RESEARCH ARTICLE

PROTEINS WILEY

DLPacker: Deep learning for prediction of amino acid side chain conformations in proteins

Mikita Misiura[1] | **Raghav Shroff**[2] | **Ross Thyer**[3] | **Anatoly B. Kolomeisky**[1,3,4]

[1]Department of Chemistry, Center for Theoretical Biological Physics, Rice University, Houston, Texas, USA

[2]CCDC Army Research Lab, Austin, Texas, USA

[3]Department of Chemical and Biomolecular Engineering, Rice University, Houston, Texas, USA

[4]Department of Physics and Astronomy, Center for Theoretical Biological Physics, Rice University, Houston, Texas, USA

Correspondence
Mikita Misiura, Department of Chemistry, Center for Theoretical Biological Physics, Rice University, Houston, TX 77005, USA.
Email: nekit@rice.edu

Abstract

Prediction of side chain conformations of amino acids in proteins (also termed "packing") is an important and challenging part of protein structure prediction with many interesting applications in protein design. A variety of methods for packing have been developed but more accurate ones are still needed. Machine learning (ML) methods have recently become a powerful tool for solving various problems in diverse areas of science, including structural biology. In this study, we evaluate the potential of deep neural networks (DNNs) for prediction of amino acid side chain conformations. We formulate the problem as image-to-image transformation and train a U-net style DNN to solve the problem. We show that our method outperforms other physics-based methods by a significant margin: reconstruction RMSDs for most amino acids are about 20% smaller compared to SCWRL4 and Rosetta Packer with RMSDs for bulky hydrophobic amino acids Phe, Tyr, and Trp being up to 50% smaller.

KEYWORDS
3DCNN, DNN, protein structure prediction, side chain restoration, U-net

1 | INTRODUCTION

De novo *protein structure prediction* is one of the key fundamental problems in structural biology and a lot of research efforts are currently focused in this area. Accurate prediction of the protein structure from its amino acid sequence is required in order to understand the molecular details of various biological phenomena and to fully uncover biological functions of large number of proteins found inside genomes of living organisms. It can enable efficient in silico protein design, allowing for the development of new enzymes catalyzing novel types of reactions, development of new target-specific protein therapeutics, and many other powerful applications. However, despite the fact that a protein's fold, in principle, is fully defined by its amino acid sequence, the prediction task is extremely complex due to the immense size of the conformational search space, which grows exponentially with the sequence length.

Existing approaches often split protein structure prediction into two steps: prediction of the protein's backbone conformation ("folding") and prediction of the side chain conformations ("packing"). Most of the currently available methods for side chain packing are physics-

based approaches that involve some sort of search inside a given sample space, often defined by a library of pre-defined rotamers and/or an optimization using a well-designed energy function.[1–6] These methods view the problem from a physico-chemical perspective and are trying to optimize interactions between side chains while avoiding steric clashes and minimizing the overall energy of the system.

There are currently multiple breakthroughs in machine learning (ML) methods that are revolutionizing multiple areas of science, including chemistry, physics, biology, and medicine. A large number of recent studies explored the potential of applying neural networks (NNs) and deep neural networks (*DNNs*) for a variety of difficult problems for which classical methods are still unsuccessful. This includes protein modeling, small molecule property predictions, drug discovery and materials design.[7–17] One of the most notable works in the field of protein structure prediction is AlphaFold and its successor AlphaFold2, which significantly outperform classical methods for protein structure prediction.[9,18]

ML methods have already been used for the task of amino acid *side chain restoration*.[13,19–22] Nagata et al.[13] trained an ensemble of 156 shallow NNs using inter-atomic distances as an input, which

resulted in a remarkably fast side chain restoration algorithm. The latest version of the OPUS side-chain modeling framework, named OPUS-RotaNN, which is based on OPUS-TASS, uses DNNs to achieve even better performance.[19] OPUS-RotaNN significantly improves rotamer sampling and can outperform other methods like OSCAR-star, SCWRL4, and FASPR.

