NeuralPLexer3: Accurate Biomolecular Complex Structure Prediction with Flow Models

Zhuoran Qiao[*]
Iambic Therapeutics
San Diego, CA 92121

Feizhi Ding[*]
Iambic Therapeutics
San Diego, CA 92121

Thomas Dresselhaus[*]
Iambic Therapeutics
San Diego, CA 92121

Mia A. Rosenfeld[*]
Iambic Therapeutics
San Diego, CA 92121

Xiaotian Han[*]
Iambic Therapeutics
San Diego, CA 92121

Owen Howell
Iambic Therapeutics
San Diego, CA 92121

Aniketh Iyengar
Iambic Therapeutics
San Diego, CA 92121

Stephen Opalenski
Iambic Therapeutics
San Diego, CA 92121

Anders S. Christensen
Iambic Therapeutics
San Diego, CA 92121

Sai Krishna Sirumalla
Iambic Therapeutics
San Diego, CA 92121

Frederick R. Manby
Iambic Therapeutics
San Diego, CA 92121

Thomas F. Miller III
Iambic Therapeutics
San Diego, CA 92121

Matthew Welborn[*†]
Iambic Therapeutics
San Diego, CA 92121

Abstract

Biomolecular structure determination is essential to a mechanistic understanding of diseases and the development of novel therapeutics. Machine-learning-based structure prediction methods have made significant advancements by computationally predicting protein and bioassembly structures from sequences and molecular topology alone. Despite substantial progress in the field, challenges remain to deliver structure prediction models to real-world drug discovery. Here, we present NeuralPLexer3 – a physics-inspired flow-based generative model that achieves state-of-the-art prediction accuracy on key biomolecular interaction types and improves training and sampling efficiency compared to its predecessors and alternative methodologies [1, 2]. Examined through existing and new benchmarks, NeuralPLexer3 excels in areas crucial to structure-based drug design, including blind docking, physical validity, and ligand-induced protein conformational changes.

1 Introduction

For decades, predicting the 3D structures of proteins has been a transformative goal in structural biology and drug discovery. Experimental techniques like X-ray crystallography and cryo-electron microscopy have provided invaluable structural data. Still, these methods are resource-intensive and time-consuming, making it challenging to scale their application across the immense diversity of

[*]Core contributors.

[†]Correspondence to: matt@iambic.ai.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

proteins and small molecules. Most drug development programs still rely on these experimental structures, leaving countless therapeutic opportunities and hypotheses unexplored.

Breakthroughs in AI-driven structure prediction, notably with AlphaFold2 (AF2) [3], brought the field closer to experimental accuracy. Despite this progress, a significant challenge remained: accurate modeling of interactions between proteins and different biomolecules, particularly small molecules, nucleic acids, and other proteins. Understanding and accurately modeling these interactions is essential for adapting structure prediction models into effective drug discovery workflows.

NeuralPLexer (NP) [4] was one of the first AI models to directly address these complex interactions by pioneering the use of AI for protein-ligand structures, with NeuralPLexer2 (NP2) [2] advancing further in prediction accuracy and covering all essential categories of biomolecules including protein complexes, nucleic acids, small molecules, and covalent modifications.

AlphaFold3 (AF3) [1] recently set a new standard in interaction modeling. While taking us another step further for drug discovery applications, this geometrical approach to interaction prediction still limits its usability. Drug discovery requires precise, atom-level details on how small molecules interact with specific protein atoms and how these interactions induce conformational changes. Rapid screening of large numbers of potential compounds is also necessary.

Specifically, several technical challenges persist in state-of-the-art structure prediction models:

- **Unphysical hallucinations**: AF3 and related methods [1, 5, 6] sometimes generate structures that are not physically plausible, such as ligands with incorrect chiral centers, unrealistic torsion angles, or misfolded chains for disordered regions.
- **Computational cost**: The resource-intensive nature of current diffusion-based structure predictors can limit their scalability in large-scale studies, such as virtual screening.
- **Performance assessment for drug discovery**: There is a need for thorough model evaluation in drug discovery contexts, particularly in predicting ligand-induced protein conformational changes and the recovery of physical interactions such as hydrogen bonds [7, 8].

Here we present NeuralPLexer3 (NP3), offering the following key contributions:

- NP3 improves protein-ligand binding structure prediction with greater accuracy than AF3 while retaining broad applicability across all categories of biomolecular interactions.
- By combining physics-informed priors, fast sampling alorithms, and hardware-aware optimizations, NP3 typically delivers a prediction within seconds of GPU time while preserving accuracy.
- We introduce novel benchmarks to evaluate conformational predictions related to binding pocket/ligand interactions and their potential effects on protein conformation across a diverse dataset. For specific protein classes, such as kinases, NP3 provides downstream functional insights, such as predicting protein inactivation from ligand binding.

We also present scaling studies for flow-based encoder-decoder structure prediction models and estimate the compute-optimal frontier, improving training efficiency relative to AF3 and related methods. This detailed breakdown of model performance offers insights about learning behaviors across various biomolecular modalities and identifies future areas for improvement.

2 Results

NeuralPLexer3 (NP3) (Figure 1A) is a generative modeling framework for the *de novo* prediction and sampling of generalized biomolecular complex structures composed of, but not limited to, proteins, nucleic acids, ligands, ions, and post-translational modifications (PTMs). The system is also designed to provide all-atom and pairwise confidence estimation of predicted complexes.

2.1 Model architecture

Underpinning NP3 is a conditional flow-based generative model that samples the 3D coordinates of all heavy atoms of the complex from a model distribution [10, 11].

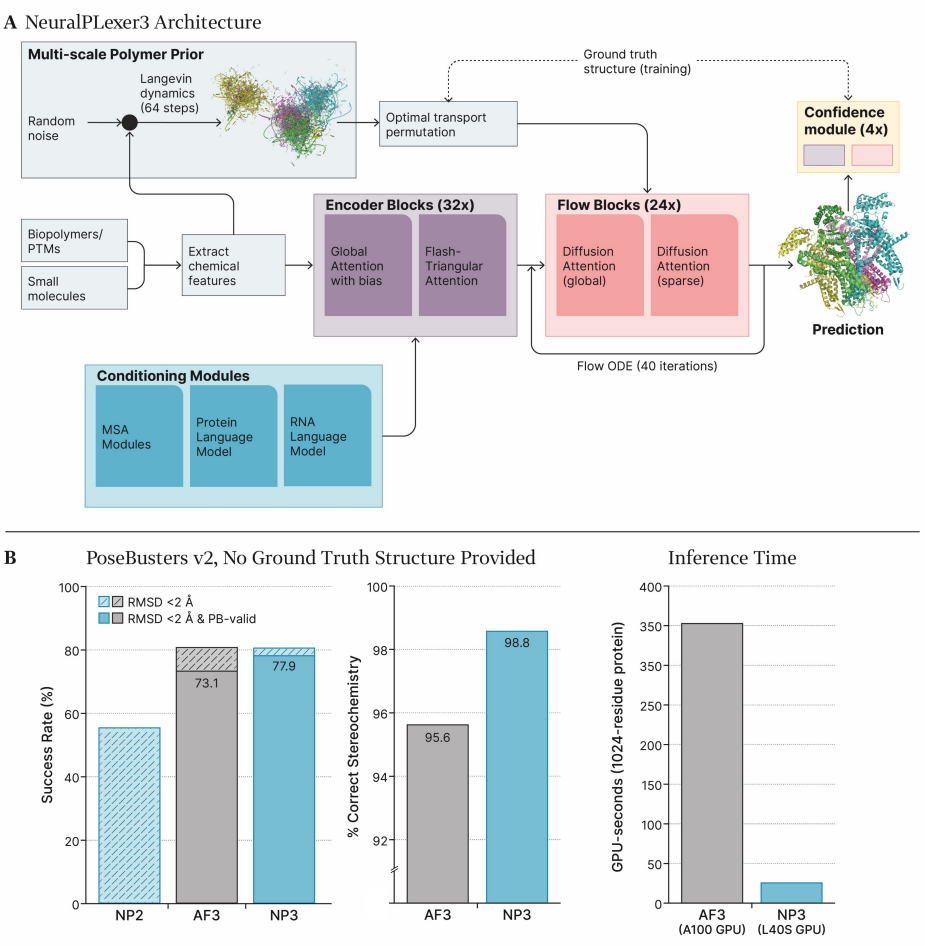

Figure 1: **NeuralPLexer3 (NP3) accurately predicts biomolecular structures with improved physical quality and prediction speed.** (**A**) Schematics of the NP3 system. To perform a prediction, NP3 uses molecular topology extracted from input biopolymer sequences and small molecule graphs as primitive inputs, with additional conditioning signals from sequence language models and multiple sequence alignments (MSAs). NP3 adopts a flow-matching framework that samples from an informative globular polymer prior (Algorithm S3). (**B**) Performance of NP3 on the PoseBusters benchmark [9]. Left panel: success rate for predicting the ligand-protein structures to within 2 Å RMSD, with and without additionally requiring that the structures are physically reasonable (PB-valid). Center panel: percentage of predicted structures where ligand stereochemistry is correct. Right panel: timing for running a single inference on a 1024-residue protein. Comparisons are made to NeuralPLexer2 (NP2) and AlphaFold 3 (AF3).

Continuous normalizing flows: The core model of NP3 is a flow-based generative model that uses continuous normalizing flows (CNFs) [10] and is trained without the use of simulations. CNFs transform simple probability distributions into complex ones through continuous-time dynamics. CNFs sample new data by integrating an ordinary differential equation (ODE) using initial conditions from the prior distribution:

$$x_t = x_0 + \int_0^t u_s(x_s)\, ds \tag{1}$$

Flow matching: Flow matching enables efficient and stable training of CNFs by aligning the model's vector field with a target vector field derived from predefined conditional probability paths:

$$\mathcal{L}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1]}\mathbb{E}_{x\sim p_t}\left[\|u_\theta(t,x) - u(t,x)\|^2\right]. \tag{2}$$

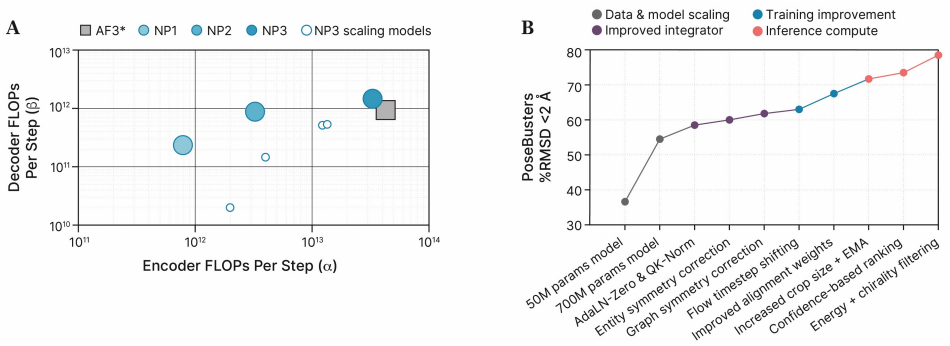

Figure 2: (**A**) Comparison of relative encoder/decoder capacity in terms of floating-point operations per second (FLOPs) among different methods. The asterisk indicates an estimate based on our reproduction of AF3. (**B**) Key model training and inference improvements included during the full course of model development and their impact on PoseBusters accuracy.

Improving flow matching for biomolecular structure prediction: We introduce several key enhancements to make flow matching more suitable for probabilistic structure prediction:

First, we take advantage of physically informed priors. Flow matching offers greater flexibility in choosing the prior distribution functional form, and we reason that more appropriate prior distributions will better capture the underlying data structure. In NP3, we introduce a physics-motivated, globular polymer prior that preserves the distance structure among linked atoms and residues belonging to the same chain (Figure 1A). This informative prior is efficiently implemented by relaxing random atom configurations with a limited number of Langevin dynamics iterations using an energy model with harmonic connectivity terms (Algorithm S3).

Second, by incorporating optimal transport principles, flow matching can create simpler flows that are more stable to train and lead to faster inference [12–14]. We introduce a simulation-free symmetry correction module to straighten the conditional flow trajectories that connect the prior samples and ground truth structures. The symmetry correction module permutes equivalent entities, then permutes local atom indices while preserving the underlying chemical graph structures (Section S.5).

Finally, instead of directly fitting the vector field u, we predict the denoised coordinates and estimate the vector field following an optimal rigid structure alignment. Following previous work [4], we also apply a rigid alignment against the previous-step structure estimate to improve trajectory continuity (Algorithm S2).

More details on model training and inference can be found in Algorithms S2 and S1. Altogether, these contributions led to two main advantages: (1) a substantial reduction in the number of integrator steps needed to sample from the model, leading to improved inference efficiency; and (2) alleviating the need for expensive diffusion rollouts [1] before each optimizer step, which both speeds up and simplifies the procedure for training the main model and confidence modules.

We have implemented several architectural enhancements to support these contributions.

Model architecture: Leveraging components from both its predecessors and recent structure prediction methods [1, 15], we adopt an encoder-decoder architecture that separates the tasks of anchor-level conditioning and atom-level structure generation (Figure S1). Anchors are selected with a prespecified budget from all atoms, with priority given to residue backbones and ligand atoms. Paired multiple sequence alignment (MSA) [16, 17] features obtained via an improved pairing algorithm are assigned to these anchor atoms. The encoder comprises embedding layers to project all atom, bond and stereochemistry features; a hierarchical MSA Module; and AF3-style PairFormer blocks. The decoder consists of flow blocks using a diffusion transformer [18] (DiT) architecture with additional geometric bias and modern normalization layers and layer initialization techniques. Each flow block operates first on anchor atoms using dense attention and then on all heavy atoms using a linear-scaling sliding window attention [19], both using bias terms from input topology and encoder representations. Refer to the Supplementary Information sections S.3 and S.4 for further details.

