DYNAMIC-BACKBONE PROTEIN-LIGAND STRUCTURE PREDICTION WITH MULTISCALE GENERATIVE DIFFUSION MODELS

A PREPRINT

Zhuoran Qiao
Caltech
zqiao@caltech.edu

Weili Nie
NVIDIA
wnie@nvidia.com

Arash Vahdat
NVIDIA
avahdat@nvidia.com

Thomas Miller
Entos, Caltech
tom@entos.ai

Anima Anandkumar
Caltech, NVIDIA
anima@caltech.edu

October 3, 2022

ABSTRACT

Molecular complexes formed by proteins and small-molecule ligands are ubiquitous, and predicting their 3D structures can facilitate both biological discoveries and the design of novel enzymes or drug molecules. Here we propose NeuralPLexer, a deep generative model framework to rapidly predict protein-ligand complex structures and their fluctuations using protein backbone template and molecular graph inputs. NeuralPLexer jointly samples protein and small-molecule 3D coordinates at an atomistic resolution through a generative model that incorporates biophysical constraints and inferred proximity information into a time-truncated diffusion process. The reverse-time generative diffusion process is learned by a novel stereochemistry-aware equivariant graph transformer that enables efficient, concurrent gradient field prediction for all heavy atoms in the protein-ligand complex. NeuralPLexer outperforms existing physics-based and learning-based methods on benchmarking problems including fixed-backbone blind protein-ligand docking and ligand-coupled binding site repacking. Moreover, we identify preliminary evidence that NeuralPLexer enriches bound-state-like protein structures when applied to systems where protein folding landscapes are significantly altered by the presence of ligands. Our results reveal that a data-driven approach can capture the structural cooperativity among protein and small-molecule entities, showing promise for the computational identification of novel drug targets and the end-to-end differentiable design of functional small-molecules and ligand-binding proteins.

1 Introduction

Protein structures are dynamically modulated by their interactions with small-molecule ligands, triggering downstream responses that are crucial to the regulation of biological functions [1–3]. Proposing ligands that selectively target protein conformations has become an increasingly important strategy in small-molecule-based therapeutics [4–6]. However, computational prediction of protein-ligand structures that are coupled to receptor conformational responses is still hampered by the prohibitive cost of physically simulating slow protein state transitions [7, 8], as well as the static nature of existing protein folding prediction algorithms [9, 10]. While several schemes have been proposed to remedy these issues [11–20], such methods often require case-specific expert interventions and lack a unified framework to predict 3D structures in a systematic and cooperative fashion.

Here we propose NeuralPLexer, a Neural framework for Protein-Ligand complex structure prediction. NeuralPLexer leverages diffusion-based generative modeling [21, 22] to sample 3D structures from a learned statistical distribution. We demonstrate that the multi-scale inductive bias in biomolecular complexes can be feasibly integrated with diffusion models by designing a finite-time stochastic differential equation (SDE) with structured drift terms. Owing to this formulation, NeuralPLexer can generalize to ligand-unbound or predicted protein structure inputs once trained solely on experimental protein-ligand complex structures that are not paired to alternative protein conformations. When applied to blind protein-ligand docking, NeuralPLexer improves both the geometrical accuracy and structure quality compared to baseline methods; when applied to ligand binding site design, an inpainting version of NeuralPLexer can

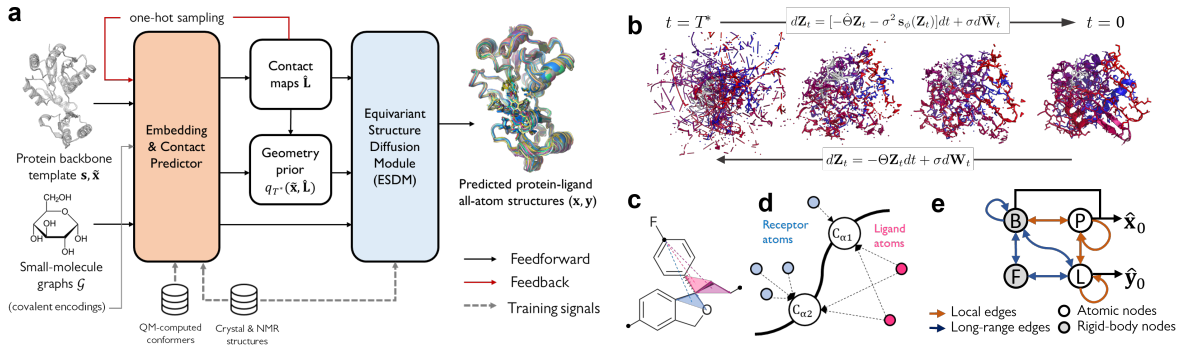

Figure 1: NeuralPLexer enables protein-ligand complex structure prediction with full receptor flexibility. (a) Method overview. (b) Sampling from NeuralPLexer. The protein (colored as red-blue from N- to C-terminus) and ligand (colored as grey) 3D structures are jointly generated from a learned SDE, with a partially-diffused initial state q_{T^*} approximated by the protein backbone template and predicted interface contact maps. (c-e) Key elements of the NeuralPLexer technical design. (c) Ligand molecules and monomeric entities are encoded as the collection of atoms, local coordinate frames (depicted as semi-transparent triangles), and stereospecific pairwise embeddings (depicted as dashed lines) representing their interactions. (d) The forward-time SDE introduces relative drift terms among protein Cα atoms, non-Cα atoms and ligand atoms, such that the SDE erases local-scale details at $t = T^*$ to enable resampling from a noise distribution. (e) Information flow in the equivariant structure diffusion module (ESDM). ESDM operates on a heterogeneous graph formed by protein atoms (P), ligand atoms (L), protein backbone frames (B) and ligand local frames (F) to predict clean atomic coordinates $\hat{\mathbf{x}}_0, \hat{\mathbf{y}}_0$ using the coordinates at a finite diffusion time $t > 0$.

accurately repack 44% of failed AlphaFold2 [9] binding sites with up to 60% success rate improvements compared to the method in Rosetta [23]. Furthermore, NeuralPLexer only requires molecular graphs as ligand inputs, therefore can enable end-to-end gradient-based design for functional small-molecules and ligand-binding proteins when coupled to recently-proposed differentiable protein sequence [24–26] and molecular graph generators [27, 28].

2 Method

We assume the model inputs are a receptor protein backbone template containing the amino acid sequence $\mathbf{s}$ and (N, Cα, C) atomic coordinates $\tilde{\mathbf{x}} \in \mathbb{R}^{n_{\text{res}} \times 3 \times 3}$, and a set of ligand molecular graphs $\{\mathcal{G}_k\}_{k=1}^K$ containing atom/bond types and stereochemistry labels (e.g., tetrahedral or E/Z isomerism [29]). We aim to sample $(\mathbf{x}, \mathbf{y}) \sim q_\phi(\cdot|\mathbf{s}, \tilde{\mathbf{x}}, \{\mathcal{G}\})$ from a generative model q_ϕ with predicted 3D heavy-atom coordinates of the protein $\mathbf{x} \in \mathbb{R}^{n \times 3}$ and that of the ligands $\mathbf{y} \in \mathbb{R}^{m \times 3}$. It can be understood as a conditional generative modeling problem for partially-observed systems.

NeuralPLexer adopts a two-stage architecture for protein-ligand structure prediction (Figure1a). The input protein backbone template and molecule graphs are first encoded and passed into a *contact predictor* that iteratively samples binding interface spatial proximity distributions for each ligand in $\{\mathcal{G}\}$; the output contact map parameterizes the *geometry prior*, a finite-time marginal of a designed SDE that progressively injects structured noise into the data distribution. An *equivariant structure diffusion module* (ESDM) then jointly generates 3D protein and ligand structures by denoising the atomic coordinates sampled from the geometry prior through a learned reverse-time SDE (Figure1b).

2.1 Protein-ligand structure generation with biophysics-informed diffusion processes

Diffusion models [22] introduce a forward SDE that diffuses data into a noised distribution and a neural-network-parameterized reverse-time SDE that generate data by reverting the noising process. To motivate the design principles for our biomolecular structure generator, we first consider a general class of linear SDEs known as the multivariate Ornstein–Uhlenbeck (OU) process [30] for point cloud $\mathbf{Z} \in \mathbb{R}^{N \times 3}$:

$$d\mathbf{Z}_t = -\Theta\mathbf{Z}_t dt + \sigma d\mathbf{W}_t \tag{1}$$

where $\Theta \in \mathbb{R}^{N \times N}$ is an invertible matrix of affine drift coefficients and $\mathbf{W}_t$ is a standard $3N$-dimensional Wiener process. The forward noising SDEs used in standard diffusion models [31, 32] can be recovered by setting $\Theta = \theta\mathbf{I}$, converging to an isotropic Gaussian prior distribution at the $t \to \infty$ (often expressed as $t \to 1$ with reparameterized t [33]) limit. In contrast, we design a multivariate SDE with data-dependent drift matrix $\Theta(\mathbf{Z}_0)$ and truncate the SDE at $t = T^* < \infty$ such that the final state of forward noising process is a partially-diffused, structured distribution q_{T*} that

can be well approximated by a coarse-scale model. We propose a set of SDEs depicted by Figure1d and detailed in Table A1, with separated lengthscale parameters σ_1, σ_2 such that the forward diffusion process erases residue-scale local details but retains global information about protein domain packing and ligand binding interfaces, yielding the following time-dependent transition kernels:

$$q_t\big(\mathbf{x}_{\mathrm{C}\alpha}(t)|\mathbf{x}(0),\mathbf{y}(0)\big) = \mathcal{N}\big(\mathbf{x}_{\mathrm{C}\alpha}(0); \sigma_1^2\tilde{\tau}\mathbf{I}\big) \tag{2}$$

$$q_t\big(\mathbf{x}_{\mathrm{nonC}\alpha}(t) - \mathbf{x}_{\mathrm{C}\alpha}(t)|\mathbf{x}(0),\mathbf{y}(0)\big) = \mathcal{N}\big(e^{-\tilde{\tau}}\big(\mathbf{x}_{\mathrm{nonC}\alpha}(0) - \mathbf{x}_{\mathrm{C}\alpha}(0)\big); 2\sigma_1^2(1-e^{-2\tilde{\tau}})\mathbf{I}\big) \tag{3}$$

$$q_t\big(\mathbf{y}(t) - \mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha}(t)|\mathbf{x}(0),\mathbf{y}(0)\big) = \mathcal{N}\big(e^{-\tilde{\tau}}\big(\mathbf{y}(0) - \mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha}(0)\big); \sigma_1^2(1-e^{-2\tilde{\tau}})(\mathbf{I} + \mathbf{c}^{\mathrm{T}}\mathbf{c})\big) \tag{4}$$

where we use an exponential schedule $\tilde{\tau} = (\sigma_{\min}^2/\sigma_1^2)e^t$ with truncation $T^* = 2\log(\sigma_2/\sigma_{\min})$. $\mathbf{c}$ is a softmax-transformed *contact map* as detailed in Sec. 2.2, which attracts the diffused ligand coordinates $\mathbf{y}(t)$ towards binding interface $\mathrm{C}\alpha$ atoms while preserving $\mathrm{SE}(3)$-equivariance. We choose $\sigma_1 = 2.0\,\text{Å}$ to match the average radius of standard amino acids with task-specific $\sigma_2 > \sigma_1$ such that at $t = T^*$: (a) the terms involving $\mathbf{x}_{\mathrm{nonC}\alpha}(0)$ and $\mathbf{y}(0)$ approximately vanishes thus are set to zeros to initialize the reverse-time SDE, and (b) the $\mathrm{C}\alpha$-atom coordinate marginal $q_{T^*}\big(\mathbf{x}_{\mathrm{C}\alpha}(t)|\mathbf{x}(0)\big)$ is sufficiently close to which approximated by the backbone template $q_{T^*}\big(\mathbf{x}_{\mathrm{C}\alpha}(t)|\tilde{\mathbf{x}}\big)$, guided by the theoretical result proposed in [34]. Proofs regarding $\mathrm{SE}(3)$-equivariance are stated in the Appendix A.1.2.

