PNAS

RESEARCH ARTICLE | BIOPHYSICS AND COMPUTATIONAL BIOLOGY

An end-to-end deep learning method for protein side-chain packing and inverse folding

Matthew McPartlon[a] and Jinbo Xu[b,c,1]

Edited by William DeGrado, University of California San Francisco, San Francisco, CA; received September 28, 2022; accepted April 24, 2023

Protein side-chain packing (PSCP), the task of determining amino acid side-chain conformations given only backbone atom positions, has important applications to protein structure prediction, refinement, and design. Many methods have been proposed to tackle this problem, but their speed or accuracy is still unsatisfactory. To address this, we present AttnPacker, a deep learning (DL) method for directly predicting protein side-chain coordinates. Unlike existing methods, AttnPacker directly incorporates backbone 3D geometry to simultaneously compute all side-chain coordinates without delegating to a discrete rotamer library or performing expensive conformational search and sampling steps. This enables a significant increase in computational efficiency, decreasing inference time by over 100× compared to the DL-based method DLPacker and physics-based RosettaPacker. Tested on the CASP13 and CASP14 native and nonnative protein backbones, AttnPacker computes physically realistic side-chain conformations, reducing steric clashes and improving both rmsd and dihedral accuracy compared to state-of-the-art methods SCWRL4, FASPR, RosettaPacker, and DLPacker. Different from traditional PSCP approaches, AttnPacker can also codesign sequences and side chains, producing designs with subnative Rosetta energy and high in silico consistency.

protein sidechain packing | machine learning | equivariant neural network | protein structure prediction

Significance

All amino acids are bound to unique chemical groups (side chains), which interact to facilitate protein folding. Thus, accurate modeling of protein side-chain conformations is essential for accurate protein structure prediction and design. Although many methods have been proposed to address this problem, their performance often suffers from simplified modeling assumptions or long inference times. This work provides a fast and precise machine learning approach that jointly models side-chain interactions and directly predicts physically realistic packings, having few atom clashes and ideal bond lengths and angles. Compared to existing methods, our approach exhibits improved accuracy and efficiency for both native and nonnative backbone structures.

Protein side-chain packing (PSCP) involves predicting the three-dimensional coordinates of a protein's side-chain atoms given the backbone conformation (coordinates) and primary sequence. This problem has important applications to protein structure prediction (1–3), design (4–7), and protein–protein interactions (8, 9). Traditional methods for PSCP rely on minimizing some energy function over a set of rotamers from a fixed library (10–17). These methods tend to differ primarily in their choice of rotamer library (18–20), energy function (21–24), and energy minimization procedure. Although many of them have shown success, the use of search heuristics coupled with a discrete sampling procedure could ultimately limit their accuracy. Currently, the fastest among them [OSCAR-star (12), FASPR (10), SCWRL4 (15)] do not employ deep learning (DL) and are rotamer library-based.

Aside from traditional approaches, several machine learning (ML) methods have been developed for PSCP (11, 16, 25–29). One of the earliest methods, SIDEPro (25), attempts to learn an additive energy function over pairwise atomic distances for each side-chain rotamer. This is achieved by training a family of 156 feedforward networks—one for each amino acid and contacting atom type. The rotamer with the lowest energy is then selected. DLPacker (26) formulates PSCP as an image-to-image transformation problem and employs a deep U-net style neural network for the task. The method iteratively predicts side-chain atom positions using a voxelized representation of the residue's local environment as input, and outputs densities for the respective side-chain atoms. To convert the network's output to coordinates, the densities are then compared to a rotamer database, and the closest conformation is selected. The most recent version of OPUS-Rota4 (28) uses a pipeline of multiple deep networks to predict side-chain coordinates. The method uses predicted side-chain dihedral angles to obtain an initial model and then applies gradient descent on predicted distance constraints to obtain a final structure. It is worth noting that OPUS-Rota4, to the best of our knowledge, is the only ML-based PSCP method that directly utilizes multiple sequence alignments (MSAs) as part of its input. Apart from methods specifically designed for PSCP, many protein structure prediction methods also produce side-chain coordinate information. AlphaFold2 (30) and RosettaFold (31) are able to produce highly accurate structures from primary sequence and MSA information along with optional template structures. Another class of structure predictors, including ESMFold (32) and OmegaFold (33),

Author affiliations: [a]Department of Computer Science, Physical Sciences, The University of Chicago, Chicago, IL 60637; [b]Toyota Technical Institute of Chicago, Chicago, IL 60637; and [c]MoleculeMind Inc., Beijing 100086, China

Author contributions: M.M. and J.X. designed research; M.M. performed research; M.M. contributed new reagents/analytic tools; M.M. and J.X. analyzed data; and M.M. wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission.

[1]To whom correspondence may be addressed. Email: jinbo.xu@gmail.com.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2216438120/-/DCSupplemental.

Published May 30, 2023.

rely on massive pretrained language models to derive structure from primary sequence alone.

In a complementary line of research, a spate of ML methods have been developed for fixed-backbone protein design (34–38) (often referred to as *protein inverse folding*). Akin to PSCP, fixed-backbone design methods attempt to find a protein sequence that folds to a given backbone structure. Although the two tasks differ in output, both problems require reasoning around sequence–structure compatibility, and we hypothesized that a single architecture could effectively model both tasks.

Here, we present AttnPacker, a deep architecture for PSCP. Our method is inspired by recent advancements in modeling three-dimensional data and architectures for protein structure prediction—most notably AlphaFold2 (30), Tensor Field Networks (TFN) (39), and the SE(3)-Transformer (40). By modifying and combining components of these architectures, we are able to significantly outperform other PSCP methods, in terms of speed, memory efficiency, side-chain atom clashes, and overall accuracy, using only features derived directly from primary sequence and backbone coordinates.

Specifically, we introduce a deep graph transformer architecture leveraging both geometric and relational aspects of PSCP. Inspired by AlphaFold2, we propose *locality-aware triangle updates* to refine our pairwise features using a graph-based framework for computing triangle attention and multiplication updates. By doing this, we are able to significantly reduce the memory and build higher capacity models. In addition, we explore several SE(3)-equivariant attention mechanisms and propose an equivariant transformer architecture for learning from 3D points.

Our method, AttnPacker, guarantees physically realistic rotamers with negligible deviation from ideal bond lengths and angles and minimal steric hindrance. AttnPacker also predicts per-residue confidence scores, which correlate strongly with side-chain rmsd and χ_1 dihedral error. AttnPacker significantly outperforms traditional PSCP methods on CASP13 and CASP14 native backbones with average reconstructed rmsd over 18% lower than the next best method on each test set. AttnPacker also surpasses the DL method DLPacker, with an 11% lower average rmsd while also significantly improving side-chain dihedral accuracy. In addition to accuracy, we also show that AttnPacker consistently produces packings with notably fewer atom clashes than other methods.

To demonstrate our method's broader applicability to protein design, we analyze AttnPacker's efficacy in 1) handling nonnative backbones and 2) simultaneously designing primary sequence and packing side chains. We show that AttnPacker performs favorably to state-of-the-art structure prediction methods AlphaFold2, RosettaFold, and OmegaFold, predicting accurate conformations from predicted backbone structures. For the second task, we train a variant of AttnPacker for codesign, which achieves native sequence recovery competitive with state-of-the-art methods, while also producing highly accurate packings. We validate these designs in silico with Rosetta and find that AttnPacker-designed structures often yield subnative Rosetta energy.

1. Methods

We now describe our model architecture, input features, and training procedure. AttnPacker is outlined schematically in Fig. 1. Additional details can be found in *SI Appendix*, as referenced in each section.