In this study, we explore the potential of DNNs for the protein side chain conformation prediction. The problem is formulated as an image-to-image transformation and a *U-net*-style DNN is utilized to solve it. We demonstrate that our method significantly outperforms other published approaches. Reconstruction RMSDs for most amino acids are about 20% smaller compared to SCWRL4 and Rosetta Packer, and RMSDs for bulky hydrophobic amino acids Phe, Tyr, and Trp are up to 50% smaller. We anticipate that our approach can be useful for a variety of structural biology-related tasks such as homology modeling, generating point mutants for in silico protein engineering and other related problems. Since the direct-coupling analysis (DCA) methods are successful in predicting contacts between amino acids and inferring the backbone conformations of proteins, it may be possible to couple our method with DCA to deliver high-quality full atom structure predictions.[23]

2 | METHODS

When training a *3DConv Neural Network* model to solve the problem of side chain packing there are three major steps: input generation, NN inference, and side chain reconstruction. The overall approach is schematically illustrated in Figure 1. The next three sections describe these steps in more detail.

2.1 | Description of the dataset preparation

Training and test datasets were created as described by Shroff et al.[24] In short, all entries in the PDB database were clustered to 50% similarity to avoid bias toward more abundant classes of proteins. From each cluster, only the single structure with the highest resolution was selected and remodeled using the PDB-redo algorithm, which rebuilds all structures using a unified algorithm.[25] Clusters containing only structures with resolution lower than 2.5 Å were discarded. If multiple alternative conformations exist inside the protein structure, only the alternative conformation with the largest sum of occupancies was used, and other conformations are discarded. Hydrogen atoms were omitted in our model and selenomethionine residues were converted into methionine.

This process yielded 19 436 structures, of which 2201 were reserved as a test set and the rest were used as a training set. This resulted in about 10 million training examples and one million test examples.

2.2 | Input data generation

The process of input generation is schematically described on the top portion of Figure 1. After a target amino acid is chosen for side chain

reconstruction, a 20-Å box containing its surrounding microenvironment is isolated from the protein. The side chain of the target amino acid (if present) is always removed from the input. For all other side chain atoms in the box, there are different options depending on what stage (training/testing/inference) is being performed by the model. Since the NN in this setup is used to restore all side chains in a sequential manner (one amino acid at a time), the method should be able to make predictions using both a pure backbone representation of a protein and a partially restored representation of a protein (with some of the side chains already present). Therefore, during the training stage the side chains of non-target amino acids were removed using the following algorithm: in 50% of training examples, all side chains were removed from the input and in the other 50%, a random fraction of the side chains were removed. All of the amino acid side chains in the PDB structure were completely removed prior to the testing stage and then restored one by one.

After the input box is defined and side chains are removed if necessary, all atoms are placed on a grid with $40 \times 40 \times 40$ dimensions and split into 28 channels: five element channels (one channel each for **C**, **N**, **O**, **S** and one for all other elements), one partial charge channel (partial charges were taken from the Amber99sb force field), 21 amino acid channels (one for each of the 20 canonical amino acids plus one for all other amino acids), and one label channel that encodes an amino acid label to restore. While the purpose of all element channels and the partial charge channel is clear, the purpose of 21 amino acid channels is to propagate the information about the sequence, which is necessary because some or even all side chains are removed. The *backbone atoms* of each amino acid in the input box are *repeated* in the respective channel (in addition to being present in their respective element channels). For example, if there is a phenylalanine residue present, its backbone (**C**, **N**, **Ca**, and **O**) atoms are copied into one phenylalanine channel. This way the information about protein sequence is not lost after removal of side chains. The output is similar to the input with two major differences: only four element channels are present (**C**, **N**, **O**, and **S** since only these four elements are present in amino acid side chains) and the *target* side chain is present. Note that only one *target* side chain is restored at a time, while all the other side chains present in the box are left as they are.