2.2 Contact map prediction and sampling from the truncated reverse-time SDE

Given protein-ligand coordinates $(\mathbf{x}, \mathbf{y})$, we define the contact map $\mathbf{L} \in \mathbb{R}^{n_{\mathrm{res}} \times m}$ with matrix elements $L_{Ai} = \log(\frac{\sum_{j\in\{A\}} e^{-2\alpha\|\mathbf{x}_j - \mathbf{y}_i\|^2}}{\sum_{j\in\{A\}} e^{-\alpha\|\mathbf{x}_j - \mathbf{y}_i\|^2}})$ where j runs over all protein atoms in amino acid residue A and $\alpha = 0.2\,\text{Å}^{-1}$. The term $\mathbf{c}$ in (4) is then defined as $c_{Ai}(\mathbf{L}) = \frac{\exp(L_{Ai})}{\sum_A \exp(L_{Ai})}$. To sample from the reverse-time SDE, we use the contact predictor to generate inferred contact maps $\hat{\mathbf{L}}$ and parameterize the geometry prior $q_{T*}(\cdot|\tilde{\mathbf{x}}, \hat{\mathbf{L}})$ - the initial condition of reverse-time SDE - by replacing $\mathbf{x}(0)$ in q_{T^*} with the backbone template $\tilde{\mathbf{x}}$ and the ligand-$\mathrm{C}\alpha$ relative drift coefficient $\mathbf{c}$ with the predicted $\mathbf{c}(\hat{\mathbf{L}})$. Note that in the general multivariate OU formulation, this corresponds to replacing the clean-data-dependent drift coefficients $\Theta(\mathbf{Z}_0)$ by a model estimation $\hat{\Theta}$. To account for the multimodal nature of protein-ligand contact distributions, the contact predictor models $\mathbf{L}$ as the logits of a categorical posterior distribution over a sequence of one-hot observations $\{\mathbf{l}\}_{k=1}^K$ sampled for individual molecules in $\{\mathcal{G}\}$. The forward pass of contact predictor ψ takes an iterative form:

$$\hat{\mathbf{L}}_k = \psi(\sum_{r=1}^{k} \mathbf{l}_r; \mathbf{s}, \tilde{\mathbf{x}}, \{\mathcal{G}\});\ \mathbf{l}_k = \mathrm{OneHot}(A_k, i_k);\ (A_k, i_k) \sim \mathrm{Categorical}_{n_{\mathrm{res}}\times m}(\hat{\mathbf{L}}_{k-1}),\ i_k \in \mathcal{G}_k \tag{5}$$

where $k \in \{1, \cdots, K\}$ and we set $\hat{\mathbf{L}} \coloneqq \hat{\mathbf{L}}_K$. All results reported in this study are obtained with $K = 1$ due to the curation scheme of standard annotated protein-ligand datasets, but we note that the model can be readily trained on more diverse structural databases with multi-ligand samples.

2.3 Architecture overview

Here we outline the key neural network design ideas and defer the featurization, architecture, and training details to the Appendix. To enable stereospecific molecular geometry generation and explicit reasoning about long-range geometrical correlations, NeuralPLexer hybridizes two types of elementary molecular representations (Figure1c): (a) atomic nodes and (b) rigid-body nodes representing coordinate frames formed by two adjacent chemical bonds. For small-molecule ligand encoding, we introduce a graph transformer with learnable chirality-aware pairwise embeddings that are constructed through graph-diffusion-kernel-like transformations [35]; such pairwise embeddings are pretrained to align with the intra-molecular 3D coordinate distributions from experimental and computed molecular conformers. The protein backbone template encoding module and the contact predictor are built upon a sparsified version of invariant point attention (IPA) adapted from AlphaFold2 [9] and are combined with standard graph attention layers [7, 36] and edge update blocks.

The architecture of ESDM (Figure1e) is inspired by prior works on 3D graph and attentional neural networks for point clouds [37, 38], rigid-body simulations [39] and biopolymer representation learning [9, 40–42]. In ESDM, each node is associated with a stack of standard scalar features $\mathbf{f}_{\mathrm{s}} \in \mathbb{R}^c$ and cartesian vector features $\mathbf{f}_{\mathrm{v}} \in \mathbb{R}^{3\times c}$ representing the displacements of a virtual point set relative to the node's Euclidean coordinate $\mathbf{t} \in \mathbb{R}^3$. A rotation matrix $\mathbf{R} \in \mathrm{SO}(3)$ is additionally attached to each rigid-body node. Geometry-aware messages are synchronously propagated among all nodes by encoding the pairwise distances among virtual point sets into graph transformer blocks.

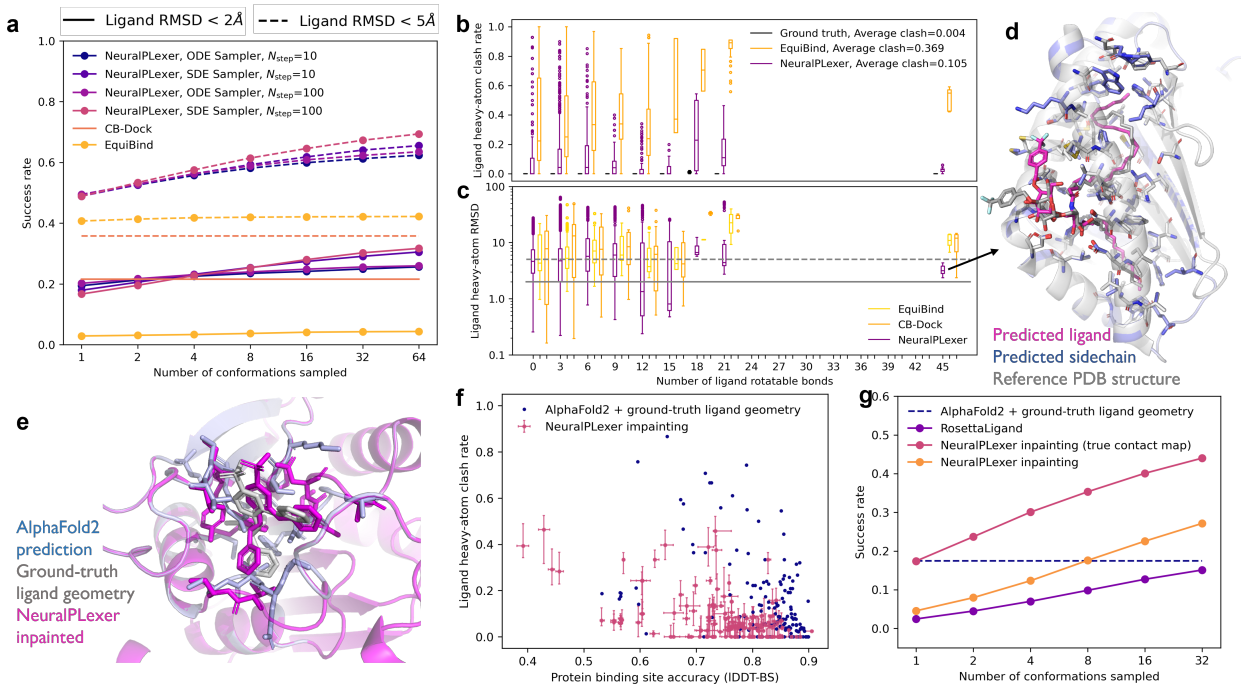

Figure 2: Model performance on benchmarking problems. (a-d) Fixed-backbone blind protein-ligand docking. (a) Success rates over the test dataset are plotted against the number of conformations sampled per protein-ligand pair; a success is defined as the ligand RMSD being lower than given threshold for at least one of the sampled conformations. Distributions of (b) the physical plausibility of sampled conformations as measured by the ligand heavy-atom steric clash rate with receptor atoms and (c) the geometrical accuracy as measured by the ligand RMSD are plotted against the number of ligand rotatable bonds, an indicatior of molecular flexibility. (d) Overlay of NeuralPLexer-predicted ligand and side-chain structures on the ground-truth for a challenging example (PDB: 6MJQ). (e-g) Ligand-coupled binding site repacking via diffusion-based inpainting. (e) A selected example (PDB:6TEL) where NeuralPLexer accurately inpaints the binding site protein-ligand structure, while directly aligning AlphaFold2 prediction to the ground-truth complex resulted in steric clashes between the ligand and binding site residues. (f) Summary of binding site accuracy (measured by the all-atom lDDT-BS score) and ligand clash rate over the test dataset. 32 conformations are sampled for each protein-ligand pair; dots indicates the median value and errorbars indicates 25% and 75% percentiles. (g) Success rates compared to baseline methods. A success is defined as: lDDT-BS > 0.7, ligand RMSD < 2.0 Å, and clash rate = 0.0. The pink "true contact map" curves are obtained by initializing the geometry prior q_{T^*} using the true protein-ligand contact map, while the gold curves are obtained by generating both protein and ligand conformations end-to-end.

Explicit non-linear transformation on vector features $\mathbf{f}_v$ is solely performed on rigid-body nodes through a coordinate-frame-inversion mechanism, 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 while the rigid-body frames $(\mathbf{t}, \mathbf{R})$ are passively reconstructed according to the updated atomic coordinates, circumventing numerical issues regarding fitting quaterion or axis-angle variables when manipulating rigid-body objects. The nontrivial actions of a parity inversion operation on rigid-body nodes ensure that ESDM can capture the correct chiral-symmetry-breaking behavior that adheres to the molecular stereochemistry constraints.

3 Results

Fixed-backbone protein-ligand docking. In this setting the ground-truth receptor protein backbone is given as input $\tilde{\mathbf{x}}$, and both ligand coordinates and protein sidechain coordinates are predicted using the input protein backbone and ligand graphs. Results are compared to a recent learning-based method EquiBind [43]; for reference, we also include results from a physics-based blind docking method CB-Dock [44] obtained with ground-truth all-atom receptor inputs and using a computing budget similar to learning-based methods. Models are trained and tested on the PDBBind-2020 [45] dataset split used in [43], with additional test dataset processing to ensure a reasonable comparison to docking-based methods (see Appendix A.8.1). As shown in Figure 2a-c, NeuralPLexer achieves both improved

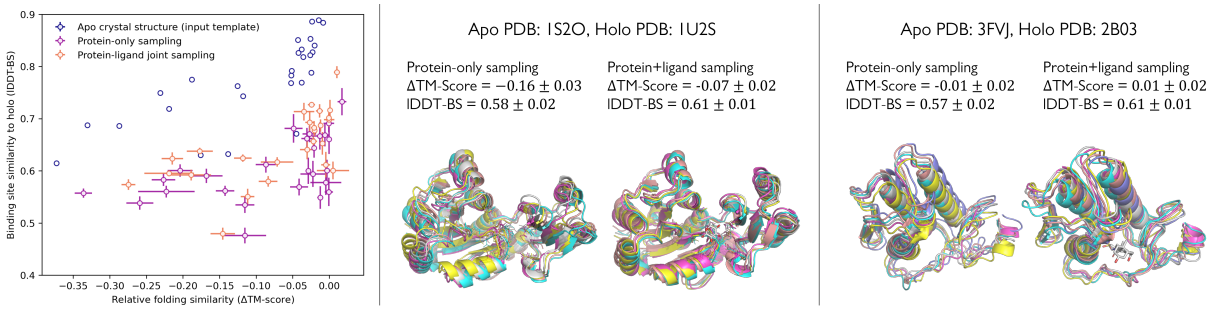

Figure 3: Assessments on systems with large binding-induced protein conformational transitions. Apo protein structures are used as the input backbone template. (a) Summary statistics of the relative protein folding similarity with respect to apo and holo PDB (measure by ΔTM-Score, the difference between TM-Scores computed against holo and apo structures) and binding site similarity with respect to holo (measured by lDDT-BS) for sampled structures. Purple dots are obtained with protein-only inputs and gold dots are obtained using protein+ligand inputs. Ligand-conditioning increases average ΔTM-Score from -9.0% to -7.7% (p=0.03), and average lDDT-BS from 0.59 to 0.63 (p<0.001). (b-c) Two examples for which neither their holo nor apo reference structures were observed during training. A marginal improvement in ΔTM-Score or lDDT-BS may indicate substantial protein conformational differences, while NeuralPLexer can qualitatively capture the correct protein state transitions.

geometrical accuracy (reported as the ligand heavy atom root-mean-squre-deviation (RMSD)) and lower steric clash rate (the fraction of ligand heavy atoms with a Lennard-Jones energy > 100 kcal/mol, using UFF [46] parameters). We found that good ligand structure quality and geometrical accuracy can be achieved using as few as 10 integrator steps (0.2 second per conformation on a single V100 GPU).