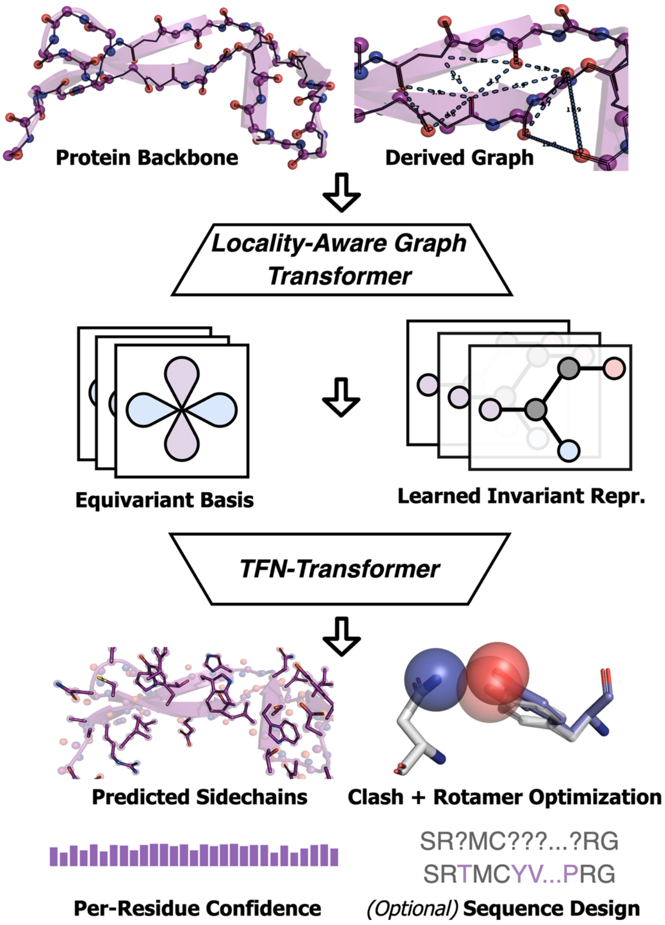

Fig. 1. Overview of AttnPacker. Our method takes protein backbone coordinates and primary sequence as input and derives a spatial feature graph and equivariant basis based on coordinate information. The feature graph is processed by an invariant graph-transformer module and then passed to an equivariant TFN-Transformer which outputs predicted side-chain coordinates, per-residue confidence scores, and optionally, a designed sequence. Predicted coordinates are postprocessed to remove any steric clashes and to ensure idealized geometry.

A. Input Representation. All input features are derived from primary sequence and backbone heavy-atom coordinates. We are given a protein backbone with L residues, indexed by i, and labeled 1 to L. The backbone is represented by its amino acid sequence $s_1, \ldots s_L$, and atom coordinates $\vec{c}^{\,a}_1, \ldots, \vec{c}^{\,a}_L \in \mathbb{R}^3$, where $a \in \{N, C\alpha, C, O\}$ is the atom type. The input to our network is a featurized graph

$$\mathcal{G}_\theta = \left(\{\mathbf{x}_i\}_{i=1..L}, \{\mathbf{e}_{ij} : d(i,j) < \theta\}_{i,j=1..L}\right) \qquad [1]$$

where $d(x, y)$ denotes the distance between $C\alpha$ atoms for residues x and y, and θ is a predefined threshold.

Residue features consist of encodings for amino acid type, backbone dihedral angles, relative sequence position, and the number of atoms in the corresponding residue's immediate microenvironment. Pair features encode interresidue orientation as defined by ref. 41, along with interatom distance, and a joint encoding of relative sequence separation and amino acid type. A detailed overview of input features and the input embedding procedure can be found in *SI Appendix*, section C.

B. Network Architecture. Our network consists of two modules, as popularized in refs. 31 and 30. The first is a *Locality Aware Graph Transformer* which selectively updates node and pair features, and the second is a TFN-based SE(3)-equivariant transformer inspired by Fuchs et al. (40). The second module differs from the first in that the TFN transformer does not update pair features. Instead, pair features are used to generate pairwise kernels transforming residue features at each attention layer. We hypothesized that enriching pair features with a separate module may help improve performance, which is justified by our ablation studies (*SI Appendix*, section I). In addition, the second module operates on a fixed basis defined by the input backbone coordinates. Since the backbone coordinates are also fixed during inference, this architecture is a natural choice for side-chain prediction. The choice of components further allows us to predict the 3D coordinates of all side-chain atoms for a given protein without relying on rotamer libraries or expensive conformational sampling. Additional details can be found in *SI Appendix*, sections G.2 and G.3.

B.1. Locality aware graph transformer. Before explicitly incorporating backbone coordinates into our model, we first generate invariant node and pair representations with a deep graph transformer network. This module utilizes the geometry of the input backbone to update node and pair features at each attention layer selectively. This component draws two submodules introduced in AlphaFold2's Evoformer for processing MSA and pair features, namely the use of pair-biased self-attention and triangle updates to revise pairwise features. Different from AlphaFold2, we make use of backbone coordinate information at each attention layer. First, we restrict residue attention to the top-k spatially closest nearest neighbors as defined by $C\alpha$ distance. Next, we introduce *Locality Aware Triangle Updates*, which nontrivially restrict multiplication and attention to a subset of predefined triangles.

Pair updates in the Evoformer module of AlphaFold2 map all triplets of residues to their corresponding pair features. This results in $\Omega(L^3)$ time complexity per update block and an additional $\Omega(L^3)$ space to store attention logits. Distinct from the protein structure prediction setting, PSCP requires the protein backbone coordinates as input. With this information, we are able to perform triangle updates in a geometrically motivated manner by considering only residue triples whose atoms are spatially close; providing significant savings in both time and space complexity. More formally, the update for pair feature e_{ij} is a function of only the edges e_{ik} and e_{jk} for which the maximum distance between the corresponding residues is bounded.

$$e_{ij}^{(\ell+1)} = f\left(e_{ij}^{(\ell)}, \left\{\left(e_{ik}^{(\ell)}, e_{jk}^{(\ell)}\right) : \max\left(d_{ij}, d_{ik}\right) < \theta\right\}\right). \qquad [2]$$

Here, $e^{(\ell)}$ denotes the pair features at layer ℓ, d_{xy} denotes the $C\alpha$ atom distance between residues x and y, and θ is a predefined distance threshold.

In *SI Appendix*, section G.1, we provide a more formal overview of our triangle update procedure and show that the procedure results in an $O(L)$ improvement over the standard implementation for both time and space complexity.

B.2. TFN transformer. The second module of our architecture is an SE(3)-equivariant neural network, which takes the output of the first one, along with the protein's backbone C_α-coordinates, to simultaneously predict all side-chain coordinates. Our equivariant architecture is derived from the SE(3)-Transformer introduced by Fuchs et al. Similar to their model, our attention blocks use TFNs to produce keys and values for scalar and point features. But our implementation differs in a few key areas. First, we use shared attention with cosine similarity (*SI Appendix*, Table S8). That is, we combine the attention logits of each feature type (e.g., scalars and points) to produce shared attention weights. Furthermore, in each attention block, we augment the input to the TFN radial kernel with pairwise distance information between hidden coordinate features and make further use of the pair features to bias attention weights and update scalar features. This process is fully described in *SI Appendix*, Algorithms 3 and 4. The architecture itself is outlined in *SI Appendix*, Fig. S5. Implementation details can be found in *SI Appendix*, section G, and ablation studies guiding these decisions are given in *SI Appendix*, section I.

B.3. Sequence design. On top of PSCP, we design a variant of AttnPacker to simultaneously predict side-chain conformations and residue types from partial sequence information. Both the PSCP and inverse-folding tasks involve reasoning over sequence and structure compatibility, and we hypothesize that a single architecture could be used for both tasks. To enable this, we randomly mask and corrupt subsets of the input sequence during training and ask our model to predict both the amino acid type and side-chain conformation for these missing residues. For sequence generation, we perform Gibbs sampling, as described in ref. 42. Full details for this modification are provided in *SI Appendix*, section J. We also describe how to condition on both partial sequence and rotamer conformations in *SI Appendix*, section C.2.

