbf{Z}}_0$ to infer the score function.

G Confidence estimation heads

The predicted IDDT (pLDDT) head of NeuralPLexer uses the same architecture as ESDM (Sec. F) with an independent set of parameters. Once a structure \mathbf{Z}_0 is generated from the main network, \mathbf{Z}_0 and an empty block contact map is passed to the pre-trained CPM network to generate the residue-scale graph embeddings to the pLDDT head (Algorithm S1, lines 41-44).

The output protein residue representations and ligand atom representations from the pLDDT head are passed to a 6-layer MLP to predict the histograms of distance error distributions between generated and reference structures, that is, for each query protein residue centroid or ligand atom i , we fit the histogram of distance deviations $\|(\mathbf{Z}_0)_i - (\mathbf{Z}_0)_j\| - \|(\mathbf{Z}_{\text{ref}})_i - (\mathbf{Z}_{\text{ref}})_j\|$ for all target point j within 15.0 Å of the query point in the reference structure \mathbf{Z}_{ref} . For such histograms, we use distance deviation bins with left boundaries of [0.0, 0.5, 1.0, 2.0, 4.0, 8.0, 16.0] Å. The pLDDT score of each protein residue or ligand atom is then computed based on the respective predicted histogram densities:

$$p\text{LDDT}[i] := 1.0 * p_{0-0.5}[i] + 0.75 * p_{0.5-1.0}[i] + 0.5 * p_{1.0-2.0}[i] + 0.25 * p_{2.0-4.0}[i]. \quad (\text{S29})$$

Table S5. Model training details. All training runs were performed on 6 NVIDIA-Tesla-V100-SXM2-32GB GPUs with a total batch size of 6, data parallelism, and automatic mixed precision (AMP) training. CA: Cosine Annealing. EMA: Exponential Moving Average.

| Task type | End-to-end structure prediction | | Rigid-backbone docking | Binding site recovery |
|--------------------------|-----------------------------------|---------------------------|-----------------------------|----------------------------|
| Model ID | A.0 | A.1 | B | C |
| Initial parameters | Random | A.0 | A.0 | A.1 |
| Training dataset | PL2019-74k (DNA/RNA unremoved) | PL2019-74k | PDBBind2020(<2019) | PDBBind2020(<2019) |
| Data sampling scheme | see text | see text | N/A | N/A |
| <i>is_rigid_receptor</i> | no | no | yes | yes (cropped binding site) |
| <i>train_plddt_head</i> | no | yes | yes | no |
| Freeze MHT parameters | no | yes | yes | yes |
| Initial learning rate | $3 \cdot 10^{-4}$ | $1 \cdot 10^{-4}$ | $2 \cdot 10^{-4}$ | $2 \cdot 10^{-4}$ |
| Learning rate schedule | CA, 4 restarts | CA | CA, 6 restarts | CA, 4 restarts |
| Dropout rate | 0.1 | 0.01 | 0.01 | 0.01 |
| Template masking rate | 0.2 | 0.2 | 0.2 | 0.2 |
| Use EMA | no | yes | yes | no |
| p_{contact} | 0.25 | 0.1 | 0.1 | 0.1 |
| λ_{FAPE} | 0.5 | 0.2 | 0.2 | 0.2 |
| λ_{dRMSD} | 0.5 | 1.0 | 1.0 | 1.0 |
| λ_{viol} | 10^{-3} | linear increase, 0 to 0.1 | linear increase, 0 to 0.1 | 0.1 |
| Num. epochs | $75 + 250 + 250 + 224$ | 332 | $36 + 4 + 8 + 16 + 32 + 60$ | $7 + 14 + 28 + 56$ |
| Training time | 10 days 20 hours | 5 days 8 hours | 7 days 16 hours | 3 days 12 hours |

H Model training details

In Algorithm S12 we summarize the general procedure for training the NeuralPLexer main network and confidence estimation heads. $\text{Bern}(p)$ denotes a Bernoulli distribution of a 0-1 boolean variable X with $Pr(X = 1) = p$. $\text{bucketize_onehot}(\cdot, \mathbf{v}_{\text{bins}})$ denotes a one-hot encoding operation using left bin boundaries defined in a vector \mathbf{v}_{bins} . $\mathcal{L}_{\text{CE}}(\cdot, \cdot)$ denotes a standard cross entropy loss between two discrete distributions: $\mathcal{L}_{\text{CE}}(\mathbf{p}, \mathbf{q}) := -\sum_{i=1}^{N_{\text{bins}}} \log p_i \cdot q_i$.