Compute-optimal scaling: The Chinchilla project [20] introduced compute-optimal scaling laws, demonstrating that an optimal balance between model size and training data can be identified by extrapolating from a series of iso-training-compute Pareto frontiers. However, applying these frameworks to structure prediction models presents unique challenges due to their (1) heterogeneous architectures; (2) the simultaneous training of conditioning and structure generation modules (Figure S1); (3) the use of decoder parallelism, where multiple independent coordinate initializations are passed to the decoder network within each training iterations to improve efficiency (Figure S2); and (4) the vast number of modalities corresponding to each category of biomolecular interaction. To resolve these challenges, we reason that the total computational cost, measured in floating-point operations (FLOPs), serves as a more accurate indicator of model capacity than parameter count.

To optimize the computational efficiency of training our encoder-decoder architecture, we conducted a series of experiments and determined the ideal balance (Figure S4): an encoder-decoder FLOP ratio of 10 with 20 decoder replicas. While intra-molecular prediction accuracy tends to saturate upon reaching a critical decoder size, inter-molecular interaction prediction continues to benefit from jointly scaling the model and data up to the production model size. We plot the relative encoder/decoder capacity among different methods, including NP3 scaling models and the production 700M-parameter model, and an estimated result for AF3 based on our reimplementation in Figure 2A).

Reaching full model scale required a custom triangular attention kernel. See Supplementary Information section S.6 for details of its implementation and computational performance.

Model ablations: We illustrate the impact of each modification included during the full course of model development on PoseBusters accuracy, categorized into data engineering, training loss improvements, improvements to the flow sampler, and better sample ranking protocols (Figure 2B). Key improvments came from weighting interface-based training data sampling and cropping, AdaLN-Zero [18], QK-Norm [21], an improved weighting scheme in rigid structure alignment that improves training (Algorithm S2), and the incorporation of clash and chirality penalties in conformer ranking.

2.2 Protein Structure and Interaction Accuracy

On the PoseBusters benchmark [9], NP3 achieves state-of-the-art accuracy in predicting protein-ligand complexes. This benchmark evaluates two critical aspects that determine whether a model truly enables structure-based drug discovery: whether the predicted structures match experimental data to within the necessary accuracy (<2 Å RMSD) and whether their molecular geometries are physically valid (PB-valid).

Given only a protein sequence and a ligand's chemical structure, without prior knowledge of the protein's structure, binding site location, or ligand conformation, NP3 achieves a 78.4% combined success rate across both metrics, outperforming AF3's 73.1% (Figure 1B). On the coordinate error (% RMSD < 2 Å), NP3 achieves similar performance to AF3 (80.2% versus 80.4%, shown in Table 1). NP3 similarly demonstrates substantial performance advantages over traditional docking methods that rely on experimental holo structures and explicitly defined pocket residues, including Vina (59.7%), GOLD (58.1%), and Uni-Mol (21.8%) [22–24]. Newer AI-based approaches, such as EquiBind, TankBind, and DiffDock [25–27], which eliminate the need for explicitly defined pockets but still depend on reference protein structures, achieved lower success rates of 1.9%, 15.9%, and 38%, respectively.

Importantly, NP3 also achieves 98.8% accuracy in predicting ligand stereochemistry (Figure 1B). This is critical for downstream drug design applications, as accurate stereochemical predictions are essential for optimizing ligand binding affinity and pharmacological activity, ensuring the development of effective and selective therapeutic compounds.

To further evaluate NP3's performance on nucleic acids, covalent ligands, and protein interaction accuracies, we introduced a new benchmarking suite, NPBench. NPBench addresses several limitations in existing benchmarks for structure prediction. Standard molecular docking benchmarks are typically limited to binary interactions (single target, single ligand) and therefore fail to capture the diversity of biomolecular interactions and stoichiometry inherent to molecular biology. Moreover, many of these benchmarks exhibit significant overlap with training structures, which can lead to an overestimation of model performance on novel targets.

Table 1: **Quantitative model performance across biomolecule and interaction types**. Protein-ligand interaction prediction accuracy is evaluated on the PoseBusters-V2 dataset [9]. Nucleic acids, covalent ligands, and protein prediction accuracies are evaluated using 1,143 chains or interfaces from low-homology, high-resolution, and deduplicated Protein DataBank (PDB) structures released after 2023. Sub- and super-scripts in the value column indicate the 95% confidence interval. Unless otherwise stated, no structural input is provided. †Holo protein structure input provided. ‡Pocket residues specified. §Requires human input.

Task	Dataset	Metric	Method	N	Value
Ligands	PoseBusters V2	% RMSD < 2 Å and PB-valid	NP3	308	$77.9^{81.4}_{72.3}$
			AF3 (2019 cutoff)	308	73.1
		% RMSD < 2 Å	NP3	308	$80.2^{84.9}_{74.6}$
			AF3 (2019 cutoff)	308	$80.5^{84.8}_{75.6}$
		% RMSD < 2 Å†	EquiBind	308	$1.9^{4.2}_{0.7}$
			TankBind	308	$15.9^{20.5}_{12.0}$
			DiffDock	308	$38.0^{43.7}_{32.5}$
		% RMSD < 2 Å‡	Vina on AF-M 2.3	308	$15.3^{19.8}_{11.4}$
			DeepDock	308	$19.5^{24.4}_{15.2}$
			Uni-Mol	308	$21.8^{26.8}_{17.3}$
			GOLD	308	$58.1^{63.7}_{52.4}$
			Vina	308	$59.7^{65.3}_{54.0}$
Low-homology Ligands	Noncovalent ligands	% RMSD < 2 Å	NP3	572	$60.7^{64.9}_{56.6}$
	Covalent ligands	% RMSD < 2 Å	NP3	584	$69.6^{75.1}_{64.1}$
	Modified residues	% RMSD < 2 Å	NP3	320	$82.6^{88.5}_{76.7}$
Proteins	Protein–Protein	% DockQ > 0.23	AF-M 2.3	239	$52.3^{58.7}_{45.9}$
			NP3	239	$52.7^{59.1}_{46.3}$
	Protein–Peptide	% DockQ > 0.23	AF-M 2.3	47	$76.6^{89.2}_{64.0}$
			NP3	47	$85.1^{95.7}_{74.5}$
	Monomers	LDDT	AF-M 2.3	123	$87.1^{95.6}_{78.5}$
			NP3	123	$86.2^{87.5}_{85.0}$
		TM-score	AF-M 2.3	123	$90.9^{93.0}_{88.7}$
			NP3	123	$89.7^{91.6}_{87.7}$
Nucleic Acids	Protein–RNA	% DockQ > 0.23	NP3	49	$32.7^{46.3}_{19.0}$
	Protein–DNA	% DockQ > 0.23	NP3	116	$56.2^{63.9}_{48.5}$
	RNA Monomers	LDDT	NP3	20	$53.3^{60.5}_{46.2}$
	CASP 15 RNA	LDDT	NP3	12	$49.9^{57.2}_{42.7}$
	CASP 15 RNA (AF3 version)	LDDT	NP3	8	$46.5^{57.3}_{35.6}$
			AF3	8	$47.3^{55.2}_{41.7}$
			Alchemy_RNA2§	8	$54.5^{62.4}_{45.3}$

While AF3 provided a RecentPDBEval evaluation protocol that covers diverse molecular interactions, the associated code is not publicly available, limiting its accessibility for broader benchmarking efforts [1]. In parallel, PLINDER introduced a comprehensive collection of protein-ligand interactions with stratified splits; however, it does not currently support nucleic acids, generalized stoichiometry, or holdout splits aligned with the time-based evaluation splits commonly used in modern structure prediction models [28].

To address these gaps, NPBench comprises 1,143 chains or interfaces derived from low-homology, high-resolution, and deduplicated PDB structures released after 2023. Further details of this benchmark are described in section S.2, and the code will be made publicly available to support reproducibility and future advancements in the field.

On this benchmark, NP3 exhibited an overall strong performance across diverse biomolecular targets and interactions:

- **Protein monomers and protein-protein interactions (PPIs)**: NP3 demonstrated comparable performance in predicting protein-protein interactions, achieving a success rate of 52.7% on low-homology interfaces with DockQ [29] scores greater than 0.23, closely matching AlphaFold2-Multimer v2.3 (AF2-M 2.3) [30], which achieved 52.3%. For monomers, NP3 and AF2-M 2.3 also showed similar results, with success rates of 87.1% and 86.2%, respectively, as measured by LDDT.
- **Protein-peptide interfaces**: On protein-peptide interfaces—where peptides are defined as standard or modified polypeptide chains with fewer than 20 amino acids—NP3 largely outperformed AF2-M, achieving an 85.1% success rate compared to AF2-M's 76.6%.
- **Noncovalent ligands**: NP3 achieved a success rate of 60.7% (ligand RMSD <2 Å) for noncovalent ligands in the evaluation set, compared to 80.2% on the PoseBusters benchmark. Notably, the evaluation set consists of ligand-target interfaces where either the binding pocket or the ligand exhibits significant structural dissimilarity from the training data. These results demonstrate NP3's ability to generalize effectively to unseen targets and novel molecules, maintaining satisfactory performance under more stringent evaluation criteria.
- **Covalent ligands and PTMs**: For covalent ligands, NP3 achieved a high success rate of 69.6%, while for modified residues it reached 82.6%. These results underscore NP3's ability to generalize to proteins with chemical modifications and covalent chemistry, making it particularly relevant for drug discovery applications targeting chemically modified proteins.
- **Nucleic acids**: On CASP15 RNA targets [31, 32], NP3 demonstrated comparable performance to AF3 (46.5% versus 47.3%, respectively) and slightly below that of Alchemy_RNA2 [33] (54.8%), which relies on explicitly curated human inputs. While AF3 conditions its predictions on RNA MSAs, NP3 uses RNA language models (LMs) and achieves similar performance—a noteworthy result, as MSAs are typically considered more effective than LMs for conditioning structure predictions. This outcome also highlights the potential of utilizing LM-based approaches in scenarios where RNA MSAs are unavailable or impractical to generate. For protein-DNA interactions, NP3 achieved accuracy comparable to protein-protein interactions (56.2% versus 52.7%, respectively) and lower, though reasonable, accuracy for protein-RNA interactions (32.7%).

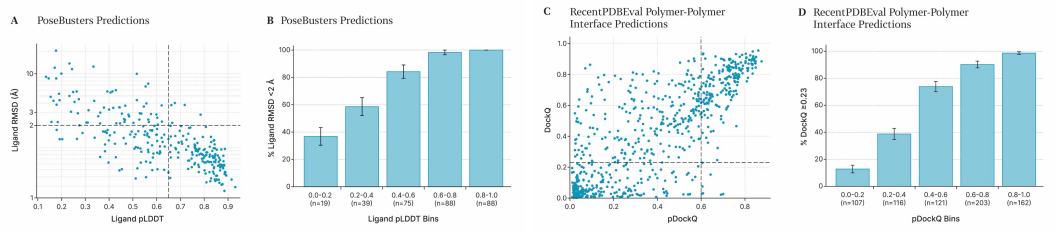

Figure 3: **Model confidence estimation.** (**A**) Scatter plot of ligand RMSD against ligand pLDDT and on PoseBusters. (**B**) PoseBusters ligand RMSD success rate statistics grouped into NP3 confidence percentiles. Prediction success rates are consistently higher for higher confidence prediction bins. (**C**) Scatter plot of DockQ score against the pDockQ score on NPBench protein-protein and protein-nucleic acid interfaces. (**D**) PPI DockQ success rate statistics grouped into NP3 confidence percentiles. Vertical lines on the scatter plot indicate the median prediction confidence. 95% confidence intervals are indicated by error bars in panels **B** and **D**.

A strong correlation was also observed between NP3's model-estimated confidence and true prediction accuracy, highlighting its ability to effectively gauge the reliability of its outputs (Figure 3). Confidence for ligand predictions was quantified using the pLDDT score, while polymer-polymer interface predictions were assessed using the newly introduced pDockQ score (see SI, S.9). To further analyze the relationship between confidence and accuracy under a cutoff that defines prediction successful-ness, prediction success rates were aggregated into percentiles based on model-estimated confidence. Results showed that higher confidence predictions achieved substantially better outcomes. Specifically, for predictions in the top 50% confidence subset, NP3 achieved a 96.7% RMSD success rate on the PoseBusters benchmark and a 93.1% DockQ success rate on the NPBench dataset.