B.4. Per-residue confidence prediction. Along with side-chain conformations, AttnPacker also outputs per-residue estimates of sequence prediction and packing quality. This is useful in the protein design setting, where the practitioner may be interested in determining the degree of sequence–structure compatibility. To estimate packing quality, we predict a per-residue local distance dissimilarity test (plDDT) score for a predefined atom in each amino acid side-chain. For each residue type, we selected the most distal atom in the χ_1 or χ_2 dihedral group, with $C\alpha$ and $C\beta$ chosen for Gly and Ala, respectively (*SI Appendix*, section D). In Fig. 2, we show that this metric correlates strongly with ground-truth side-chain plDDT as well as side-chain rmsd and χ_1 dihedral error. Furthermore, we remark that confidence predictions can be used to produce accurate side-chain packings of large proteins. In *SI Appendix*, section F, we describe how AttnPacker can be used to pack proteins of arbitrary length by segmenting the input backbone into overlapping regions and selecting rotamers based on predicted confidence scores.

When using AttnPacker to design sequence and rotamers jointly, we also output sequence scores obtained by averaging normalized negative log probabilities of predicted residue types. Shown in *SI Appendix*, Fig. S8, we find a strong correlation between the sequence score and native sequence recovery ($\rho = 0.90$) and between predicted residue lDDT and sequence recovery ($\rho = 0.83$).

B.5. Rotamer and clash optimization. As our method directly predicts side-chain atom coordinates, there is no guarantee that bond lengths and angles between predicted atoms will be physically realistic. To ensure stereochemical feasibility, we develop a postprocessing procedure that projects side-chain coordinates onto a continuous rotamer library and simultaneously minimizes steric clashes. To avoid limitations inherent in discrete rotamer sampling, we develop a continuous mapping from dihedrals to rotamers and from atomic distances to steric clashes. This enables us to use gradient descent in optimizing our conformations while also providing guarantees of idealized rotamer geometry. Although our objective function is nonconvex, we show that proper rotamer initialization results in a minimal increase in per-residue rmsd. Further details on this procedure can be found in *SI Appendix*, section E. A detailed analysis of running time is shown in *SI Appendix*, Fig. S2.

C. Training Details. We trained and validated all models using the BC40 dataset.* The dataset contains roughly 39k proteins which are selected from the PDB database by 40% sequence identity cutoff. We filtered entries from this dataset so that no single target had greater than 40% sequence identity with any target in our test set and used a 90–10 split for training and validation. We train our models for ten epochs using the Adam (43) optimizer with default settings and a learning rate of 10^{-3}. To save memory, we crop all training examples to at most 400 residues by randomly selecting a single contiguous region. A complete summary of training data, optimization settings, and training procedure is given in *SI Appendix*, section B.

D. Test Datasets. Our native backbone data consist of protein backbones downloaded from the 13th and 14th critical assessment of techniques for protein structure prediction (CASP) database. This includes 82 regular targets in CASP13 and 64 regular targets in CASP14 (see *SI Appendix*, Table S16). We also consider CASP13 and CASP14 free modeling targets and provide results in *SI Appendix*, Table S14. We chose to use the CASP13 and CASP14 test sets as there is no canonical training or validation sets for PSCP, and in most cases, these test sets have little overlap with the training sets used for the methods in comparison.

We generated a set of nonnative backbone structures for the CASP13 and CASP14 targets, using AlphaFold2 from ColabFold (44) with default MSA settings. We selected all nonnative backbones with 1) at most 600 residues and 2) $C\alpha$ rmsd less than 2.5 Å from native. The rationale for this decision is given in Section 3C.2.

2. Results

When native backbones are given, we compare our method against several popular PSCP methods: DLPacker, RosettaPacker, SCWRL4, and FASPR. We use bold fonts and underlining in each table to mark the best and second-best performers in each category, respectively. For nonnative backbones, we compare our method with DLPacker, RosettaPacker, RosettaFold, Omegafold, and AlphaFold2. Results for AlphaFold2 are split

*BC40 is publicly available at https://drug.ai.tencent.com/protein/bc40/download.html

