nature machine intelligence

Article

https://doi.org/10.1038/s42256-024-00792-z

State-specific protein–ligand complex structure prediction with a multiscale deep generative model

In the format provided by the authors and unedited

Supplementary Information

Contents

A Supplementary results

In Tables S1-S2 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 S1-S2, "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 S1. 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 [S1]	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	-	-
	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 [S1]	mean	0.909	1.893	0.943	1.387	9.115	10.00
		std	0.089	0.684	0.084	0.559	9.466	11.11
		25%	0.898	1.370	0.943	1.008	3.101	3.394
		50%	0.933	1.780	0.967	1.260	5.398	5.390
		75%	0.958	2.310	0.978	1.680	9.808	10.30
	LSA-SDE	mean	**0.934**	**1.486**	**0.950**	**1.236**	8.985	9.812
		std	0.091	0.750	0.087	0.656	9.943	10.75
		25%	0.938	0.900	0.955	0.720	2.674	2.583
		50%	0.966	1.290	0.975	1.095	5.010	5.246
		75%	0.978	1.870	0.987	1.613	9.388	10.54

Table S2. 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 [S1]	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	-	-
	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 [S1]	mean	0.867	2.418	0.921	1.818	14.22	13.79
		std	0.110	0.908	0.089	0.948	13.01	12.50
		25%	0.839	1.730	0.918	1.162	3.567	3.156
		50%	0.903	2.280	0.953	1.555	7.410	7.878
		75%	0.941	2.940	0.971	2.218	23.55	24.27
	LSA-SDE	mean	**0.893**	**2.026**	**0.926**	**1.676**	14.03	12.72
		std	0.110	0.982	0.091	0.961	12.94	12.23
		25%	0.866	1.360	0.916	1.055	3.353	3.293
		50%	0.929	1.870	0.958	1.425	8.146	5.907
		75%	0.965	2.570	0.978	2.118	22.97	21.60

B 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, LinearNoBias$(\cdot)$ denotes a standard linear transformation with a trainable weight matrix right-multiplied to the last input tensor dimension; MLP$(\cdot)$ denotes a standard 3-layer multilayer perceptron with GELU activation function [S3] and with layer normalization [S4] applied to the output activations.

C Diffusion SDEs

C.1 Preliminaries

For the temporally homogeneous forward-time SDE

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

or equivalently, in the space of latent coordinates

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

the corresponding reverse-time SDE is given by

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

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

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

or equivalently,

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

C.2 Euclidean and chiral symmetries

Given group G, a function $f: X \to Y$ is said to be equivariant if for all $g \in G$ and $x \in X$, $f(\varphi_X(g)\cdot x) = \varphi_Y(g)\cdot f(x)$, and f is said to be invariant if $f(\varphi_X(g)\cdot x) = f(x)$. We are interested in the special Euclidean group $G = \mathrm{SE}(3)$ 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 $\mathbf{R}\in\mathrm{SO}(3)$. To adhere to the physical constraint that p_{data} is always SE(3)-invariant, the transition kernels of the forward-time SDE need to satisfy 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:

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

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

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

Algorithm S1 NeuralPLexer Inference

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

$\{\mathbf{f}_{\text{PLM}}\} \leftarrow$ Compute ESM-2-650M features for all input chains in $\{\mathbf{s}\}$ — Ref. [S2]
if *use_template* **then**
  $\{\mathbf{f}_{\text{template}}\} \leftarrow$ Retrieve template protein structure and compute template features
end if
for $i \in \{1, \cdots, N_{\text{conformations}}\}$ **do**
  $T = \text{DiffusionTimeSchedule}(\tau = 1.0)$ — Equation (8)
  Sample initial protein coordinates $\mathbf{x}_T$ from the prior q_T — Equation (S5)
  residue_graph_0 $\leftarrow$ Generate the residue-scale graph based on $(\mathbf{x}_T)_{C\alpha}$ — Section E
  $\mathbf{F}_{\text{P}} \leftarrow$ Sample N_{P} protein backbone frame nodes and gather embeddings — $N_{\text{P}} = 96$
  $\tilde{\mathbf{I}} \leftarrow \mathbf{0}$
  # Generating contact maps and block-adjacency matrices (Equation (1))
  if $\{G\}$ contains more than 0 ligands **then**
    Compute MHT embeddings for all input ligand molecular graphs in $\{G\}$ — Algorithm S2
    $\mathbf{F}_{\text{L}} \leftarrow$ Sample N_{L} ligand frame nodes, gather and symmetrize MHT embeddings — $N_{\text{L}} = 32$
    for $k \in \{1, \cdots, N_{\text{L}}\}$ **do**
      residue_graph_k $\leftarrow \text{CPMForward}(\textit{residue\_graph\_0}, \tilde{\mathbf{I}}, \tau = 1.0)$ — Section F
      $\hat{P} = \text{LinearNoBias}(\text{MLP}(\text{GetEdges}_{\text{BF}}(\textit{residue\_graph\_k})))$ — $\hat{P} \in \mathbb{R}^{N_{\text{res}} \times N_{\text{L}} \times N_{\text{bins}}}, N_{\text{bins}} = 32$
      $\tilde{L}_k \leftarrow \text{DistogramToContactMap}(\hat{P})$ — Equation (4)
      $\hat{\mathbf{L}}_k \leftarrow \text{SumIntoPatches}(\tilde{L}_k) \odot [1 - \max_{\text{column-wise}}(\tilde{\mathbf{I}})]$ — $\hat{\mathbf{L}}_k \in \mathbb{R}^{N_{\text{P}} \times N_{\text{L}}}$
      $\mathbf{l}_k = \text{OneHot}(i_k, j_k);\ (i_k, j_k) \sim \text{Categorical}_{N_{\text{P}} \times N_{\text{L}}}(\hat{\mathbf{L}}_k)$ — $\mathbf{l}_k \in \{0,1\}^{N_{\text{P}} \times N_{\text{L}}}$
      $\tilde{\mathbf{I}} \leftarrow \tilde{\mathbf{I}} + \mathbf{l}_k$ — $\tilde{\mathbf{I}} \in \{0,1\}^{N_{\text{P}} \times N_{\text{L}}}$
    end for
    Compute $\mathbf{c}$ and $\mathbf{U}$ for ligand degrees of freedom using predicted $\hat{\mathbf{L}} \coloneqq \hat{\mathbf{L}}_{N_{\text{L}}}$ — Equation (7)
    Sample initial ligand coordinates $\mathbf{y}_T$ from the prior q_T — Equation (S5)
    $\mathbf{Z}_T \leftarrow \text{concat}(\mathbf{x}_T, \mathbf{y}_T)$
  else
    $\mathbf{Z}_T \leftarrow \mathbf{x}_T$
  end if
  # Generating the 3D structure for all heavy atoms (Equation (2))
  Compute MHT embeddings for all amino acid molecular graphs — Algorithm S2
  for $\tau \in \{1, 1 - \Delta\tau, \cdots, \Delta\tau\}$ **do** — $\Delta\tau = 1/N_{\text{steps}}$
    $t \leftarrow \text{DiffusionTimeSchedule}(\tau)$ — Equation (8)
    $\Delta t \leftarrow \text{DiffusionTimeSchedule}(\tau) - \text{DiffusionTimeSchedule}(\tau - \Delta\tau)$ — Equation (8)
    residue_graph, *atomic_graph* $\leftarrow$ Regenerate the residue-scale and atomic-scale graph based on $\mathbf{Z}_t$ — Section E
    residue_graph $\leftarrow \text{CPMForward}(\textit{residue\_graph}, \tilde{\mathbf{I}}, \tau)$ — Section F
    graph_rep $\leftarrow \text{Collate}(\textit{residue\_graph}, \textit{atomic\_graph}, \textit{cross\_scale\_graph})$ — Table S6
    $\hat{\mathbf{Z}}_0 \leftarrow \text{ESDMForward}(\mathbf{Z}_t; \textit{graph\_rep}, \tau)$ — Section G
    $\beta_{t-\Delta t} \leftarrow \text{InvTempSchedule}(\tau - \Delta\tau)$ — See Methods, LSA-SDE
    $\mathbf{Z}_{t-\Delta t} = \text{IntegratorStep}(\mathbf{Z}_t, \hat{\mathbf{Z}}_0, \beta_{t-\Delta t}, \Delta t)$ — Equation (12)
  end for
  # Assigning confidence estimations (per-residue and per-ligand-atom pLDDT)
  if *compute_plddt* **then**
    residue_graph, *atomic_graph* $\leftarrow$ Regenerate the residue-scale and atomic-scale graph based on $\mathbf{Z}_t$ — Section E
    residue_graph $\leftarrow \text{CPMForward}(\textit{residue\_graph}, \mathbf{0}, \tau = 0.0)$ — Section F
    graph_rep $\leftarrow \text{Collate}(\textit{residue\_graph}, \textit{atomic\_graph}, \textit{cross\_scale\_graph})$ — Table S6
    pLDD_gram $= \text{ConfidenceEstimationHead}(\mathbf{Z}_0; \textit{graph\_rep})$ — Section H
    pLDDT_prot, *pLDDT_lig* $= \text{compute\_plDDT}(\textit{pLDD\_gram})$ — Equation S29
    yield $\mathbf{Z}_0, (\textit{pLDDT\_gram}, \textit{pLDDT\_prot}, \textit{pLDDT\_lig})$
  else
    yield $\mathbf{Z}_0$
  end if
end for

While the forward SDE is 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 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 $\nabla_{-\mathbf{Z}} \log q_t(-\mathbf{Z}) \neq -\nabla_{\mathbf{Z}} \log q_t(\mathbf{Z})$.