2.3 | NN architecture and training

We conceptualized the problem of side chain restoration as an image-to-image transformation and opted for a U-net architecture, which is frequently used in this field.[26–30] One major difference, however, is that our inputs are 3D images, which require the use 3D convolutions in our architecture. The final architecture is shown on the right in Figure 1. First, amino acid labels are added as an additional channel through a fully connected (FC) layer, whose output is then reshaped and concatenated with the rest of the input. The encoder has three-strided 3D convolutional layers with 3×3 kernel size, stride 2, and ReLU (Rectified Linear Unit) nonlinearities. The number of convolutional filters are 128, 256, and 512, respectively. After the encoder, six residual blocks follow with identical architecture shown
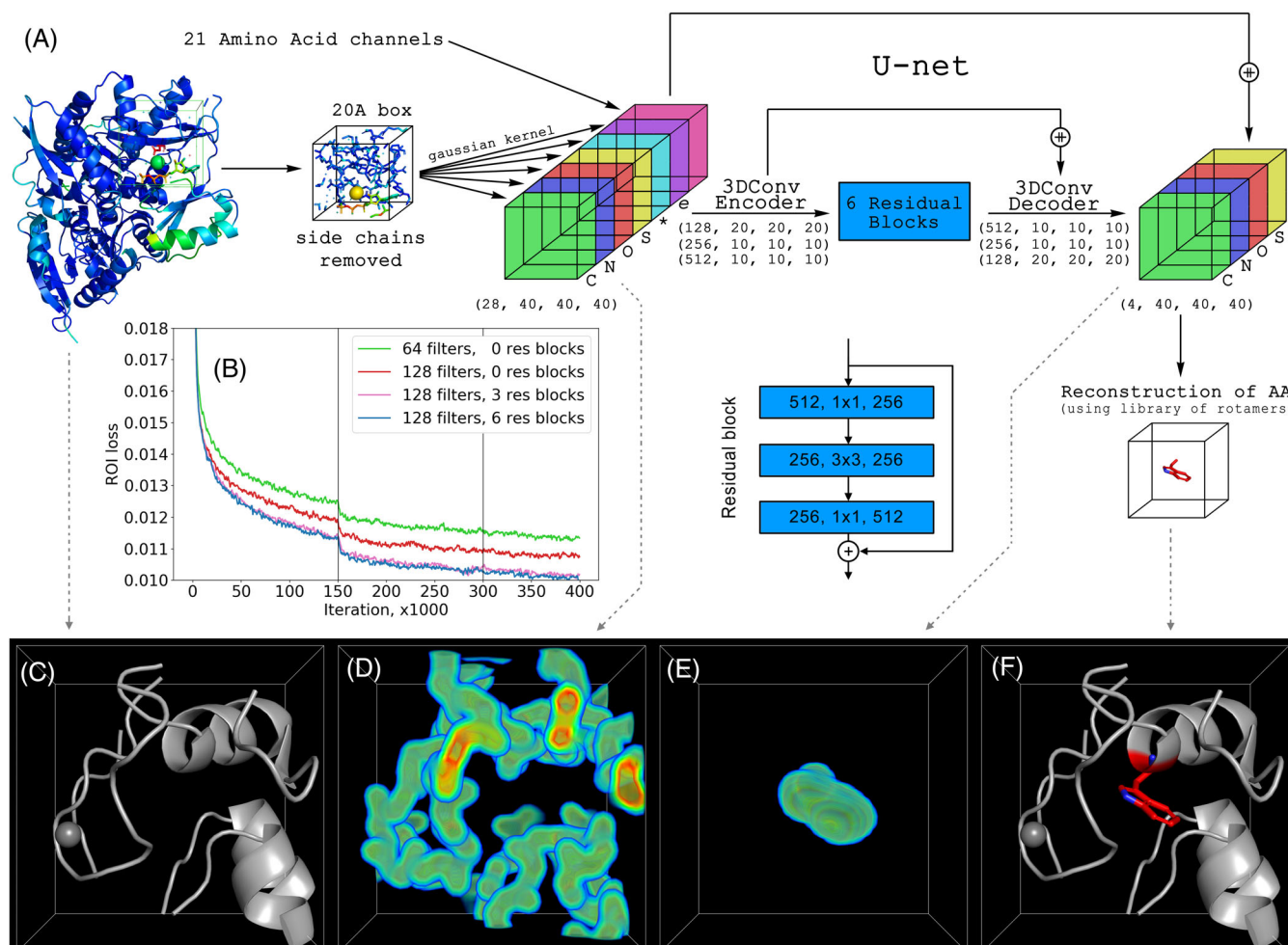
FIGURE 1 Overview of our side chain restoration algorithm. (A) Input data generation and U-net architecture. After an amino acid is chosen, a cube comprising its microenvironment is isolated from the protein and atom positions are placed on a grid and split into a number of different input channels. The input is then fed into the 3DConv U-net, consisting of encoder, six residual blocks and a decoder. Structure of a residual block is shown in blue. The output is a 3D density, outlining predicted side chain conformation. This prediction is then turned into side chain atom positions by fitting a pre-calculated library of side chain conformations. (B) Training loss curves for different NN architectures with different numbers of filters in the first 3DConv layer and different numbers of residual blocks. Vertical lines correspond to step-wise decreases of the learning rate. Bottom row: Illustration of input and output. (C) Cartoon representation of the amino acid's microenvironment. Only the backbone is left at this stage. (D) Encoded atom positions. Each atom is represented as a 3D Gaussian density kernel. Only the sum of four element channels (**C, N, O, S**) is shown for simplicity. (E) Output from the U-net. Sum of four element channels (**C, N, O, S**) is shown. (F) Reconstructed amino acid side chain (Trp in this case, shown in red). All images were generated using PyMol