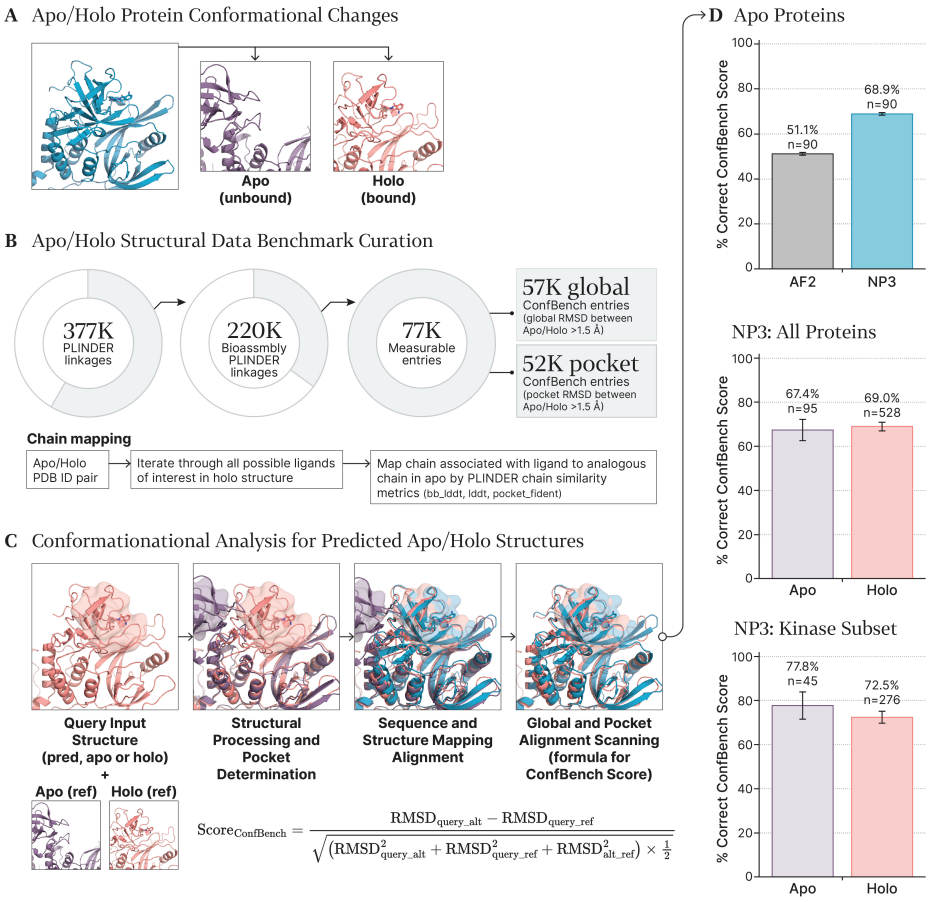

Figure 4: **ConfBench conformational prediction**. (**A**) Schematic illustration of types of conformational changes of interest. (**B**) Benchmark target statistics. 57k global and 52k pocket rearrangement entries with meaningful motions and biologically relevant ligands are identified. (**C**) The ConfBench scoring protocol. (**D**) Conformational prediction success rate statistics comparison. A correct conformational prediction is defined as a ConfBench score greater than 0. NP3 outperforms AF2-M 2.3 in predicting pocket conformations on apo targets, while maintaining similar success rates on apo targets and holo targets. 95% confidence intervals are indicated by error bars in panel **D**. In the top panel, the $n = 90$ apo structures correspond to the subset of the $n = 95$ apo structures in ConfBench for which AF2 predictions were possible.

2.3 Ligand-induced Conformational Change Prediction

To this point, we have considered benchmarks of biomolecular systems in isolation. Also of interest is assessment of accuracy in predicting the structural change of proteins from apo (non-ligand-bound) to holo (ligand-bound) structures. Accurate conformational predictions, both globally and locally around the binding site, are of acute importance in drug discovery by assisting in identifying allosteric binding sites and optimizing interactions for improved efficacy and selectivity.

Several groups have discussed AlphaFold's limitations in predicting complete conformational ensembles of proteins of interest [34–38]. Even so, there is a lack of standard benchmarks to quantitatively measure the prediction accuracy of proteins with induced conformational changes.

To address this need, we developed ConfBench, a conformational benchmark that systematically evaluates ligand-induced conformational changes across the proteome and assesses model conformational biases by comparing predictions against reference apo and holo states (Figure 4). ConfBench consists of a rigorously curated selection of chain and ligand-specified apo/holo PDB pairs that

exhibit measurable conformational differences in one or more of the following ways: global RMSD > 1.5Å, pocket alpha carbon RMSD > 1.5Å, or pocket alpha carbon plus sidechain heavy atom RMSD > 1.5Å. Additionally, we propose a conformational scoring function that is a measure of conformational accuracy which is insensitive to absolute variance between reference structures and also agnostic of structural accuracy of conformationally irrelevant domains.

To enable the PDB-wide measurement of conformational change down to an atomic level of accuracy, a stringent alignment protocol was required prior to RMSD calculation. Starting from PLINDER [28], we implemented an exhaustive, bidirectional search of all linkages, examining all possible chain combinations and distinct ligand binding sites in multi-liganded complexes. Optimal chain mapping was comprehensively validated with PLINDER chain similarity metrics, namely pocket sequence identity (pocket_fident), chain-wise local distance difference test (LDDT), and chain-wise backbone local distance difference test (bbLDDT).

For apo/holo pairs that have conformational changes greater than 1.5Å, i.e. changes that are measurable and relevant, NP3 outperforms AF2-M in conformational prediction success rates. NP3 outperformed AF2-M for 54 unique linkages with a correctness rate of 51.9% of predictions, compared to 29.6% of AF2-M predictions. This is true across the board, whether that be global conformational changes, pocket alpha carbon atom conformational changes, or pocket including sidechain conformational changes. Compared to context-unaware methods like AF2-M, this result demonstrates the potential of using all-atom structure prediction algorithms to capture conformational variations induced by ligand binding and change in physiological conditions.

NP3 achieves consistent performance for apo and holo structures. Across all evaluated proteins, NP3 achieved correctness rates of 67.4% for apo structures ($N_{\text{linkages}} = 95$) and 69.0% ($N_{\text{linkages}} = 528$) for holo structures. To further evaluate performance within a high-value therapeutic target class, the benchmark dataset was refined to focus on kinases. Within this subset, NP3 achieved correctness rates of 77.8% for apo structures and 72.5% for holo structures.

There are some limitations: (1) imperfect ConfBench scores for NP predictions and (2) apo structures used for scoring do not always imply a fully ligand-free structure. We recognize that ligand-free structures in the PDB are relatively limited [28, 39]; we expect more work to enable a conformational benchmark that is diverse, biologically relevant, and leakage free.

3 Discussion

Accurately determining protein-ligand structures and their conformational distributions remains a significant challenge, especially at the fine scale required for drug discovery. Along these lines, one notable contribution of NP3 is its high accuracy on the PB-valid metrics that assess the prediction of physically plausible structures. This capability is essential for generating meaningful insights and avoiding unreliable results, as physically realistic predictions directly impact the translational success of computational models in drug discovery. NP3's high accuracy in predicting PTMs further represents a step forward in modeling biological processes such as signaling, immune response, and protein folding [40]. These targets are historically underexplored in structure-based drug discovery due to experimental challenges, where NP3 may open new opportunities for drug targeting.

In modeling specific conformations, particularly apo and holo states, NP3 also demonstrates improved accuracy compared to AF-M 2.3. Notably, NP3 shows a similar fraction of correct conformational predictions between its predictions on apo targets and holo targets. These precise predictions, such as the successful modeling of kinase conformations, are critical for optimizing lead compounds and deepening the understanding of target protein behavior in structure-based drug design, and we anticipate the evaluation framework of this work to be a basis for fairly measuring and improving conformation modeling ability.

Finally, NP3 delivers predictions in seconds. This speed advantage may enable researchers to virtually screen extensive chemical libraries and iterate on ligand designs with significantly greater scalability. Moreover, NP3 achieves this efficiency while maintaining compatibility with standard hardware, reducing computational resource requirements and making advanced structure prediction more accessible.

NP3's performance is limited by its training data: PDB entries are resolved under artificial experimental conditions, whereas the model's intended use is to predict structures in native biological

environments. Model capacity is a second limiting factor. Figure S4 shows that scaling the network would boost accuracy, yet such expansion is challenging on current GPUs. Finally, this study did not evaluate NP3's accuracy on antibody structures.

4 Broader Impacts

NP3 offering a powerful in-silico complement to existing wet-lab methods in biology and medicine. Potential positive impacts include (i) accelerating the design of more selective and effective therapeutics, (ii) reducing experimental screening cycles, thereby lowering cost and environmental footprint, and (iii) enabling biologists without extensive structural resources to generate testable hypotheses. We anticipate downstream benefits for health.

The same capabilities also pose dual-use concerns. High-fidelity complex modeling could aid design of novel toxins, immune-evasive variants, or delivery vehicles. We will not release model weights, substantially mitigating these concerns.

NP3 will be used in virtual screening of drug molecules. Compared to existing co-folding models, NP3 is much more computationally and therefore environmentally efficient.

5 Code and Data Availability

Code for the NPBench benchmark is available on GitHub at `https://github.com/iambic-therapeutics/np-bench`. NPBench reference structures and corresponding model inferences are available on Zenodo at `https://zenodo.org/records/14503936`.

6 Acknowledgments

This work used computational resources provided by the National Energy Research Scientific Computing Center (NERSC) under Contract No. ERCAP0029432.

References

1. Abramson, J. *et al.* Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature* **630,** 493–500. `https://www.nature.com/articles/s41586-024-07487-w` (June 2024).
2. *Transforming Computational Drug Discovery with NeuralPLexer2* 2024. `https://www.iambic.ai/post/np2`.
3. Jumper, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* **596,** 583–589. `https://www.nature.com/articles/s41586-021-03819-2` (Aug. 2021).
4. Qiao, Z., Nie, W., Vahdat, A., Miller III, T. F. & Anandkumar, A. State-specific protein–ligand complex structure prediction with a multiscale deep generative model. *Nature Machine Intelligence* **6,** 195–208. `https://www.nature.com/articles/s42256-024-00792-z` (2024).
5. Krishna, R. *et al.* Generalized biomolecular modeling and design with RoseTTAFold All-Atom. *Science* **384,** eadl2528. `https://www.science.org/doi/10.1126/science.adl2528` (2024).
6. Lu, W. *et al.* DynamicBind: Predicting ligand-specific protein-ligand complex structure with a deep equivariant generative model. *Nature Communications* **15,** 1071. `https://www.nature.com/articles/s41467-024-45461-2` (2024).
7. Errington, D., Schneider, C., Bouysset, C. & Dreyer, F. *Assessing interaction recovery of predicted protein-ligand poses* Sept. 2024. `https://zenodo.org/doi/10.5281/zenodo.13843798`.
8. Mirdita, M. *et al.* ColabFold: making protein folding accessible to all. *Nature Methods* **19,** 679–682. `https://www.nature.com/articles/s41592-022-01488-1` (June 2022).

9. Buttenschoen, M., M. Morris, G. & M. Deane, C. PoseBusters: AI-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science* **15,** 3130–3139. https://pubs.rsc.org/en/content/articlelanding/2024/sc/d3sc04185a (2024).
10. Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M. & Le, M. *Flow Matching for Generative Modeling* in *International Conference on Learning Representations* (Sept. 2022). https://openreview.net/forum?id=PqvMRDCJT9t.
11. Albergo, M. S., Boffi, N. M. & Vanden-Eijnden, E. *Stochastic Interpolants: A Unifying Framework for Flows and Diffusions* Nov. 2023. http://arxiv.org/abs/2303.08797.
12. Tong, A. *et al. Improving and generalizing flow-based generative models with minibatch optimal transport* in *Transactions on Machine Learning Research* (Nov. 2023). https://openreview.net/forum?id=CD9Snc73AW.
13. Kornilov, N., Mokrov, P., Gasnikov, A. & Korotin, A. *Optimal Flow Matching: Learning Straight Trajectories in Just One Step* Nov. 2024. http://arxiv.org/abs/2403.13117.
14. Liu, X., Gong, C. & Liu, Q. *Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow* in *International Conference on Learning Representations* (Sept. 2022). https://openreview.net/forum?id=XVjTT1nw5z.
15. Bryant, P., Kelkar, A., Guljas, A., Clementi, C. & Noé, F. Structure prediction of protein-ligand complexes from sequence information with Umol. *Nature Communications* **15,** 4536. https://www.nature.com/articles/s41467-024-48837-6 (May 2024).
16. Bacon, D. J. & Anderson, W. F. Multiple sequence alignment. *Journal of Molecular Biology* **191,** 153–161. https://www.sciencedirect.com/science/article/pii/0022283686902524 (Sept. 1986).
17. Bryant, P., Pozzati, G. & Elofsson, A. Improved prediction of protein-protein interactions using AlphaFold2. *Nature Communications* **13,** 1265. https://www.nature.com/articles/s41467-022-28865-w (Mar. 2022).
18. Peebles, W. & Xie, S. *Scalable Diffusion Models with Transformers* Mar. 2023. http://arxiv.org/abs/2212.09748.
19. Beltagy, I., Peters, M. E. & Cohan, A. *Longformer: The Long-Document Transformer* Dec. 2020. http://arxiv.org/abs/2004.05150.
20. Hoffmann, J. *et al. Training Compute-Optimal Large Language Models* Mar. 2022. http://arxiv.org/abs/2203.15556.
21. Henry, A., Dachapally, P. R., Pawar, S. & Chen, Y. *Query-Key Normalization for Transformers* Oct. 2020. http://arxiv.org/abs/2010.04245.
22. Eberhardt, J., Santos-Martins, D., Tillack, A. F. & Forli, S. AutoDock Vina 1.2.0: New Docking Methods, Expanded Force Field, and Python Bindings. *Journal of Chemical Information and Modeling*. https://pubs.acs.org/doi/abs/10.1021/acs.jcim.1c00203 (July 2021).
23. Verdonk, M. L., Cole, J. C., Hartshorn, M. J., Murray, C. W. & Taylor, R. D. Improved protein–ligand docking using GOLD. *Proteins: Structure, Function, and Bioinformatics* **52,** 609–623. https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.10465 (2003).
24. Zhou, G. *et al. Uni-Mol: A Universal 3D Molecular Representation Learning Framework* in *International Conference on Learning Representations* (Sept. 2022). https://openreview.net/forum?id=6K2RM6wVqKu.
25. Stärk, H., Ganea, O., Pattanaik, L., Barzilay, D. R. & Jaakkola, T. *EquiBind: Geometric Deep Learning for Drug Binding Structure Prediction* in *Proceedings of the 39th International Conference on Machine Learning* (PMLR, June 2022), 20503–20521. https://proceedings.mlr.press/v162/stark22b.html.
26. Corso, G., Stärk, H., Jing, B., Barzilay, R. & Jaakkola, T. S. *DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking* in *International Conference on Learning Representations* (Sept. 2022). https://openreview.net/forum?id=kKF8_K-mBbS.
27. Lu, W. *et al. TANKBind: Trigonometry-Aware Neural NetworKs for Drug-Protein Binding Structure Prediction* in *Advances in Neural Information Processing Systems* (eds Koyejo, S. *et al.*) **35** (2022), 7236–7249. https://proceedings.neurips.cc/paper_files/paper/2022/file/2f89a23a19d1617e7fb16d4f7a049ce2-Paper-Conference.pdf.
28. Durairaj, J. *et al. PLINDER: The protein-ligand interactions dataset and evaluation resource* July 2024. https://www.biorxiv.org/content/10.1101/2024.07.17.603955v3.