C.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 $\mathbf{z}_t \in \mathbb{R}^3$, $t \in (0,\infty)$

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

where

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

thus

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

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 $\tau \in [t,s)$:

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

or equivalently,

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

where

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

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

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

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

which implies

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

Computing the integrals analytically yields

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

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

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

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

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

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

For simplicity, we adapt the variance term to that of the forward SDE: $\mathrm{Var}(\mathbf{z}_t) \approx \frac{\sigma^2}{2\theta\beta}(e^{-2\theta t} - e^{-2\theta s})$. 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 [S1] 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].

D Molecular graph featurization and encoder details

D.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 $\mathbb{R}^{18+7} \to \mathbb{R}^{c}$;
- 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 $\mathbb{R}^{4*2+18+7} \to \mathbb{R}^{c}$ that embeds the bond type encodings (defined as $[\text{is\_single}, \text{is\_double}, \text{is\_triple}, \text{is\_aromatic}]$) 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 $\mathbf{S} \in \mathbb{R}^{N_{\text{frame}} \times N_{\text{frame}} \times c_{\text{s}}}$, as detailed in Table S3. $\mathbf{S}$ 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 $\mathbf{S}_{\text{uv}}$ are included as model inputs.
- 3-hop edge representations $\mathbf{f}_{\text{aa}} \in \mathbb{R}^{N_{\text{atom}} \times N_{\text{atom}} \times c_{\text{p}}}$. For each pair of atoms (i, j), the element $(\mathbf{f}_{\text{aa}})_{ij}$ is initialized by a linear layer $\mathbb{R}^{4+4} \to \mathbb{R}^{c_{\text{p}}}$ 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 $\mathbf{f}_{\text{aa}}$ are included as model inputs.
- Pair representations $\mathbf{f}_{\text{fa}} \in \mathbb{R}^{N_{\text{frame}} \times N_{\text{atom}} \times c_{\text{p}}}$. For each frame-atom pair (u, l), the element $(\mathbf{f}_{\text{fa}})_{ul}$ is initialized by a linear layer $\mathbb{R}^{3*4} \to \mathbb{R}^{c_{\text{p}}}$ 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 $\mathbf{X}_{\text{s}}$ as the $N_{\text{frame}} \times N_{\text{frame}}$ binary adjacency matrix of edges among frame nodes, and $\mathbf{X}_{\text{a}}$ as the $N_{\text{atom}} \times N_{\text{atom}}$ binary adjacency matrix of 3-hop edges among atomic nodes. We additionally denote $\mathbf{H}_{\text{fa}}$ 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 S3. Stereochemistry encoding definitions.

Feature	Definition
# Relative topological orientation between two frames	
$\mathbf{S}_{uv,0}$	$\text{common\_bond}(\text{u}, \text{v}) = \text{incoming\_bond}(\text{u})$
$\mathbf{S}_{uv,1}$	$\text{common\_bond}(\text{u}, \text{v}) = \text{incoming\_bond}(\text{v})$
$\mathbf{S}_{uv,2}$	$\text{common\_bond}(\text{u}, \text{v}) = \text{outgoing\_bond}(\text{u})$
$\mathbf{S}_{uv,3}$	$\text{common\_bond}(\text{u}, \text{v}) = \text{outgoing\_bond}(\text{v})$
# Detect small ring structures	
$\mathbf{S}_{uv,4}$	$\text{i}(\text{v}) \in \{\text{i}(\text{u}), \text{j}(\text{u}), \text{k}(\text{u})\}$
$\mathbf{S}_{uv,5}$	$\text{j}(\text{v}) \in \{\text{i}(\text{u}), \text{j}(\text{u}), \text{k}(\text{u})\}$
$\mathbf{S}_{uv,6}$	$\text{k}(\text{v}) \in \{\text{i}(\text{u}), \text{j}(\text{u}), \text{k}(\text{u})\}$
# Polyhedral chiral center stereochemistry	
$\mathbf{S}_{uv,7}$	$(\text{j}(\text{u}) = \text{j}(\text{v})) \wedge \text{is\_above\_plane}(\text{u}, \text{v})$
$\mathbf{S}_{uv,8}$	$(\text{j}(\text{u}) = \text{j}(\text{v})) \wedge \text{is\_below\_plane}(\text{u}, \text{v})$
# Planar stereochemistry for double and π bonds	
$\mathbf{S}_{uv,9}$	$\text{is\_double\_or\_aromatic}(\text{common\_bond}(\text{u}, \text{v})) \vee \text{is\_same\_side}(\text{u}, \text{v})$
$\mathbf{S}_{uv,10}$	$\text{is\_double\_or\_aromatic}(\text{common\_bond}(\text{u}, \text{v})) \vee \text{not\_same\_side}(\text{u}, \text{v})$

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.

D.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 $\mathbf{H}_{\text{L}}$ as the $N_{\text{L}} \times N_{\text{frame}}$ incidence matrix indicating whether a frame node is selected.

Algorithm S2 Molecular Heat Transformer (MHT) Inference

def MHT($\mathbf{f}_{\text{atom}}, \mathbf{f}_{\text{frame}}, \mathbf{f}_{\text{aa}}, \mathbf{f}_{\text{fa}}, \mathbf{S}, N_{\text{blocks}} = 8, K = 8, c = 512, c_{\text{p}} = 64$)

for $i = 1$ to N_{blocks} **do**
  # Pair update block (Line 3-7)
  $\mathbf{f}_{\text{K}}, \mathbf{f}_{\text{Q}}, \mathbf{b} = \text{LinearNoBias}(\mathbf{f}_{\text{frame}}), \text{LinearNoBias}(\mathbf{f}_{\text{frame}}), \text{LinearNoBias}(\mathbf{S})$
  # Computing a normalized affinity matrix for nearest-neighbor frames
  $\mathbf{U}_l = \text{Softmax}_{\text{row-wise}}\left(\frac{1}{\sqrt{c_{\text{p}}}}(\mathbf{f}_{\text{K}} \cdot \mathbf{f}_{\text{Q}}^{\top}) + \mathbf{b} - \text{Inf} \cdot (1 - \mathbf{X}_{\text{s}})\right)$
  # Propagating to all frame-frame pairs via heat kernel expansion
  # Approximating the matrix exponential $\exp(\mathbf{U}) \approx (\mathbf{1} + \frac{1}{K}\mathbf{U})^K$
  $\tilde{\mathbf{U}} = (\mathbf{I} + \frac{1}{K}\mathbf{U}_l)^K$
  # Updating frame-atom pairs using broadcasted kernel matrices
  $\mathbf{g} = \tilde{\mathbf{U}} \cdot \text{LinearNoBias}(\mathbf{f}_{\text{fa}})$
  $\mathbf{f}_{\text{fa}} \leftarrow \text{MLP}(\text{concat}(\mathbf{g}, \mathbf{f}_{\text{fa}})) + \mathbf{f}_{\text{fa}}$
  # Graph attention block (Line 8-12)
  $\mathbf{f}_{\text{node}} = \text{concat}_{\text{column-wise}}(\mathbf{f}_{\text{atom}}, \mathbf{f}_{\text{frame}})$
  $\mathbf{f}_{\text{edge}} = \text{concat}_{\text{column-wise}}(\text{concat}_{\text{row-wise}}(\mathbf{f}_{\text{aa}}, \mathbf{f}_{\text{fa}}^{\top}), \text{concat}_{\text{row-wise}}(\mathbf{f}_{\text{fa}}, \mathbf{S}))$
  $\mathbf{X} = \text{concat}_{\text{column-wise}}(\text{concat}_{\text{row-wise}}(\mathbf{X}_{\text{a}}, \mathbf{1}), \text{concat}_{\text{row-wise}}(\mathbf{1}, \mathbf{X}_{\text{s}}))$
  $\mathbf{f}_{\text{out}}, \_, \mathbf{z}_{\text{out}} = \text{MHAwithEdgeBias}(\mathbf{f}_{\text{node}}, \mathbf{f}_{\text{node}}, \mathbf{f}_{\text{edge}}, \mathbf{X}, n_{\text{heads}} = 8, c_{\text{head}} = 8)$
  $\mathbf{f}_{\text{edge}} \leftarrow \text{MLP}(\text{LinearNoBias}(\mathbf{z}_{\text{out}}) + \mathbf{f}_{\text{edge}}) + \mathbf{f}_{\text{edge}}$
  # Note update block
  $\mathbf{f}_{\text{node}} \leftarrow \text{MLP}_{\text{hidden\_dim=2048}}(\mathbf{f}_{\text{out}} + \mathbf{f}_{\text{node}}) + \mathbf{f}_{\text{node}}$
  Update $\mathbf{f}_{\text{frame}}$, $\mathbf{f}_{\text{fa}}$ from merged embeddings $\mathbf{f}_{\text{node}}$, $\mathbf{f}_{\text{edge}}$
end for
if Subsample ligand pair representations **then**
  $\mathbf{F}_{\text{L}} \leftarrow \frac{1}{2}\left(\mathbf{H}_{\text{L}} \cdot \mathbf{f}_{\text{fa}} + (\mathbf{H}_{\text{L}} \cdot \mathbf{f}_{\text{fa}})^{\top}\right)$ $\quad \mathbf{F}_{\text{L}} \in \mathbb{R}^{N_{\text{L}} \times N_{\text{L}} \times c_{\text{p}}}$
end if
return $\mathbf{f}_{\text{atom}}, \mathbf{f}_{\text{frame}}, \mathbf{f}_{\text{fa}}, \mathbf{f}_{\text{aa}}, \mathbf{F}_{\text{L}}$