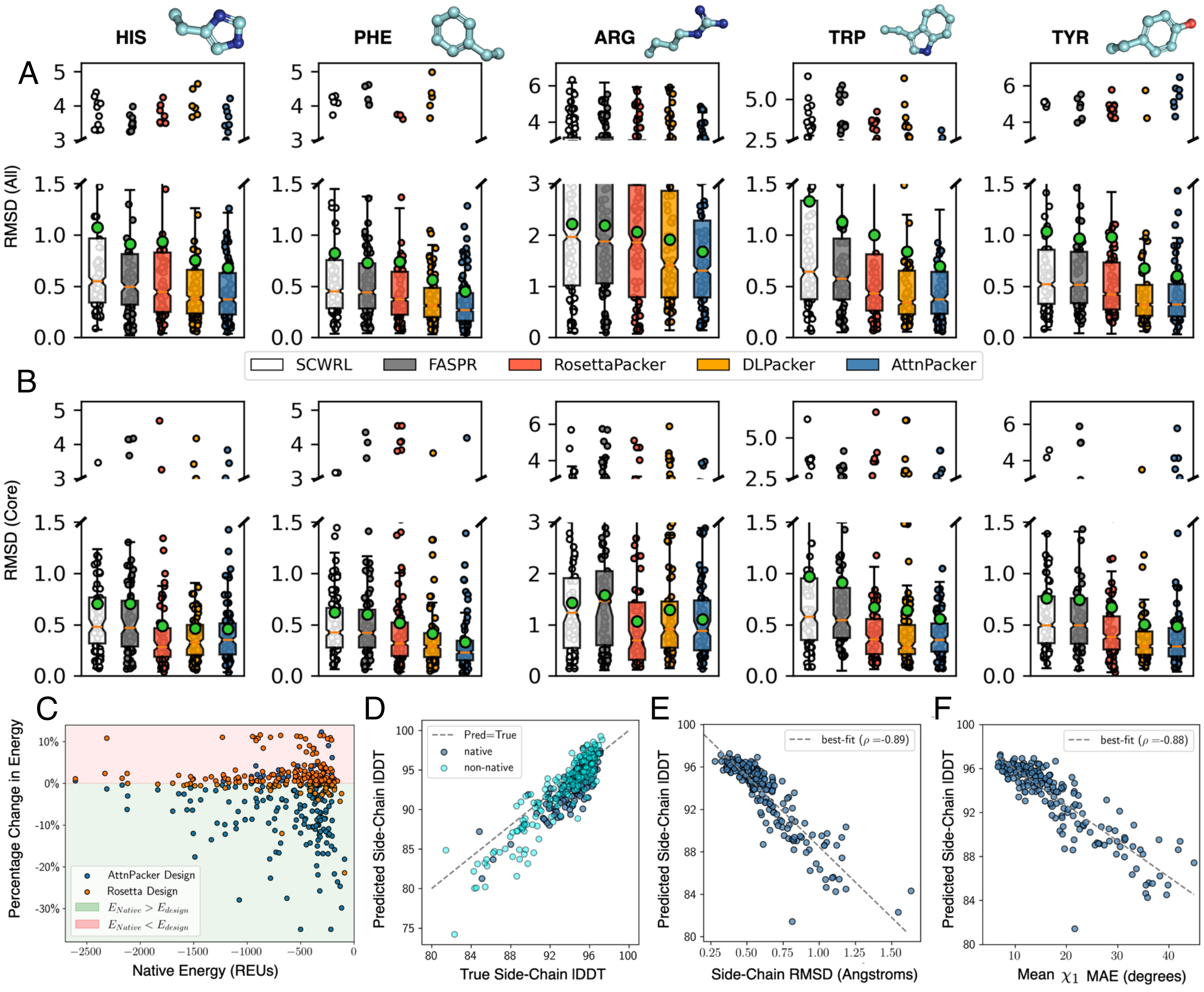

Fig. 2. Results for residue-level rmsd, confidence prediction, and sequence design. (*A*) and (*B*) show rmsd box plots for bulky amino acids on the CASP13 targets. Each box plot shows the average rmsd (Å) (*y*-axis) of each method (*x*-axis) on His, Phe, Arg, Trp, and Tyr. (*A*) shows rmsd values over all centrality categories, and (*B*) restricts to the protein core. Each box extends from the lower to upper quartile values of the data, with an orange line at the median. Note that 95% CIs around the median are shown with a notch, and the mean is shown with a green circle. For each residue type, a random sample of 80 rmsd values is plotted vertically with each bar. (*C*) Scatter plot of Rosetta energy for Rosetta-relaxed native structures (*x*-axis) against percentage change in Rosetta energy for designed sequences and side-chain conformations (*y*-axis). Orange dots correspond to Rosetta designs, and blue dots correspond to designs with AttnPacker. Energy scores were gathered after running Rosetta's FastRelax protocol (*SI Appendix*, section J). (*D*) Scatter plot of predicted side-chain lDDT (*y*-axis) against ground-truth side-chain lDDT for native and nonnative CASP targets. (*E* and *F*) Predicted side-chain lDDT (*y*-axis), against average side-chain rmsd and χ_1-MAE for CASP13 and CASP14 native targets.

further by information provided at inference time. We consider MSA and backbone templates as input (AF2+MSA+Temp.), and without MSA, given only backbone templates (AF2+Temp.). Results for RosettaFold are included only for the CASP14 targets. Complete details on data collection are provided in *SI Appendix*, section A.

When reporting results, we explicitly differentiate between the variant of AttnPacker trained only for PSCP and AttnPacker trained for PSCP and sequence design (+Design). Although trained with partial sequence information, the design variant of AttnPacker is provided with complete sequence information when assessing MAE and rmsd. We explicitly mention instances where sequence information is withheld. In the design setting, we do not give this variant any sequence information unless otherwise specified. We use the same hyperparameters for each model (*SI Appendix*, section B.1). We also remark that all results for our methods are taken after postprocessed according to the procedure described in Section 2B.5. Additional results, including a comparison of CASP test sets, are given in *SI Appendix*, section K, Tables S13–S15, and Figs. S10–S14.

A. Evaluation Criteria. We consider residue-level rmsd and dihedral angle deviations between predicted and native. Residue rmsd is calculated over nonhydrogen side chain atoms and excludes residues Ala and Gly. Overall rmsd is computed by averaging over all residues in all proteins for a dataset. For a side-chain dihedral χ_i, mean absolute error (MAE) is calculated analogously to rmsd. Some residue types have bimodal MAE distributions, and the average MAE tends to be much lower than the median. We also report dihedral accuracy to better illustrate instances where entire side chains are correctly predicted. Overall accuracy (χ_{1-4} Acc.) is defined as the fraction of residues having all dihedral angles within 20° of the corresponding native angles. The accuracy of a fixed dihedral χ_i is defined analogously over all residues containing this angle. As pointed out by Zhang et al. (10), the prediction accuracy of side-chain dihedrals is much sharper when all side-chain dihedrals are considered, and a 20° cutoff, and hence, we opt to use this criterion for all reported accuracy scores. Following ref. 10, in computing rmsd and angle MAE statistics, the symmetry of residues Asp, Glu, Phe, Arg, and Tyr were considered. We also consider flipping Asn, Gln, and

Table 1. Side-chain packing results for the CASP13 and CASP14 targets while native backbones are given

	rmsd (Å)↓			χ−MAE°↓				χ_{1-4}Acc. (%)↑			Clash (%VdW)↓		
Method	All	Surface	Core	χ_1	χ_2	χ_3	χ_4	All, (%)	Surface, (%)	Core, (%)	100%	90%	80%
CASP13													
SCWRL	0.934	1.200	0.597	27.64	28.97	49.57	61.54	56.2	45.2	71.2	115.3	20.6	4.6
FASPR	0.910	1.167	0.604	27.04	28.41	50.30	60.89	56.4	45.5	70.3	112.8	23.3	5.6
RosettaPacker	0.872	1.171	0.509	25.88	28.25	48.13	59.82	58.6	45.9	75.3	73.8	7.9	2.6
DLPacker	0.772	1.018	0.483	22.18	27.00	51.22	70.04	58.8	47.3	73.9	64.3	7.3	2.0
AttnPacker	**0.669**	**0.881**	**0.414**	18.92	**23.17**	**44.89**	**58.98**	**62.1**	**51.5**	**75.9**	46.1*	1.4*	0.3*
+Design	0.673	0.887	0.424	**18.77**	23.44	46.12	59.57	61.2	50.7	75.5	46.4*	1.2*	0.2*
Residue Count	19118	6046	8880	19118	14333	4887	2268	19118	6046	8880			
CASP14													
SCWRL	1.062	1.331	0.677	33.50	33.05	51.61	55.28	45.4	35.1	62.3	124.0	24.6	6.5
FASPR	1.048	1.304	0.696	33.04	32.49	50.15	54.82	46.3	36.3	62.3	130.5	29.5	8.7
RosettaPacker	1.006	1.384	0.716	31.79	33.00	50.54	56.16	47.5	35.3	66.1	100.6	10.1	3.4
DLPacker	0.929	1.197	0.562	29.01	31.69	53.98	72.88	48.0	36.3	66.8	74.1*	10.5	3.0
AttnPacker	0.823	1.067	0.502	25.34	28.19	**48.77**	**51.92**	50.9	39.1	68.2	57.1*	3.0*	0.7*
+Design	**0.815**	**1.058**	**0.466**	**24.75**	**27.56**	48.96	55.06	**51.6**	**39.8**	**70.6**	52.3*	2.5*	0.8*
Residue Count	17693	4979	8476	17693	13588	4613	2194	17693	4979	8476			

For rmsd and χ_{1-4} accuracy (χ_{1-4}Acc), columns divide results by residue degree centrality (All, Core, and Surface). Residue counts for each category are shown as a separate row for each dataset. We also report the average number of clashes for each target using a 100%, 90%, and 80% fraction of the van der Waals radius in the rightmost columns. An asterisk indicates an average clash value below that of native structures: 56.0, 5.9, and 0.4 for CASP13 and 80.4, 7.9, and 2.5 for CASP14. Results for AttnPacker+Design are obtained with the native sequence provided.

His terminal groups when computing rmsd due to difficulties distinguishing these atoms (17). A complete list of symmetries can be found in *SI Appendix*, Table S2.

A.1. Steric clashes. To assess the quality of packed solutions, we also consider the average number of steric clashes in packing solutions. Two atoms are considered to clash if their distance is smaller than a fixed percentage of the sum of their van der Waals (vdW) radii, taken from the AMBER force field (45). For this metric, we consider only atom pairs separated by at least four bonds, where at least one atom belongs to a residue side chain. To account for the fact that atoms sharing a hydrogen bond can favorably approach distances within the sum of their vdW radii (46), we reduce the clash cutoff by 0.4 Å for atom pairs composed of donor-borne hydrogen and acceptor (*SI Appendix*, Table S6).

A.2. Definition of core and surface residues. In some instances, we divide our results based on *residue centrality*; the number of $C\beta$ atoms within a 10 Å ball of the query residue's $C\beta$ atom. For this measure, $C\beta$ atoms from the native conformation are used. *Core residues* are defined as those amino acids with a centrality of at least 20, and *surface residues* are those with as most 15 $C\beta$ atoms within the same region of interest.

B. Network Architecture. Much of AttnPacker's success can be attributed to robust architectural components, described in Section 2B. Our stand-alone TFN-Transformer outperforms DLPacker with respect to both overall rmsd and dihedral angle accuracy on native backbones. Integrating local triangle updates boosts accuracy further, especially for core residues, as shown in *SI Appendix*, Fig. S7 and Table S10. Some of this improvement may come from increased depth, which effectively increases the network's receptive field. This is supported by the fact that performance differences are minimal for residues on the protein surface. Compared to the global triangle updates implemented in AlphaFold2, our local approach roughly matches the performance while considering only the 30 nearest neighbors of each residue. These results and several other ablation studies are shown in *SI Appendix*, section I.