29. Mirabello, C. & Wallner, B. *DockQ v2: Improved automatic quality measure for protein multimers, nucleic acids, and small molecules* June 2024. https://www.biorxiv.org/content/10.1101/2024.05.28.596225v1.
30. Evans, R. *et al. Protein complex prediction with AlphaFold-Multimer* Mar. 2022. https://www.biorxiv.org/content/10.1101/2021.10.04.463034v2.
31. Kryshtafovych, A., Schwede, T., Topf, M., Fidelis, K. & Moult, J. Critical assessment of methods of protein structure prediction (CASP)—Round XV. *Proteins: Structure, Function, and Bioinformatics* **91,** 1539–1549. https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.26617 (2023).
32. Das, R. *et al.* Assessment of three-dimensional RNA structure prediction in CASP15. *bioRxiv,* 2023.04.25.538330. https://www.biorxiv.org/content/10.1101/2023.04.25.538330v3 (Oct. 2023).
33. Chen, K., Zhou, Y., Wang, S. & Xiong, P. RNA tertiary structure modeling with BRiQ potential in CASP15. *Proteins: Structure, Function, and Bioinformatics* **91,** 1771–1778. https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.26574 (2023).
34. Lazou, M. *et al.* Predicting multiple conformations of ligand binding sites in proteins suggests that AlphaFold2 may remember too much. *Proceedings of the National Academy of Sciences* **121,** e2412719121. https://www.pnas.org/doi/10.1073/pnas.2412719121 (Nov. 2024).
35. Sala, D., Engelberger, F., Mchaourab, H. S. & Meiler, J. Modeling conformational states of proteins with AlphaFold. *Current Opinion in Structural Biology* **81,** 102645. https://www.sciencedirect.com/science/article/pii/S0959440X23001197 (Aug. 2023).
36. Saldaño, T. *et al.* Impact of protein conformational diversity on AlphaFold predictions. *Bioinformatics* **38,** 2742–2748. https://doi.org/10.1093/bioinformatics/btac202 (May 2022).
37. Meller, A. *et al.* Predicting locations of cryptic pockets from single protein structures using the PocketMiner graph neural network. *Nature Communications* **14,** 1177. https://www.nature.com/articles/s41467-023-36699-3 (Mar. 2023).
38. Riccabona, J. R. *et al.* Assessing AF2's ability to predict structural ensembles of proteins. *Structure* **32,** 2147–2159.e2. https://www.sciencedirect.com/science/article/abs/pii/S0969212624003708 (11 2024).
39. Feidakis, C. P., Krivak, R., Hoksza, D. & Novotny, M. AHoJ-DB: A PDB-wide Assignment of apo & holo Relationships Based on Individual Protein–Ligand Interactions. *Journal of Molecular Biology. Computation Resources for Molecular Biology* **436,** 168545. https://www.sciencedirect.com/science/article/pii/S0022283624001402 (Sept. 2024).
40. Lane, T. J. Protein structure prediction has reached the single-structure frontier. *Nature Methods* **20,** 170–173. https://www.nature.com/articles/s41592-022-01760-4 (2023).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Claims are supported fully. Evidence is provided primarily in Section 2.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations of our introduced ConfBench benchmark in Section 2.3. We discuss limitations of our model and its training data in Section 3.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Full details of the model architecture and training procedure are provided in the Supplementary Information. The code and data corresponding to new benchmarks introduced in this work are publicly available.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The model code is not available. The code and data corresponding to new benchmarks introduced in this work are publicly available. We provide details necessary to reproduce the models in this work in the Supplementary Information.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Full details of the model architecture and training procedure are provided in the Supplementary Information.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Error bars and confidence intervals are reported with statistical bootstrap over the test set. Only one model training run was feasible, so variance with respect to model initialization is not reported.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Details about the computational resources used for model training and inference are provided in the Supplementary Information section S.7.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Potential positive and negative societal impacts of the work are discussed in Section 4.

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not release any data or models with high risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The following existing assets were cited and used in this work (corresponding licenses are provided in parenthesis): RCSB PDB (CC0), OpenProteinSet (CC BY 4.0), ESM2 (MIT), RiNalMo (Apache-2), HHSuite (GPL-3), and HMMER (BSD-3c). All code libraries used in this work were checked for license violations, and none were found.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: New assets are provided as code with a documentation README.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Supplementary Information

S.1 Comparison to previous work

NeuralPLexer2

NeuralPLexer2 is a diffusion-based model using an equivariant graph-based denoising decoder, based on a continual development of the original NeuralPLexer model [1]. NeuralPLexer3 instead is a flow matching model with a multimodal conditioning module and a conditional diffusion transformer decoder. NeuralPLexer2 employs a Gaussian prior while NeuralPLexer3 employs a more physical globular polymer prior (described in Algorithm S.4). Compared to NeuralPLexer2, NeuralPLexer3 is trained with greatly expanded training data and model size, and shows substantially improved performance.

AlphaFold 3

The main architectural components of AlphaFold 3 and NeuralPLexer3 are the same: MSA and Pairformer are used by both models. NeuralPLexer3 conditions additionally on protein and RNA language model embeddings. From scaling studies (Figure S4), we identified a different pareto-optimal choice for encoder/decoder size. We also using sliding window attention instead of ad-hoc block-based operations for token-to-atom communication.

NeuralPlexer3 employs flow matching to allow sampling from any prior distribution, and introduces a novel physics-based prior (Algorithm S.4). NeuralPLexer3 does not use AlphaFold 3's hand-crafted positional encoding among entities and atoms. NeuralPLexer3 does not rely on reference RDKit conformers, allowing it to encode molecules of all chemistries.

During training, no mini-rollouts are needed in NeuralPLexer3 because of the prior alignment step, which dramatically reduces the complexity of the training pipeline compared to Alphafold 3. We also introduce a fused triangular attention kernel absent in AlphaFold 3, leading to faster and more memory-efficient training and inference.

S.2 Methods

Structure Prediction Benchmarks: We evaluated NP3's structural prediction performance on three benchmarks—the PoseBusters-V2 dataset [2] for protein-ligand interaction prediction accuracy, NPBench for diverse interaction types and stoichiometry on low-homology structures, and CASP15 for RNAs [3].

Baselines: We conducted comparisons across multiple benchmark datasets to evaluate performance. For the PoseBusters ligand dataset [2], we compared NP3 to traditional docking methods that require holo receptor structures, including Vina [4] and GOLD [5], as well as machine learning-based methods with similar holo structure requirements, such as EquiBind [6], TankBind [7], and DiffDock [8]. Additionally, we benchmarked against structure prediction methods that use only sequence and graph information, including AF3. Baseline numbers for these methods were derived from Abramsom et al (Extended Data Table 1) [9].

For protein targets from the dataset of recent PDB structures (NPBench), NP3's performance was compared to AlphaFold-Multimer 2.3 using ColabFold [10]. Five seeds were generated per target, and the top-ranked conformers were scored using both NP3 metrics and the "multimer" metric provided by AlphaFold-Multimer [11].

For CASP15 RNA targets, NP3 was benchmarked against AF3 and Alchemy_RNA2 [12]. Baseline metrics for these methods were sourced from the AF3 publication, which originally evaluated eight CASP15 RNA targets [9]. With all CASP15 RNA targets now publicly available, NP3's average RNA LDDT scores were reported across both the original subset and the complete dataset, with no significant differences in performance between subsets.

Evaluation Criteria: To evaluate model performance across this diverse set of biomolecules and structures, we utilized several well-established metrics:

- RMSD <2 Å success rate: The fraction of predictions with pocket-aligned ligand RMSD below 2 Angstrom, a widely used metric for assessing structural prediction accuracy.

- PB-valid success rate: The proportion of PoseBusters predictions that pass physical validity checks implemented in PoseBusters (PB).
- Local distance difference test (LDDT): An alignment-free metric for assessing the accuracy of local structures across cutoff all biomolecules. A 25 Å inclusion cutoff is used for DNAs and RNAs, and a 15 Å is used for proteins.
- DockQ >0.23 success rate: The fraction of DockQ score greater than 0.23 across all interfaces for evaluation, a metric for evaluating the accuracy of polymer-polymer interaction predictions using acceptable thresholds from CAPRI [13].

Conformational Benchmarks: To develop the ConfBench benchmark, which assesses conformational accuracy, all PLINDER [14] linkages were filtered for entries with only experimental structures from the PDB. These linkages were further distilled to curate a dataset of protein-ligand systems exhibiting measurable conformational changes between apo and holo states (4A and B). Systems with "lock-and-key" binding, where the protein structure remains largely unchanged upon ligand binding, or those with minimal allosteric shifts, were excluded. This step ensures the benchmark focuses on clear, quantifiable ligand-induced structural transitions that are significant enough to measure and therefore evaluate prediction accuracy and enable effective model performance assessment.

We next employed a systematic approach to identify precise chain-level correspondences between apo and holo protein structures—a critical metric for reliably evaluating ligand-induced conformational shifts in multichain, multi-liganded oligomeric systems:

1. Extract chain information from PLINDER's structured identifiers;
2. For each protein chain with an associated ligand chain, identify all possible mappings between holo protein chain and linked apo protein chains;
3. Rank these mappings based on quantitative structural metrics:
   - Pocket_fident: Sequence identity within the binding pocket
   - LDDT: Local distance difference test
   - bb_LDDT: Backbone local distance difference test

To evaluate structural predictions, we employed a scoring formula designed to quantify the similarity between predicted (query) and ground truth (reference) structures. The score is calculated as:

$$\text{score} = \frac{\text{RMSD}_{\text{query\_alt}} - \text{RMSD}_{\text{query\_ref}}}{\sqrt{\left(\text{RMSD}^2_{\text{query\_alt}} + \text{RMSD}^2_{\text{query\_ref}} + \text{RMSD}^2_{\text{alt\_ref}}\right) \times \frac{1}{2}}} \tag{3}$$

where:

- score = 1 when the query RMSD is 0, indicating the prediction is identical to the reference state
- score = -1 indicates the prediction is identical to the linked alternative state
- score = 0 indicates the prediction is equally distant from the reference as the apo/holo states.

Between 0 and 1 the score scales smoothly, reflecting the degree of similarity to the reference state.

S.3 Data and Input Pipelines

Molecular Topology Featurization: The initial structural features passed to the NP3 inference pipeline are a complete representation of the molecular topology. These input features include:

- Atom-wise features: atom types and residue types. Residue types include a one-hot encoding of the 1-letter residue name for atoms from standard amino acids and nucleotides, and a one-hot encoding for any other residues.
- Atom-pair features: bond orders, and bond orientations:

- For a bonded atom pair (i, j), we evaluate the bond orientations by first constructing a local frame based on two nearest bonded atoms (ik, il) using the generalized Gram-Schmidt procedure described in [1], and assign the bond orientation feature as the direction of the displacement vector between atom i and atom j expressed in the assigned local frame.
- We assign the bond order to 0 and bond orientation features to (0, 0, 0) for any non-bonded atom pairs.
- Disulfides, ionic bonds, hydrogen bonds, and halogen bonds are excluded from featurization and solely inferred by the model.

Heavy atoms of a bioassembly are split into two levels of treatment: atoms and anchors. Only the anchors and anchor pairs among them are being processed by the encoder block to produce conditionings, while all atoms are passed through the flow blocks to produce denoised coordinates of the entire assembly. Given a fixed context size budget, anchors are selected as follows:

1. Biopolymer backbones are prioritized. We label all Cα atoms of amino acid residues and C1' atoms for nucleic acids as anchors, unless the budget is filled in which case we perform a sampling without replacement;
2. Then we perform additional sampling without replacement from all atoms from ligands, non-standard residues, and PTMs;
3. If any budget is left, we perform atom sampling from the remaining standard protein, DNA, and RNA residues until the budget limit is reached.

Evolutionary sequence features: NP3 additionally leverages conditioning from MSAs, protein language models, and nucleic acid language models. For training, we follow the MSA generation protocol from AF3, combining jackhmmer MSAs [15] on uniref30 [16], reduced BFD, and uniref100; and hhblits MSAs [17] for BFD [18]. We do not include RNA MSAs. We use a 3B-parameter language model for proteins (ESM-2) [19] and a 650M-parameter language model for RNAs (RiNalMo) [20].

We introduce a compactified MSA pairing algorithm to capture cross-chain coevolutionary genetics while efficiently utilizing a fixed token context window of at most 16,384 sequences. The method proceeds by first sorting sequences from all databases with decreasing sequence similarity w.r.t the query sequence, then pairing sequences by taxonomical ID [21], and finally backfilling the blank spaces with unpaired monomer sequences.

Data filtering and preprocessing pipeline: For structures obtained from the PDB, we generally follow the mmcif-level filtering protocol that has been described for AlphaFold 3 [9], with the following minor differences:

- We skip structures where either the asymmetric unit cif file size or the bioassembly cif file size exceeds 20 MB;
- We do not need to remove leaving group atoms based on information about the residue in the CCD;
- We do not limit the structures passed to the training pipeline to 20 chains.