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($\mathbf{f}_{\text{src}}, \mathbf{f}_{\text{dst}}, \mathbf{f}_{\text{edge}}, \mathbf{X}, c = 512, c_{\text{p}} = 64, n_{\text{head}}, c_{\text{head}}$) $\quad \mathbf{f}_{\text{src}} \in \mathbb{R}^{N_{\text{src}} \times c}, \mathbf{f}_{\text{dst}} \in \mathbb{R}^{N_{\text{dst}} \times c}, \mathbf{f}_{\text{edge}} \in \mathbb{R}^{N_{\text{edges}} \times c_{\text{p}}}$

$\mathbf{f}_{\text{K}}, \mathbf{f}_{\text{Vs}} = \text{LinearNoBias}(\mathbf{f}_{\text{src}})$ $\quad \mathbf{f}_{\text{K}}, \mathbf{f}_{\text{Vs}} \in \mathbb{R}^{N_{\text{src}} \times n_{\text{heads}} \times c_{\text{heads}}}$
$\mathbf{f}_{\text{Q}}, \mathbf{f}_{\text{Vd}} = \text{LinearNoBias}(\mathbf{f}_{\text{dst}})$ $\quad \mathbf{f}_{\text{Q}}, \mathbf{f}_{\text{Vd}} \in \mathbb{R}^{N_{\text{dst}} \times n_{\text{heads}} \times c_{\text{heads}}}$
$\mathbf{b} = \text{LinearNoBias}(\mathbf{f}_{\text{edge}})$ $\quad \mathbf{b} \in \mathbb{R}^{N_{\text{edges}} \times n_{\text{heads}} \times 1}$
$\mathbf{z}_{ij} = \frac{1}{\sqrt{c_{\text{head}}}}(\mathbf{f}_{\text{K},i} \cdot \mathbf{f}_{\text{Q},j}^{\top}) + \mathbf{b}_{ij}$ $\quad \mathbf{z} \in \mathbb{R}^{N_{\text{edges}} \times n_{\text{heads}} \times 1}$
$\mathbf{a}_{ij} = \text{Softmax}_j^{X_{ij}=1}\, \mathbf{z}_{ij}$ $\quad \mathbf{a} \in \mathbb{R}^{N_{\text{edges}} \times n_{\text{heads}} \times 1}$
$\mathbf{f}_{\text{os},i} \leftarrow \text{LinearNoBias}(\sum_j^{X_{ij}=1} \mathbf{a}_{ij} \odot \mathbf{f}_{\text{Vd},j})$ $\quad \mathbf{f}_{\text{os}} \in \mathbb{R}^{N_{\text{src}} \times c}$
$\mathbf{f}_{\text{od},j} \leftarrow \text{LinearNoBias}(\sum_j^{X_{ij}=1} \mathbf{a}_{ij} \odot \mathbf{f}_{\text{Vs},i})$ $\quad \mathbf{f}_{\text{od}} \in \mathbb{R}^{N_{\text{dst}} \times c}$
return $\mathbf{f}_{\text{os}}, \mathbf{f}_{\text{od}}, \mathbf{z}$

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

def rigidFrom3Points($\mathbf{x}_1 \in \mathbb{R}^3, \mathbf{x}_2 \in \mathbb{R}^3, \mathbf{x}_3 \in \mathbb{R}^3, \varepsilon = 0.01\,\text{Å}$)

$\mathbf{v}_1 \leftarrow \mathbf{x}_3 - \mathbf{x}_2$
$\mathbf{v}_2 \leftarrow \mathbf{x}_1 - \mathbf{x}_2$
$\mathbf{e}_1 \leftarrow \frac{\mathbf{v}_1}{\sqrt{\|\mathbf{v}_1\|_2^2 + \varepsilon^2}}$
$\mathbf{u}_2 \leftarrow \mathbf{v}_2 - (\mathbf{e}_1^\top \mathbf{v}_2)\mathbf{e}_1$
$\mathbf{e}_2 \leftarrow \frac{\mathbf{u}_2}{\sqrt{\|\mathbf{u}_2\|_2^2 + \varepsilon^2}}$
$\mathbf{e}_3 \leftarrow \mathbf{e}_1 \times \mathbf{e}_2$
$\mathbf{R} \leftarrow \text{vstack}(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ $\quad \mathbf{R} \in \mathbb{R}^{3\times 3}$
$\mathbf{t} \leftarrow \mathbf{x}_2$
return $\mathbf{t}, \mathbf{R}$

D.3 MHT model pretraining

Table S4. Composition of the dataset used for pretraining the MHT chemical encoder.

Data source	Num. samples collected	Sampling weight	$\mathscr{L}_{\text{3D}}$	$\mathscr{L}_{\text{REG}}$	$\mathscr{L}_{\text{MLM}}$
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 S4 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:

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

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 $\mathbf{y}$:

$$\mathscr{L}_{\text{3D-marginal}} = \sum_{u}^{N_{\text{frame}}} \sum_{i}^{N_{\text{atom}}} \log \Big[\sum_{l}^{N_{\text{modes}}} \frac{\exp(w_{iul}) \cdot q_{\text{3D}}(T_u^{-1} \circ \mathbf{y}_i | \mathbf{m}_{iul})}{\sum_l^{N_{\text{modes}}} \exp(w_{iul})} \Big] \tag{S22}$$

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

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

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 Å 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

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

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

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

where 3DMixtureDensityHead is a 3-layer MLP.

Using an equivariant graph transformer similar to ESDM (see Sec. G) 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]:

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

where

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

$\mathscr{L}_{\text{CC-regression}}$ is a SmoothL1 loss for fitting the "level 1" chemical checker (CC) [S12] embeddings which represents harmonized and integrated bioactivity data, and $\mathscr{L}_{\text{CC-ismask}}$ is an auxiliary binary cross entropy loss for classifying whether a specific CC entry is available for any molecule in the chemical checker dataset. $\mathscr{L}_{\text{MLM}}$ 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.

E Protein-ligand graph featurization

Table S5. 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. E.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 $\mathbf{f}_{\text{atom}}$
L	N_{ligatm}	All ligand atoms	MHT atomic node embeddings $\mathbf{f}_{\text{atom}}$

Table S6. 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. 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 E.1
BS	local	B	S	Intra-patch, dense	See Section E.1
BF	long-range	B	F	Dense	Node embedding outer sum
SS	long-range	S	S	Dense	See Section E.1
SF	long-range	S	F	Dense	Zero tensors
FF	long-range	F	F	Dense	Symmetrized MHT embeddings $\mathbf{F}_{\text{L}}$
# *Atomic-scale subgraph*					
As	local	P+L	P+L	Exponential-kNN($\mathbf{Z}$), $k_{\text{NN}} = 8$	$\text{RBF}_{32}(\|\mathbf{Z}_{src} - \mathbf{Z}_{dst}\|/10\,\text{Å})$
Ac	local	P+L	P+L	Molecular graph n-hop, n=3	Gathered MHT embeddings $\mathbf{f}_{\text{aa}}$
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 $\mathbf{f}_{\text{fa}}$
FL	local	F	L	Intra-ligand, dense	Gathered MHT embeddings $\mathbf{f}_{\text{fa}}$

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 S5-S6. The protein anchor frame nodes (Table S5 S) are selected by first sequentially segmenting the input protein sequence into $N_{\text{P}} = 96$ 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 S6 BS) then connect each protein residue node to the anchor node within the same patch.