FAPE denotes the frame-aligned point error introduced in Ref. [S6] for which we used all $N_{\text{res}} + N_{\text{frame}}$ backbone and ligand frames for relative point coordinate alignment; FAPE_{scaled} is a modified variant of FAPE to encourage learning the overall molecular chirality, where we scaled all aligned point coordinates by its vector norm and removed the clamping term.

dRMSD $_{\mathcal{X}}$ denotes the distance-based RMSD introduced in Ref. [S16] computed for all atoms in a selected set \mathcal{X} ; we denote $\mathcal{X} = \text{global}$ for all atoms in the protein-ligand complex, $\mathcal{X} = \text{site}$ for all atoms within 8.0 Å of the binding ligands, $\mathcal{X} = \text{pli}$ for all pairs between protein residue centroids and ligand atoms, and $\mathcal{X} = \text{weighted}$ for all pairs for which the residue-residue distance deviation between the reference structure \mathbf{x}_{ref} and the template structure $\mathbf{x}_{\text{template}}$ is greater than 2.0 Å.

The distance-geometry loss $\mathcal{L}_{\text{distgeom}}$ is defined as

$$\mathcal{L}_{\text{distgeom}}(\hat{\mathbf{Z}}, \mathbf{Z}) = \text{mean}_A \sum_{i,j \in \{A\}} ||\|\mathbf{Z}_i - \mathbf{Z}_j\| - \|\hat{\mathbf{Z}}_i - \hat{\mathbf{Z}}_j\|| \cdot \mathbb{1}_{\|\mathbf{Z}_i - \mathbf{Z}_j\| < 2.8 \text{ Å}} \quad (\text{S30})$$

where $\mathbb{1}$ denotes a 0-1 indicator function, all coordinates are in the Angstrom unit, index A runs over all residues and ligand molecules in the structure, and $i, j \in \{A\}$ indicates all atom pairs in the residue or ligand A .

The steric clash loss $\mathcal{L}_{\text{clash}}$ is defined as

$$\mathcal{L}_{\text{clash}}(\hat{\mathbf{Z}}, \mathbf{Z}) = \sum_{i,j} \max(\sigma_{ij} - ||\hat{\mathbf{Z}}_i - \hat{\mathbf{Z}}_j||, 0) \cdot \mathbb{1}_{\|\mathbf{Z}_i - \mathbf{Z}_j\| > 1.2\sigma_{ij}}. \quad (\text{S31})$$

where $\sigma_{ij} = \frac{1}{2}(\sigma_i + \sigma_j)$ is the average VdW radius of atom i and atom j .

The NeuralPLexer main network (including the MHT (51.6M), CPM (8.3M), ESDM (5.0M), and projection layers) has 65.8×10^6 trainable parameters in total, while the pLDDT head has 5.0×10^6 parameters in addition.