Ligand-coupled binding site repacking. Here we apply a diffusion-based inpainting strategy to jointly sample ligand and protein structure for a cropped region within 6.0 Å of the ligand conditioning on the uncropped parts of the protein. Protein binding site accuracy is measured by the lDDT-BS metric [47] with cutoff parameters consistent with CAMEO [48]. Input backbones are obtained using template-free AlphaFold2 (AF2) predictions of 154 selected chains whose TM-score [49]>0.8 and lDDT-BS<0.9 out of the abovementioned PDBBind test set, a subset representing cases where AF2 correctly predicts the global protein folding but unable to reproduce the exact bound-state binding site structure. We found 82% of structures contain steric clash with the ligand when directly aligned to reference complex structure in PDB, while NeuralPLexer is able to rescue 44% of these AF2 binding sites with joint protein-ligand inpainting (Figure 2e-g). Comparing to an energy-based flexible ligand-receptor modeling method RosettaLigand [23], NeuralPLexer increases success rate by up to 60% on the combined metric for ligand accuracy, binding site accuracy and physical plausibility.

Cryptic pockets and binding-induced protein conformation transitions. Lastly, we assessed NeuralPLexer-sampled structures for 31 systems from the PocketMiner dataset [50] which represents proteins with substantial ligand-binding-induced conformation changes. As a preliminary examination, we use the ligand-unbound (apo) crystal structure from PDB as the input backbone template and fix the ligand conformation to ground-truth coordinates along sampling. We found NeuralPLexer shifts the sampled ensemble toward bound-state (holo) structures when performing joint protein-ligand generation, compared to unconditioned protein-only sampling results (Figure 3a). Human evaluations reveal that NeuralPLexer correctly predicts biologically-relevant motions as illustrated by examples in Figure 3b-c, but a more systematic examination is currently hampered by the sensitivity of TM-Score and lDDT-BS to binding-irrelevant fluctuations. We note that native contact analysis algorithms [51] may provide improved metrics for interpreting protein generative models and consider that a future direction.

4 Discussion

We have presented a learning-based method for dynamic-backbone protein-ligand structure prediction, establishing an accuracy and sampling efficiency advantage relative to baseline approaches. We anticipate the incorporation of state-of-the-art protein representation learning techniques such as the use of sequence evolutionary signals, pretrained language models or higher-level attention mechanisms [9, 24, 25] and training on large-scale structure datasets to further improve the methodology and facilitate applications in various downstream molecular design problems.

Acknowledgements

Z.Q. acknowledges graduate research funding from Caltech and partial support from the Amazon–Caltech AI4Science fellowship. T.M. acknowledge partial support from the Caltech DeLogi fund, and A.A. acknowledges support from a Caltech Bren professorship.

References

[1] Katherine Henzler-Wildman and Dorothee Kern. "Dynamic personalities of proteins". In: *Nature* 450.7172 (2007), pp. 964–972.

[2] Nataliya Popovych et al. "Dynamically driven protein allostery". In: *Nature structural & molecular biology* 13.9 (2006), pp. 831–838.

[3] Ruth Nussinov and Chung-Jung Tsai. "Allostery in disease and in drug discovery". In: *Cell* 153.2 (2013), pp. 293–305.

[4] Alastair DG Lawson. "Antibody-enabled small-molecule drug discovery". In: *Nature Reviews Drug Discovery* 11.7 (2012), pp. 519–525.

[5] Amanda R Moore et al. "RAS-targeted therapies: is the undruggable drugged?" In: *Nature Reviews Drug Discovery* 19.8 (2020), pp. 533–552.

[6] Christopher J Draper-Joyce et al. "Positive allosteric mechanisms of adenosine A1 receptor-mediated analgesia". In: *Nature* 597.7877 (2021), pp. 571–576.

[7] David E Shaw et al. "Atomic-level characterization of the structural dynamics of proteins". In: *Science* 330.6002 (2010), pp. 341–346.

[8] Yibing Shan et al. "How does a small molecule bind at a cryptic binding site?" In: *PLoS computational biology* 18.3 (2022), e1009817.

[9] John Jumper et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (2021), pp. 583–589.

[10] 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.

[11] 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.

[12] Jue Wang et al. "Scaffolding protein functional sites using deep learning". In: *Science* 377.6604 (2022), pp. 387–394.

[13] 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.

[14] 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.

[15] Lim Heo and Michael Feig. "Multi-State Modeling of G-protein Coupled Receptors at Experimental Accuracy". In: *Proteins: Structure, Function, and Bioinformatics* (2022).

[16] Yuqi Zhang et al. "Benchmarking Refined and Unrefined AlphaFold2 Structures for Hit Discovery". In: (2022).

[17] Marta Amaral et al. "Protein conformational flexibility modulates kinetics and thermodynamics of drug binding". In: *Nature communications* 8.1 (2017), pp. 1–14.

[18] 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.

[19] Richard A Stein and Hassane S Mchaourab. "Modeling alternate conformations with alphafold2 via modification of the multiple sequence alignment". In: *bioRxiv* (2021).

[20] Lucas SP Rudden, Mahdi Hijazi, and Patrick Barth. "Deep learning approaches for conformational flexibility and switching properties in protein design". In: *Frontiers in Molecular Biosciences* (2022), p. 840.

[21] Jascha Sohl-Dickstein et al. "Deep unsupervised learning using nonequilibrium thermodynamics". In: *International Conference on Machine Learning*. PMLR. 2015, pp. 2256–2265.

[22] Yang Song et al. "Score-based generative modeling through stochastic differential equations". In: *arXiv preprint arXiv:2011.13456* (2020).

[23] 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.

[24] Tristan Bepler and Bonnie Berger. "Learning the protein language: Evolution, structure, and function". In: *Cell systems* 12.6 (2021), pp. 654–669.

[25] 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.

[26] 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).

[27] 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.

[28] Tianfan Fu et al. "Differentiable scaffolding tree for molecular optimization". In: *arXiv preprint arXiv:2109.10469* (2021).

[29] Ernest L Eliel and Samuel H Wilen. *Stereochemistry of organic compounds*. John Wiley & Sons, 1994.

[30] Attilio Meucci. "Review of statistical arbitrage, cointegration, and multivariate Ornstein-Uhlenbeck". In: *Cointegration, and Multivariate Ornstein-Uhlenbeck (May 14, 2009)* (2009).

[31] Yang Song and Stefano Ermon. "Generative modeling by estimating gradients of the data distribution". In: *Advances in Neural Information Processing Systems* 32 (2019).

[32] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6840–6851.

[33] Tero Karras et al. "Elucidating the Design Space of Diffusion-Based Generative Models". In: *arXiv preprint arXiv:2206.00364* (2022).

[34] Weili Nie et al. "Diffusion Models for Adversarial Purification". In: *arXiv preprint arXiv:2205.07460* (2022).

[35] 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.

[36] Petar Veličković et al. "Graph attention networks". In: *arXiv preprint arXiv:1710.10903* (2017).

[37] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. "E (n) equivariant graph neural networks". In: *International conference on machine learning*. PMLR. 2021, pp. 9323–9332.

[38] Johannes Brandstetter et al. "Geometric and physical quantities improve e (3) equivariant message passing". In: *arXiv preprint arXiv:2110.02905* (2021).

[39] Yunzhu Li et al. "Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids". In: *arXiv preprint arXiv:1810.01566* (2018).

[40] Bowen Jing et al. "Learning from protein structure with geometric vector perceptrons". In: *arXiv preprint arXiv:2009.01411* (2020).

[41] 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).

[42] Namrata Anand and Tudor Achim. "Protein Structure and Sequence Generation with Equivariant Denoising Diffusion Probabilistic Models". In: *arXiv preprint arXiv:2205.15019* (2022).

[43] Hannes Stärk et al. "Equibind: Geometric deep learning for drug binding structure prediction". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 20503–20521.

[44] Yang Liu et al. "CB-Dock: a web server for cavity detection-guided protein–ligand blind docking". In: *Acta Pharmacologica Sinica* 41.1 (2020), pp. 138–144.

[45] Renxiao Wang et al. "The PDBbind database: methodologies and updates". In: *Journal of medicinal chemistry* 48.12 (2005), pp. 4111–4119.

[46] 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.

[47] Valerio Mariani et al. "lDDT: a local superposition-free score for comparing protein structures and models using distance difference tests". In: *Bioinformatics* 29.21 (2013), pp. 2722–2728.

[48] 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.

[49] 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.

[50] Artur Meller et al. "Predicting the locations of cryptic pockets from single protein structures using the PocketMiner graph neural network". In: *bioRxiv* (2022).

[51] 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.

[52] Pat Vatiwutipong and Nattakorn Phewchean. "Alternative way to derive the distribution of the multivariate Ornstein–Uhlenbeck process". In: *Advances in Difference Equations* 2019.1 (2019), pp. 1–7.

[53] Jiaming Song, Chenlin Meng, and Stefano Ermon. "Denoising diffusion implicit models". In: *arXiv preprint arXiv:2010.02502* (2020).

[54] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. "Layer normalization". In: *arXiv preprint arXiv:1607.06450* (2016).

[55] 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.

[56] 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.

[57] Alexander G Donchev et al. "Quantum chemical benchmark databases of gold-standard dimer interaction energies". In: *Scientific data* 8.1 (2021), pp. 1–9.

[58] 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.

[59] 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.

[60] Weihua Hu et al. "OGB-LSC: A Large-Scale Challenge for Machine Learning on Graphs". In: *arXiv preprint arXiv:2103.09430* (2021).

[61] 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.

[62] Nicola De Cao and Wilker Aziz. "The power spherical distribution". In: *arXiv preprint arXiv:2006.04437* (2020).

[63] Helen M Berman et al. "The protein data bank". In: *Nucleic acids research* 28.1 (2000), pp. 235–242.

[64] Andrea Scarpino, György G Ferenczy, and György M Keserű. "Comparative evaluation of covalent docking tools". In: *Journal of Chemical Information and Modeling* 58.7 (2018), pp. 1441–1458.

[65] Gaoqi Weng et al. "Comprehensive evaluation of fourteen docking programs on protein–peptide complexes". In: *Journal of chemical theory and computation* 16.6 (2020), pp. 3959–3969.