C. Overall Reconstruction Accuracy. We begin by comparing average rmsd and dihedral accuracy over CASP13 and CASP14 test sets. Additional results for CASP free-modeling targets can be found in *SI Appendix*, Table S14. Example side-chain packings are shown in *SI Appendix*, Figs. S13 and S14.

C.1. Native backbones. As shown in Table 1, our method (AttnPacker and AttnPacker+Design) consistently achieves the lowest rmsd in each centrality category on both datasets. The performance carries over into dihedral prediction accuracy, where our method also achieves top performance for both datasets, regardless of residue centrality. Notably, physics-based RosettaPacker performs well on core residues, while its accuracy for surface residues ultimately hinders its overall performance. Compared to the DL method DLPacker, we obtain notably lower rmsd scores in all centrality categories, with the largest improvement on surface residues. We also improve overall dihedral accuracy over DLPacker by more than 3% for each test set.

We now focus on dihedral angle MAE across $\chi_1 - \chi_4$ degrees of freedom. As shown in Table 1, in terms of $\chi_1 - \chi_4$ MAE, our method achieves top 1 performance on CASP13 targets, and top 1 or top 2 performance on CASP14 native backbones. In line with the results reported by Misiura et al. (26), when compared to traditional methods, DL methods recover χ_1 dihedral angles considerably closer to those of the native structure. For our method, this improvement carries over to χ_2 angle prediction, where we obtain a 10% and 8% improvements over the next best method on CASP13 and CASP14 targets.

As shown in Table 1, AttnPacker also yields much fewer steric clashes and is the only method that generates fewer clashes than experimental structures. Rosetta also includes the capability to perform side-chain minimization after packing, which could remove many of the clashes in RosettaPacker's output. Finally, our method runs significantly faster than all other methods except for FASPR. Table 2 shows the cumulative and relative time spent by each method for reconstructing side chains of all the CASP13 targets.

C.2. Nonnative backbones. Comparing PSCP methods directly to protein structure prediction methods is difficult because the

Table 2. Time comparison of PSCP methods

Method	Ours	DLPack	RosPack	FASPR	SCWRL4
Rel. Time	1.0	124.4	151.7	0.5	14.7

Relative times for reconstructing the side-chain atoms of all 83 CASP13 targets. We exclude the time for running our postprocessing procedure, as this can be done in parallel on CPU. DLPacker (DLPack) and AttnPacker (Ours) were run on a single RTXA6000 GPU. RosettaPacker (RosPack), SCWRL4, and FASPR were run on a single AMD EPYC 7742 processor. With local triangle updates, our method is able to reconstruct all side-chain atoms for all the CASP13 targets in 68 s.

predicted tertiary structure can deviate far from that of the native. Side-chain MAE and rmsd statistics also lose interpretability as the predicted backbone structure deviates from the ground truth and native contacts are not conserved. Huang et al. (10) noticed that χ_{1-4} recovery rates decrease significantly for nonnative side chain packing when backbone rmsd is larger than 2.38 Å. Cao et al. also find a strong correlation between de novo designability and backbone $C\alpha$ rmsd in protein binder design. In an attempt to fairly compare structure prediction methods to PSCP methods, we restrict to predicted backbones with $C\alpha$ rmsd at most 2.5 Å from native in the main text and include a comparison of side-chain rmsd and χ_1 MAE against a broader range of backbone rmsds in *SI Appendix*, Fig. S10*A*. Since nonnative backbones can deviate from ground truth, we compute a transformation to align backbone heavy atoms N, CA, and C for each residue in the predicted structure to the corresponding residue in the native structure. After applying this per-residue transformation, we can compute side-chain rmsd and dihedral MAE as usual.

As a consequence of choosing low-rmsd backbones, we remark that the comparison may be biased in favor of PSP methods since we consider only those targets for which AlphaFold2 already produces accurate backbone conformations. Furthermore, the results may be biased against OmegaFold and RosettaFold, since target selection was based exclusively on AlphaFold2 accuracy.

For simplicity, we exclude FASPRand SCWRL and include only the top-performing traditional method RosettaPacker. As shown in Table 3, discarding MSA information from AlphaFold2 input causes a drastic reduction in performance. RosettaFold (with MSA) performs slightly better than AlphaFold2 (without MSA), having a slightly higher average rmsd but better dihedral accuracy. The relatively poor performance of these methods is likely attributed to inaccurate backbone predictions, where the average rmsds are 13 Å and 15 Å for RosettaFold and AF2+Temp, respectively. It is likely that MSA information also contributes to AlphaFold2's success on CASP14 targets, where the nearly identical architecture used in OmegaFold falls short in terms of dihedral accuracy. Part of this reduction can be attributed to OmegaFold predicting more accurate backbone conformations for the CASP13 targets, with an average rmsd of 4 Å increasing to 6 Å for the CASP14 targets. The two methods perform similarly in terms of χ_1-MAE and rmsd on both datasets, with OmegaFold having a slight edge in terms of χ_1 prediction, despite predicting backbone conformations with higher rmsd on average.

For PSCP methods, AttnPacker achieves top performance in terms of rmsd and χ_1 MAE, with significantly lower overall rmsd and a smaller improvement in χ_1-MAE over OmegaFold. For several targets, AttnPacker produces packings with nearly 30% lower rmsd than AlphaFold2, an example of which is given in *SI Appendix*, Fig. S13*B*. For nonnative backbones, RosettaPacker is more competitive with our method, having dihedral accuracy within a percentage point of the best score on both test sets and a slightly smaller gap in terms of rmsd.

Overall, we see fewer steric clashes on nonnative backbones than on native backbones. Looking into this, we found a correlation between protein length and the average number of clashes. This difference is likely attributed to nonnative backbones being restricted to less than 600 amino acids in length.

OmegaFold and non-MSA-based AlphaFold2 have the highest per-target average. This may contribute to OmegaFold's accuracy in terms of χ_1 MAE and rmsd, as it frequently predicts packings with large amounts of steric hindrance. This highlights the importance of considering steric clashes alongside traditional metrics when analyzing modeling performance. AttnPacker has fewer steric clashes than MSA-based AlphaFold2 and corresponding native structures, indicating that our method is

Table 3. Results for nonnative backbone structures predicted by AlphaFold2

	rmsd↓	MAE° ↓	Acc.↑	Clash (%VdW)↓		
Method	All	χ_1	χ_{1-4}, (%)	100%	90%	80%
CASP13 Nonnative (N = 7,490)						
AF2+Temp.	1.401	45.00	32.9	80.0	30.5	13.2
AF2+MSA+Temp.	0.944	30.12	54.8	39.7	1.1*	0.0*
OmegaFold	0.926	28.15	54.6	52.3	10.7	2.8
RosettaPacker	0.953	30.16	54.6	45.9	3.9	0.6
DLPacker	0.925	29.56	53.3	39.4	3.3	0.4*
AttnPacker	**0.871**	28.00	**55.0**	28.8*	0.4*	0.0*
+Design	0.876	**27.87**	54.6	30.7*	0.4*	0.0*
CASP14 Nonnative (N = 4,327)						
AF2+Temp.	1.431	47.40	28.4	77.2	29.4	12.3
AF2+MSA+Temp.	0.948	30.36	**52.2**	38.8*	1.7*	0.0*
OmegaFold	0.949	29.32	49.3	54.7	10.8	3.0
RosettaFold	1.51	48.45	34.8	40.6	1.3*	0.0*
RosettaPacker	0.980	30.85	51.2	46.7	3.3	0.4*
DLPacker	0.955	30.39	49.9	38.8*	4.4	0.7*
AttnPacker	0.902	29.14	51.2	30.8*	0.44*	0.0*
+Design	**0.884**	**28.18**	51.1	32.6*	0.48*	0.0*

Values are derived from a total of 47 CASP13 and 27 CASP14 targets, all having at most 600 residues and $C\alpha$ rmsd less than 2.5 Å from native. An asterisk indicates an average clash value below that of corresponding native structures—34.6, 2.2, and 0.5 for CASP13 and 40.0, 2.7, and 0.7 for CASP14. Results for AttnPacker+Design are obtained with the native sequence provided.

effective at producing physically realistic packings. It is well known that AlphaFold2 structures can contain considerable steric hindrance between side-chain atoms. According to the AlphaFold Structure Database (47), this typically occurs only in low-confidence regions. Since we selected only accurately modeled backbones, these numbers may be biased. This could also explain why the average number of clashes is much higher when excluding MSA information. We also note that AlphaFold2 is fine-tuned with steric clash loss, likely contributing to lower clash values.