In Table S5 we summarize the training procedure and hyperparameters for all models reported in this study. During the training of model A.0, we randomly subsampled 5% of the entire PL2019-74k dataset at each epoch using a relative weight of the inverse-square-root of the UniRef50 cluster index occurrence frequency of the training protein chain. During the training of model A.1, we subsampled 5% of the PL2019-74k dataset at each epoch with an additional relative sampling weight of (a) $(1 +$

Algorithm S12 NeuralPLexer Training

Require: $\{\mathbf{s}\}, \{G\}$, reference structure \mathbf{Z}_{ref} , is_rigid_receptor , train_plddt_head , loss weights $p_{\text{contact}}, \lambda_{\text{FAPE}}, \lambda_{\text{dRMSD}}, \lambda_{\text{viol}}$

```

1: for  $i \in \{1, \dots, N_{\text{training\_iter}}\}$  do
2:   if  $\text{is\_rigid\_receptor}$  then
3:      $\mathbf{x}_{\text{template}} \leftarrow \mathbf{x}_{\text{ref}}$ 
4:   else
5:      $\text{use\_template} \sim \text{Bern}(0.5)$ 
6:     if  $\text{use\_template}$  then
7:        $\mathbf{x}_{\text{template}} \leftarrow \text{Retrieve a random structure from all AF2 predictions and PL2019-74k structures of } \{\mathbf{s}\}$ 
8:     else
9:        $\mathbf{x}_{\text{template}} \leftarrow \text{None}$ 
10:    end if
11:   end if
12:   if not  $(\text{train\_plddt\_head} \text{ and } (i \mid 10))$  then
13:     Compute MHT embeddings for all input ligand molecular graphs in  $\{G\}$  and standard amino acids Algorithm S2
14:      $N_P \leftarrow \min(N_{\text{res}}, 96), N_L \sim \{\text{floorRound}(\sqrt{N_{\text{frames}}}), \dots, 33, 32\}$ 
15:      $\text{is\_prior\_training} \sim \text{Bern}(p_{\text{contact}})$ 
16:     if  $\text{is\_prior\_training}$  then
17:        $\tau \leftarrow 1.0, k \sim \{0, \dots, N_L - 1\}$ 
18:     else
19:        $\tau \sim \text{Uniform}(0, 1), k \leftarrow N_L$ 
20:     end if
21:      $t \leftarrow \text{DiffusionTimeSchedule}(\tau)$  Equation (8)
22:      $\mathbf{z}_t \leftarrow \text{Perturb the structure using the forward SDE transition kernel } q_{t:0}(\cdot | \mathbf{Z}_{\text{ref}})$  Equation (9)
23:      $\text{residue\_graph}, \text{atomic\_graph} \leftarrow \text{Generate the residue-scale and atomic-scale graph based on } \mathbf{Z}_t$  Section D
24:      $\mathbf{D}, \mathbf{L} \leftarrow \text{Compute ground-truth distance and contact map based on } \mathbf{Z}_{\text{ref}}$  Equation (3),  $\mathbf{D}, \mathbf{L} \in \mathbb{R}^{N_{\text{res}} \times N_L}$ 
25:      $\tilde{\mathbf{l}} \leftarrow \mathbf{0}$ 
26:     for  $r \in \{1, \dots, k\}$  do
27:        $\mathbf{L}_r \leftarrow \text{SumIntoPatches}(L) \odot [1 - \max_{\text{column-wise}}(\tilde{\mathbf{l}})]$   $\mathbf{L}_r \in \mathbb{R}^{N_P \times N_L}$ 
28:        $\mathbf{l}_r = \text{OneHot}(i_r, j_r); (i_r, j_r) \sim \text{Categorical}_{N_P \times N_L}(\mathbf{L}_r)$   $\mathbf{l}_r \in \{0, 1\}^{N_P \times N_L}$ 
29:        $\tilde{\mathbf{l}} \leftarrow \tilde{\mathbf{l}} + \mathbf{l}_r$   $\tilde{\mathbf{l}} \in \{0, 1\}^{N_P \times N_L}$ 
30:     end for
31:      $\text{residue\_graph} \leftarrow \text{CPMForward}(\text{residue\_graph}, \tilde{\mathbf{l}}, \tau)$  Section E