To obtain correct connectivity and bond types, we make use of the information provided in the CCD for non-standard residue names via ParmED's 'all_residue_template_match' [22]. Furthermore, we read the bonds present in the 'struct_conn' section in the cif file for the asymmetric unit, which is usually missing in cif files for bioassemblies. We then identify matching atom pairs in the bioassemblies and add the corresponding bonds.

S.4 Model architecture and inference

In S1, we outline NP3's structure sampling procedure. CNFs integrate a learned velocity field to transform a prior distribution and sample new data. Our conformer sampling protocol iteratively refines structures through time-conditioned decoding, rigid-body structure between integrator steps alignment to reduce discretization errors, and graph–preserving atom permutation corrections. In practice, we also apply a timestep shifting trick using a polynomial transform $t^* = t^{1.15}$, which is found beneficial to reduce exposure bias [23].

Table S1: Comparison of data processing pipeline details.

Training Data Processing Step	**AF3**	**NP3 (Ours)**
Keep structures of reported resolution of 9 Å or less	✓	✓
Remove structures with 300 or more polymer chains	✓	✓
Remove polymer chains containing fewer than 4 resolved residues	✓	✓
Remove hydrogen atoms	✓	✓
Remove clashing chains (those with >30% of atoms within 1.7 Å of an atom in another chain)	✓	✓
Remove atoms outside of the CCD code's defined set of atom names	✓	✓
Remove leaving atoms for covalent ligands	✓	Not needed
Filter out protein chains with consecutive Cα atoms greater than 10 Å distance	✓	✓
Truncate closest 20 chains relative to random token for large bioassemblies	✓	Keep all chains
For structures from crystallography, remove crystallization aid molecules	✓	✓

Algorithm S1 Sampling from NP3 with symmetry-corrected flow. CFM: Conditional flow matching.

function NP3SAMPLESTRUCTURE(NP3Model, $\mathbf{c}$, $T = 100$) ▷ T: Number of integrator steps
  $\mathbf{x}_0 \leftarrow \text{MultichainPolymerPriorSample}(\mathbf{c})$ ▷ Sample initializations from the physics-inspired prior
  $\mathbf{x}_1^{\text{last}} \leftarrow \mathbf{x}_0$, $\Delta t \leftarrow 1/T$, $t \leftarrow 0$
  $\mathbf{z}_{\text{atom}}, \mathbf{z}_{\text{anchor}}, \mathbf{z}_{\text{pair-anchor}}, \mathbf{z}_{\text{pair-local}} = \text{NP3Encoder}(\mathbf{c})$
  for timestep **in** 0 **to** $T - 1$ **do**
    $t_{\text{next}} \leftarrow t + \Delta t$
    $\mathbf{x}_1^{\text{pred}} \leftarrow \text{NP3Decoder}(\mathbf{x}_t, t^* | \mathbf{z}_{\text{atom}}, \mathbf{z}_{\text{anchor}}, \mathbf{z}_{\text{pair-anchor}}, \mathbf{z}_{\text{pair-local}}), \quad t^* = t^{1.15}$
    $\mathbf{r}_{\text{opt}}, \mathbf{t}_{\text{opt}} \leftarrow \text{GetKabschTransform}(\mathbf{x}_1^{\text{pred}}, \mathbf{x}_1^{\text{last}}, \mathbf{w}_{\text{align}})$ ▷ Global SE(3)-superposition to improve continuity
    $\mathbf{x}_1^{\text{pred}} \leftarrow \mathbf{x}_1^{\text{pred}} \cdot \mathbf{r}_{\text{opt}} + \mathbf{t}_{\text{opt}}$
    $\mathbf{x}_1^{\text{pred}} \leftarrow \text{OptimalGraphPermutation}(\mathbf{x}_1^{\text{pred}}, \mathbf{x}_1^{\text{last}})$ ▷ Atom permutation correction using graph isomorphism
    $\mathbf{x}_1^{\text{last}} \leftarrow \mathbf{x}_1^{\text{pred}}$
    $\mathbf{x}_{t_{\text{next}}} \leftarrow \text{CFMSampleStep}(\mathbf{x}_1^{\text{pred}}, \mathbf{x}_t, t, t_{\text{next}})$ $\quad \text{CFMSampleStep}(x_1, x_t, t, t_{\text{next}}) = t_{\text{next}} \cdot x_1 + (1 - t_{\text{next}}) \cdot \frac{x_t - t \cdot x_1}{1 - t}$
    $t \leftarrow t_{\text{next}}$
  end for
  return x_1
end function

When sampling initial coordinates from the prior, we incorporate constraints at multiple scales. Bonded connectivity ensures that linked atoms form coherent polymer-like chains; Entity-level and residue-level scale constraints guide atomic positions to cluster in a manner consistent with coarse-grained structural motifs; and a global spherical confinement maintains a compact, globular arrangement.

Algorithm S2 NP3 main model training iteration.

function NP3TRAININGITER(NP3Model, $\mathbf{x}_1^{\text{ref}}$, t, $\mathbf{c}$, $\sigma_{\text{data}} = 16.0$Å, $\epsilon = 0.01$) ▷ $\mathbf{x}_1^{\text{ref}}$: Ground truth structure
  $\mathbf{x}_0 \leftarrow \text{MultichainPolymerPriorSample}(\mathbf{c})$ ▷ Sample initializations from the physics-inspired prior
  $\mathbf{x}_0 \leftarrow \text{OptimalEntityPermutation}(\mathbf{x}_0, \mathbf{x}_1^{\text{ref}})$ ▷ Apply entity permutation on prior sample
  $\mathbf{x}_t \leftarrow (1-t) \cdot \mathbf{x}_0 + t \cdot \mathbf{x}_1^{\text{ref}}$
  $\mathbf{x}_1^{\text{pred}} \leftarrow \text{NP3Model}(\mathbf{x}_t, t|\mathbf{c})$ ▷ Encoder + decoder forward pass
  $\mathbf{r}_{\text{opt}}, \mathbf{t}_{\text{opt}} \leftarrow \text{GetKabschTransform}(\mathbf{x}_1^{\text{pred}}, \mathbf{x}_1^{\text{ref}}, \mathbf{w}_{\text{align}})$ ▷ Global SE(3)-superposition using backbone trace
  $\mathbf{x}_1^{\text{ref}} \leftarrow \mathbf{x}_1^{\text{ref}} \cdot \mathbf{r}_{\text{opt}} + \mathbf{t}_{\text{opt}}$
  $\mathbf{x}_1^{\text{pred}} \leftarrow \text{OptimalGraphPermutation}(\mathbf{x}_1^{\text{pred}}, \mathbf{x}_1^{\text{ref}})$ ▷ Atom permutation correction using graph isomorphism
  $\mathbf{x}_1^{\text{ref}} \leftarrow \text{STOP\_GRAD}(\mathbf{x}_1^{\text{ref}})$
  $w(t) \leftarrow 1/(\epsilon + (1-t)\sigma_{\text{data}})$ ▷ Loss weighting following [24]
  $\mathcal{L} = w(t) \cdot \text{PseudoHuberLoss}(\mathbf{x}_1^{\text{pred}}, \mathbf{x}_1^{\text{ref}}) + \text{w}_2 \cdot \text{SmoothLDDT}(\mathbf{x}_1^{\text{pred}}, \mathbf{x}_1^{\text{ref}}) + w_3 \cdot \text{FAPE}(\mathbf{x}_1^{\text{pred}}, \mathbf{x}_1^{\text{ref}})$
  Backpropagate on $\mathcal{L}$
end function

Algorithm S3 Sampling from a Globular Polymer Prior via Short Langevin Dynamics

Require: $X_0 \in \mathbb{R}^{N_{\text{atoms}} \times 3}$, $dt = 0.25$, sphere_r, res_r $= 4.0$, ent_r $= 10.0$
Require: Matrices $S_{\text{bond}}, S_{\text{entity}}, S_{\text{residue}}$ define neighbor and group structure
for $i = 1 \to 64$ **do**
  $d_{\text{bond}} \leftarrow (S_{\text{bond}} X_0) - X_0$
  $d_{\text{entity}} \leftarrow (S_{\text{entity}} X_0) - X_0$
  $d_{\text{res}} \leftarrow (S_{\text{residue}} X_0) - X_0$
  $\text{drift} \leftarrow 2\, d_{\text{bond}} + \frac{d_{\text{entity}}}{\text{ent\_r}^2} + \frac{d_{\text{res}}}{\text{res\_r}^2} - \frac{X_0}{\text{sphere\_r}^2}$
  $X_0 \leftarrow X_0 + dt \cdot \text{drift} + 2\sqrt{dt}\,\epsilon \quad \text{with } \epsilon \sim \mathcal{N}(0, I)$
end for
return X_0

Starting from random atomic positions, the algorithm iteratively applies drift updates derived from these constraints and adds small thermal noise. After a brief relaxation period, the resulting coordinates serve as a well-structured initialization that respects chemical connectivity, coarse geometric organization, and global compactness. This initialization naturally breaks inter-chain symmetry for copies of the same biopolymer. During training, at the end of the initialization we apply a greedy permutation of the prior entities such that the nearest neighbor entity index of each entity maximally agrees with the nearest neighbors derived from the ground truth.

In S1, we provide a detailed view of the NP3 model architecture. The production model uses the following configuration: 350M encoder parameters, 350M decoder parameters, 50M confidence module parameters. We highlight the following **architecture details:**

- **Standard block details**: We use a single representation embedding size of 384, pair representation embedding size of 128, atom representation embedding size of 256, and atom pair representation size of 128. For MLP blocks, we use a hidden size that is 4 times the embedding size similar to the design choices made in Llama-2 [25]. RMSNorm [26] is used for all MLP blocks; LayerNorm [27] without learnable bias is used for all MSA module blocks, pair bias attention, triangular attention, and triangular multiplication blocks.
- **U-Shaped MSA Module**: NP3 uses an MSA Module consisting of 6 blocks, each with the architecture described in [9]. The MSA latent dimension progressively scales from 32 to 512 with a multiplier of 2 after each block, optimizing memory usage with minimal representation capacity tradeoff.

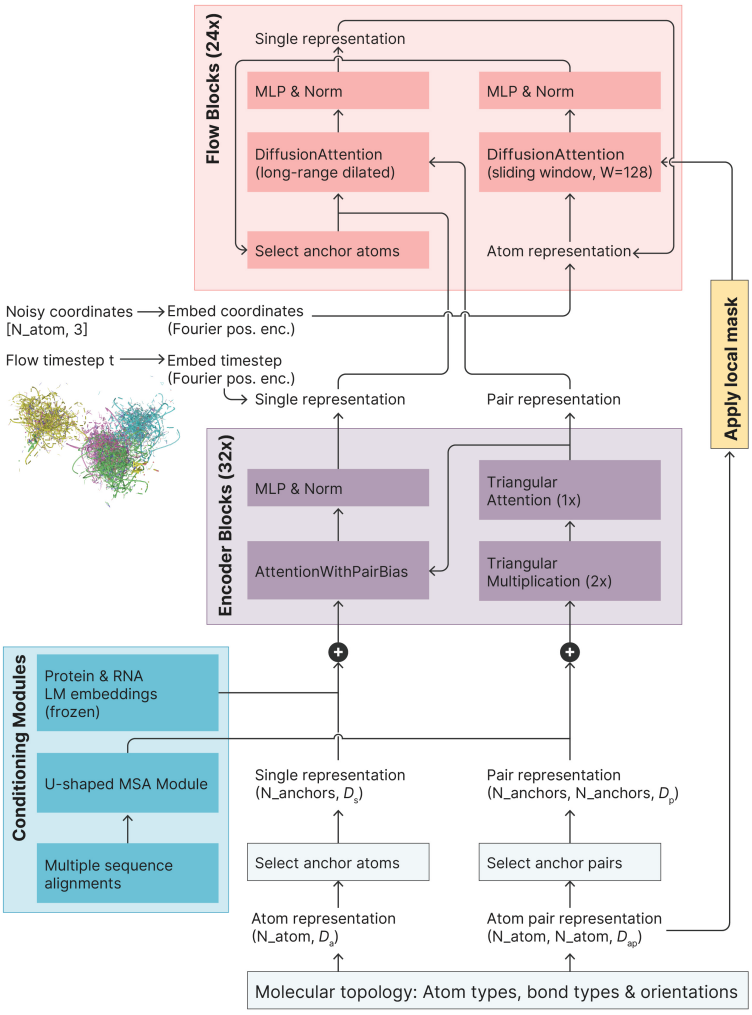

Figure S1: Architecture details. D_{a}: atom representative latent dimension; D_{s}: single representation latent dimension; D_{p}: pair representation latent dimension; D_{ap}: sliding window atom pair representation latent dimension.

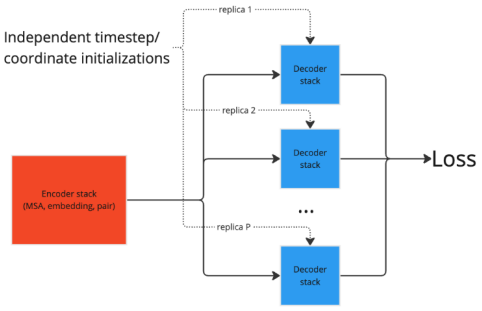

Figure S2: Schematics of decoder parallelism for training encoder-decoder structure prediction models.