D. Residue-Wise Reconstruction Accuracy.

D.1. rmsd for bulky amino acids. In terms of residue-level rmsd, the most significant improvements are achieved for Arg and His, each of which have large positively charged side chains, as well as bulky hydrophobic amino acids Phe, Trp, and Tyr.

In Fig. 2 *A* and *B*, we break the *y*-axis for all residues except Arg but maintain the split-axis formatting for consistency. Aside from Arg, all rmsd distributions have two major peaks around 0.5 Å, corresponding to correct predictions, and around 4 to 5 Å, corresponding to mirrored conformations.

As shown in Fig. 2*A*, AttnPacker improves the overall rmsd of each amino acid type by 10 to 20%, In *SI Appendix*, Fig. S11, we show that part of this reduction can be attributed to more accurate modeling of surface residues, where DL methods outperform traditional approaches by 20% for these residue types. In the protein core, DL methods lose their edge on His, Arg, and Trp. Incorrect modeling of bulky amino acids in the protein core could cause adverse effects on neighboring amino acids, leading to inaccurate downstream predictions. The density of atoms in the protein core also provides more packing constraints, which is favorable for traditional energy-based methods like RosettaPacker. Extended results on residue-level rmsd for the CASP13 and CASP14 targets can be found in *SI Appendix*, Table S15.

D.2. Dihedral accuracy for large amino acids with high degrees of freedom. To better understand the instances where our method loses its advantage to traditional PSCP algorithms, we evaluate dihedral predictions on four large amino acids with high side-chain dihedral degrees of freedom. We consider dihedral prediction for charged and polar amino acids Lys, Arg, Glu, and Gln in Table 4.

For these residue types, we restrict the comparison to the top-performing classical method RosettaPacker. For these amino acids, our method is still competitive with RosettaPacker, achieving top performance in $\chi_1 - \chi_4$ MAE for all but χ_3 of Lys. Surprisingly, low dihedral MAE does not always translate to high dihedral accuracy. Although AttnPacker almost always obtains lower rmsd and χ_1–χ_4 MAE than RosettaPacker, the physics-based method, achieves top 1 χ_{1-4} accuracy for both Arg and Lys. In *SI Appendix*, Fig. S12, we consider accuracy conditioned on average B-factor and show that inaccuracies in 3D models are unlikely to be the cause of this discrepancy. We hypothesize that this is a side effect of using rmsd as a training objective, which places more importance on accurately predicting lower-order dihedrals. This suggests that effective training methods or loss functions which improve higher-order dihedral accuracy are important areas for future research.

E. Confidence Prediction and Sequence Design.

We briefly discuss side-chain plDDT prediction and sequence design. Extended results and a comparison with other fixed backbone design methods can be found in *SI Appendix*, section J. An overview of AttnPacker's confidence prediction scheme is provided in *SI Appendix*, section D.

Illustrated in 3D, AttnPacker's side-chain lDDT predictions correlate strongly with ground truth lDDT for both native ($\rho = 0.85$) and nonnative ($\rho = 0.92$) input backbones. Fig. 2 *E* and *F* shows that this metric is also consistent with per-target side-chain rmsd and χ_1-MAE, suggesting that confidence predictions can be used to accurately assess rotamer quality.

Apart from rotamer prediction, we find that AttnPacker's sequence designs strongly encode native backbones through in silico validation with ESMFold (32). We generated designs for CASP13 native backbones and assessed their quality using self-consistency TM-score (scTM) and predicted lDDT (plDDT). scTM indicates how well sequences encode structure by measuring the TM-score (48) between predicted and ground-truth backbones. Predicted lDDT indicates how confidently a sequence is predicted to fold into the predicted structure. As shown in Fig. 3, AttnPacker achieves native-like scores for both metrics. More details are provided in *SI Appendix*, J.1, along with additional results for CASP14 targets and an equivalent in silico analysis of ProteinMPNN-designed sequences. Results for native sequence recovery and perplexity in various design settings are given in *SI Appendix*, Tables S11 and S12.

To jointly design rotamers and primary sequence, we tested the design variant of AttnPacker on the CASP13 and CASP14 native backbones. In Fig. 2*C*, we compare the Rosetta energy of AttnPacker designs to analogous designs using Rosetta's *FastDesign* protocol (49, 50) and the Rosetta ref_2015 energy function (14). A detailed overview of data collection is provided in *SI Appendix*, section A. Finally, we remark that other software has also been developed and applied to this task, including Osprey (51), protCAD (52), and SCADS (53). On average, designs from AttnPacker yield lower energy than both native structures and Rosetta designs, indicating that our method captures biologically meaningful aspects of sequence–structure compatibility.

Table 4. χ-Dihedral MAE° and accuracy on the CASP13 targets for charged and polar amino acids Arg, Lys, Gln, and Glu

	MAE° ↓				Acc.(%)↑	
Method	χ_1	χ_2	χ_3	χ_4	χ_{1-2}, (%)	χ_{1-4}, (%)
Arg						
Ros. Pack.	32.0	32.21	59.3	63.3	**52.5**	**22.2**
DLPack.	28.6	32.5	61.1	67.2	49.1	14.5
AttnPack.	**26.2**	**30.1**	**57.0**	**61.8**	50.5	13.3
Lys						
Ros. Pack.	31.3	37.9	**45.0**	56.5	51.5	**21.6**
DLPack.	27.6	37.6	51.8	72.7	46.1	9.6
AttnPack.	**24.1**	**33.8**	45.9	**56.3**	**54.2**	16.0
Gln						
Ros. Pack.	32.2	42.4	54.0		50.9	29.7
DLPack.	30.5	39.9	50.4		51.9	26.0
AttnPack.	**24.8**	**34.6**	**43.0**		**56.3**	**34.0**
Glu						
Ros. Pack.	38.9	43.7	31.1		43.9	24.9
DLPack.	34.6	42.4	36.5		42.9	22.0
AttnPack.	**29.0**	**38.1**	**27.9**		**49.2**	**30.1**

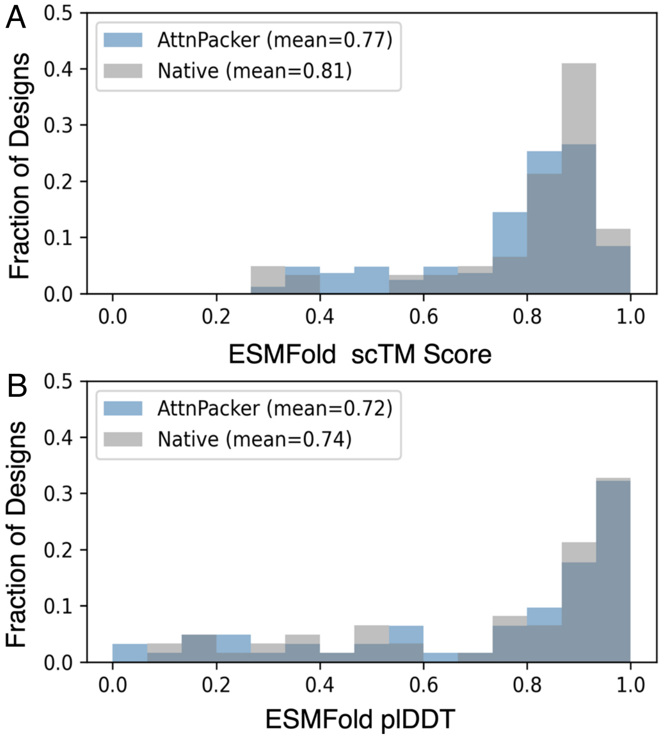

Fig. 3. In silico evaluation of AttnPacker generated sequence designs for CASP13 targets. ESMFold scTM-scores (*A*) and plDDT scores (*B*) for native (gray) and AttnPacker-generated (blue) sequences.