32:      $\hat{\mathbf{P}} = \text{LinearNoBias}(\text{MLP}(\text{GetEdgesBF}(\text{residue\_graph}_k)))$   $\hat{\mathbf{P}} \in \mathbb{R}^{N_{\text{res}} \times N_L \times N_{\text{bins}}}, N_{\text{bins}} = 32$ 
33:      $\text{graph\_rep} \leftarrow \text{Collate}(\text{residue\_graph}, \text{atomic\_graph}, \text{cross\_scale\_graph})$  Table S4
34:      $\hat{\mathbf{Z}}_0 \leftarrow \text{ESDMForward}(\mathbf{z}_t; \text{graph\_rep}, \tau)$  Section F
35:     # Distogram and contact prediction losses
36:      $\mathcal{L}_{\text{dgram}} = \text{mean}_{A,J} \mathcal{L}_{\text{CE}}(\hat{\mathbf{P}}_{AJ}, \text{bucketize\_onehot}(D_{AJ}, \mathbf{v}_{\text{bins}}))$ 
37:      $\mathcal{L}_{\text{cmap}} = \mathcal{L}_{\text{CE}}(\text{DistogramToContactMap}(\hat{\mathbf{P}}), \mathbf{L})$ 
38:     # Invariant denoising structure prediction losses
39:      $\mathcal{L}_{\text{FAPE}} = \text{FAPE}(\hat{\mathbf{Z}}_0, \mathbf{Z}_{\text{ref}}) + \text{FAPE}_{\text{scaled}}(\hat{\mathbf{Z}}_0, \mathbf{Z}_{\text{ref}})$ 
40:      $\mathcal{L}_{\text{dRMSD}} = \text{dRMSD}_{\text{global}}(\hat{\mathbf{Z}}_0, \mathbf{Z}_{\text{ref}}) + \text{dRMSD}_{\text{site}}(\hat{\mathbf{Z}}_0, \mathbf{Z}_{\text{ref}}) + \text{dRMSD}_{\text{pli}}(\hat{\mathbf{Z}}_0, \mathbf{Z}_{\text{ref}}) + \text{dRMSD}_{\text{weighted}}(\hat{\mathbf{x}}_0, \mathbf{x}_{\text{ref}}; \mathbf{x}_{\text{template}})$ 
41:      $\lambda(t) = \frac{\sqrt{\sigma(t)^2 + \sigma_{\text{data}}^2}}{\sigma(t) \cdot \sigma_{\text{data}}}, \sigma(t) = \sigma \sqrt{t}$  Loss weighting adapted from Ref. [S15],  $\sigma = 12.25 \text{ \AA}$ ,  $\sigma_{\text{data}} = 5.0 \text{ \AA}$ 
42:     # Distance-geometry and clash regularizers
43:      $\mathcal{L}_{\text{distgeom}}, \mathcal{L}_{\text{clash}} \leftarrow \text{Compute structure violation losses}$  See Equations (S30)-(S31)
44:      $\mathcal{L} \leftarrow 0.1 \cdot (\mathcal{L}_{\text{dgram}} + \mathcal{L}_{\text{cmap}}) + \lambda(t) \lambda_{\text{FAPE}} \cdot \mathcal{L}_{\text{FAPE}} + \lambda(t) \lambda_{\text{dRMSD}} \cdot \mathcal{L}_{\text{dRMSD}} + \lambda(t) \lambda_{\text{viol}} \cdot (\mathcal{L}_{\text{distgeom}} + 0.1 \cdot \mathcal{L}_{\text{clash}})$ 
45:      $\mathcal{L} \leftarrow \text{is\_prior\_training} \cdot (\mathcal{L}_{\text{dgram}} + \mathcal{L}_{\text{cmap}}) + (1 - 0.9 \cdot \text{is\_prior\_training}) \cdot \mathcal{L}$ 
46:   else
47:     # Confidence estimation losses
48:      $\mathbf{Z}_0, (p\text{LDD\_gram}, \_, \_) \leftarrow \text{NeuralPLexerInference}(\{\mathbf{s}\}, \{G\}, 1, N_{\text{steps}} = 10, \text{use\_template}, \text{True})$  Alg. S1
49:      $\text{LDD\_gram} \leftarrow \text{Compute reference distance deviation histograms between } \mathbf{Z}_0 \text{ and } \mathbf{Z}_{\text{ref}}$  See Section G text
50:      $\mathcal{L} \leftarrow \text{mean}_{A \in \{\mathbf{s}\}} \mathcal{L}_{\text{CE}}(p\text{LDD\_gram}_A, \text{LDD\_gram}_A) + \text{mean}_{J \in \{G\}} \mathcal{L}_{\text{CE}}(p\text{LDD\_gram}_J, \text{LDD\_gram}_J)$ 
51:   end if
52:   Computing gradients w.r.t.  $\mathcal{L}$  and update model parameters
53: end for