The geometry-dependent local edges (Table S6 BB, As) are generated using a randomized k-nearest-neighbor (kNN) scheme with an exponentially-decaying attachment probability $p(\text{add\_edge}(i,j)) \propto \exp(-\|\mathbf{Z}_i - \mathbf{Z}_j\|/5.0\,\text{Å})$ 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 $D = 5.0\,\text{Å}$

for $i \in \{1, \cdots, N_{\text{nodes}}\}$ **do**
  for $j \in \{1, \cdots, N_{\text{nodes}}\}$ **do**
    $s_{ij} \leftarrow -\|\mathbf{Z}_i - \mathbf{Z}_j\|/D$
    $u_{ij} \sim \text{Uniform}(0,1)$
    $\tilde{s}_{ij} \leftarrow s_{ij} - \log(-\log(u_{ij}))$ ▷ Gumbel distribution over logits s_{ij}
  end for
  $\{src\_idx\}_i, \{dst\_idx\}_i \leftarrow \text{ArgTopK}_j(\{\tilde{s}_{ij}\}_i), i$ ▷ Row-wise top-k node indices, k=k_{NN}
end for
return $\{src\_idx\}, \{dst\_idx\}$

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

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

The inter-molecular covalent edges (Table S6 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.

E.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 [S2] 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: $\mathbb{R}^{N_{\text{res}} \times (1280+21+37*3)} \to \mathbb{R}^{N_{\text{res}} \times c}$.

As detailed in Algorithm S6, the initial edge features of all protein residue-residue edges (Table S6 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 $\mathbf{f}_{\text{B}}$, input protein coordinates $\mathbf{x}_t$, input template coordinates $\mathbf{x}_{\text{template}}$ (optional)

$\mathbf{f}_{\text{osum}} = (\mathbf{f}_{\text{B}})_{src\_idx} + (\mathbf{f}_{\text{B}})_{dst\_idx}$ — Eq. S28
$\mathbf{f}_{\text{seq}} = \text{RBF}_{16}(residue\_idx(src\_idx) - residue\_idx(dst\_idx)) \cdot \text{IsSameChain}(src\_idx, dst\_idx)$
$\mathbf{f}_{\text{geom}} = \text{RelGeomEnc}(src\_idx, dst\_idx, \mathbf{x}_t, N_{\text{basis}} = 15)$ — Alg. S7
$\mathbf{f}_{\text{edge}} \leftarrow \text{MLP}(\text{concat}(\mathbf{f}_{\text{osum}}, \mathbf{f}_{\text{seq}}, \mathbf{f}_{\text{geom}}))$
if $\mathbf{x}_{\text{template}}$ is not None **then**
    $\mathbf{f}_{\text{edge}} \leftarrow \mathbf{f}_{\text{edge}} + \text{LinearNoBias}(\text{RelGeomEnc}(src\_idx, dst\_idx, \mathbf{x}_{\text{template}}, N_{\text{basis}} = 15))$ — Alg. S7
end if
return $\mathbf{f}_{\text{edge}}$

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

def $\text{RelGeomEnc}(src\_idx, dst\_idx, \mathbf{x}, N_{\text{basis}}, D_0 = 10.0\,\text{Å})$

$\{\mathbf{x}_{\text{N}}, \mathbf{x}_{\text{C}\alpha}, \mathbf{x}_{\text{C}}\} \leftarrow$ Get protein backbone (N, Cα, C) coordinates from $\mathbf{x}$
$\mathbf{t}_{\text{src}}, \mathbf{R}_{\text{src}} = \text{rigidFrom3Points}((\mathbf{x}_{\text{N}})_{src\_idx}, (\mathbf{x}_{\text{C}\alpha})_{src\_idx}, (\mathbf{x}_{\text{C}})_{src\_idx})$ — Alg. S4
$\mathbf{t}_{\text{dst}}, \mathbf{R}_{\text{dst}} = \text{rigidFrom3Points}((\mathbf{x}_{\text{N}})_{dst\_idx}, (\mathbf{x}_{\text{C}\alpha})_{dst\_idx}, (\mathbf{x}_{\text{C}})_{dst\_idx})$ — Alg. S4
$d = \|\mathbf{t}_{\text{src}} - \mathbf{t}_{\text{dst}}\|$
$\mathbf{f}_{\text{dist}} = \text{RBF}_{N_{\text{basis}}}(d / D_0)$
$\mathbf{f}_{\text{dirs}} = \mathbf{R}_{\text{src}}^\top \cdot (\mathbf{t}_{\text{dst}} - \mathbf{t}_{\text{src}}) / (d + 1.0\,\text{Å})$
$\mathbf{f}_{\text{dird}} = \mathbf{R}_{\text{dst}}^\top \cdot (\mathbf{t}_{\text{src}} - \mathbf{t}_{\text{dst}}) / (d + 1.0\,\text{Å})$
$\mathbf{f}_{\text{ori}} = \text{flatten}(\mathbf{R}_{\text{src}}^\top \cdot \mathbf{R}_{\text{dst}})$
$\mathbf{f}_{\text{geom}} = \text{concat}(\mathbf{f}_{\text{dist}}, \mathbf{f}_{\text{dirs}}, \mathbf{f}_{\text{dird}}, \mathbf{f}_{\text{ori}})$ — $\mathbf{f}_{\text{geom}} \in \mathbb{R}^{N_{\text{basis}}+15}$
return $\mathbf{f}_{\text{geom}}$

F 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 $\mathbf{f}_{\text{SF}} \in \mathbb{R}^{N_{\text{P}} * N_{\text{L}} \times c_{\text{P}}}$.

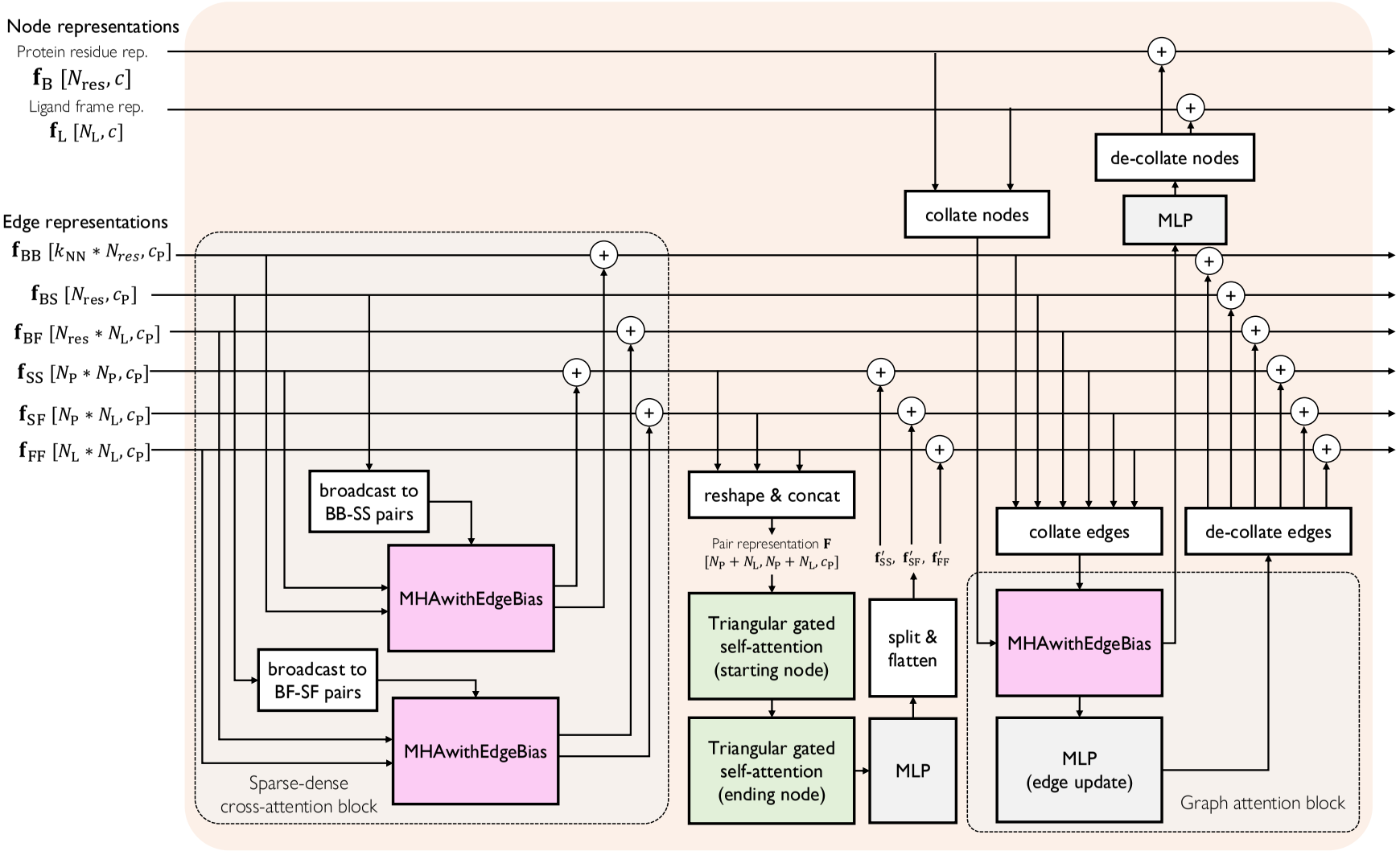


Figure S1. Network architecture of a single contact prediction module (CPM) block. Arrows indicate information flow directions, and "+" indicates an element-wise tensor summation. The Triangular gated self-attention (around starting/ending node) blocks refer to Algorithm 13-14 of Ref.[S6]. k_{NN} denotes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