3. Concluding Discussion

We have developed AttnPacker, an SE(3)-equivariant model for the direct prediction of sequence and side-chain coordinates. AttnPacker uses spatial information derived from protein backbone coordinates to efficiently model residue and pairwise neighborhoods. This, coupled with an SE(3)-equivariant architecture, allows for the simultaneous prediction of all side-chain rotamers without conformational sampling or discrete rotamer selection.

Components of our DL model were inspired by AlphaFold2 and the SE(3)-Transformer. By generalizing and carefully evaluating ideas from these architectures, we were able to achieve comparable or better accuracy while drastically improving efficiency. Specifically, we generalized two components of AlphaFold2 to spatial graphs specific to protein backbones. We also modified the attention heads of the SE(3)-Transformer to incorporate pair bias, distance information between hidden points, and shared attention weights.

Even without mechanisms from AlphaFold2, our modified TFN-Transformer outperforms all other PSCP methods on native backbones in terms of average rmsd on the CASP13 and CASP14 targets. On the other hand, the baseline SE(3)-Transformer of Fuchs et al. falls short of DLPacker for residues in the protein core (*SI Appendix*, section I). Part of this improvement is achieved by augmenting edge features with pairwise distance information between hidden points. We hypothesize that this information is especially important for deeper TFN-based architectures because information about relative distances between hidden coordinates is lost as a result of using spherical basis functions.

On top of outperforming other popular methods, our model presents several other advantages. First, it is extremely fast. We are able to predict all side-chain conformations for a 600-residue protein in less than a second using a single Nvidia RTXA6000 GPU. On the other hand, DLPacker must be run iteratively for each amino acid side-chain creating a strong dependence on protein length at inference time. Our model is also very simple to use—it requires only a protein data bank (PDB) file to run. In contrast, OPUS-Rota4 (28) requires voxel representations of atomic environments derived from DLPacker, logits from trRosetta100, secondary structure, and constraint files derived from the output of OPUS-CM. Obtaining the requisite input data was too burdensome for us to compare our method with this method.

Furthermore, since our method directly predicts side-chain coordinates, the output is fully differentiable, which benefits downstream prediction tasks such as refinement or protein–protein interaction. This also circumvents the use of engineered energy functions and rotamer libraries and places the emphasis on architectural innovations and better loss functions.

AttnPacker also succeeds in efficiently modeling residue-level local environments by using locality-based graph attention during feature and structure generation stages. On the other hand, DLPacker and OPUS-Rota4 use 3D-voxelized representations of each amino acid's microenvironment—requiring space $O\left(v^3cd\right)$, where v is the voxelized width (40 in the case of DLPacker and OPUS), c is the number of channels, and d is the channel dimension. Although this choice of representation has helped facilitate good performance for each method, the memory requirements prevent simultaneous modeling of all amino acid side chains which could ultimately hinder reconstruction accuracy. We hypothesize that simultaneously modeling all side chains helps contribute to our method's success.

Although AttnPacker yields significant improvements in residue-level rmsd, we remark that traditional PSCP methods SCWRL, FASPR, and RosettaPacker still perform comparably in terms of χ_3 and χ_4 angle prediction. We guess that incorporating dihedral information into our model—either directly or through an appropriate loss function—could help improve performance.

In addition to side-chain packing, the design variant of AttnPacker is able to generate sequences with both high recovery (mean NSR of 47.8%, CASP13) and high in silico success rates (mean scTM of 0.77, CASP13). These results are competitive with the inverse-folding method ProteinMPNN, which achieves a mean NSR of 48.5% and ESMFold scTM of 0.83 for the CASP13 targets. Unlike AttnPacker, ProteinMPNN does not explicitly model coordinates and is primarily featurized by pairwise distances between backbone atoms. In training ProteinMPNN, backbone atoms are perturbed with a small amount of Gaussian noise (SD = 0.02 Å). This data augmentation strategy is found to increase in silico success rates and may explain some of the performance differences between the two methods. Furthermore, ProteinMPNN is trained on a combination of monomeric and multimeric inputs. The training set is clustered at 30% sequence identity and contains 23k clusters of protein assemblies in the PDB as of August 2021. At every training epoch, representatives from each cluster are resampled, which may also favorably impact performance.

Data, Materials, and Software Availability. Pretrained model, source code, and inference scripts are available at https://github.com/MattMcPartlon/AttnPacker (54). All study data are included in the article and/or supporting information.

ACKNOWLEDGMENTS. Research reported in this publication was supported by the Department of Health and Human Services of the NIH under award number R01 GM089753 (J.X.). M.M. would like to thank Phil Wang for some support code and discussions on SE(3)-equivariant architectures.