```

$0.5 * \exp(1.0 * \text{BackboneRMSD}(\mathbf{x}_{\text{ref}}, \mathbf{x}_{\text{template}}))$ for AF2 templates and (b) $(1 + 0.3 * \exp(1.1 * \text{BackboneRMSD}(\mathbf{x}_{\text{ref}}, \mathbf{x}_{\text{template}})))$ for experimental templates, multiplied to the sampling weights used for training model A.0. All PDBBind fine-tuning runs (models B and C) were executed with standard data shuffling and no subsampling.

I Computational details

For results reported in blind protein-ligand docking and binding site recovery tasks, we used a diffusion model sampling setting of $N_{\text{step}} = 25$ integrator steps; for all end-to-end structure prediction results, we used $N_{\text{step}} = 100$ integrator steps.

The AlphaFold2 structures used for binding site structure recovery (Figure 3) and for templates in end-to-end structure prediction (Figure 4-5) are predicted using ColabFold [S17] using default recycling and AMBER relaxation settings, and without templates in order to best reflect the prediction fidelity of AlphaFold2 on new targets. The input sequences for all protein chains are obtained from <https://www.ebi.ac.uk/pdbe/api/pdb/entry/molecules/> to avoid issues related to unresolved residues and to represent a realistic testing scenario where the protein backbone models are obtained from the full sequence.

In Figure 3b, EquiBind [S18] is launched with the default configuration file, and for each protein-ligand pair 16 ligand conformations are generated using different random RDKit [S19] input conformers.

RosettaLigand [S20] runs are launched with a configuration modified from the standard protocol. We set the receptor Calpha constraint parameter to 100.0 to enable a fully flexible receptor; the ligand coordinates are initialized using the aligned-ground-truth conformation as obtained by TM-Align [S21], with randomized torsion angles using the BCL [S22] library as described in the standard protocol. We set the docking box width to 4.0 Å and remove the ligand center perturbation step to ensure the ligand search space during the low-resolution docking stage is constrained to the binding site location. We set the number of docking cycles of the high-resolution docking stage to 64 to converge the receptor structure during backbone relaxation; for each protein-ligand pair in the test set, generating 32 ligand poses took on average 150 minutes on a single Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz CPU.

All protein structure alignments and TM-score calculations are performed using TMAlign [S21]. All reported TM-scores are normalized by the chain length of the reference PDB structure. The per-residue all-atom IDDT score is computed using OpenStructure [S23] with an inclusion radius of 10.0 Å; the IDDT-BS score is then computed by averaging the per-residue scores for binding site residues within 4.0 Å of the ligand. The symmetry-corrected heavy-atom RMSD for ligand structure comparison is computed using the obrms function in OpenBabel [S24]. A standard 6-12 Lennard-Jones energy functional form is used for computing the clash rate statistics; the L-J energy and VdW radius parameters are obtained from the UFF parameter file retrieved from <https://github.com/kbsezginel/lammps-data-file/blob/master/uff-parameters.csv>.