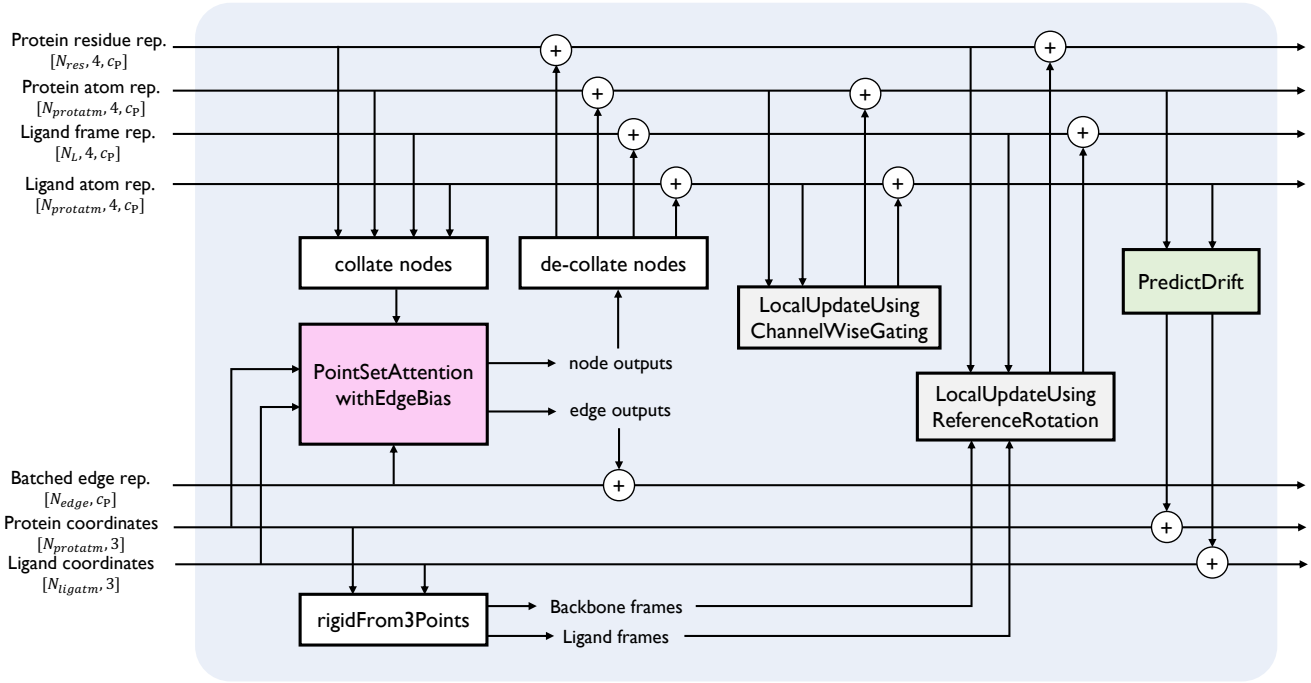


Figure S2. Network architecture of a single block in the equivariant structure diffusion module (ESDM). Arrows indicate information flow directions, and "+" indicates an element-wise tensor summation.

The first 2 CPM blocks are executed on the protein sub-graph only (i.e., BB, BS, and SS edges as defined in Table S6), 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 S6).

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

G 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 $N_{\text{ESDM}} = 4$ 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 $\text{RBF}(\tau)$ 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($\mathbf{f}_{\text{s}}, \mathbf{f}_{\text{v}}, \mathbf{f}_{\text{e}}, \mathbf{t}$, $c = 64$) $\quad \mathbf{f}_{\text{s}} \in \mathbb{R}^{N_{\text{nodes}} \times c}, \mathbf{f}_{\text{v}} \in \mathbb{R}^{N_{\text{nodes}} \times 3 \times c}, \mathbf{f}_{\text{e}} \in \mathbb{R}^{N_{\text{edges}} \times c}, \mathbf{t} \in \mathbb{R}^{N_{\text{nodes}} \times 3}$

$\mathbf{f}_{\text{Q}}, \mathbf{f}_{\text{K}}, \mathbf{f}_{\text{V}} = \text{LinearNoBias}(\mathbf{f}_{\text{s}})$ $\quad \mathbf{f}_{\text{Q}}, \mathbf{f}_{\text{K}}, \mathbf{f}_{\text{V}} \in \mathbb{R}^{N_{\text{nodes}} \times n_{\text{heads}} \times c_{\text{head}}}$

$\mathbf{t}_{\text{Q}}, \mathbf{t}_{\text{K}}, \mathbf{t}_{\text{V}} = \mathbf{t}/D_{\text{points}} + \text{LinearNoBias}(\mathbf{f}_{\text{v}})$ $\quad \mathbf{t}_{\text{Q}}, \mathbf{t}_{\text{K}}, \mathbf{t}_{\text{V}} \in \mathbb{R}^{N_{\text{nodes}} \times n_{\text{heads}} \times c_{\text{point}} \times 3}, D_{\text{points}} = 10\,\text{Å}$

$\mathbf{b} = \text{LinearNoBias}(\mathbf{f}_{\text{e}})$ $\quad \mathbf{b} \in \mathbb{R}^{N_{\text{edges}} \times c_{\text{heads}}}$

 # Computing attention weights on all edges of the graph

$\mathbf{z}_{ij} = \frac{1}{\sqrt{c_{\text{head}}}}(\mathbf{f}_{\text{Q},i}^{\top} \cdot \mathbf{f}_{\text{K,j}}) + \mathbf{b}_{ij} - \frac{\mathbf{w}_{ij}}{\sqrt{18 c_{\text{head}}}} \|\mathbf{t}_{\text{Q},i} - \mathbf{t}_{\text{K},j}\|_2^2$ $\quad \mathbf{z} \in \mathbb{R}^{N_{\text{edges}} \times n_{\text{heads}}}$

$\boldsymbol{\alpha}_{ij} = \text{Softmax}_{j \in \{i\}}(\mathbf{z}_{ij})$

$\mathbf{f}'_{\text{s}} \leftarrow \sum_{j \in \{i\}} \boldsymbol{\alpha}_{ij} \odot \mathbf{f}_{\text{V}}$

$\mathbf{f}'_{\text{v}} \leftarrow (\sum_{j \in \{i\}} \boldsymbol{\alpha}_{ij} \odot \mathbf{t}_{\text{V}}) - \mathbf{t}/D_{\text{points}}$

$\mathbf{f}'_{\text{e}} \leftarrow \text{MLP}(\mathbf{z}_{ij}) + \mathbf{f}_{\text{e}}$

return $\mathbf{f}'_{\text{s}}, \mathbf{f}'_{\text{v}}, \mathbf{f}'_{\text{e}}$

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

Algorithm S9 LocalUpdateUsingChannelWiseGating

def LocalUpdateUsingChannelWiseGating($\mathbf{f}_\mathrm{s}, \mathbf{f}_\mathrm{v}$) $\quad \mathbf{f}_\mathrm{s} \in \mathbb{R}^{N_\mathrm{nodes} \times c}, \mathbf{f}_\mathrm{v} \in \mathbb{R}^{N_\mathrm{nodes} \times 3 \times c}$

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

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

Algorithm S10 LocalUpdateUsingReferenceRotation

def LocalUpdateUsingReferenceRotation($\mathbf{f}_\mathrm{s}, \mathbf{f}_\mathrm{v}, \mathbf{R}$) $\quad \mathbf{f}_\mathrm{s} \in \mathbb{R}^{N_\mathrm{nodes} \times c}, \mathbf{f}_\mathrm{v} \in \mathbb{R}^{N_\mathrm{nodes} \times 3 \times c}, \mathbf{R} \in \mathrm{SO}(3)^{N_\mathrm{nodes}}$

1: $(\mathbf{f}_\mathrm{vloc})_i = (\mathbf{R}_i)^\mathrm{T} \cdot (\mathbf{f}_\mathrm{v})_i$ $\quad i \in \{1, \cdots, N_\mathrm{nodes}\}$
2: $\mathbf{f}_\mathrm{loc} = \mathrm{concat}(\mathbf{f}_\mathrm{s}, \mathbf{f}_\mathrm{vloc}, \|\mathbf{f}_\mathrm{v}\|_2)$ $\quad \mathbf{f}_\mathrm{loc} \in \mathbb{R}^{N_\mathrm{nodes} \times 5c}$
3: $\mathbf{f}'_\mathrm{s}, \mathbf{f}_\mathrm{vloc} \leftarrow \mathrm{MLP}(\mathbf{f}_\mathrm{loc})$
4: $(\mathbf{f}'_\mathrm{v})_i \leftarrow \mathbf{R}_i \cdot (\mathbf{f}_\mathrm{vloc})_i$ $\quad i \in \{1, \cdots, N_\mathrm{nodes}\}$
5: **return** $\mathbf{f}'_\mathrm{s}, \mathbf{f}'_\mathrm{v}$

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 $\mathbf{f}_\mathrm{vloc}$ 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($\mathbf{f}_\mathrm{s}, \mathbf{f}_\mathrm{v}$) $\quad \mathbf{f}_\mathrm{s} \in \mathbb{R}^{N_\mathrm{nodes} \times c}, \mathbf{f}_\mathrm{v} \in \mathbb{R}^{N_\mathrm{nodes} \times 3 \times c}$

1: $\mathbf{o}_\mathrm{scale} \leftarrow \mathrm{Softplus}(\mathrm{MLP}(\mathbf{f}_\mathrm{s}))$ $\quad \mathbf{o}_\mathrm{scale} \in \mathbb{R}^{N_\mathrm{nodes} \times 1}$
2: $\Delta\mathbf{t} \leftarrow \mathrm{LinearNoBias}(\mathbf{f}_\mathrm{v}) \odot \mathbf{o}_\mathrm{scale}$ $\quad \Delta\mathbf{t} \in \mathbb{R}^{N_\mathrm{nodes} \times 3}$
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.