1. G. Faure, A. Bornot, A. de Brevern, Erratum: Protein contacts, inter-residue interactions and side-chain modelling. *Biochimie* **90**, 626–639 (2007).
2. S. Farokhirad *et al.*, "3.13. Computational methods related to molecular structure and reaction chemistry of biomaterials" in *Comprehensive Biomaterials II*, P. Ducheyne, Ed. (Elsevier, Oxford, UK, 2017), pp. 245–267.
3. G. Chinea, G. Padron, R. W. Hooft, C. Sander, G. Vriend, The use of position-specific rotamers in model building by homology. *Proteins* **23**, 415–421 (1995).
4. N. Ollikainen, R. M. de Jong, T. Kortemme, Coupling protein side-chain and backbone flexibility improves the re-design of protein–ligand specificity. *PLoS Comput. Biol.* **11**, 111–222 (2015).
5. D. Simoncini, K. Y. J. Zhang, T. Schiex, S. Barbe, A structural homology approach for computational protein design with flexible backbone. *Bioinformatics* **35**, 2418–2426 (2019).
6. J. R. Desjarlais, T. M. Handel, Side-chain and backbone flexibility in protein core design. *J. Mol. Biol.* **290**, 305–318 (1999).
7. A. M. Watkins, T. W. Craven, P. D. Renfrew, P. S. Arora, R. Bonneau, Rotamer libraries for the high-resolution design of β-amino acid foldamers. bioRxiv [Preprint] (2016). https://doi.org/10.1101/086389 (Accessed November 2022).
8. A. M. Watkins, R. Bonneau, P. S. Arora, Side-chain conformational preferences govern protein–protein interactions. *J. Am. Chem. Soc.* **138**, 10386–10389 (2016).
9. H. Hogues *et al.*, Propose: Direct exhaustive protein–protein docking with side chain flexibility. *J. Chem. Theory Comput.* **14**, 4938–4947 (2018).
10. X. Huang, R. Pearce, Y. Zhang, Faspr: An open-source tool for fast and accurate protein side-chain packing. *Bioinformatics* **36**, 3758–3765 (2020).
11. C. Yanover, O. Schueler-Furman, Y. Weiss, Minimizing and learning energy functions for side-chain prediction. *J. Comput. Biol.* **15**, 899–911 (2008).
12. S. Liang, D. Zheng, C. Zhang, D. Standley, Fast and accurate prediction of protein side-chain conformations. *Bioinformatics* **20**, 2913–2914 (2011).
13. A. Badaczewska-Dawid, A. Kolinski, S. Kmiecik, Computational reconstruction of atomistic protein structures from coarse-grained models. *Comput. Struct. Biotechnol. J.* **18**, 162–176 (2020).
14. R. F. Alford *et al.*, The Rosetta all-atom energy function for macromolecular modeling and design. *J. Chem. Theory Comput.* **13**, 3031–3048 (2017).
15. R. Dunbrack, M. Shapovalov, G. Krivov, Improved prediction of protein side-chain conformations with SCWRL4. *Proteins* **77**, 778–795 (2009).
16. J. Xu, B. Berger, Fast and accurate algorithms for protein side-chain packing. *J. ACM* **53**, 533–557 (2006).
17. Y. Cao *et al.*, Improved side-chain modeling by coupling clash-detection guided iterative search with rotamer relaxation. *Bioinformatics* **27**, 785–790 (2010).
18. P. D. Renfrew, T. W. Craven, G. L. Butterfoss, K. Kirshenbaum, R. Bonneau, A rotamer library to enable modeling and design of peptoid foldamers. *J. Am. Chem. Soc.* **136**, 8772–8782 (2014).
19. M. V. Shapovalov, R. L. Dunbrack Jr, A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions. *Structure* **19**, 844–858 (2011).
20. R. L. Dunbrack, Rotamer libraries in the 21st century. *Curr. Opin. Struct. Biol.* **12**, 431–440 (2002).
21. J. M. Jumper, N. F. Faruk, K. F. Freed, T. R. Sosnick, Accurate calculation of side chain packing and free energy with applications to protein molecular dynamics. *PLOS Comput. Biol.* **14**, 1–25 (2018).
22. C. Yanover, O. Schueler-Furman, Y. Weiss, Minimizing and learning energy functions for side-chain prediction. *J. Comput. Biol.* **15**, 899–911 (2008).
23. A. Fahmy, G. Wagner, Optimization of van der Waals energy for protein side-chain placement and design. *Biophys. J.* **101**, 1690–1698 (2011).
24. T Gaillard, N Panel, T Simonson, Protein side chain conformation predictions with an MMGBSA energy function. *Proteins* **84**, 803–819 (2016).
25. K. Nagata, A. Randall, P. Baldi, SIDEpro: A novel machine learning approach for the fast and accurate prediction of side-chain conformations. *Proteins* **80**, 142–153 (2012).
26. M. Misiura, R. Shroff, R. Thyer, A. B. Kolomeisky, DLPacker: Deep learning for prediction of amino acid side chain conformations in proteins. bioRxiv [Preprint] (2021). https://doi.org/10.1101/2021.05.23.445347 (Accessed November 2022).
27. G. Xu, Q. Wang, J. Ma, OPUS-Rota3: Improving protein side-chain modeling by deep neural networks and ensemble methods. *J. Chem. Inf. Mod.* **60**, 6691–6697 (2020).
28. G. Xu, Q. Wang, J. Ma, OPUS-Rota4: A gradient-based protein side-chain modeling framework assisted by deep learning-based predictors. bioRxiv [Preprint] (2021). https://doi.org/10.1101/2021.07.22.453446 (Accessed November 2022).
29. K. Liu *et al.*, Prediction of amino acid side chain conformation using a deep neural network. arXiv [Preprint] (2017). https://doi.org/10.48550/arXiv.1707.08381 (Accessed November 2022).
30. J. Jumper *et al.*, Highly accurate protein structure prediction with alphafold. *Nature* **596**, 583–589 (2021).
31. M. Baek *et al.*, Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **373**, 871–876 (2021).
32. Z. Lin *et al.*, Language models of protein sequences at the scale of evolution enable accurate structure prediction. bioRxiv [Preprint] (2022). https://doi.org/10.1101/2022.07.20.500902 (Accessed November 2022).
33. R. Wu *et al.*, High-resolution de novo structure prediction from primary sequence. bioRxiv [Preprint] (2022). https://doi.org/10.1101/2022.07.21.500999 (Accessed November 2022).
34. Y. Qi, J. Z. H. Zhang, DenseCPD: Improving the accuracy of neural-network-based computational protein sequence design with DenseNet. *J. Chem. Inf. Mod.* **60**, 1245–1252 (2020).
35. B. Jing, S. Eismann, P. Suriana, R. J. L. Townshend, R. Dror, Learning from protein structure with geometric vector perceptrons. arXiv [Preprint] (2020). https://doi.org/10.48550/arXiv.2009.01411 (Accessed November 2022).
36. Z. Lin *et al.*, Language models of protein sequences at the scale of evolution enable accurate structure prediction. bioRxiv [Preprint] (2022). https://doi.org/10.1101/2022.07.20.500902 (Accessed November 2022).
37. K. K. Yang, N. Zanichelli, H. Yeh, Masked inverse folding with sequence transfer for protein representation learning. bioRxiv [Preprint] (2022). https://doi.org/10.1101/2022.05.25.493516 (Accessed November 2022).
38. J. Dauparas *et al.*, Robust deep learning based protein sequence design using ProteinMPNN. *Science* **378**, 49–56 (2022).
39. N. Thomas *et al.*, Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds. arXiv [Preprint] (2018). https://doi.org/10.48550/arXiv.1802.08219 (Accessed November 2022).
40. F. Fuchs, D. Worrall, V. Fischer, M. Welling, Se(3)-transformers: 3D roto-translation equivariant attention networks in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, H. Lin, Eds. (Curran Associates, Inc., 2020), vol. 33, pp. 1970–1981.
41. J. Yang *et al.*, Improved protein structure prediction using predicted interresidue orientations. *Proc. Natl. Acad. Sci. U.S.A.* **117**, 1496–1503 (2020).
42. S. R. Johnson, S. Monaco, K. Massie, Z. Syed, Generating novel protein sequences using Gibbs sampling of masked language models. bioRxiv [Preprint] (2021). https://doi.org/10.1101/2021.01.26.428322 (Accessed November 2022).
43. D. P. Kingma, J. Ba, Adam: A method for stochastic optimization in *3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, May 7–9, 2015, Conference Track Proceedings*, Y. Bengio, Y. LeCun, Eds. (2015).
44. M. Mirdita *et al.*, ColabFold: Making protein folding accessible to all. *Nat. Methods* **19**, 679–682 (2022).
45. D. A. Case *et al.*, The amber biomolecular simulation programs. *J. Comput. Chem.* **26**, 1668–1688 (2005).
46. A. J. Li, R. Nussinov, A set of van der Waals and coulombic radii of protein atoms for molecular and solvent-accessible surface calculation, packing evaluation, and docking. *Proteins* **32**, 111–127 (1998).
47. M. Varadi *et al.*, AlphaFold Protein Structure Database: Massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Res.* **50**, D439–D444 (2021).
48. J. Xu, Y. Zhang, How significant is a protein structure similarity with TM-score = 0.5? *Bioinformatics* **26**, 889–895 (2010).
49. S. Chaudhury, S. Lyskov, J. J. Gray, PyRosetta: A script-based interface for implementing molecular modeling algorithms using Rosetta. *Bioinformatics* **26**, 689–91 (2010).
50. A. L. Loshbaugh, T. Kortemme, Comparison of Rosetta flexible-backbone computational protein design methods on binding interactions. *Proteins* **88**, 206–226 (2020).
51. P. Gainza *et al.*, OSPREY: Protein design with ensembles, flexibility, and provable algorithms. *Methods Enzymol.* **523**, 87–107 (2013).
52. C. M. Summa, W. F. DeGrado, protCAD: Protein Computer Aided Design (University of Pennsylvania Schol of Medicine, Philadelphia, 2002).
53. X. Fu, J. R. Apgar, A. E. Keating, Modeling backbone flexibility to achieve sequence diversity: The design of novel alpha-helical ligands for Bcl-xL. *J. Mol. Biol.* **371**, 1099–1117 (2007).
54. M. Mcpartlon, J. Xu, AttnPacker Source Code. *Github*. https://github.com/MattMcPartlon/AttnPacker. Deposited 17 March 2023.