Table S2: **Notation used in the symmetry-corrected alignment.**

Symbol	Meaning
$\mathbf{x}_T$	Noisy prior sample drawn at diffusion timestep T
$\hat{\mathbf{x}}_0$	One-step denoised coordinates predicted from $\mathbf{x}_T$
$\mathbf{x}$	Ground-truth (noise-free) label structure
$\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_m\}$	Set of entity groups; each $\mathcal{D}_k$ contains indices of chemically identical chains or ligands
$\mathrm{RMSD}(a, b)$	Root-mean-square deviation between coordinate sets a, b
$\mathbf{R}$, $\mathbf{t}$	Rotation matrix and translation vector returned by Kabsch alignment
$\mathbf{P}$	Block-diagonal permutation matrix acting inside each $\mathcal{D}_k$

- **Sliding window attention with sparse pair bias**: We employ an efficient implementation of sliding window attention compatible with pairwise attention bias, maintaining linear asymptotic inference time and memory scaling with respect to the number of heavy atoms. The atom pair features (bonds and orientations) are projected by a linear layer and collated to the sliding window mask. The biased sliding window masks are shared across all decoder replicas when using decoder parallelism during training.

S.5 Flow Trajectory Symmetry Correction

We incorporate a simulation-free symmetry correction module motivated to align the conditional flow trajectories with displacement interpolation paths that approximate the optimal transport geodesics between the prior and target distributions. The details are as follows:

1. **Denoise the prior with pre-computed conditioning:**

$$\hat{\mathbf{x}}_0 \;=\; f_\theta(\mathbf{x}_T,\, T).$$

2. **Rigidly align each indistinguishable entity.** For every entity group $\mathcal{D}_k$:
   (a) For each index pair $(i, j) \in \mathcal{D}_k \times \mathcal{D}_k$ (e.g., $(0,0)$, $(0,1)$, …, $(2,1)$ when entities $0, 1, 2$ are identical):
      i. Compute $(\mathbf{R}_{ij}, \mathbf{t}_{ij}) = \mathrm{Kabsch}\Big(\hat{\mathbf{x}}_0^{(i)},\, \mathbf{x}^{(j)}\Big)$
      ii. Obtain an aligned label copy $\mathbf{x}^{(j)\star} = \mathbf{R}_{ij}\,\mathbf{x}^{(j)} + \mathbf{t}_{ij}$.
   (b) *Greedy in-group permutation search* (cf. [11], Sec. 7.3): for each entity group $\mathcal{D}_\ell$, choose the permutation $\pi_\ell^\star$ that minimises

$$\sum_{i \in \mathcal{D}_\ell} \mathrm{RMSD}\Big(\mathbf{x}^{(\pi_\ell(i))\star},\, \hat{\mathbf{x}}_0^{(i)}\Big)\,.$$

3. **Global objective.** For each global alignment $(\mathbf{R}_{ij}, \mathbf{t}_{ij})$, compute

$$\mathrm{RMSD}_{\mathrm{tot}} \;=\; \mathrm{RMSD}(\mathbf{P}^\star\mathbf{x},\, \hat{\mathbf{x}}_0)\,,$$

   and retain the permutation/alignment pair $\big(\mathbf{P}^\star,\, \{\mathbf{R}_{ij}^\star,\, \mathbf{t}_{ij}^\star\}\big)$ with the lowest $\mathrm{RMSD}_{\mathrm{tot}}$.

4. **Apply the optimal permutation to the label:**

$$\tilde{\mathbf{x}} \;=\; \mathbf{P}^\star\mathbf{x}.$$

5. **Final alignment applied to the prior (optional with SE(3)-invariant losses).** Align the permuted label $\tilde{\mathbf{x}}$ to the original noisy sample $\mathbf{x}_T$:

$$(\mathbf{R}_T, \mathbf{t}_T) \leftarrow \mathrm{Kabsch}(\tilde{\mathbf{x}},\, \mathbf{x}_T)\,,$$
$$\tilde{\mathbf{x}}' = \mathbf{R}_T\tilde{\mathbf{x}} + \mathbf{t}_T.$$

The resulting $\tilde{\mathbf{x}}'$ (or $\tilde{\mathbf{x}}$ in an invariant setting) is used as the target in the flow-matching loss, reducing the transport cost between the prior and target data distribution and maintaining a consistent frame across sampling iterations.

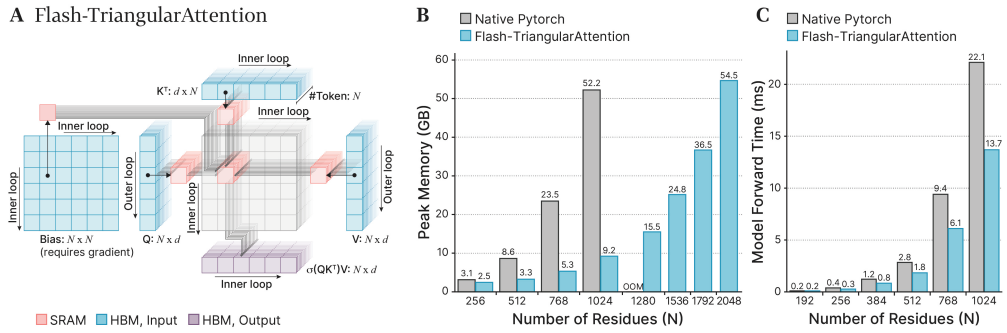

Figure S3: **Illustration of Flash-TriangularAttention kernel and experimental comparison on the peak memory usage and the inference time.** (**A**) The workflow for Flash-TriangularAttention. The main goal is to reduce peak memory usage, which is the bottleneck of triangular attention. Our implementation avoids this explicit broadcast, which is required while using other memory-efficient implementations, thus enabling longer crop size training. (**B**) Peak memory usage comparison between Flash-TriangularAttention and PyTorch built-in SDPA. Our implementation significantly reduces peak memory usage for longer residues, enabling samples with residues exceeding 1024 to run on one H100 GPU (80GB). (**C**) the inference time of the Flash-TriangularAttention and Pytorch built-in SDPA. The figure shows that our implementation significantly reduces inference runtime—for example, by 38% at a residue of 1024.

S.6 Computational Efficiency and Prediction Speed

To address the computational challenges of molecular structure prediction, we introduced Flash-TriangularAttention, a novel attention mechanism optimized for efficient training and inference (1A and S3). The method leverages implicit bias broadcasting within the kernel during the $\mathbf{QK}^T + \mathbf{Bias}$ operation, which eliminates unnecessary bias replication, significantly reducing memory overhead and enabling scalability for large structures.

In more detail, the triangular attention mechanism requires a gradient-tracked $\mathbf{Bias}$ term. The query ($\mathbf{Q}$), key ($\mathbf{K}$), and value ($\mathbf{V}$) tensors each have the shape $(B * H, N, N, d)$, and the $\mathbf{Bias}$ tensor has the shape $(B * H, N, N)$, where B is the batch size, H is the number of attention heads, N is the crop size, and d is the head dimension. In a naive approach, during the $\mathbf{QK}^T + \mathbf{Bias}$ operation, $\mathbf{Bias}$ is broadcasted from $(B * H, N, N)$ to $(B * H, N, N, N)$. Current optimized solutions, such as FlashAttention [28] and PyTorch's scaled dot-product attention (SDPA) and FlexAttention [29], either do not support gradient-tracked $\mathbf{Bias}$ or require explicit bias broadcasting which dramatically increases peak memory usage. To train models with larger structural crop sizes, we seek to avoid this broadcasting step.

Flash-TriangularAttention leverages implicit bias broadcasting within the kernel during the $\mathbf{QK}^T + \mathbf{Bias}$ operation. For each channel, the bias is loaded once from the bias matrix during the forward pass, while the backward pass accumulates gradients into a shared bias matrix. This eliminates unnecessary bias replication, significantly reducing memory overhead and enabling scalability for large values. Our Triton kernel implementation, a language and compiler for writing highly efficient custom Deep-Learning primitives, enhances efficiency, allowing for training on larger sequences and crop sizes while staying within GPU memory constraints. Experimental results underscore the effectiveness of Flash-TriangularAttention (S3). Peak memory usage decreases by $5\times$ compared to a naive implementation as crop size increases, enabling the model to scale effectively, while the forward pass inference time is improved by 50% on average. These improvements make NP3 capable of processing large, complex structures without compromising computational efficiency.

Building on these advancements, NP3 delivers predictions $15\times$ faster than inference timing statistics reported by AF3 (1B). Unlike AF3 [9] which requires approximately six A100-minutes per prediction, NP3 generates results in approximately 30 seconds using a single L40S GPU. This efficiency is also attributed to removing the expensive trunk recycling operations and cutting the number of sampling steps to 40 by leveraging optimal-transport flow samplers (S1).

S.7 Model training

Dataset Splitting and sampling:

- NP3 is trained on all PDB structures deposited **before September 1, 2020**, along with additional synthetic datasets. The cutoff date is chosen to ensure no overlap with evaluation set structures. We process training samples based on contiguous and spatial cropping centered around each to ensure maximal training data mixture diversity. In our implementation, we always sort the chain numbers based on the minimum distance to the chain or interface of interest before applying contiguous cropping.
- **Weighted sampling** of structure crops is employed to ensure inclusion of underrepresented molecule classes (e.g., nucleic acids, ligands). We mainly follow AF3 for clustering all chains and interfaces including a 40% maximum similarity threshold for sequence clustering, with the notable difference of using 60% Tanimoto similarity cutoff for ligand clustering rather than the 100% CCD identity cutoff employed in AF3.
- **Data augmentation** strategies include spatial translation, rotation, MSA row-wise subsampling, and 15% residue masking rate for language model embeddings, and 15% masking rate for atom type, bond order, and bond orientation features.

Several synthetic datasets are used to augment model training and improve generalization, with examples drawn from computational predictions:

- Protein monomer distillation: With 30% probability, we sample a structure from the OpenProteinSet dataset of OpenFold [30] predictions (N=270,000).
- Disordered distillation set for protein multimers. With 1% probability, we sample a structure from NP2 predictions of bioassemblies containing more than one polymer chain, with the motivation to reduce hallucination for intrinsically disordered regions. These samples were generated by NP2 with full-length sequence inputs and template coordinates guidance, for PDB structures with 50 or more contiguous missing residues (N=2,801).
- OrbNet conformers: With 2% probability, we sample a single molecular conformer. These conformers are obtained from a comprehensive dataset of small molecule conformations refined at the ωB97X-D3/Def2-TZVP level of DFT theory, expanded from OrbNet Denali training data (N=12,706,496). [31]

Model weight initialization: Within all attention layers, we adopt QK-normalization [32] to stabilize model training. For all PairFormer blocks, we employ Zero-residual initialization motivated by recent works [33, 34]. For Flow blocks, we employ AdaLN-Zero initialization [35] to both the attention and MLP layers, with scale and shift parameters dependent on timestep embedding.

Optimization Details: In S2, we outline the procedure of a single optimization step for training NP3. Following Lee et al. [24], we found that pseudo-Huber loss combined with SmoothLDDT introduced in AlphaFold 3 yields the most stable training. Additional loss terms include a distance-normalized Frame-Aligned Point Error (FAPE) and a distance-geometry loss to ensure accurate local and global structural alignments [1].

We modified the alignment weights $\mathbf{w}_{\mathrm{align}}$ used in the weighted Kabsch-Umeyama algorithm [36, 37], which we found important to reduce numerical variations in late-stage training:

- Atom weight = 10.0 for the ligand of interest, when the training structure is generated based on a contiguous or spatial crop around a ligand-polymer interface;
- Atom weight = 1.0 for Cα and C1" atoms, and all rest ligand atoms;
- Atom weight = 0.0 for all side-chain atoms.

Scaling studies

Optimization algorithm and training stages: The NP3 production model was trained on 56 H100 GPUs hosted on a cluster that implements the NVIDIA NCP Reference Architecture for 24 days. Training is carried out using PyTorch FSDP [39] under BF16 automatic mixed precision. Over the full course of model training, we progressively scale the structure crop size to gradually capture longer contexts. The number of decoder replicas P is tuned for each stage:

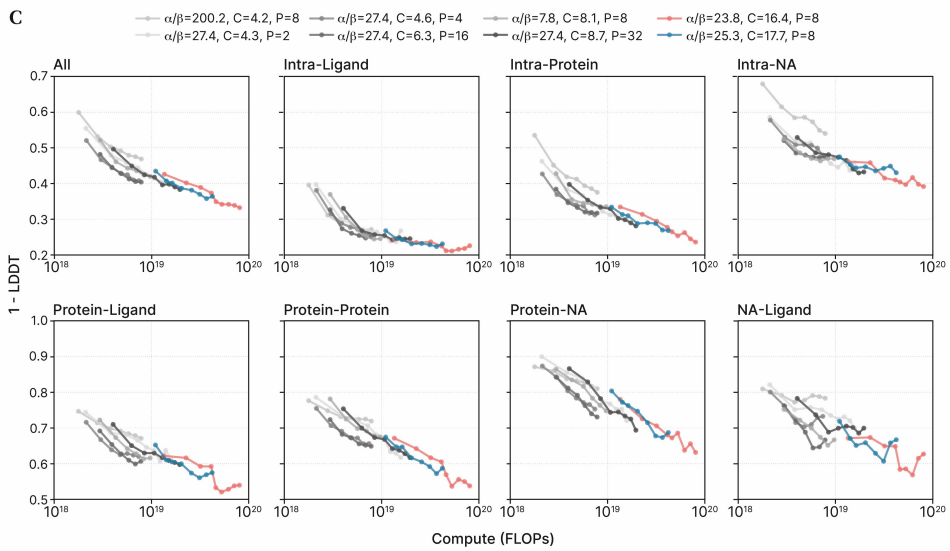

Figure S4: Model scaling behavior across molecule and interface types as reported by the relation between the average validation set local distance difference test (LDDT) [38] scores and the total training floating-point operation count (FLOPs). While intra-molecular prediction accuracy tends to saturate upon reaching a critical model size, inter-molecular interaction prediction benefits from jointly scaling the model and data up to the production model size. C: Compute in GFLOPs; P: number of decoder replicas.