in the middle of Figure 1. Residual blocks also use ReLU nonlinearities between convolutions. The final part is the decoder, which essentially mirrors the structure of the encoder, but uses ×2 upsampling to scale tensor sizes up and Leaky ReLUs as nonlinearities ($\alpha = .2$). As the U-net architecture implies, the respective outputs from encoder layers were concatenated with outputs of decoder layers before feeding into the next layer. The very last 3D convolution used four filters to output the four element channels. This architecture was implemented in Python using TensorFlow 2.4. The NN was trained for about 1.5 epochs with a step-wise learning rate schedule and a batch size of 32. The initial learning rate was 1×10^{-5} and was decreased by a factor of 10 each time loss stopped improving. The loss was mean absolute error (MAE) between prediction and ground truth. Additionally, we

added what we call a region of interest loss, which is just MAE in the voxels with side chain atoms being restored, which was given larger weight (100 in the final version) to force the NN to put more attention into that region.

2.4 | Side chain restoration

The output from our model is not atom positions, but a 3D density map, where values of each voxel (the 3D analog of a pixel) are proportional to the probability of finding an amino acid's side chain atom at that location (Figure 1E, bottom row). Converting this output into actual atom positions can be done in a number of ways. We settled

on an approach that uses a library of side chain conformations because this approach guarantees valid conformation as a result. We used our training dataset to build a library of side chain conformations for each amino acid. The library is backbone-independent and created in such a way that it contains a list of side chain conformations in which RMSD between any individual conformation and all other conformations exceeds some threshold value. The threshold values for each amino acid were manually tuned so that the final library includes a decent, but not overwhelming number of different conformations. Threshold values that we used as well as the final numbers of side chain conformations are presented in Table S1.