J Supplementary results

In Tables S6-S7 we summarize the performance statistics for end-to-end structure prediction on targets reported in this study. For each target, we generate six AF2 structure models with the following setup:

- For model 0, we use an MSA cluster size = 256, extra MSA size = 512, 3 recycling cycles, and the AF2 model version "AlphaFold-ptm:3". This set of results is also used to generate the visualizations in Figure 4a-c.
- For models 1-5, we use an MSA cluster size = 512, extra MSA size = 1024, 3 recycling cycles, and AF2 model versions "AlphaFold-ptm:1", "AlphaFold-ptm:2", "AlphaFold-ptm:3", "AlphaFold-ptm:4", "AlphaFold-ptm:5", respectively.

These AF2 structure models are used as the template inputs for NeuralPLexer to generate six independent sets of predictions; for each apo protein or protein-ligand pair, we sample 8 NeuralPLexer structures using independent random seeds. On both datasets, NeuralPLexer achieves the highest average prediction accuracy (as shown by the highest TM-score and lowest backbone RMSD).

We observe that NeuralPLexer achieves a further improvement against AF2 when averaged against the sampled structures of the highest TM-score for each target (Tables S6-S7, "per-target top-1"), suggesting that the multi-structure generative model formulation naturally enables better coverage of the experimental structure compared to heuristic approaches that are based on randomizing AF2.

Table S6. Model predictions on the PocketMiner dataset (33 holo structures + 29 apo structures).

| Method | Sampler | TM-score (all predictions) | Backbone RMSD (all predictions) | TM-score (per-target top-1) | Backbone RMSD (per-target top-1) | Ligand RMSD (all predictions) | Ligand RMSD (per-target top-1) |
|-----------------------------|-----------|-------------------------------|------------------------------------|--------------------------------|-------------------------------------|----------------------------------|-----------------------------------|
| AlphaFold2 | - | mean | 0.929 | 1.548 | 0.943 | 1.323 | - |
| | | std | 0.095 | 0.813 | 0.090 | 0.721 | - |
| | | 25% | 0.931 | 0.897 | 0.948 | 0.755 | - |
| | | 50% | 0.962 | 1.350 | 0.969 | 1.095 | - |
| | | 75% | 0.978 | 2.032 | 0.986 | 1.710 | - |
| NeuralPlexer (no ligand) | DDIM [S4] | mean | 0.895 | 2.082 | 0.938 | 1.501 | - |
| | | std | 0.092 | 0.778 | 0.079 | 0.567 | - |
| | | 25% | 0.877 | 1.480 | 0.934 | 1.062 | - |
| | | 50% | 0.920 | 1.950 | 0.959 | 1.470 | - |
| | | 75% | 0.951 | 2.620 | 0.973 | 1.812 | - |
| NeuralPlexer (ours) | LSA-SDE | mean | 0.925 | 1.623 | 0.946 | 1.305 | - |
| | | std | 0.091 | 0.784 | 0.087 | 0.647 | - |
| | | 25% | 0.910 | 0.970 | 0.952 | 0.788 | - |
| | | 50% | 0.958 | 1.470 | 0.970 | 1.195 | - |
| | | 75% | 0.973 | 2.172 | 0.982 | 1.625 | - |
| NeuralPlexer (ours) | DDIM [S4] | mean | 0.909 | 1.893 | 0.943 | 1.387 | 9.115 |
| | | std | 0.089 | 0.684 | 0.084 | 0.559 | 10.00 |
| | | 25% | 0.898 | 1.370 | 0.943 | 1.008 | 9.466 |
| | | 50% | 0.933 | 1.780 | 0.967 | 1.260 | 11.11 |
| | | 75% | 0.958 | 2.310 | 0.978 | 1.680 | 3.101 |
| | LSA-SDE | mean | 0.934 | 1.486 | 0.950 | 1.236 | 5.398 |
| | | std | 0.091 | 0.750 | 0.087 | 0.656 | 5.390 |
| | | 25% | 0.938 | 0.900 | 0.955 | 0.720 | 2.674 |
| | | 50% | 0.966 | 1.290 | 0.975 | 1.095 | 2.583 |
| | | 75% | 0.978 | 1.870 | 0.987 | 1.613 | 5.010 |
| | | mean | 0.934 | 1.486 | 0.950 | 1.236 | 9.388 |
| | | std | 0.091 | 0.750 | 0.087 | 0.656 | 10.30 |
| | | 25% | 0.938 | 0.900 | 0.955 | 0.720 | 9.808 |
| | | 50% | 0.966 | 1.290 | 0.975 | 1.095 | 9.812 |
| | | 75% | 0.978 | 1.870 | 0.987 | 1.613 | 10.75 |