H Confidence estimation heads

The predicted lDDT (pLDDT) head of NeuralPLexer uses the same architecture as ESDM (Sec. G) 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 $|\|(\mathbf{Z}_0)_i - (\mathbf{Z}_0)_j\| - \|(\mathbf{Z}_\mathrm{ref})_i - (\mathbf{Z}_\mathrm{ref})_j\||$ for all target point j within 15.0 Å of the query point in the reference structure $\mathbf{Z}_\mathrm{ref}$. 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:

$$pLDDT[i] \coloneqq 1.0 * p_{0-0.5}[i] + 0.75 * p_{0.5-1.0}[i] + 0.5 * p_{1.0-2.0}[i] + 0.25 * p_{2.0-4.0}[i]. \tag{S29}$$

Table S7. 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
is_rigid_receptor	no	no	yes	yes (cropped binding site)
train_plddt_head	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

I 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 $\mathbf{v}_{\text{bins}}$. $\mathscr{L}_{\text{CE}}(\cdot,\cdot)$ denotes a standard cross entropy loss between two discrete distributions: $\mathscr{L}_{\text{CE}}(\mathbf{p},\mathbf{q}) \coloneqq -\sum_{i=1}^{N_{\text{bins}}} \log p_i \cdot q_i$.

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; $\text{FAPE}_{\text{scaled}}$ 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.

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

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

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

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 $\mathscr{L}_{\text{clash}}$ is defined as

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

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 S7 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 $\mathbf{Z}_{\text{ref}}$, *is_rigid_receptor*, *train_plddt_head*, loss weights $p_{\text{contact}}, \lambda_{\text{FAPE}}, \lambda_{\text{dRMSD}}, \lambda_{\text{viol}}$

for $i \in \{1, \cdots, N_{\text{training\_iter}}\}$ **do**
  if *is_rigid_receptor* **then**
    $\mathbf{x}_{\text{template}} \leftarrow \mathbf{x}_{\text{ref}}$
  else
    use_template $\sim \text{Bern}(0.5)$
    if *use_template* **then**
      $\mathbf{x}_{\text{template}} \leftarrow$ Retrieve a random structure from all AF2 predictions and PL2019-74k structures of $\{\mathbf{s}\}$
    else
      $\mathbf{x}_{\text{template}} \leftarrow$ None
    end if
  end if
  if not (*train_plddt_head* and $(i \mid 10)$) **then**
    Compute MHT embeddings for all input ligand molecular graphs in $\{G\}$ and standard amino acids Algorithm S2
    $N_{\text{P}} \leftarrow \min(N_{\text{res}}, 96)$, $N_{\text{L}} \sim \{\text{floorRound}(\sqrt{N_{\text{frames}}}), \cdots, 33, 32\}$
    is_prior_training $\sim \text{Bern}(p_{\text{contact}})$
    if *is_prior_training* **then**
      $\tau \leftarrow 1.0$, $k \sim \{0, \cdots, N_{\text{L}} - 1\}$
    else
      $\tau \sim \text{Uniform}(0, 1)$, $k \leftarrow N_{\text{L}}$
    end if
    $t \leftarrow \text{DiffusionTimeSchedule}(\tau)$ Equation (8)
    $\mathbf{z}_t \leftarrow$ Perturb the structure using the forward SDE transition kernel $q_{t:0}(\cdot|\mathbf{Z}_{\text{ref}})$ Equation (9)
    residue_graph, *atomic_graph* $\leftarrow$ Generate the residue-scale and atomic-scale graph based on $\mathbf{Z}_t$ Section E
    $\mathbf{D}, \mathbf{L} \leftarrow$ Compute ground-truth distance and contact map based on $\mathbf{Z}_{\text{ref}}$ Equation (3), $\mathbf{D}, \mathbf{L} \in \mathbb{R}^{N_{\text{res}} \times N_{\text{L}}}$
    $\tilde{\mathbf{I}} \leftarrow \mathbf{0}$
    for $r \in \{1, \cdots, k\}$ **do**
      $\mathbf{L}_r \leftarrow \text{SumIntoPatches}(L) \odot [1 - \max_{\text{column-wise}}(\tilde{\mathbf{I}})]$ $\mathbf{L}_r \in \mathbb{R}^{N_{\text{P}} \times N_{\text{L}}}$
      $\mathbf{l}_r = \text{OneHot}(i_r, j_r);\ (i_r, j_r) \sim \text{Categorical}_{N_{\text{P}} \times N_{\text{L}}}(\mathbf{L}_r)$ $\mathbf{l}_r \in \{0,1\}^{N_{\text{P}} \times N_{\text{L}}}$
      $\tilde{\mathbf{I}} \leftarrow \tilde{\mathbf{I}} + \mathbf{l}_r$ $\tilde{\mathbf{I}} \in \{0,1\}^{N_{\text{P}} \times N_{\text{L}}}$
    end for
    residue_graph $\leftarrow \text{CPMForward}(\textit{residue\_graph}, \tilde{\mathbf{I}}, \tau)$ Section F
    $\hat{P} = \text{LinearNoBias}(\text{MLP}(\text{GetEdges}_{\text{BF}}(\textit{residue\_graph\_k})))$ $\hat{P} \in \mathbb{R}^{N_{\text{res}} \times N_{\text{L}} \times N_{\text{bins}}}, N_{\text{bins}} = 32$
    graph_rep $\leftarrow \text{Collate}(\textit{residue\_graph}, \textit{atomic\_graph}, \textit{cross\_scale\_graph})$ Table S6
    $\hat{\mathbf{Z}}_0 \leftarrow \text{ESDMForward}(\mathbf{Z}_t; \textit{graph\_rep}, \tau)$ Section G
    # Distogram and contact prediction losses
    $\mathscr{L}_{\text{dgram}} = \text{mean}_{A,J}\, \mathscr{L}_{\text{CE}}(\hat{P}_{AJ}, \text{bucketize\_onehot}(D_{AJ}, \mathbf{v}_{\text{bins}}))$
    $\mathscr{L}_{\text{cmap}} = \mathscr{L}_{\text{CE}}(\text{DistogramToContactMap}(\hat{P}), \mathbf{L})$
    # Invariant denoising structure prediction losses
    $\mathscr{L}_{\text{FAPE}} = \text{FAPE}(\hat{\mathbf{Z}}_0, \mathbf{Z}_{\text{ref}}) + \text{FAPE}_{\text{scaled}}(\hat{\mathbf{Z}}_0, \mathbf{Z}_{\text{ref}})$
    $\mathscr{L}_{\text{dRMSD}} = \text{dRMSD}_{\text{global}}(\hat{\mathbf{Z}}_0, \mathbf{Z}_{\text{ref}}) + \text{dRMSD}_{\text{site}}(\hat{\mathbf{Z}}_0, \mathbf{Z}_{\text{ref}}) + \text{dRMSD}_{\text{pli}}(\hat{\mathbf{Z}}_0, \mathbf{Z}_{\text{ref}}) + \text{dRMSD}_{\text{weighted}}(\hat{\mathbf{x}}_0, \mathbf{x}_{\text{ref}}; \mathbf{x}_{\text{template}})$
    $\lambda(t) = \frac{\sqrt{\sigma(t)^2 + \sigma_{\text{data}}^2}}{\sigma(t) \cdot \sigma_{\text{data}}},\ \sigma(t) = \sigma\sqrt{t}$ Loss weighting adapted from Ref. [S15], $\sigma = 12.25\,\text{Å}, \sigma_{\text{data}} = 5.0\,\text{Å}$
    # Distance-geometry and clash regularizers
    $\mathscr{L}_{\text{distgeom}}, \mathscr{L}_{\text{clash}} \leftarrow$ Compute structure violation losses See Equations (S30)-(S31)
    $\mathscr{L} \leftarrow 0.1 \cdot (\mathscr{L}_{\text{dgram}} + \mathscr{L}_{\text{cmap}}) + \lambda(t)\lambda_{\text{FAPE}} \cdot \mathscr{L}_{\text{FAPE}} + \lambda(t)\lambda_{\text{dRMSD}} \cdot \mathscr{L}_{\text{dRMSD}} + \lambda(t)\lambda_{\text{viol}} \cdot (\mathscr{L}_{\text{distgeom}} + 0.1 \cdot \mathscr{L}_{\text{clash}})$
    $\mathscr{L} \leftarrow \textit{is\_prior\_training} \cdot (\mathscr{L}_{\text{dgram}} + \mathscr{L}_{\text{cmap}}) + (1 - 0.9 \cdot \textit{is\_prior\_training}) \cdot \mathscr{L}$
  else
    # Confidence estimation losses
    $\mathbf{Z}_0, (\textit{pLDD\_gram}, \_, \_) \leftarrow \text{NeuralPLexerInference}(\{\mathbf{s}\}, \{G\}, 1, N_{\text{steps}} = 10, \textit{use\_template}, \text{True})$ Alg. S1
    LDD_gram $\leftarrow$ Compute reference distance deviation histograms between $\mathbf{Z}_0$ and $\mathbf{Z}_{\text{ref}}$ See Section H text
    $\mathscr{L} \leftarrow \text{mean}_{A \in \{\mathbf{s}\}} \mathscr{L}_{\text{CE}}(\textit{pLDD\_gram}_A, \textit{LDD\_gram}_A) + \text{mean}_{J \in \{G\}} \mathscr{L}_{\text{CE}}(\textit{pLDD\_gram}_J, \textit{LDD\_gram}_J)$
  end if
  Computing gradients w.r.t. $\mathscr{L}$ and update model parameters
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.