[66] Milot Mirdita et al. "ColabFold: making protein folding accessible to all". In: *Nature Methods* (2022), pp. 1–4.

[67] Oleg Trott and Arthur J Olson. "AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading". In: *Journal of computational chemistry* 31.2 (2010), pp. 455–461.

[68] Greg Landrum et al. "RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling". In: *Greg Landrum* (2013).

[69] 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.

[70] 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.

[71] Noel M O'Boyle et al. "Open Babel: An open chemical toolbox". In: *Journal of cheminformatics* 3.1 (2011), pp. 1–14.

Table A1: Summary of the forward-time SDEs with a constant effective diffusion coefficient ($\sigma(\tau) = \sigma$).

Atom type	SDE Expression	Approximate marginal at $t = T^*$
Receptor Cα	$d\mathbf{x}_{\mathrm{C}\alpha} = \sigma d\mathbf{w}_1$	$q_{T^*}\big(\mathbf{x}_{\mathrm{C}\alpha}\vert\mathbf{x}(0),\mathbf{y}(0)\big) = \mathcal{N}(\mathbf{x}_{\mathrm{C}\alpha}(0); \sigma_2^2\mathbf{I})$
Receptor non-Cα	$d\mathbf{x}_{\mathrm{nonC}\alpha} = \theta(\mathbf{x}_{\mathrm{C}\alpha} - \mathbf{x}_{\mathrm{nonC}\alpha})d\tau + \sigma d\mathbf{w}_2$	$q_{T^*}\big(\mathbf{x}_{\mathrm{nonC}\alpha} - \mathbf{x}_{\mathrm{C}\alpha}\vert\mathbf{x}(0),\mathbf{y}(0)\big) = \mathcal{N}(\mathbf{0}; 2\sigma_1^2\mathbf{I})$
Ligand atoms	$d\mathbf{y} = \theta(\mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha} - \mathbf{y})d\tau + \sigma d\mathbf{w}_3$	$q_{T^*}\big(\mathbf{y} - \mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha}\vert\mathbf{x}(0),\mathbf{y}(0)\big) = \mathcal{N}(\mathbf{0}; \sigma_1^2(\mathbf{I} + \mathbf{c}^{\mathrm{T}}\mathbf{c}))$

A Appendix

A.1 The forward-time and reverse-time SDEs

The forward-time SDEs in NeuralPLexer are summarized in Table A1. For generality, we introduce an effective time stamp τ such that the drift and diffusion coefficients are constant $\theta(t) = \theta, \sigma(\tau) = \sigma$. The symbolic conventions are as following:

- $\mathbf{x}_{\mathrm{C}\alpha} \in \mathbb{R}^{n_{\mathrm{res}}\times 3}$ denotes the collection of alpha-carbon coordinates in the protein, following the standard nomenclature for amino acid atom types:
- $\mathbf{x}_{\mathrm{nonC}\alpha} \in \mathbb{R}^{(n-n_{\mathrm{res}})\times 3}$ denotes the set of coordinates for all non-alpha-carbon protein atoms (backbone N, C, O, and all side-chain heavy atoms);
- $\mathbf{y} \in \mathbb{R}^{m\times 3}$ denotes all ligand heavy atom coordinates. Note that $m \coloneqq \sum_{k=1}^{K} m_k$ with m_k being the number of heavy atoms in each ligand molecule $\mathcal{G}_k$.

A.1.1 Transition kernel densities and sampling

Following the general result for Ornstein–Uhlenbeck processes [52]

$$q_{0:t}(\mathbf{x}_t) = \mathcal{N}(\exp(-\Theta t)\mathbf{x}_0;\ \int_0^t e^{\Theta(s-t)}\boldsymbol{\sigma}\boldsymbol{\sigma}^{\mathrm{T}}e^{\Theta^{\mathrm{T}}(s-t)}ds) \tag{6}$$

given the effective time-homogeneous diffusion process described in Table A1, for internal coordinates $\mathbf{x}_{\mathrm{nonC}\alpha} - \mathbf{x}_{\mathrm{C}\alpha}$:

$$d(\mathbf{x}_{\mathrm{nonC}\alpha} - \mathbf{x}_{\mathrm{C}\alpha}) = -\theta(\mathbf{x}_{\mathrm{nonC}\alpha} - \mathbf{x}_{\mathrm{C}\alpha})d\tau + \sigma d\mathbf{w}_2 - \sigma d\mathbf{w}_1 \tag{7}$$

since the Brownian motions $\mathbf{w}_1, \mathbf{w}_2$ are independent, we obtain the transition kernel for the a finite time interval s:

$$\begin{aligned} &q(\mathbf{x}_{\mathrm{nonC}\alpha}(\tau+s) - \mathbf{x}_{\mathrm{C}\alpha}(\tau+s)\vert\mathbf{x}_{\mathrm{nonC}\alpha}(t) - \mathbf{x}_{\mathrm{C}\alpha}(\tau)) \\ &= \mathcal{N}\big(e^{-\theta s}(\mathbf{x}_{\mathrm{nonC}\alpha}(\tau) - \mathbf{x}_{\mathrm{C}\alpha}(\tau)); (1-e^{-2\theta s})\frac{\sigma^2}{\theta^2}\mathbf{I}\big) \end{aligned} \tag{8}$$

Similarly, for the ligand degrees of freedom

$$d(\mathbf{y} - \mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha}) = -\theta(\mathbf{y} - \mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha})dt + \sigma d\mathbf{w}_3 - \sigma\mathbf{c}^{\mathrm{T}}d\mathbf{w}_1 \tag{9}$$

the transition kernel is

$$\begin{aligned} &q(\mathbf{y}(\tau+s) - \mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha}(\tau+s)\vert\mathbf{y}(\tau) - \mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha}(\tau)) \\ &= \mathcal{N}\big(e^{-\theta s}(\mathbf{y}(\tau) - \mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha}(\tau)); (1-e^{-2\theta s})\frac{\sigma^2}{2\theta^2}(\mathbf{I} + \mathbf{c}^{\mathrm{T}}\mathbf{c})\big) \end{aligned} \tag{10}$$

The transition kernel for alpha-carbon atoms is a standard Gaussian

$$q(\mathbf{x}_{\mathrm{C}\alpha}(\tau+s)\vert\mathbf{x}_{\mathrm{C}\alpha}(\tau)) = \mathcal{N}\big(\mathbf{x}_{\mathrm{C}\alpha}(\tau); \sigma^2 s\mathbf{I}\big). \tag{11}$$

Defining $\sigma_1^2 = \frac{\sigma^2}{2\theta}$, $\sigma_2^2 = \sigma^2 \cdot \tau(T^*)$, and $\tilde{\tau} = 2\theta\tau$, we recover (2-4). For model training in practice, we use an exponential noise schedule defined by $\tau = \tau_0 e^t$ and $\tau_0 = \frac{\sigma_{\min}^2}{\sigma^2}$ with $\sigma_{\min}$ being a minimum perturbation scale as commonly adopted in variance-exploding (VE) [22] SDEs. For completeness, the SDEs defined in the transformed time horizon $t \in [0, T^*]$ is given by replacing the drift coefficient θ and the diffusion coefficient σ with the following time-dependent counterparts:

$$\theta(t) = \theta \cdot \frac{d\tau}{dt} = \frac{\sigma_{\min}^2}{2\sigma_1^2}e^t \tag{12}$$

and

$$\sigma(t) = \sqrt{\sigma^2 \cdot \frac{d\tau}{dt}} = \sigma_{\min} e^{\frac{1}{2}t}. \tag{13}$$

To sample from the marginal distribution $q_t \coloneqq p_{\text{data}} * q_{0:t}$ derived from the forward SDEs:

$$\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3 \sim \mathcal{N}(0; \mathbf{I}) \tag{14}$$

$$(\mathbf{x}, \mathbf{y}) \sim p_{\text{data}} \tag{15}$$

$$\mathbf{x}_{\text{C}\alpha}(t) = \mathbf{x}_{\text{C}\alpha} + \sigma\sqrt{\tau(t)}\mathbf{z}_1 \tag{16}$$

$$\mathbf{x}_{\text{nonC}\alpha}(t) = \mathbf{x}_{\text{C}\alpha}(t) + \sqrt{\alpha(t)}(\mathbf{x}_{\text{nonC}\alpha} - \mathbf{x}_{\text{C}\alpha}) + \sqrt{1-\alpha(t)}\sigma_1(\mathbf{z}_2 - \mathbf{z}_1) \tag{17}$$

$$\mathbf{y}(t) = \mathbf{c}^{\mathrm{T}}\mathbf{x}_{\text{C}\alpha}(t) + \sqrt{\alpha(t)}(\mathbf{y} - \mathbf{c}^{\mathrm{T}}\mathbf{x}_{\text{C}\alpha}) + \sqrt{1-\alpha(t)}\sigma_1(\mathbf{z}_3 - \mathbf{c}^{\mathrm{T}}\mathbf{z}_1) \tag{18}$$

where $\alpha(t) = e^{-2\theta\tau(t)}$.

For the reverse-time SDE

$$d\mathbf{Z}_t = [-\Theta(t)\mathbf{Z}_t - \sigma^2(t)\nabla_{\mathbf{Z}_t}\log q_t(\mathbf{Z}_t)]dt + \sigma(t)d\mathbf{W}_t \tag{19}$$

the ESDM ϕ predicts the denoised observations $\hat{\mathbf{x}}(0), \hat{\mathbf{y}}(0)$ using $\hat{\mathbf{x}}(t), \hat{\mathbf{y}}(t)$ which is formally equivalent to estimating the score function $\nabla_{\mathbf{Z}}\log q_t(\mathbf{Z})$ [53]. Given a time discretization schedule with interval s, we obtain the expression for the predicted observation mean $\bar{\mathbf{Z}}(\phi, t-s)$ in one denoising step $\mathbf{Z}(t) \mapsto \mathbf{Z}(t-s)$:

$$\bar{\mathbf{x}}_{\text{C}\alpha}(\phi, t-s) = -(\mathbf{x}_{\text{C}\alpha}(t) - \hat{\mathbf{x}}_{\text{C}\alpha}(0))\frac{\sigma(t-s)}{\sigma(t)} + \mathbf{x}_{\text{C}\alpha}(t) \tag{20}$$

$$\begin{aligned}\bar{\mathbf{x}}_{\text{nonC}\alpha}(\phi, t-s) = &-\frac{(\mathbf{x}_{\text{nonC}\alpha}(t) - \mathbf{x}_{\text{C}\alpha}(t))/\sqrt{\alpha(t)} - (\hat{\mathbf{x}}_{\text{nonC}\alpha}(0) - \hat{\mathbf{x}}_{\text{C}\alpha}(0))}{\sqrt{1-\alpha(t)}}\sqrt{1-\alpha(t-s)} \\ &+ \bar{\mathbf{x}}_{\text{C}\alpha}(t-s) + \sqrt{\alpha(t-s)}(\hat{\mathbf{x}}_{\text{nonC}\alpha}(0) - \hat{\mathbf{x}}_{\text{C}\alpha}(0))\end{aligned} \tag{21}$$

$$\begin{aligned}\bar{\mathbf{y}}(\phi, t-s) = &-\frac{(\mathbf{y}(t) - \mathbf{c}^{\mathrm{T}}\mathbf{x}_{\text{C}\alpha}(t))/\sqrt{\alpha(t)} - (\hat{\mathbf{y}}(0) - \mathbf{c}^{\mathrm{T}}\hat{\mathbf{x}}_{\text{C}\alpha}(0))}{\sqrt{1-\alpha(t)}}\sqrt{1-\alpha(t-s)} \\ &+ \mathbf{c}^{\mathrm{T}}\bar{\mathbf{x}}_{\text{C}\alpha}(t-s) + \sqrt{\alpha(t-s)}(\hat{\mathbf{y}}(0) - \mathbf{c}^{\mathrm{T}}\hat{\mathbf{x}}_{\text{C}\alpha}(0))\end{aligned} \tag{22}$$

standard ODE-based or SDE-based integrators can then be adapted to sample from (19).