Table S7. Model predictions on 118 recent targets with ligand-induced conformational changes.

| Method | Sampler | TM-score (all predictions) | Backbone RMSD (all predictions) | TM-score (per-target top-1) | Backbone RMSD (per-target top-1) | Ligand RMSD (all predictions) | Ligand RMSD (per-target top-1) |
|-----------------------------|-----------|-------------------------------|------------------------------------|--------------------------------|-------------------------------------|----------------------------------|-----------------------------------|
| AlphaFold2 | - | mean | 0.891 | 2.049 | 0.911 | 1.881 | - |
| | | std | 0.111 | 1.007 | 0.099 | 1.057 | - |
| | | 25% | 0.852 | 1.368 | 0.865 | 1.103 | - |
| | | 50% | 0.930 | 1.910 | 0.951 | 1.615 | - |
| | | 75% | 0.969 | 2.592 | 0.975 | 2.322 | - |
| NeuralPlexer (no ligand) | DDIM [S4] | mean | 0.844 | 2.695 | 0.908 | 1.995 | - |
| | | std | 0.112 | 0.924 | 0.092 | 0.878 | - |
| | | 25% | 0.802 | 2.010 | 0.898 | 1.390 | - |
| | | 50% | 0.877 | 2.570 | 0.942 | 1.755 | - |
| | | 75% | 0.920 | 3.270 | 0.961 | 2.370 | - |
| NeuralPlexer (ours) | LSA-SDE | mean | 0.877 | 2.261 | 0.916 | 1.847 | - |
| | | std | 0.110 | 1.018 | 0.093 | 1.005 | - |
| | | 25% | 0.833 | 1.580 | 0.891 | 1.218 | - |
| | | 50% | 0.916 | 2.070 | 0.949 | 1.585 | - |
| | | 75% | 0.950 | 2.790 | 0.971 | 2.173 | - |
| NeuralPlexer (ours) | DDIM [S4] | mean | 0.867 | 2.418 | 0.921 | 1.818 | 14.22 |
| | | std | 0.110 | 0.908 | 0.089 | 0.948 | 13.79 |
| | | 25% | 0.839 | 1.730 | 0.918 | 1.162 | 13.01 |
| | | 50% | 0.903 | 2.280 | 0.953 | 1.555 | 12.50 |
| | | 75% | 0.941 | 2.940 | 0.971 | 2.218 | 3.567 |
| | LSA-SDE | mean | 0.893 | 2.026 | 0.926 | 1.676 | 7.410 |
| | | std | 0.110 | 0.982 | 0.091 | 0.961 | 7.878 |
| | | 25% | 0.866 | 1.360 | 0.916 | 1.055 | 23.55 |
| | | 50% | 0.929 | 1.870 | 0.958 | 1.425 | 24.27 |
| | | 75% | 0.965 | 2.570 | 0.978 | 2.118 | 22.97 |