J 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. For the GPU times reported in Table 1, NeuralPLexer model inferences are performed on a single NVIDIA RTX A5000 GPU; to facilitate a direct comparison with existing methods, we normalized the inference GPU time by the ratio between the average per-pose wall-clock time of running DiffDock [S20] under the same hardware configuration and the time reported in Ref. [S20].

RosettaLigand [S21] runs are launched with a configuration modified from the standard protocol to enable sufficient relaxation of the receptor conformation. 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 [S22], with randomized torsion angles using the BCL [S23] 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.

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`.

K Additional discussions

K.1 Background and related works

We note that in the past decades, several schemes have been proposed to remedy issues regarding sampling protein conformational changes and slow motions. Those include methods based on molecular dynamics simulation and enhanced sampling techniques [S25–S28], molecular docking guided by template-based modeling [S29–S31] and iterative refinement protocols [S32–S35], as well as recently proposed modifications to structure prediction networks [S36–S39]. However, such methods often require case-specific expert interventions or constraints from experimental data, and are still not a unified 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 for which the ligand identity cannot be inferred from protein motifs.

NeuralPLexer is a deep generative model with a model architecture deeply informed by biophysical inductive biases. As supported by our empirical results, we claim that the integration of these inductive biases into the deep neural network, which combines both auto-regressive and diffusion-generative modeling, is a 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.

K.2 Future extension of the method

Although the current presentation of NeuralPLexer is trained solely on structural data, we identify a weak statistical correlation between the model-assigned confidence scores and experimental binding affinities (Supplementary Information, K.5) suggesting that the model has learned representations related to binding strength. Therefore, we hypothesize that fine-tuning the model for affinity prediction in a supervised manner may yield better performance compared to standard approaches. Furthermore, comprehensively analyzing the compound dependence of model confidence estimations (such as ligand pLDDT) against

empirical and physical measures including force fields and docking score functions, with potential calibrations, may yield improved workflows for compound prioritization and virtual screening. We consider those as promising directions for future studies.

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 [S40] and high-throughput mass spectrometry signals [S41], 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 [S6, S42, S43]. 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 [S26].

K.3 Q-factor analysis

Apart from standard Cα-based metrics such as TM-score, we introduce a weighted version of the Q-factor from Best and Hummer [S44] based on the inter-residue native contacts that are not conserved between two distinct-state reference structures (Methods, Structure accuracy metrics). 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 a more marked improvement from AF2 (average TM-score=0.927) to NeuralPLexer (average TM-score=0.942) when evaluated on the subset at which model is most confident (protein pLDDT > 0.8).

K.4 Template dependency of generated structures

Using the test dataset of recently determined structures, we plot the TM-scores computed against AF2 template (x-axis) and the ground-truth structure from PDB (y-axis), for structures generated without template (red) and with the template input (blue). As shown in Figure S3a, the predictions are generally not biased to the AlphaFold2 template structures (blue dots), as an exact match to the templates would result in a vertical line of TM-score=1.0 with respect to the AF2 structures.

In Figure S3b, we also found that when compared to the predictions for when no template structure input is used, the relative structure similarity with respect to AF2 structures and the reference structures (as manifested by the TM-score difference $\Delta\text{TMscore} = \text{TMscore}_{\text{true}} - \text{TMscore}_{\text{AF2}}$) is not substantially shifted.

K.5 Relationship between ligand structure confidence score and binding affinity

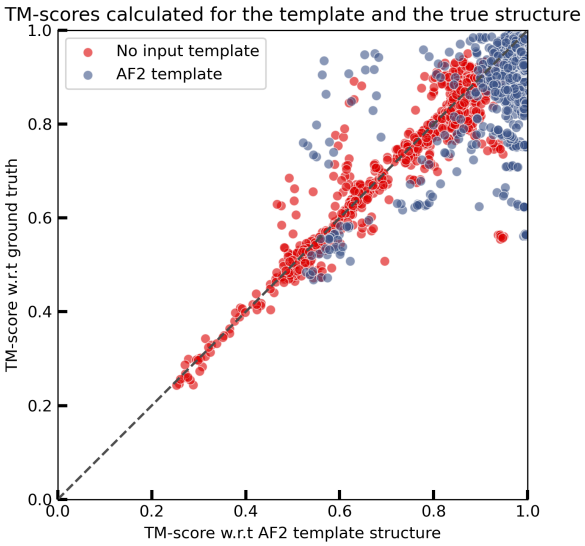
In Figures S4-S5, we examine the relationship between the model-assigned confidence score (pLDDT), ligand RMSD relative to the true structure, and the experimental binding affinity values reported in the PDBBind2020 database [S45]. In Figure S4 we have plotted the experimental binding affinity value for all 363 protein-ligand pairs in the PDBBind2020 test dataset against the ligand pLDDT score and ligand RMSD, grouped over the experiment type (Ki, Kd, or IC50 assays). For the Ki subset and the IC50 subset, we observe a weak positive statistical correlation between both (a) ligand pLDDT and (b) ligand RMSD against the negative log of the binding affinities (Spearman r $\approx$ 0.2).

Moreover, in Figure S5 we have plotted the ligand pLDDT scores binned over intervals of binding affinities of the corresponding protein-ligand pairs. We note that for all three assay types, protein-ligand pairs within the lowest affinity and highest affinity bins are consistently assigned with lower and higher pLDDT scores, repsectively, compared to samples in the middle of the distribution.

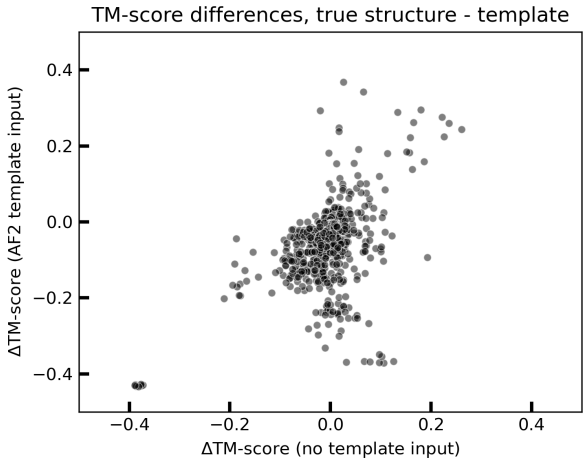
Therefore, we conclude that while the correlation between NeuralPLexer confidence head and binding affinity is not strong enough to be used for virtual screening, the results do suggest that the model has learned representations related to affinity by solely training on structures. We note that fine-tuning the model for affinity prediction in a supervised manner may yield better performance compared to standard ML-based affinity prediction models, and we consider that as a promising direction for future studies.

K.6 Relationship between ligand and binding site structure prediction accuracy

In the data presented in Figure 4h and Figure 5e, we recognize the absence of a robust correlation between ligand RMSD and lDDT-BS. However, within both the PocketMiner dataset and the recently determined structures dataset, we note that a high lDDT-BS is a precondition to achieve a low ligand RMSD, as supported by the vacancy in the lower-left regions of both

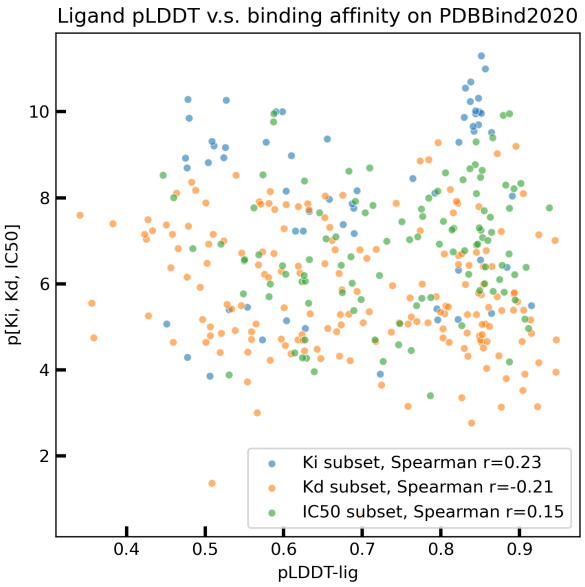


(a) Scatter plots of TM-scores calculated for both the template structures and the corresponding true structures.