A.1.2 Euclidean equivariance

Given group G, a function $f : X \to Y$ is said to be equivariant if for all $g \in G$ and $x \in X$, $f(\varphi_X(g)\cdot x) = \varphi_Y(g)\cdot f(x)$. Specifically f is said to be invariant if $f(\varphi_X(g)\cdot x) = f(x)$. We are interested in the special Euclidean group $G = \mathrm{SE}(3)$ consists of all global rigid translation and rotation operations $g \cdot \mathbf{Z} \coloneqq \mathbf{t} + \mathbf{Z}\cdot\mathbf{R}$ where $\mathbf{t} \in \mathbb{R}^3$ and $\mathbf{R} \in \mathrm{SO}(3)$. To adhere to the physical constraint that p_{data} is always $\mathrm{SE}(3)$-invariant, the transition kernels of forward-time SDE should satisfy $\mathrm{SE}(3)$-equivariance $q(\mathbf{Z}_{t+s}|\mathbf{Z}_t) = q(g\cdot\mathbf{Z}_{t+s}|g\cdot\mathbf{Z}_t)$ such that the marginals are invariant $q_t(\mathbf{Z}_t) = q_t(g\cdot\mathbf{Z}_t)$ for any time t. The proofs are straightforward:

For receptor $\mathrm{C}\alpha$ degrees of freedom

$$\begin{aligned}&q(\mathbf{t} + \mathbf{x}_{\text{C}\alpha}(\tau+s)\cdot\mathbf{R}|\mathbf{t} + \mathbf{x}_{\text{C}\alpha}(\tau)\cdot\mathbf{R}) \\ &= \mathcal{N}\big(\mathbf{t} + \mathbf{x}_{\text{C}\alpha}(\tau+s)\cdot\mathbf{R}; \mathbf{t} + \mathbf{x}_{\text{C}\alpha}(\tau)\cdot\mathbf{R}, \sigma^2 s\mathbf{I}\big) \\ &= \mathcal{N}\big((\mathbf{x}_{\text{C}\alpha}(\tau+s) - \mathbf{x}_{\text{C}\alpha}(\tau))\cdot\mathbf{R}\mathbf{R}^{\mathrm{T}}; 0, \sigma^2 s\mathbf{R}\cdot\mathbf{I}\cdot\mathbf{R}^{\mathrm{T}}\big) \\ &= \mathcal{N}\big((\mathbf{x}_{\text{C}\alpha}(\tau+s) - \mathbf{x}_{\text{C}\alpha}(\tau)); 0, \sigma^2 s\mathbf{I}\big) \\ &= q(\mathbf{x}_{\text{C}\alpha}(\tau+s)|\mathbf{x}_{\text{C}\alpha}(\tau)).\end{aligned}$$

For receptor non-$\mathrm{C}\alpha$ degrees of freedom

$$\begin{aligned}&q((\mathbf{t} + \mathbf{x}_{\text{nonC}\alpha}(\tau+s)\cdot\mathbf{R} - \mathbf{t} - \mathbf{x}_{\text{C}\alpha}(\tau+s)\cdot\mathbf{R})|(\mathbf{t} + \mathbf{x}_{\text{nonC}\alpha}(\tau)\cdot\mathbf{R} - \mathbf{t} - \mathbf{x}_{\text{C}\alpha}(\tau)\cdot\mathbf{R})) \\ &= \mathcal{N}\big((\mathbf{x}_{\text{nonC}\alpha}(\tau+s)\cdot\mathbf{R} - \mathbf{x}_{\text{C}\alpha}(\tau+s)\cdot\mathbf{R}); e^{-\theta s}(\mathbf{x}_{\text{nonC}\alpha}(\tau)\cdot\mathbf{R} - \mathbf{x}_{\text{C}\alpha}(\tau)\cdot\mathbf{R}), (1-e^{-2\theta s})\frac{\sigma^2}{\theta^2}\mathbf{I}\big) \\ &= \mathcal{N}\big((\mathbf{x}_{\text{nonC}\alpha}(\tau+s) - \mathbf{x}_{\text{C}\alpha}(\tau+s)); e^{-\theta s}(\mathbf{x}_{\text{nonC}\alpha}(\tau) - \mathbf{x}_{\text{C}\alpha}(\tau)), (1-e^{-2\theta s})\frac{\sigma^2}{\theta^2}\mathbf{R}\cdot\mathbf{I}\cdot\mathbf{R}^{\mathrm{T}}\big) \\ &= q((\mathbf{x}_{\text{nonC}\alpha}(\tau+s) - \mathbf{x}_{\text{C}\alpha}(\tau+s)|(\mathbf{x}_{\text{nonC}\alpha}(\tau) - \mathbf{x}_{\text{C}\alpha}(\tau))).\end{aligned}$$

For ligand degrees of freedom

$$
\begin{aligned}
&q(\mathbf{t}+\mathbf{y}(\tau+s)\cdot\mathbf{R}-\mathbf{c}^{\mathrm{T}}(\mathbf{t}+\mathbf{x}_{\mathrm{C}\alpha}(\tau+s)\cdot\mathbf{R})|\mathbf{t}+\mathbf{y}(\tau)\cdot\mathbf{R}-\mathbf{c}^{\mathrm{T}}(\mathbf{t}+\mathbf{x}_{\mathrm{C}\alpha}(\tau)\cdot\mathbf{R}))\\
&=q(\mathbf{t}+\mathbf{y}(\tau+s)\cdot\mathbf{R}-\mathbf{c}^{\mathrm{T}}\mathbf{t}-\mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha}(\tau+s)\cdot\mathbf{R}|\mathbf{t}+\mathbf{y}(\tau)\cdot\mathbf{R}-\mathbf{c}^{\mathrm{T}}\mathbf{t}-\mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha}(\tau)\cdot\mathbf{R})\\
&=q(\mathbf{y}(\tau+s)\cdot\mathbf{R}-\mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha}(\tau+s)\cdot\mathbf{R}|\mathbf{y}(\tau)\cdot\mathbf{R}-\mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha}(\tau)\cdot\mathbf{R})\\
&=\mathcal{N}\big(e^{-\theta s}(\mathbf{y}(\tau)-\mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha}(\tau));(1-e^{-2\theta s})\frac{\sigma^2}{2\theta^2}\mathbf{R}\cdot(\mathbf{I}+\mathbf{c}^{\mathrm{T}}\mathbf{c})\cdot\mathbf{R}^{\mathrm{T}}\big)\\
&=q(\mathbf{y}(\tau+s)-\mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha}(\tau+s)|\mathbf{y}(\tau)-\mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha}(\tau))
\end{aligned}
$$

where we have used $\mathbf{c}^{\mathrm{T}}\mathbf{t}=\mathbf{t}$ up to a column-wise broadcasting operation based on the row-wise normalization property of the softmax-transformed contact map $\mathbf{c}$.

Since all transition kernels are $\mathrm{SE}(3)$-equivariant, it then follows that the score $\nabla_{\mathbf{Z}}\log q_t(\mathbf{Z})$ is also $\mathrm{SE}(3)$-equivariant: $\nabla_{\mathbf{Z}'}\log q_t(\mathbf{Z}')=\nabla_{\mathbf{Z}}\log q_t(\mathbf{Z})\cdot\mathbf{R}$ where $\mathbf{Z}'=\mathbf{t}+\mathbf{Z}\cdot\mathbf{R}$ and thus the reverse-time SDE is equivariant. While the forward SDE is also $\mathrm{E}(3)$-equivariant as the noising process satisfies $q(-\mathbf{Z}(\tau+s)|-\mathbf{Z}(\tau))=q(\mathbf{Z}(\tau+s)|\mathbf{Z}(\tau))$, it is worth noting that the reverse SDE is only $\mathrm{SE}(3)$-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: $\exists\mathbf{Z}$ such that $\nabla_{-\mathbf{Z}}\log q_t(-\mathbf{Z})\neq-\nabla_{\mathbf{Z}}\log q_t(\mathbf{Z})$.

A.2 Small-molecule featurization and encoding

We consider two types of nodes to construct a graph-based molecular representation: (a) heavy-atoms $i\in\{1,2,\cdots,N_{\mathrm{atom}}\}$ and (b) local coordinate frames $u\in\{1,2,\cdots,N_{\mathrm{frame}}\}, u\coloneqq u(ijk)$ formed by atom triplets (i,j,k) that are connected by bonds (ij) and (jk). We introduce Path-integral Graph Transformer (PiFormer), an attentional neural network with edge-level operations inspired by the path-integral formulation of quantum mechanics, to infer the long-range inter-atomic geometrical correlations for small molecules based on their graph-topological properties. PiFormer operates on the collection of following classes of embeddings:

- Atom representations $\mathbf{H}\in\mathbb{R}^{N_{\mathrm{atom}}\times c}$. The input atom representations 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 $\mathbb{R}^{18+7}\to\mathbb{R}^{c}$;
- Frame representations $\mathbf{F}\in\mathbb{R}^{N_{\mathrm{frame}}\times c}$. For a given frame u, $\mathbf{F}_u$ is initialized by a 2-layer MLP $\mathbb{R}^{4*2+18+7}\to\mathbb{R}^{c}$ that embed the bond type encodings (defined as $[\mathrm{is\_single},\mathrm{is\_double},\mathrm{is\_triple},\mathrm{is\_aromatic}]$) 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 encodings $\mathbf{S}\in\mathbb{R}^{N_{\mathrm{frame}}\times N_{\mathrm{frame}}\times c_{\mathrm{s}}}$. $\mathbf{S}$ is a sparse tensor where an element $\mathbf{S}_{uv}$ is nonzero only if the pair of frames (u,v) is adjacent, i.e., u and v sharing a common incoming or outgoing bond;
- Pair representations $\mathbf{G}\in\mathbb{R}^{N_{\mathrm{frame}}\times N_{\mathrm{atom}}\times c_{\mathrm{p}}}$. $\mathbf{G}$ is initialized by an outer sum of $\mathbf{H}$ and $\mathbf{F}$ which is added to linear-projected $\mathbf{S}$ and passed to a 2-layer MLP.

Elements of the stereochemistry encoding tensor $\mathbf{S}$ are defined as

$$
\begin{aligned}
\mathbf{S}_{uv,0}&\coloneqq(\mathrm{common\_bond(u,v)}=\mathrm{incoming\_bond(u)})\\
\mathbf{S}_{uv,1}&\coloneqq(\mathrm{common\_bond(u,v)}=\mathrm{incoming\_bond(v)})\\
\mathbf{S}_{uv,2}&\coloneqq(\mathrm{common\_bond(u,v)}=\mathrm{outgoing\_bond(u)})\\
\mathbf{S}_{uv,3}&\coloneqq(\mathrm{common\_bond(u,v)}=\mathrm{outgoing\_bond(v)})\\
\mathbf{S}_{uv,4}&\coloneqq\mathrm{i(v)}\in\{\mathrm{i(u)},\mathrm{j(u)},\mathrm{k(u)}\}\\
\mathbf{S}_{uv,5}&\coloneqq\mathrm{j(v)}\in\{\mathrm{i(u)},\mathrm{j(u)},\mathrm{k(u)}\}\\
\mathbf{S}_{uv,6}&\coloneqq\mathrm{k(v)}\in\{\mathrm{i(u)},\mathrm{j(u)},\mathrm{k(u)}\}\\
\mathbf{S}_{uv,7}&\coloneqq(\mathrm{j(u)}=\mathrm{j(v)})\wedge\mathrm{is\_above\_plane(u,v)}\\
\mathbf{S}_{uv,8}&\coloneqq(\mathrm{j(u)}=\mathrm{j(v)})\wedge\mathrm{is\_below\_plane(u,v)}\\
\mathbf{S}_{uv,9}&\coloneqq\mathrm{is\_double\_or\_aromatic(common\_bond(u,v))}\vee\mathrm{is\_same\_side(u,v)}\\
\mathbf{S}_{uv,10}&\coloneqq\mathrm{is\_double\_or\_aromatic(common\_bond(u,v))}\vee\mathrm{not\_same\_side(u,v)}
\end{aligned}
$$

Table A2: Composition of the dataset used for pretraining the small-molecule encoder.