- Maximum cropping size for first 80% training iterations: 384 anchors or 3072 atoms, P=32;
- For 10% of training iterations: 512 anchors or 4096 atoms, P=32;
- For 6% of training iterations: 640 anchors or 6400 atoms, P=24;
- For the last 4% of training iterations: 1024 anchors or 15360 atoms, P=8.

Activation checkpointing is employed for all training stages with anchor crop size larger than 384. Note that training on structure crops of 1024 anchors or greater sizes was only possible when using the Flash-TriangularAttention kernel, otherwise we observe GPU OOM.

All inference results used either: one H100 GPU hosted on a cluster that implements the NVIDIA NCP Reference Architecture, or one L40S GPU hosted on an AWS g6e.xlarge instance.

S.8 Training Strategy for Confidence Module

Algorithm S4 Schematics of NP3 Confidence Module Training

Require: NP3Model, ground truth $\mathbf{x}_{\text{ref}}$, conditions $\mathbf{c}$

for each iteration n **do**
  Sample $t \sim \text{Uniform}[0, 1)$ ▷Sample timestep from noise schedule
  Sample noised structure $\mathbf{x}_t \leftarrow \text{SampleFromPrior}(t|\mathbf{c})$
  Compute denoised structure $\mathbf{x}_{\text{denoised}} \leftarrow \text{NP3Model}(\mathbf{x}_t)$
  Compute flow matching loss: $\mathcal{L}_{\text{flow}}(\mathbf{x}_{\text{ref}}, \mathbf{x}_{\text{denoised}})$
  LDDT $\leftarrow \text{LDDT}(\mathbf{x}_{\text{denoised}}, \mathbf{x}_{\text{ref}})$ ▷Compute per-atom LDDT score
  DistanceError $\leftarrow |\text{Dmap}(\mathbf{x}_{\text{denoised}}) - \text{Dmap}(\mathbf{x}_{\text{ref}})|$ ▷Compute anchor distance map errors
  Pass $\mathbf{x}_{\text{denoised}}$ through confidence module and extract confidence_logits
  $\mathcal{L}_{\text{confidence}} \leftarrow \text{CrossEntropy}(\text{confidence\_logits}, \{\text{LDDT}, \text{DistanceError}\})$ ▷Compute confidence loss
  $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{flow}} + \mathcal{L}_{\text{confidence}}$ ▷ Update model with total loss
  if $n \mid 10$ **then**
    $\mathbf{x}_{\text{sampled}} \leftarrow \text{NP3SampleStructure}(\text{NP3Model}, \mathbf{c}, T = 10)$ ▷Apply rollout with limited steps
    Pass $\mathbf{x}_{\text{sampled}}$ through confidence module and extract confidence_logits
    $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{total}} + \mathcal{L}_{\text{confidence}}(\mathbf{x}_{\text{sampled}}, \mathbf{x}_{\text{ref}}, \text{confidence\_logits})$
  end if
end for

NP3 provides two types of confidence scores:

- Predicted local distance difference test (pLDDT) for each heavy atom;
- Predicted distance error (pDE) for each pair among anchors.

In NP v1-2 and AF3, confidence module training is performed by sampling bioassembly structures via full diffusion Stochastic Differential Equations (SDE) at predefined intervals N, followed by optimization steps to minimize deviations between model-estimated errors and true distance errors. We note that AF3 uses N=1 (perform diffusion rollout and optimize at every iteration), while NP1, NP2, and NP3 use N=10.

Analysis of preliminary models revealed limited sensitivity of confidence scores to sampled conformations, despite good correlation between accuracy and confidence scores over diverse targets. We reasoned that the insensitivity of confidence scores stems from under-training and insufficient paired conformational data for the same molecular topology.

The improved training algorithm integrates additional optimization steps into the standard flow-matching procedure. This approach involves sampling noisy conformations, denoising them through the model, and optimizing confidence scores based on cross-entropy loss with calculated structural errors. Noise levels are varied, producing multiple hypotheses for the same molecular topology, enabling robust confidence estimation. The algorithm thus ensures efficiency while minimizing the need for costly resampling or simulation-based inference. In the fine-tuning stages, we also incorporate a InfoNCE [40] loss commonly adopted in contrastive learning to encourage the model to discern minor prediction quality differences across multiple conformations.

S.9 Aggregated Confidence Metrics and Sample Ranking

Aggregated confidence metrics are provided for entities and interfaces:

- pLDDT for atomic-level predictions, averaged against atoms from the entity of interest.
- pDockQ for self- or inter-molecular interactions between a chain pair.

pDockQ is calculated based on the following formula:

$$\text{pDockQ} = (\text{iRMS\_scaled\_lr} + \text{iRMS\_scaled\_sr} + \text{Predicted\_Fnat})/3$$

where:

- iRMS_scaled_lr = 1 / (1 + (mean(pDE^2)) / 8.5^2)
- iRMS_scaled_sr = 1 / (1 + (mean(pDE^2)) / 1.5^2)
- Predicted_Fnat = mean(pDE < 5.0)

Ranking Methodology: Sample ranking leverages adjusted pLDDT and pDockQ values. Scores are penalized for steric clashes, ensuring physically plausible configurations.

- For a ligand of interest, LG: Score = pLDDT(LG) – 1000 * (is_clash + is_chirality_violation)
- For chain A of interest: Score = pDockQ(A, A)
- For chain pair (A, B) of interest: Score = pDockQ(A, B).

S.10 Evaluation datasets and benchmarking (NPBench)

Curation of the recent PDB dataset is loosely based on the strategy introduced by AF3 Supplemental Information Section 6.1-6.2 [9] The dataset construction started by taking all 10,192 PDB entries deposited between 2022-05-01 and 2023-01-12. We filter to non-NMR entries with resolution better than 4.5A. To choose the most representative sample from these candidates with strong protection against overlap from training set, we perform a PDB clustering on chain/ligand/interface index for all structures released before 2023-01-12, with a maximum similarity cutoff of 40% sequence identity for polymers and 0.6 Tanimoto similarity for ligands and PTMs. Then, for the recent structures, we:

- Include monomers from novel clusters (not covered by PDB clusters used during training);
- Include polymer-polymer interfaces if at least one of the polymers is in novel clusters;
- Include polymer-peptide interfaces if the receptor chain is in novel clusters;
- Include all polymer-ligand, ion, and metal interfaces with additional labels: is_ligand_unseen_CCD, is_plsite_unseen_cluster, is_covalent, is_modified_residue.
- Exclude all exotic cases not covered by criteria above. In particular, we exclude all branch entities such as multi-residue glycosylations in this benchmark release as there is no clear consensus on their representations in the context of structure prediction model inputs.

We then created the benchmark input mmCIF files by removing solvents and crystallography artifacts from the original samples. Finally, we deduplicate the dataset by only keeping the first occurrence of chain or interface from each cluster and removing all samples on which ColabFold inference did not complete within 1 GPU hour, yielding 1,143 evaluation targets in total.

Benchmarking metrics The following metrics were used to assess the accuracy of the predictions on the benchmarking datasets:

- Pocket-aligned ligand RMSD: We used a rigorous alignment protocol to compute the pocket-aligned ligand RMSD which is described as follows:
  - We define the reference pocket as all Cα atoms within 10 of the ligand atoms in the ground-truth structure. If the ligand-polymer interface is not specified (as in Posebusters benchmarking), the Cα atoms that belong to the chain with the most residues in the reference pocket are selected for alignment.
  - The same set of Cα atoms in the predicted structure are selected based on the residue indices and the optimal chain mapping between the predicted and the ground-truth structures obtained using DockQ v2. If the optimal chain mapping is not provided (as in Posebusters benchmarking), we will align each chain in the predicted pocket, defined in the same way as the reference pocket, and find the minimal RMSD.
  - We use PyMOL [41] to align the selected Cα atoms and all ligand atoms in the predicted structure to those in the ground-truth structure with zero refinement cycles.
  - The pocket-aligned ligand RMSD is computed using RDKit [42] CalcRMS between the aligned ligand and the reference ligand structures.
- DockQ v2: The DockQ scores are computed for all polymer-polymer interfaces defined in the Recent PDB Evaluation Set using DockQ v2 [43] package.

- Generalized RMSD: This is computed for each ligand-polymer interface defined in the Recent PDB Evaluation Set as follows:
  - We first find the optimal chain mapping between the predicted and the ground-truth structures using DockQ v2.
  - We use the same algorithm as described above to compute the pocket-aligned ligand RMSD for a given ligand-polymer interface.
  - For structures with multiple identical ligands, we perform an exhaustive search over all pairwise permutations between the predicted and reference ligands (up to 10 ligands) and find the minimal pocket-aligned ligand RMSD. This leads to N^2 pocket-aligned RMSD calculations where N is the number of ligands corresponding to the same CCD code in a ligand-polymer interface.
- TM-score: This is computed using the USalign package. The TM-scores for both entire protein and each chain are calculated using the optimal chain mapping obtained from DockQ.
- LDDT (local distance difference test): We compute the all-atom and backbone-only LDDT scores for both entire structure and each chain in the predictions with a custom implementation.

S.11 Conformational change benchmarking (ConfBench)

S.11.1 ConfBench Dataset Curation

The development of ConfBench required establishing a systematic protocol for identifying and validating apo-holo protein structure pairs that exhibit measurable conformational changes. Here we detail the technical implementation of our structure pair identification pipeline and the specific criteria used for quality control.

Structure Pair Database Integration: The curation pipeline was built on PLINDER's structured database schema, leveraging its system_id indexing that encodes both PDB identifiers and chain information in a standardized format (PDB__model__protein.chain__ligand.chain). We developed a two-pass search algorithm that first identifies potential apo-holo relationships through system_id queries, then validates these relationships through detailed structural analysis.

Chain Mapping Protocol: For structures with multiple chains, our mapping algorithm evaluates all possible chain combinations using a hierarchical sorting approach. The best mapping is selected by first sorting by `pocket_fident` (binding site sequence conservation), followed by `lddt` (overall structural similarity) and `bb_lddt` (backbone conformation) as successive tiebreakers. This prioritizes binding site similarity while still considering global structural features when discriminating between similar candidates.

Conformational Change Validation: Structure pairs were required to exhibit at least one of the following:

- Global RMSD > 1.5Å
- Pocket Cα RMSD > 1.5Å
- Pocket all-atom RMSD > 1.5Å

S.11.2 ConfBench Scoring Protocol

The ConfBench scoring protocol introduces several key technical advances to address quantifying the quality of prediction of ligand-induced conformational changes:

- A robust protocol for identifying and mapping binding site residues between structures that accounts for numbering discontinuities and chain breaks
- A multi-level structural comparison approach that evaluates both global and local conformational changes
- A novel scoring function that enables fair comparison across protein systems with varying magnitudes of conformational change

Pocket Detection and Mapping: Pocket residues were identified using a distance-based approach in PyMOL, selecting protein residues within 10Å of the ligand of interest. To ensure consistent pocket mapping between structures, we implemented a robust sequence alignment-based protocol that accounts for potential numbering discontinuities and chain breaks. The mapping algorithm uses local sequence alignment with the Bio.Align.PairwiseAligner module, prioritizing sequence identity while handling insertions and deletions. Pocket mappings were validated by requiring at least 50% of the reference pocket residues to be successfully mapped to maintain structural context.

RMSD Calculations and Structural Alignment: Three levels of structural comparison were implemented:

- Global alignment using all Cα atoms
- Pocket-specific alignment using only pocket Cα atoms
- Complete pocket alignment including both backbone and sidechain heavy atoms

Each alignment was performed using PyMOL's align command with cycles=0 to prevent local optimization bias. For multi-chain structures, chain mapping was determined through a hierarchical approach - first attempting sequence-based matching, then falling back to spatial proximity to the ligand if necessary.

Conformational Score Calculation: The conformational scoring function was designed to be:

- Symmetric with respect to the reference structures
- Normalized to account for varying magnitudes of conformational change
- Invariant to global rigid body movements

For each alignment type i (global, pocket, pocket+sidechains), the score is calculated as:

$$\text{score\_apo\_i} = \frac{\text{RMSD\_holo} - \text{RMSD\_apo}}{\sqrt{\frac{1}{2}(\text{RMSD\_holo}^2 + \text{RMSD\_apo}^2 + \text{RMSD\_ref}^2)}} \tag{4}$$

$$\text{score\_holo\_i} = \frac{\text{RMSD\_apo} - \text{RMSD\_holo}}{\sqrt{\frac{1}{2}(\text{RMSD\_apo}^2 + \text{RMSD\_holo}^2 + \text{RMSD\_ref}^2)}} \tag{5}$$

where RMSD_ref is the RMSD between reference apo and holo structures, and RMSD_apo/RMSD_holo are RMSDs between the query structure and respective references. This formulation ensures scores are bounded and comparable across different protein systems regardless of their absolute conformational differences.

S.11.3 Additional NP3 and AF2-M results

When limiting to "big" ConfBench wins (i.e. ConfBench scores » 0), NP3 achieves 47.7% of wins above ConfBench score of 0.5, whereas AF2-M achieves 32.22%.

Detailed distributions of prediction accuracy are presented through histograms of ConfBench scores and subsequent kernel density estimation plots comparing AlphaFold2-Multimer and NP3 performance, providing comprehensive visualization of the complete prediction landscape beyond aggregate statistics.

For additional clarity, histograms of NP3 raw ConfBench score distribution data are shared below, detailing the magnitude of win rates for the apo and holo dataset discussed in the main text.