References

- [S1] Zeming Lin et al. “Evolutionary-scale prediction of atomic-level protein structure with a language model”. In: *Science* 379.6637 (2023), pp. 1123–1130.
- [S2] Dan Hendrycks and Kevin Gimpel. “Gaussian error linear units (gelus)”. In: *arXiv preprint arXiv:1606.08415* (2016).
- [S3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. “Layer normalization”. In: *arXiv preprint arXiv:1607.06450* (2016).
- [S4] Jiaming Song, Chenlin Meng, and Stefano Ermon. “Denoising diffusion implicit models”. In: *arXiv preprint arXiv:2010.02502* (2020).
- [S5] Yang Song et al. “Score-based generative modeling through stochastic differential equations”. In: *arXiv preprint arXiv:2011.13456* (2020).
- [S6] John Jumper et al. “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596.7873 (2021), pp. 583–589.
- [S7] Jianyi Yang, Ambrish Roy, and Yang Zhang. “BioLiP: a semi-manually curated database for biologically relevant ligand–protein interactions”. In: *Nucleic acids research* 41.D1 (2012), pp. D1096–D1103.
- [S8] Simon Axelrod and Rafael Gomez-Bombarelli. “GEOM, energy-annotated molecular conformations for property prediction and molecular generation”. In: *Scientific Data* 9.1 (2022), pp. 1–14.
- [S9] Viki Kumar Prasad, Alberto Otero-de-La-Roza, and Gino A DiLabio. “PEPCONF, a diverse data set of peptide conformational energies”. In: *Scientific data* 6.1 (2019), pp. 1–9.
- [S10] Maho Nakata and Tomomi Shimazaki. “PubChemQC project: a large-scale first-principles electronic structure database for data-driven chemistry”. In: *Journal of chemical information and modeling* 57.6 (2017), pp. 1300–1308.
- [S11] Weihua Hu et al. “OGB-LSC: A Large-Scale Challenge for Machine Learning on Graphs”. In: *arXiv preprint arXiv:2103.09430* (2021).
- [S12] Miquel Duran-Frigola et al. “Extending the small-molecule similarity principle to all levels of biology with the Chemical Checker”. In: *Nature Biotechnology* 38.9 (2020), pp. 1087–1096.
- [S13] Nicola De Cao and Wilker Aziz. “The power spherical distribution”. In: *arXiv preprint arXiv:2006.04437* (2020).
- [S14] John Ingraham et al. “Illuminating protein space with a programmable generative model”. In: *bioRxiv* (2022). DOI: [10.1101/2022.12.01.518682](https://doi.org/10.1101/2022.12.01.518682). eprint: [https://www.biorxiv.org/content/early/2022/12/02/2022.12.01.518682](https://www.biorxiv.org/content/early/2022/12/02/2022.12.01.518682.full.pdf). URL: <https://www.biorxiv.org/content/early/2022/12/02/2022.12.01.518682>.
- [S15] Tero Karras et al. “Elucidating the Design Space of Diffusion-Based Generative Models”. In: *arXiv preprint arXiv:2206.00364* (2022).
- [S16] Mohammed AlQuraishi. “End-to-end differentiable learning of protein structure”. In: *Cell systems* 8.4 (2019), pp. 292–301.
- [S17] Milot Mirdita et al. “ColabFold: making protein folding accessible to all”. In: *Nature Methods* (2022), pp. 1–4.
- [S18] Hannes Stärk et al. “Equibind: Geometric deep learning for drug binding structure prediction”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 20503–20521.
- [S19] Greg Landrum et al. “RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling”. In: *Greg Landrum* (2013).
- [S20] Ian W Davis and David Baker. “RosettaLigand docking with full ligand and receptor flexibility”. In: *Journal of molecular biology* 385.2 (2009), pp. 381–392.
- [S21] Yang Zhang and Jeffrey Skolnick. “TM-align: a protein structure alignment algorithm based on the TM-score”. In: *Nucleic acids research* 33.7 (2005), pp. 2302–2309.
- [S22] Benjamin Brown et al. “Introduction to the BioChemical Library (BCL): An application-based open-source toolkit for integrated cheminformatics and machine learning in computer-aided drug discovery”. In: *Frontiers in pharmacology* (2022), p. 341.
- [S23] Marco Biasini et al. “OpenStructure: an integrated software framework for computational structural biology”. In: *Acta Crystallographica Section D: Biological Crystallography* 69.5 (2013), pp. 701–709.
- [S24] Noel M O’Boyle et al. “Open Babel: An open chemical toolbox”. In: *Journal of cheminformatics* 3.1 (2011), pp. 1–14.