Data source	Num. samples collected	Sampling weight	$\mathcal{L}_{\mathrm{3D}}$	$\mathcal{L}_{\mathrm{CC}}$	$\mathcal{L}_{\mathrm{MLM}}$
BioLip [55] ligands (deposited date<2019.1.1)	160k	2.0	+	-	+
GEOM [56]	450k * 5	0.4	+	-	+
DES370k [57]	370k	1.0	+	-	+
PEPCONF [58]	3775	5.0	+	-	+
PCQM4Mv2 [59, 60]	3.4M	0.1	+	-	+
Chemical Checker [61]	800k	1.0	-	+	+

$\mathrm{is\_above\_plane}(u, v)$ is defined as one of the three atoms in frame v is above the plane formed by frame u with normal vector $\mathbf{v}_u = \frac{(\mathbf{r}_{j(u)}-\mathbf{r}_{i(u)})\times(\mathbf{r}_{k(u)}-\mathbf{r}_{j(u)})}{\|\mathbf{r}_{j(u)}-\mathbf{r}_{i(u)}\|\|\mathbf{r}_{k(u)}-\mathbf{r}_{j(u)}\|}$; $\mathrm{is\_same\_side}(u, v)$ is defined as the two bonds not shared between u, v being on the same side of the common bond, equivalent to $\mathbf{v}_u \cdot \mathbf{v}_v > 0$, vice versa. Our current technical implementations for $\mathrm{is\_above\_plane}$ and $\mathrm{is\_same\_side}$ 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 explicit coordinate generations. We additionally denote $\mathrm{MASK_s}$ as a $N_{\mathrm{frame}} \times N_{\mathrm{frame}}$ logical matrix defined as the adjacency matrix of frame pairs (u, v).

The notion of "frames" in a coordinate-free topological molecular graph is justified by the inductive bias 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 forms a consistent molecular representation without prior knowledge on 3D coordinates. PiFormer operates solely on the molecular representation defined by the input graph, and the frame coordinates $(\mathbf{t}, \mathbf{R})$ are initialized right before the ESDM blocks.

The forward pass of single PiFormer block is expressed as:

$$\mathbf{U}_l = \mathrm{Softmax}_{\mathrm{row-wise}}\big(\frac{(\mathbf{F}\cdot\mathbf{W}_{\mathrm{K},l})\cdot(\mathbf{F}\cdot\mathbf{W}_{\mathrm{Q},l})^{\mathrm{T}} + \mathbf{S}\cdot\mathbf{W}_{\mathrm{S},l}}{\sqrt{c_{\mathrm{P}}}} + \mathrm{Inf}\cdot\mathrm{MASK_s}\big) \tag{23}$$

$$\mathbf{G}_{\mathrm{out}} = (\mathbf{1} + \frac{1}{K}\mathbf{U}_l)^K \cdot (\mathbf{G}_l \cdot \mathbf{W}_{\mathrm{G},l}), \quad \mathbf{G}_{l+1} = \mathrm{MLP}([\mathbf{G}_{\mathrm{out}}||(\mathbf{F}_l)^{\mathrm{T}}\cdot\mathbf{H}_l||\mathbf{G}_l]) + \mathbf{G}_l \tag{24}$$

$$\mathbf{F}_{\mathrm{out}} = \mathrm{MHAwithEdgeBias}(\mathbf{F}_l, \mathbf{H}_l, (\mathbf{G}_{l+1})^{\mathrm{T}}), \quad \mathbf{F}_{l+1} = \mathrm{MLP}(\mathbf{F}_{\mathrm{out}} + \mathbf{F}_l) + \mathbf{F}_l \tag{25}$$

$$\mathbf{H}_{\mathrm{out}} = \mathrm{MHAwithEdgeBias}(\mathbf{H}_l, \mathbf{F}_{l+1}, \mathbf{G}_{l+1}), \quad \mathbf{H}_{l+1} = \mathrm{MLP}(\mathbf{H}_{\mathrm{out}} + \mathbf{H}_l) + \mathbf{H}_l \tag{26}$$

where K denotes the propagation length truncation for the learnable graph kernel $\exp(\mathbf{U}_l) \approx (\mathbf{1} + \frac{1}{K}\mathbf{U}_l)^K$ in a single PiFormer block, MLP denotes a 3-layer multilayer perceptron combined with layer normalization [54]. $\mathbf{W}_{\mathrm{K}}, \mathbf{W}_{\mathrm{Q}}, \mathbf{W}_{\mathrm{S}}, \mathbf{W}_{\mathrm{G}}$ are trainable linear weight matrices. $\mathrm{MHAwithEdgeBias}(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_{\mathrm{edge}})$ denotes a multi-head cross-attention layer between source node embeddings $\mathbf{X}_1$ and target node embeddings $\mathbf{X}_1$, with edge embeddings $\mathbf{X}_{\mathrm{edge}}$ entering attention computation as a relative positional encoding term as in the relation-aware transformer introduced in [7]. For all models descibed in this study, we set $l_{\max} = 6$ and $K = 8$.

A.2.1 PiFormer model pretraining

In Table A2 we summarize the small-molecule datasets used for training the PiFormer encoder used in the reported NeuralPLexer model. The loss function used in PiFormer pretraining is the following:

$$\mathcal{L}_{\mathrm{lig-pretraining}} = \mathcal{L}_{\mathrm{3D-marginal}} + \mathcal{L}_{\mathrm{3D-DSM}} + \mathcal{L}_{\mathrm{CC-regression}} + 0.01\cdot\mathcal{L}_{\mathrm{CC-ismask}} + 0.1\cdot\mathcal{L}_{\mathrm{MLM}} \tag{27}$$

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 $\mathbf{y}$:

$$\mathcal{L}_{\mathrm{3D-marginal}} = \sum_{u}^{N_{\mathrm{frame}}}\sum_{i}^{N_{\mathrm{atom}}} \log\Big[\sum_{l}^{N_{\mathrm{modes}}} \frac{\exp(w_{iul})\cdot q_{\mathrm{3D}}(T_u^{-1}\circ\mathbf{y}_i|\mathbf{m}_{iul})}{\sum_l^{N_{\mathrm{modes}}}\exp(w_{iul})}\Big] \tag{28}$$

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

$$(\mathbf{R}_u, \mathbf{t}_u) = \mathrm{rigidFrom3Points}(\mathbf{y}_{i(u)}, \mathbf{y}_{j(u)}, \mathbf{y}_{k(u)}) \tag{29}$$

where $\mathrm{rigidFrom3Points}$ is the Gram–Schmidt-based frame construction operation described in Ref. [9], Alg. 21; we additionally add a numerical stability factor of 0.01 Å to the vector-norm calculations to handle edge cases when

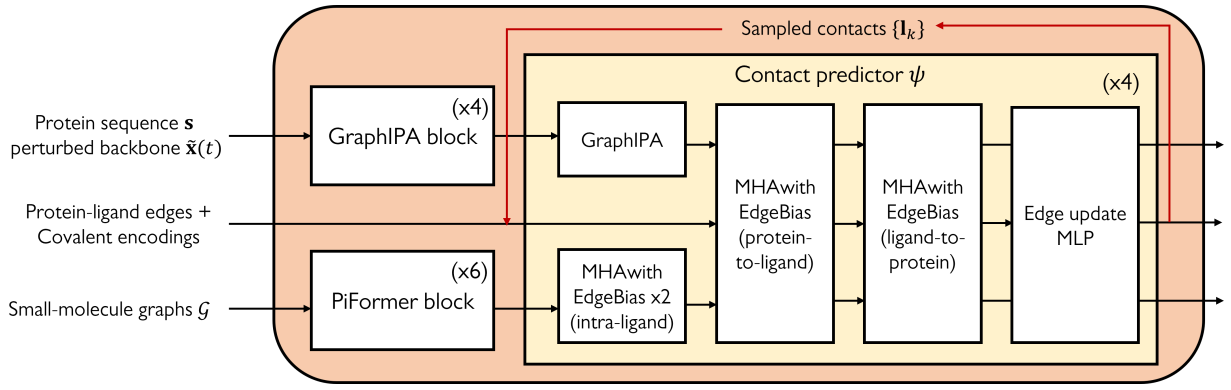

Figure 1: Network architecture schematics for the encoders and contact prediction modules.

computing the rotation matrices from perturbed coordinates. Each component the 3D distance-angle distribution q^{3D} is parameterized by

$$q_{\mathrm{3D}}(\mathbf{t}|\mu,\sigma,\mathbf{v}) = \mathrm{Gaussian}(\|\mathbf{t}\|_2|\mu,\sigma) \times \mathrm{PowerSpherical}(\frac{\mathbf{t}}{\|\mathbf{t}\|_2}|\mathbf{v}, d=3) \tag{30}$$

where PowerSpherical is a power spherical distribution introduced in [62]; $\mathbf{m}_{iul} \coloneqq (\mu,\sigma,\mathbf{v})_{iul}$, and

$$[\mathbf{w}_{iu}, \mathbf{m}_{iu}] = \mathrm{3DMixtureDensityHead}\big(\mathbf{G}_{l_{\max}}\big)_{iu}. \tag{31}$$

whre 3DMixtureDensityHead is a 3-layer MLP.

Using an equivariant graph transformer similar to ESDM (see Sec. A.6) 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 [22] and introduce a $\mathrm{SE}(3)$-invariant denoising score matching loss based on the Frame Aligned Point Error (FAPE) [9]:

$$\mathcal{L}_{\mathrm{3D-DSM}} = \mathbb{E}_{t\sim(0,1],\mathbf{y}_t\sim q_{0:t}(\cdot|\mathbf{y})}\left[\mathrm{mean}_{u,i}\min(\|T_u^{-1}\circ\mathbf{y}_i - \hat{T}_u^{-1}\circ\hat{\mathbf{y}}_i\|_2, 10\,\text{Å})\cdot\sqrt{\alpha_t}\right] \tag{32}$$

where

$$\hat{\mathbf{y}} = \mathrm{GeometryPredictionHead}(\mathbf{y}_t; \mathbf{H}_{l_{\max}}, \mathbf{F}_{l_{\max}}, \mathbf{G}_{l_{\max}}) \tag{33}$$

$\mathcal{L}_{\mathrm{CC-regression}}$ is a mean squared loss for fitting the "level 1" chemical checker (CC) [61] embeddings which represents harmonized and integrated bioactivity data, and $\mathcal{L}_{\mathrm{CC-ismask}}$ is an auxiliary binary cross entropy loss for classifying whether a specific CC entry is available for any molecule in the chemical checker dataset. Model is trained for 20 epochs with 15% masking ratio for atom and bond encodings, 40% masking ratio for stereochemistry encodings, and dropout=0.1; $\mathcal{L}_{\mathrm{MLM}}$ is a standard cross-entropy loss for predicting the masked tokens which is added to encourage learning on molecular graph topology distributions, but empirically we found $\mathcal{L}_{\mathrm{MLM}}$ converged within the first two epochs and did not find it to influence the learning dynamics of other tasks.