(b) Scatter plots of the relative TM-score differences calculated for both the template structures and the corresponding true structures. X-axis: ΔTMscores computed using template-free NeuralPLexer predictions. Y-axis: ΔTMscores computed using template-based NeuralPLexer predictions.

Figure S3. Analyzing the dependence of predicted structures on templates.



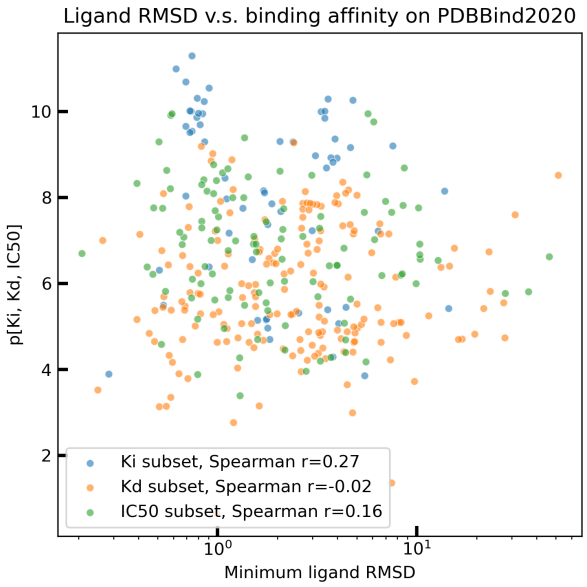


Figure S4. Scatter plot of the negative log of binding affinity as reported in the PDBBind2020 dataset (y-axis) against (a) the average ligand pLDDT score of each protein-ligand pair and (b) the minimum ligand RMSD with respect to the true structure (x-axis), computed over all 363 samples in the holdout test set.

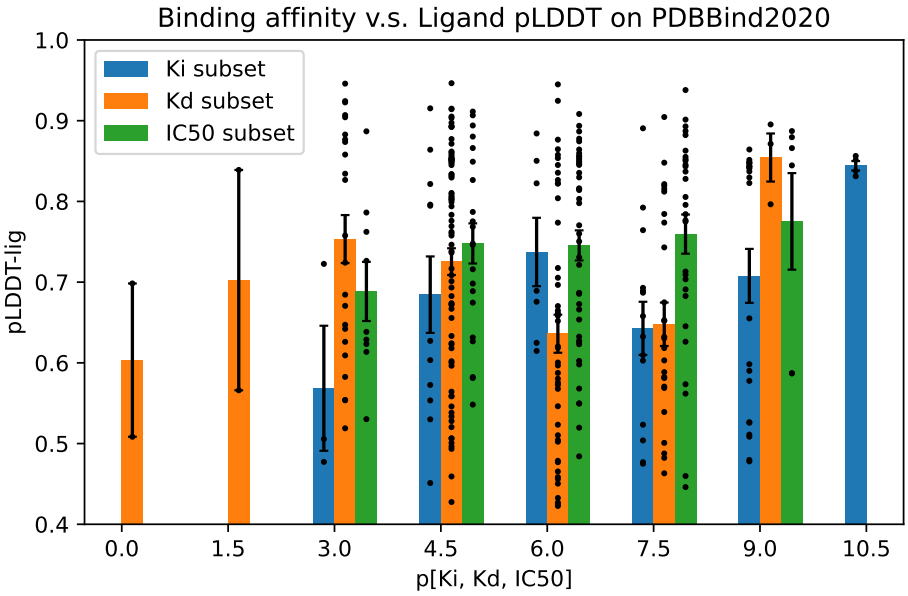


Figure S5. Bar charts of the average ligand pLDDT scores grouped over the binding affinity of the corresponding protein-ligand pair in the PDBBind2020 dataset. Statistics are computed over (a) the subset for which the original affinity data are Ki values (n=61), (b) the subset for which the original affinity data are Kd values (n=183), and (c) the subset for which the original affinity data are IC50 values (n=107). Bar heights indicate the mean value for each bin and error bars indicate the standard error of the mean.

Figure 4h and Figure 5e. Several contributing factors could explain the instances where NeuralPLexer predictions yield high ligand RMSDs alongside high lDDT-BS scores, including (a) the misidentification of binding sites on correctly predicted target structures, (b) less ideal accuracy in binding pose generation, or (c) critical deviations in the predicted binding site shape despite a good overall geometrical accuracy caused by either the misprediction of important residue conformations or sensitivity in the scoring function to prediction errors, as discussed in Ref. [S46]. Results from Figure 3e-g also suggest that improved metrics that systematically account for important physical interactions are necessary to evaluate the fidelity of predicted binding sites, compared to geometrical measurements such as lDDT and RMSD.

We hypothesize that for generative-model-based structure prediction methods minor errors in the predicted binding pockets may not substantially result in decreased pose accuracy, unlike in traditional docking methods. Systematically delineating these underlying factors with comparisons to the statistics in traditional docking methods discussed above will facilitate our understanding of the generalization behavior of generative-model-based structure predictors over different target classes, and we consider that as a promising future research direction.

References

[S1] Jiaming Song, Chenlin Meng, and Stefano Ermon. "Denoising diffusion implicit models". In: *arXiv preprint arXiv:2010.02502* (2020).

[S2] Zeming Lin et al. "Evolutionary-scale prediction of atomic-level protein structure with a language model". In: *Science* 379.6637 (2023), pp. 1123–1130.

[S3] Dan Hendrycks and Kevin Gimpel. "Gaussian error linear units (gelus)". In: *arXiv preprint arXiv:1606.08415* (2016).

[S4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. "Layer normalization". In: *arXiv preprint arXiv:1607.06450* (2016).

[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. 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.

[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] Gabriele Corso et al. "Diffdock: Diffusion steps, twists, and turns for molecular docking". In: *arXiv preprint arXiv:2210.01776* (2022).

[S21] 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.

[S22] 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.

[S23] 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.

[S24] Noel M O'Boyle et al. "Open Babel: An open chemical toolbox". In: *Journal of cheminformatics* 3.1 (2011), pp. 1–14.

[S25] 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.

[S26] 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.

[S27] Yuqi Zhang et al. “Benchmarking Refined and Unrefined AlphaFold2 Structures for Hit Discovery”. In: (2022).

[S28] Bodhi P Vani et al. “From sequence to Boltzmann weighted ensemble of structures with AlphaFold2-RAVE”. In: *bioRxiv* (2022), pp. 2022–05.

[S29] 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.

[S30] 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.

[S31] Maarten L Hekkelman et al. “AlphaFill: enriching AlphaFold models with ligands and cofactors”. In: *Nature Methods* 20.2 (2023), pp. 205–213.

[S32] 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.

[S33] 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.

[S34] Marta Amaral et al. “Protein conformational flexibility modulates kinetics and thermodynamics of drug binding”. In: *Nature communications* 8.1 (2017), pp. 1–14.

[S35] Jue Wang et al. “Scaffolding protein functional sites using deep learning”. In: *Science* 377.6604 (2022), pp. 387–394.

[S36] Richard A Stein and Hassane S Mchaourab. “Modeling alternate conformations with alphafold2 via modification of the multiple sequence alignment”. In: *bioRxiv* (2021).

[S37] Diego Del Alamo et al. “Sampling alternative conformational states of transporters and receptors with AlphaFold2”. In: *Elife* 11 (2022), e75751.

[S38] Lim Heo and Michael Feig. “Multi-State Modeling of G-protein Coupled Receptors at Experimental Accuracy”. In: *Proteins: Structure, Function, and Bioinformatics* (2022).

[S39] 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.

[S40] 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.

[S41] Ilaria Piazza et al. “A map of protein-metabolite interactions reveals principles of chemical communication”. In: *Cell* 172.1-2 (2018), pp. 358–372.

[S42] Sergey Ovchinnikov et al. “Protein structure determination using metagenome sequence data”. In: *Science* 355.6322 (2017), pp. 294–298.

[S43] 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.

[S44] 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.

[S45] Renxiao Wang et al. “The PDBbind database: methodologies and updates”. In: *Journal of medicinal chemistry* 48.12 (2005), pp. 4111–4119.

[S46] Masha Karelina, Joseph J. Noh, and Ron O. Dror. “How accurately can one predict drug binding modes using AlphaFold models?” In: (Aug. 2023). DOI: 10.7554/elife.89386.1. URL: https://doi.org/10.7554/elife.89386.1.