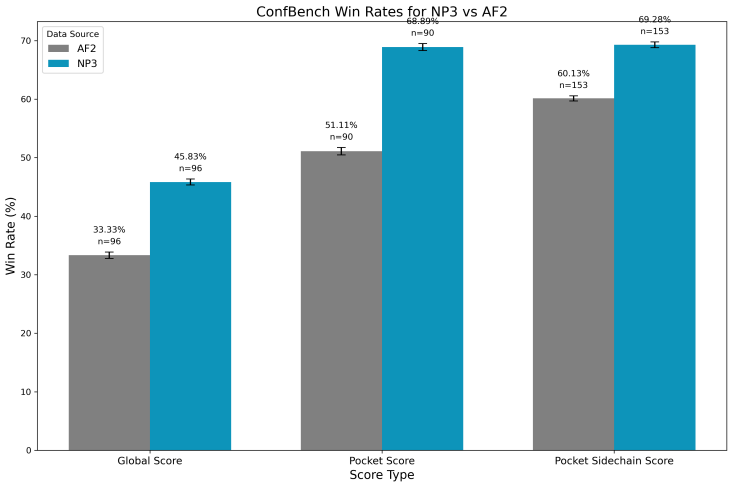

Figure S5: Win rates for all ConfBench score types on query conformation = apo linkages where reference RMSD > 1.5 Å. A ConfBench win is defined as query structure score > 0.

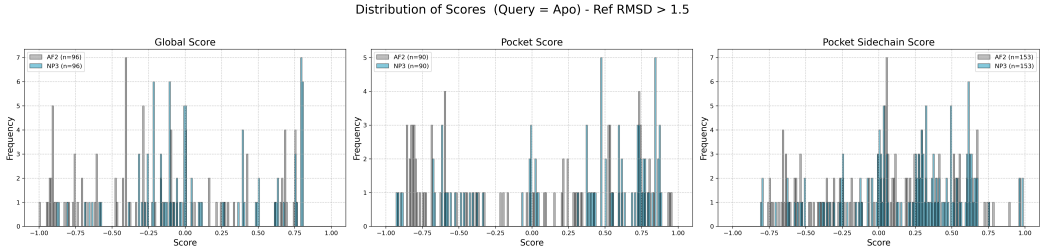

Figure S6: Distribution of ConfBench score values for NP3 and AF2 multimer models with query conformation = apo and a reference RMSD > 1.5 Å. The histograms display the frequency of scores across all metrics (global, pocket, and pocket-sidechain) within the range of -1 to 1.

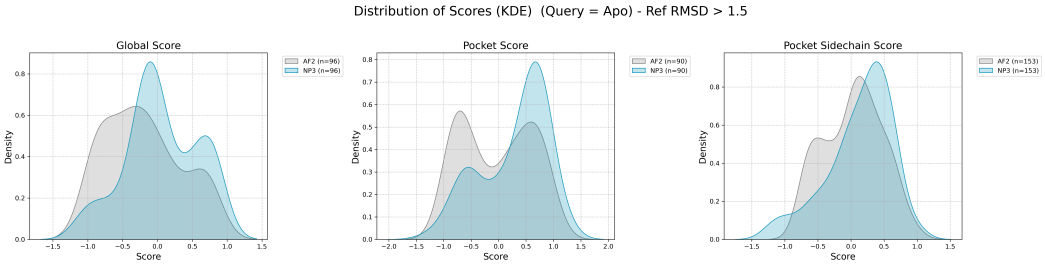

Figure S7: Kernel density estimation plots depicting the score probability densities for NP3 and AF2 models where query conformation = apo and reference RMSD > 1.5 Å.

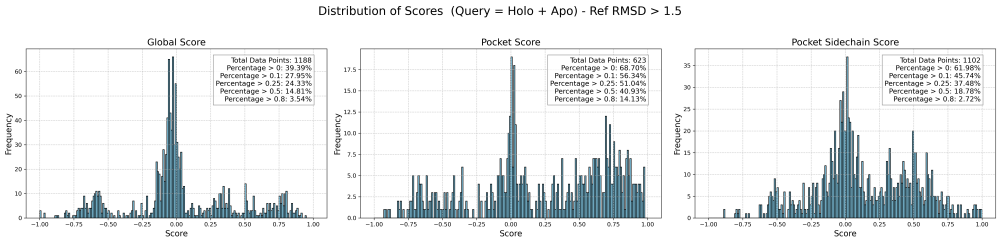

Figure S8: Distribution of ConfBench score values for NP3 with query conformation = apo or holo and a reference RMSD > 1.5 Å. The histograms display the frequency of scores across all metrics (global, pocket, and pocket-sidechain) within the range of -1 to 1.

References

1. Qiao, Z., Nie, W., Vahdat, A., Miller III, T. F. & Anandkumar, A. State-specific protein–ligand complex structure prediction with a multiscale deep generative model. *Nature Machine Intelligence* **6,** 195–208. https://www.nature.com/articles/s42256-024-00792-z (2024).
2. Buttenschoen, M., M. Morris, G. & M. Deane, C. PoseBusters: AI-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science* **15,** 3130–3139. https://pubs.rsc.org/en/content/articlelanding/2024/sc/d3sc04185a (2024).
3. Das, R. *et al.* Assessment of three-dimensional RNA structure prediction in CASP15. *bioRxiv,* 2023.04.25.538330. https://www.biorxiv.org/content/10.1101/2023.04.25.538330v3 (Oct. 2023).
4. Eberhardt, J., Santos-Martins, D., Tillack, A. F. & Forli, S. AutoDock Vina 1.2.0: New Docking Methods, Expanded Force Field, and Python Bindings. *Journal of Chemical Information and Modeling*. https://pubs.acs.org/doi/abs/10.1021/acs.jcim.1c00203 (July 2021).
5. Verdonk, M. L., Cole, J. C., Hartshorn, M. J., Murray, C. W. & Taylor, R. D. Improved protein–ligand docking using GOLD. *Proteins: Structure, Function, and Bioinformatics* **52,** 609–623. https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.10465 (2003).
6. Stärk, H., Ganea, O., Pattanaik, L., Barzilay, D. R. & Jaakkola, T. *EquiBind: Geometric Deep Learning for Drug Binding Structure Prediction* in *Proceedings of the 39th International Conference on Machine Learning* (PMLR, June 2022), 20503–20521. https://proceedings.mlr.press/v162/stark22b.html.
7. Lu, W. *et al.* TANKBind: Trigonometry-Aware Neural NetworKs for Drug-Protein Binding Structure Prediction. *Advances in Neural Information Processing Systems* **35,** 7236–7249. https://proceedings.neurips.cc/paper_files/paper/2022/hash/2f89a23a19d1617e7fb16d4f7a049ce2-Abstract-Conference.html (Dec. 2022).
8. Corso, G., Stärk, H., Jing, B., Barzilay, R. & Jaakkola, T. S. *DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking* in *International Conference on Learning Representations* (Sept. 2022). https://openreview.net/forum?id=kKF8_K-mBbS.
9. Abramson, J. *et al.* Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature* **630,** 493–500. https://www.nature.com/articles/s41586-024-07487-w (June 2024).
10. Mirdita, M. *et al.* ColabFold: making protein folding accessible to all. *Nature Methods* **19,** 679–682. https://www.nature.com/articles/s41592-022-01488-1 (June 2022).
11. Evans, R. *et al. Protein complex prediction with AlphaFold-Multimer* Mar. 2022. https://www.biorxiv.org/content/10.1101/2021.10.04.463034v2.
12. Chen, K., Zhou, Y., Wang, S. & Xiong, P. RNA tertiary structure modeling with BRiQ potential in CASP15. *Proteins: Structure, Function, and Bioinformatics* **91,** 1771–1778. https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.26574 (2023).
13. Janin, J. *et al.* CAPRI: a critical assessment of predicted interactions. *Proteins: Structure, Function, and Bioinformatics* **52,** 2–9. https://onlinelibrary.wiley.com/doi/10.1002/prot.10381 (2003).
14. Durairaj, J. *et al. PLINDER: The protein-ligand interactions dataset and evaluation resource* July 2024. https://www.biorxiv.org/content/10.1101/2024.07.17.603955v3.
15. Eddy, S. R. Accelerated Profile HMM Searches. *PLOS Computational Biology* **7,** e1002195. https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1002195 (Oct. 2011).
16. Suzek, B. E. *et al.* UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics* **31,** 926–932. https://doi.org/10.1093/bioinformatics/btu739 (Mar. 2015).
17. Steinegger, M. *et al.* HH-suite3 for fast remote homology detection and deep protein annotation. *BMC Bioinformatics* **20,** 473. https://doi.org/10.1186/s12859-019-3019-7 (Sept. 2019).
18. Steinegger, M., Mirdita, M. & Söding, J. Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nature Methods* **16,** 603–606. https://www.nature.com/articles/s41592-019-0437-4 (July 2019).

19. Lin, Z. *et al.* Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379,** 1123–1130. https://www.science.org/doi/10.1126/science.ade2574 (Mar. 2023).
20. Penić, R. J., Vlašić, T., Huber, R. G., Wan, Y. & Šikić, M. *RiNALMo: General-Purpose RNA Language Models Can Generalize Well on Structure Prediction Tasks* Nov. 2024. http://arxiv.org/abs/2403.00043.
21. Bryant, P., Pozzati, G. & Elofsson, A. Improved prediction of protein-protein interactions using AlphaFold2. *Nature Communications* **13,** 1265. https://www.nature.com/articles/s41467-022-28865-w (Mar. 2022).
22. Shirts, M. R. *et al.* Lessons learned from comparing molecular dynamics engines on the SAMPL5 dataset. *Journal of Computer-Aided Molecular Design* **31,** 147–161. https://doi.org/10.1007/s10822-016-9977-1 (Jan. 2017).
23. Esser, P. *et al. Scaling Rectified Flow Transformers for High-Resolution Image Synthesis* in *International Conference on Machine Learning* (June 2024). https://openreview.net/forum?id=FPnUhsQJ5B.
24. Lee, S., Lin, Z. & Fanti, G. *Improving the Training of Rectified Flows* Oct. 2024. http://arxiv.org/abs/2405.20320.
25. Touvron, H. *et al. Llama 2: Open Foundation and Fine-Tuned Chat Models* July 2023. http://arxiv.org/abs/2307.09288.
26. Zhang, B. & Sennrich, R. *Root Mean Square Layer Normalization* in *Advances in Neural Information Processing Systems* **32** (Curran Associates, Inc., 2019). https://proceedings.neurips.cc/paper/2019/hash/1e8a19426224ca89e83cef47f1e7f53b-Abstract.html.
27. Ba, J. L., Kiros, J. R. & Hinton, G. E. *Layer Normalization* July 2016. http://arxiv.org/abs/1607.06450.
28. Dao, T., Fu, D. Y., Ermon, S., Rudra, A. & Ré, C. *FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness* in (arXiv, June 2022). https://papers.nips.cc/paper_files/paper/2022/hash/67d57c32e20fd0a7a302cb81d36e40d5-Abstract-Conference.html.
29. Li, J. *et al. FlexAttention for Efficient High-Resolution Vision-Language Models* in *Computer Vision – ECCV 2024* (eds Leonardis, A. *et al.*) (Springer Nature Switzerland, Cham, 2025), 286–302. ISBN: 978-3-031-72698-9. https://link.springer.com/chapter/10.1007/978-3-031-72698-9_17.
30. Ahdritz, G. *et al.* OpenFold: retraining AlphaFold2 yields new insights into its learning mechanisms and capacity for generalization. *Nature Methods* **21,** 1514–1524. https://www.nature.com/articles/s41592-024-02272-z (Aug. 2024).
31. Christensen, A. S. *et al.* OrbNet Denali: A machine learning potential for biological and organic chemistry with semi-empirical cost and DFT accuracy. *The Journal of Chemical Physics* **155,** 204103. https://doi.org/10.1063/5.0061990 (Nov. 2021).
32. Henry, A., Dachapally, P. R., Pawar, S. & Chen, Y. *Query-Key Normalization for Transformers* Oct. 2020. http://arxiv.org/abs/2010.04245.
33. Jumper, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* **596,** 583–589. https://www.nature.com/articles/s41586-021-03819-2 (Aug. 2021).
34. Zhao, J., Schaefer, F. T. & Anandkumar, A. *ZerO initialization: Initializing residual networks with only zeros and ones* in *Transactions on Machine Learning Research* (2022). https://openreview.net/forum?id=1AxQpKmiTc.
35. Peebles, W. & Xie, S. *Scalable Diffusion Models with Transformers* Mar. 2023. http://arxiv.org/abs/2212.09748.
36. Kabsch, W. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A* **32,** 922–923. https://onlinelibrary.wiley.com/doi/abs/10.1107/S0567739476001873 (1976).
37. Umeyama, S. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13,** 376–380. https://ieeexplore.ieee.org/document/88573 (Apr. 1991).

38. Mariani, V., Biasini, M., Barbato, A. & Schwede, T. lDDT: a local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics* **29,** 2722–2728. https://doi.org/10.1093/bioinformatics/btt473 (2013).
39. Ansel, J. *et al. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation* in *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2* **2** (Association for Computing Machinery, New York, NY, USA, Apr. 2024), 929–947. ISBN: 9798400703850. https://dl.acm.org/doi/10.1145/3620665.3640366.
40. Van den Oord, A., Li, Y. & Vinyals, O. *Representation Learning with Contrastive Predictive Coding* 2018. http://arxiv.org/abs/1807.03748.
41. DeLano, W. Pymol: An open-source molecular graphics tool. *CCP4 Newsletter on Protein Crystallography* **40,** 82–92. http://www.ccp4.ac.uk/newsletters/newsletter40/11_pymol.pdf (2002).
42. Landrum, G. RDKit: Open-Source Cheminformatics Software. https://github.com/rdkit/rdkit/releases/tag/Release_2024_09_1 (2024).
43. Mirabello, C. & Wallner, B. *DockQ v2: Improved automatic quality measure for protein multimers, nucleic acids, and small molecules* June 2024. https://www.biorxiv.org/content/10.1101/2024.05.28.596225v1.