A.3 Protein sequence and backbone encoding

The inputs to the protein encoder are (i) the one-hot amino-acid type (20 standard residues + 1 "unknown" token) encoding of the 1D sequence s, (ii) the backbone $(\mathrm{N}, \mathrm{C}\alpha, \mathrm{C})$ coordinates of a perturbed protein structure $\mathbf{x}(t)$ sampled from the forward SDEs described in Table A1, and (iii) a random Fourier encoding of the diffusion time step t. To reduce memory cost, the protein backbone is represented as a sparse graph with each node mapped to each amino acid residue and randomized edges according to the inclusion probability $p(\mathrm{add\_edge}(i,j)) = \exp(-\|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|/10.0\,\text{Å})$ for all residue pairs (i,j). The edge representations are initialized as a random Fourier encoding of the signed sequence distance between two residues (i,j) if i and j are located on the same chain, and are initialized as zeros if (i,j) are located on different chains.

The protein encoder is composed of 4 stacks of invariant point attention (IPA) [9] blocks with two technical modifications:

- The attention scores are computed on the sparsified protein graph, instead of the densely-connected graph as in standard self-attention layers;

- Each node i is associated with n_{head} replicas of coordinate frames $\{T\}_i$, instead of a single frame as in a static structure representation. $\{T\}_i$ is initialized as n_{head} copies of the backbone frames constructed by $\text{rigidFrom3Points}(\mathbf{x}_{\text{N},i}, \mathbf{x}_{\text{C}\alpha,i}, \mathbf{x}_{\text{C},i})$. The layer output is $n_{\text{head}} \times 7$ scalars representing the translation vector and the quaternion variable to update the frame associated with each attention head.

the multi-replica design is found to moderately improve model convergence at a fixed network size. For conciseness, we refer to the modified invariant point attention as GraphIPA.

A.4 Contact predictor

As illustrated in Figure 1, the embeddings from the protein and small-molecule ligand graph encoders are passed to the contact predictor to estimated the contact maps $\mathbf{L}$. A protein-ligand graph is created before the contact predictor forward pass, with pairwise intermolecular edges connecting all protein residues and ligand atoms. The contact predictor is composed of 4 modules each comprises of an intra-protein GraphIPA block, a bidirectional intra-ligand-graph self-attention layer, a bidirectional self-attention layer on the protein-ligand intermolecular edges, and a MLP to update protein-ligand edge representations using the attention maps and previous-layer edge representations. The final edge representations are used to predict $\mathbf{L}$ as described by Equation 5. The contact predictor weights are shared across all one-hot contact matrix sampling iterations.

A.5 All-atom graph featurization

All protein heavy-atoms nodes (features and 3D coordinates) and the ligand 3D coordinates sampled from the geometry prior q_{T^*} are added to the network inputs right before the ESDM block forward pass. Each protein atom representation is initialized as the concatenation of:

- The residue-wise representation from the protein backbone encoder;
- An one-hot encoding of its atom type as defined by the 37 standard amino acid heavy atom symbols in the PDB format [63];
- A random Fourier encoding of the diffusion time step t.

A random Fourier encoding of the diffusion time step t is also concatenated to the ligand atom representations from the ligand graph encoder and are transformed by a 2-layer MLP.

Given the noised all-atom protein coordinates at diffusion time t, the following edges are added to the protein-ligand graph:

- Edges connecting a protein atom node and the residue node that the protein atom belongs to;
- Edges connecting two protein atom nodes that are within the same residue;
- Edges connecting two protein atom nodes that are within $6.0\,\text{Å}$ distance;
- Edges connecting a protein atom node and a ligand atom node that are within $8.0\,\text{Å}$ distance;

The protein-atom-involving edges are initialized as a concatenation of the following features:

- A boolean code indicating whether the source node and target node belong to the same residue or the same ligand molecule;
- A boolean code indicating whether there is a covalent bond between the source and target nodes. The covalent bonding information for protein-ligand edges are resolved based on the reference protein-ligand complex structure, where an atom pair (i, j) is considered as a covalent bond if the distance satisfies $d_{ij} < 1.2\sigma_{ij}$ where $\sigma_{ij} = \frac{1}{2}(\sigma_i + \sigma_j)$ is the average Van der Waals (VdW) radius for the atom pair.

To focus the learning problem on binding-site parts of the protein-ligand complex structure, the following *native contact encoding* features are added to the protein sub-graph that do not involve residues that are within $6.0\,\text{Å}$ of any ligand heavy atom; given two amino acid residues, we define the native contact encoding as the concatenation of clean-protein-structure $\text{N} - \text{N}$, $\text{C}\alpha - \text{C}\alpha$, and $\text{C} - \text{C}$ distances discretized into $[2.0\,\text{Å}, 4.0\,\text{Å}, 6.0\,\text{Å}, 8.0\,\text{Å}]$ bins. Such features are embedded by a 2-layer MLP and added to the residue-residue edge representations. Note that at training time the native contact encodings are computed from the protein structure in the ground-truth protein-ligand complex, while at sampling time they are computed from the input backbone template.

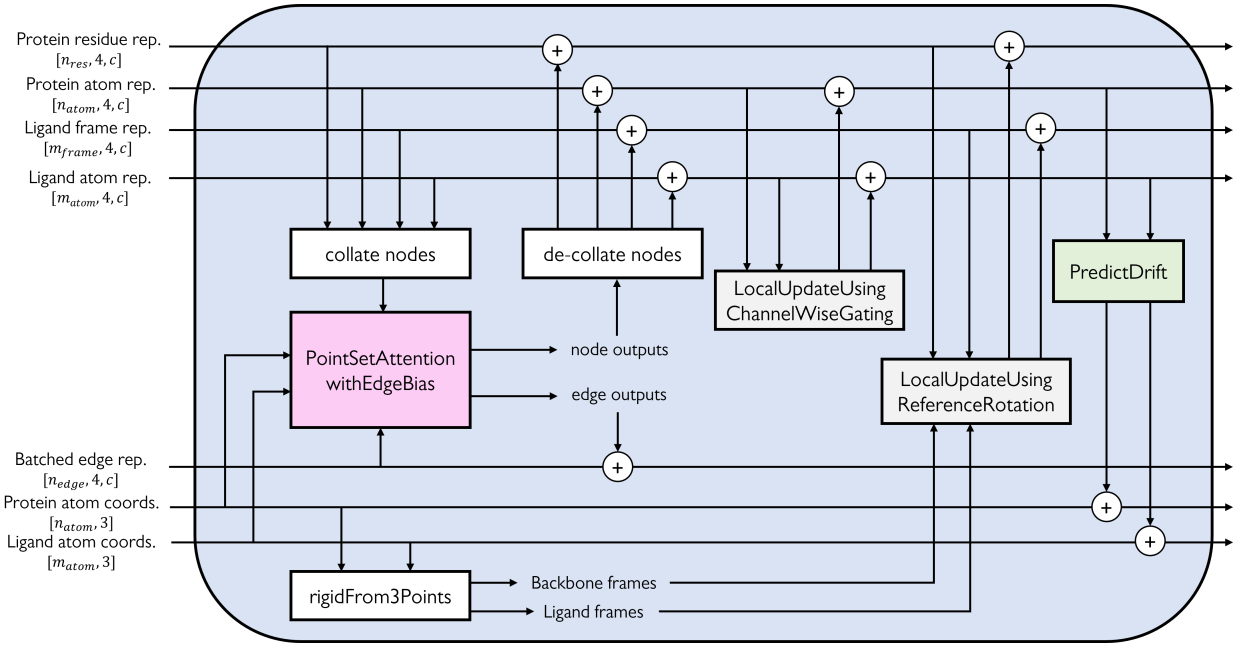

Figure 2: Network architecture of a single block in the equivariant structure diffusion module (ESDM). Arrows indicate information flow directions, and "+" indicates an element-wise tensor summation.

A.6 The ESDM architecture

The neural network architecture of the proposed equivariant structure diffusion module (ESDM) is summarized in Figure 2. The forward pass expression of the trainable modules PointSetAttentionwithEdgeBias, LocalUpdateUsingChannelWiseGating, LocalUpdateUsingReferenceRotation, PredictDrift are defined as:

$$\mathbf{f}'_{\mathrm{s}}, \mathbf{f}'_{\mathrm{v}}, \mathbf{e}' = \mathrm{PointSetAttentionwithEdgeBias}(\mathbf{f}_{\mathrm{s}}, \mathbf{f}_{\mathrm{v}}, \mathbf{e}, \mathbf{t}) \quad \text{where} \tag{34}$$

$$\mathbf{f}_{\mathrm{Q}}, \mathbf{f}_{\mathrm{K}}, \mathbf{f}_{\mathrm{V}} = \mathbf{W}_{\mathrm{s}} \cdot \mathbf{f}_{\mathrm{s}}, \quad \mathbf{t}_{\mathrm{Q}}, \mathbf{t}_{\mathrm{K}}, \mathbf{t}_{\mathrm{V}} = (\mathbf{t}/10\,\text{Å} + \mathbf{f}_{\mathrm{v}} \cdot \mathbf{W}_{\mathrm{v}}) \tag{35}$$

$$\mathbf{z}_{ij} = \frac{1}{\sqrt{c_{\mathrm{head}}}}(\mathbf{f}^{\mathrm{T}}_{\mathrm{Q,i}} \cdot \mathbf{f}_{\mathrm{K,j}}) + \mathbf{W}_{\mathrm{e}} \cdot \mathbf{e}_{ij} - \frac{\mathbf{w}_{ij}}{\sqrt{18 c_{\mathrm{head}}}} \|\mathbf{t}_{\mathrm{Q}} - \mathbf{t}_{\mathrm{K}}\|_2^2 \tag{36}$$

$$\boldsymbol{\alpha}_{ij} = \mathrm{Softmax}_{j\in\{i\}}(\mathbf{z}_{ij}), \quad \mathbf{e}' = \mathrm{MLP}(\mathbf{z}_{ij}) \tag{37}$$

$$\mathbf{f}'_{\mathrm{s}} = \sum_{j\in\{i\}} \boldsymbol{\alpha}_{ij} \odot \mathbf{f}_{\mathrm{V}}, \quad \mathbf{f}'_{\mathrm{v}} = (\sum_{j\in\{i\}} \boldsymbol{\alpha}_{ij} \odot \mathbf{t}_{\mathrm{V}}) - \mathbf{t}/10\,\text{Å} \tag{38}$$

where $\mathbf{f}_{\mathrm{s}} \in \mathbb{R}^{N_{\mathrm{nodes}} \times c}, \mathbf{f}_{\mathrm{v}} \in \mathbb{R}^{N_{\mathrm{nodes}} \times 3 \times c}, \mathbf{e} \in \mathbb{R}^{N_{\mathrm{edges}} \times c}, \mathbf{t} \in \mathbb{R}^{N_{\mathrm{nodes}} \times 3}$. Note that the expression for computing attention weights $\mathbf{z}$ is directly adapted from IPA.

$$\mathbf{f}'_{\mathrm{s}}, \mathbf{f}'_{\mathrm{v}} = \mathrm{LocalUpdateUsingChannelWiseGating}(\mathbf{f}_{\mathrm{s}}, \mathbf{f}_{\mathrm{v}}) \quad \text{where} \tag{39}$$

$$\mathbf{f}'_{\mathrm{s}}, \mathbf{f}_{\mathrm{gate}} = \mathrm{MLP}(\mathbf{f}_{\mathrm{s}} \oplus \|\mathbf{f}_{\mathrm{v}}\|_2) \tag{40}$$

$$\mathbf{f}'_{\mathrm{v}} = (\mathbf{f}_{\mathrm{v}} \cdot \mathbf{W}_{\mathrm{v}}) \odot \mathbf{f}_{\mathrm{gate}} \tag{41}$$

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

$$\mathbf{f}'_{\mathrm{s}}, \mathbf{f}'_{\mathrm{v}} = \mathrm{LocalUpdateUsingReferenceRotation}(\mathbf{f}_{\mathrm{s}}, \mathbf{f}_{\mathrm{v}}, \mathbf{R} \in \mathrm{SO}(3)) \quad \text{where} \tag{42}$$

$$\mathbf{f}'_{\mathrm{s}}, \mathbf{f}_{\mathrm{vloc}} = \mathrm{MLP}(\mathbf{f}_{\mathrm{s}} \oplus \mathbf{R}^{\mathrm{T}} \cdot \mathbf{f}_{\mathrm{v}} \oplus \|\mathbf{f}_{\mathrm{v}}\|_2) \tag{43}$$

$$\mathbf{f}'_{\mathrm{v}} = \mathbf{R} \cdot \mathbf{f}_{\mathrm{vloc}} \tag{44}$$

Since the third row of $\mathbf{R}$ is a pseudovector as described in $\mathrm{rigidFrom3Points}$, 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 $\mathbf{f}_{\mathrm{vloc}}$ is $\mathrm{SE}(3)$-invariant but in general **not** invariant under parity inversion i. This property ensures that ESDM can learn the correct chiral symmetry breaking behaviors in molecular 3D conformation distributions.

$$\Delta\mathbf{t} = \mathrm{PredictDrift}(\mathbf{f}_{\mathrm{s}}, \mathbf{f}_{\mathrm{v}}) \quad \text{where} \tag{45}$$

$$\mathbf{o}_{\mathrm{scale}} = \mathrm{Softplus}(\mathrm{MLP}(\mathbf{f}_{\mathrm{s}})) \tag{46}$$

$$\Delta\mathbf{t} = (\mathbf{f}_{\mathrm{v}} \cdot \mathbf{W}_{\mathrm{drift}}) \odot \mathbf{o}_{\mathrm{scale}}. \tag{47}$$

The predicted drift vectors $\Delta\mathbf{t}$ are added to the input node coordinates; the final coordinate outputs are taken as the predicted denoised observations $\hat{\mathbf{x}}(0), \hat{\mathbf{y}}(0)$.

A.7 Model training and hyperparameters

The loss function for NeuralPLexer training is:

$$\mathcal{L}_{\mathrm{training}} = \mathbb{E}_{t\sim(0,1]}\big[\mathcal{L}_{\mathrm{contact}}(t) + \mathcal{L}_{\mathrm{gp-mean}}(t) + \mathcal{L}_{\mathrm{DSM-prot}}(t) + \mathcal{L}_{\mathrm{DSM-ligand}}(t) + \mathcal{L}_{\mathrm{DSM-site}}(t)\big] \tag{48}$$

We train the contact predictor ψ to match the posterior distribution defined by the observed contact map $q_{\mathbf{L}} \coloneqq \mathrm{Categorical}_{n_{\mathrm{res}}\times m}(\mathbf{L})$ where $\mathbf{L} \coloneqq \bigoplus_k \mathbf{L}_k$ with intermediate ligand-wise one-hot matrices $\mathbf{l}_k$ sampled from $q_{\mathbf{L}_k}$:

$$\mathcal{L}_{\mathrm{contact}}(t) = \mathrm{KL}(q_{\mathbf{L}} \| q_\psi(\cdot|\mathbf{0}, \mathbf{s}, \tilde{\mathbf{x}}(t), \mathcal{G})) + \sum_{k=1}^{K} \mathbb{E}_{\mathbf{l}_k\sim q_{\mathbf{L}_k}}\big[\mathrm{JS}(q_{\mathbf{L}_k} \| q_{\psi,k}(\cdot|\sum_{r=1}^{k}\mathbf{l}_r, \mathbf{s}, \tilde{\mathbf{x}}(t), \mathcal{G}))\big] \tag{49}$$

where KL denotes a Kullback–Leibler divergence and JS denotes a Jensen–Shannon divergence. An auxiliary loss is added to the mean term in the predicted geometry prior:

$$\mathcal{L}_{\mathrm{gp-mean}}(t) = \mathbb{E}_{\mathbf{l}_k\sim q_{\mathbf{L}_k}}\big[\|\mathbf{c}_{\psi,k}^{\mathrm{T}}(\sum_{r=1}^{k}\mathbf{l}_r, \mathbf{s}, \tilde{\mathbf{x}}(t), \mathcal{G}) \cdot \tilde{\mathbf{x}}(t) - \mathbf{c} \cdot \tilde{\mathbf{x}}(t)\|\big] \tag{50}$$

The denoising score matching (DSM) loss expressions are given by

$$\mathcal{L}_{\mathrm{DSM-prot}} = \mathbb{E}_{\mathbf{x}(t),\mathbf{y}(t)\sim q_{0:t}(\cdot|\mathbf{x}(0),\mathbf{y}(0))}\big[\frac{1}{n}\sum_i \|\mathbf{x}_i(0) - \hat{\mathbf{x}}_i(0)\|_2/\sigma(t)\big] \tag{51}$$

$\mathcal{L}_{\mathrm{DSM-site}}$ is defined analogously but averaged for residues that are within $6.0\,\text{Å}$ of the ligand in the ground-truth structure. Lastly

$$\mathcal{L}_{\mathrm{DSM-ligand}} = \mathbb{E}_{\mathbf{x}(t),\mathbf{y}(t)\sim q_{0:t}(\cdot|\mathbf{x}(0),\mathbf{y}(0))}\big[\frac{1}{m}\sum_i \|\mathbf{y}_i(0) - \hat{\mathbf{y}}_i(0)\|_2/\sigma(t)\big]. \tag{52}$$

For the ligand graph encoder, we use 6 PiFormer blocks with a embedding dimension of 512 for atom representation and frame representations, and a dimension of 128 for pair representations. For the protein encoder, we use 4 GraphIPA blocks with a node embedding dimension of 256 and edge embedding dimension of 64. For the contact predictor we use 4 blocks with the same embeddings sizes (256, 64) as in the protein encoder; linear layers are added to project the ligand representations to the length of protein representations before they are passed to the contact predictor. For ESDM, we use a stack of 4 blocks with a embedding dimension of 64 for both node and edge representations, that is, each node i is associated with scalar representations $\mathbf{f}_{\mathrm{s},i}$ of size 64 and vector representations $\mathbf{f}_{\mathrm{v},i}$ of size [3, 64].

The pretrained small-molecule encoder weights are frozen during training. Model is trained with batch size of 8 for 40 epochs, using dropout=0.05, an inital learning rate of 3E-4 with 1000 warmup steps followed by a cosine annealing learning rate decay schedule. On the PDBBind 2020 training set (170k samples), the training run took 20 hours a single NVIDIA-Tesla-V100-SXM2-32GB GPU.

A.7.1 Task-specific fine-tuning

The model used for fixed-backbone protein-ligand docking is fine-tuned on the original PDBBind training dataset, while all backbone atoms (N, $\mathrm{C}\alpha$, C, O) and $\mathrm{C}\beta$ atoms are set to the ground-truth coordinates. Fine-tuning is performed for 20 epochs with a batch size of 8 without teacher forcing for the geometry prior (i.e., sampling the one-hot matrix $\mathbf{l}$ from the

observed contact map $q_{\mathbf{L}} = \mathrm{Categorical}_{n_{\mathrm{res}} \times m}(\mathbf{L})$, using the predicted contact map $\psi(\mathbf{l}, \mathbf{s}, \tilde{\mathbf{x}}, \mathcal{G})$ to parameterize the finite-time transition kernels $q_t(\mathbf{Z}(t)|\mathbf{Z}(0))$ during model forward pass, and then backpropagating the model end-to-end) using a cosine annealing schedule with a initial learning rate of $1E-4$.

The model used for binding-site inpainting is fine-tuned on all split-chain samples from the original PDBBind training dataset. A protein-chain/ligand pair is included in the fine-tuning dataset if any heavy atom of the ligand is within $10\,\text{Å}$ of any heavy atom of the protein chain. All receptor residues that are not within $6.0\,\text{Å}$ of the ligand are set to the ground-truth coordinates with the residue-wise and protein-atom-wise time-step encoding set to zeros. Fine-tuning is performed for 40 epochs with a batch size of 10 without teacher forcing for the geometry prior using a cosine annealing schedule with a initial learning rate of $1E-4$.

A.8 Computational details

A.8.1 Test datasets and post-processing

While the time-split-based PDBBind 2020 dataset has been used in previous works for studying model generalization to novel protein-ligand pairs, we noticed that the 363-sample test set curated by [43] contains samples with improperly removed alternative ligand conformation ground truths or deleted adjacent chains that strongly interact with the ligand molecule in the full structure (e.g., binding sites near protein-protein interfaces). To ensure a reasonable comparison to docking-based methods, for the test dataset used fixed-backbone ligand conformation prediction experiments we keep all protein chains that are within $10\,\text{Å}$ of the ligand from the original PDB file instead of using the receptor PDB files curated by PDBBind; we further removed all covalent ligands and pipetide binders from the test set as such cases are usually tackled by specialized algorithms [64, 65], resulting in 275 test samples in total to produce the results presented in Figure 2a-d.

The AlphaFold2 structures used in the ligand-coupled binding site repacking task are predicted using ColabFold [66] with default MSA, recycling, and AMBER relaxation settings, and without using templates in order to best reflect the prediction fidelity of AlphaFold2 on novel targets (since all PDBBind test set samples are deposited before year 2021). 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.

A.8.2 Baseline method configurations

We run CB-Dock [44] with a heuristic low-sampling-intensity configuration (exhaustiveness=1, number of clustered binding sites to start local docking = 1) such that the execution time (43 seconds per ligand on average on single core of an Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz CPU) is comparable to deep-learning-based methods that were proposed to perform docking at a low computing budget. The top-scored ligand conformations collected for each protein-ligand pair as ranked by Autodock Vina [67] are used to obtain the success rate results in Figure 2a. EquiBind [43] are launched with the default configuration file, and for each protein-ligand pair 64 ligand conformations are generated using different random RDKit [68] input conformers. We note that the incorporation of side-chain flexibility as provided by AutoDock Vina and the systematic tuning of sampling intensity in docking-based methods may offer a more accurate comparision regarding the accuracy/computational time relationship among physics-based and learning-based methods.

RosettaLigand [23] 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 [49], with randomized torsion angles using the BCL [69] library as described in the standard protocol. We set the docking box width to $4.0\,\text{Å}$ 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. While high-fideltiy physics-based methods such as IFD-MD [18] have been proposed for flexible-receptor ligand docking, such algorithms often incurs orders-of-magnitude higher computational cost thus are not included within the scope of this study.

A.9 Evaluation metric details

All protein structure alignments and TM-Score calculations are performed using TMAlign [49]. All reported TM-Scores are normalized by the chain length of the reference PDB structure. The per-residue all-atom lDDT score is computed using OpenStructure [70]; the lDDT-BS score is then computed by averaging the per-residue scores for ligand binding site residues with a cutoff of $4.0\,\text{Å}$ as used in CAMEO [48]. The symmetry-corrected heavy-atom RMSD for ligand structure comparison is computed using the obrms function in OpenBabel [71]. 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`.