After NN inference, all side chain conformations in the library for a target amino acid are placed on a grid (see Section 2.2) and compared to the NN's prediction by calculating a fitness score as the mean absolute difference between the prediction and the current conformation. The conformation with the lowest score is then chosen as the final prediction. An advantage of this procedure is that it guarantees the prediction to be a valid conformation. The final fitness score can also be used as a measure of prediction quality: higher scores correspond to poorer prediction quality and therefore higher uncertainty.

2.5 | Detection of salt bridges

We consider that a salt bridge between Arg or Lys and Asp or Glu is formed if any of the side chain nitrogen atoms of Arg or Lys are within 3.2 Å of any side chain oxygen atoms of Asp or Glu.

2.6 | SCWRL4 and Rosetta Packer validation

SCWRL4 (ver 4.02) was run in default configuration. For Rosetta Packer (ver 3.12) the maximum number of rotamers were considered by passing −EX1, −EX2, −EX3, and −EX4 flags and invoking *fixbb* using the default scoring function (*ref2015*) unless stated otherwise.[31,32] We tried further increasing the number of rotamers for surface residues by using *extrachi_cutoff 0*, but that resulted in marginal improvement of RMSDs while leading to much longer calculation times.

3 | RESULTS

3.1 | Architecture of NN

We first started by optimizing some of the hyperparameters of our model, including the size of the microenvironment in relation to the network input and output, number of convolutional filters in the NN, and number of residual layers, as presented in Figure 1B. We found that increasing the number of convolutional filters gives significant improvements in performance up to 128 filters in the first convolutional layer (the subsequent layers of decoder always double the number of filters of the previous layer), but further increases did

not yield further improvements, perhaps due to larger memory consumption and smaller batch size during training. We then found that the addition of residual blocks between the encoder and decoder also leads to significant improvement in performance. We tried adding up to nine residual blocks in increments of three. Six residual blocks gave the best performance boost and further increase did not seem to improve the loss further in our experiments. Finally, we hypothesized that using a larger microenvironment (compared to our initial 20-Å box) could yield further improvements. The initial cutoff value of 20 Å was chosen because it was already used in the literature and it is just enough to fit all side chains of all amino acids. It also seems to include enough of the surrounding atoms. Larger cutoff values could provide more information to predict the larger amino acid side chains like Arg and Lys. We therefore tried using 30-Å box as the input and output size, but that did not lead to any measurable gains in performance, while requiring about three times longer training and predictions times. Additionally, the use of BatchNorm layers also did not give any performance boost in our experiments. Our final architecture is therefore illustrated in Figure 1.

3.2 | Performance analysis

Figure 1 illustrates our best performing architecture. We found that residual blocks in the middle significantly boost performance, with six blocks delivering the best performance. Further increase in the number of blocks did not lead to further improvement in our experiments. Increasing the number of convolutional filters also improves the performance at the cost of longer training and inference times. BatchNorm layers also did not give any performance boost in our experiments.

We used RMSD as a key metric to measure the performance of our algorithm and to compare it to existing models for amino acid side chain prediction. Our method's performance is compared with that of SCWRL4 and Rosetta Packer, as both are well known and widely utilized approaches.[1,31,33–37] Each algorithm was used on ∼1000 PDB structures from our test dataset. The results of these comparisons are presented in Figure 2 and Table S2. In all results shown and discussed below, we report the performance on full protein packing: we first remove all the side chains from the protein and then restore them one by one in a sequential fashion. Several restoration ordering strategies are devised and discussed as well.

As one can see from Figure 2, DLPacker significantly outperforms both SCWRL4 and Rosetta Packer for all amino acids. The most significant improvement is achieved for large hydrophobic amino acids like Trp, Phe, and Tyr. The improvement is also significant for a number of charged and polar amino acids such as Arg, Lys, His, Asp, and Glu. To further validate the model's performance, an additional test was performed using PDB structures deposited in 2020 and 2021 (i.e., after construction of our original dataset) using PISCES server, which yielded almost identical results, details can be found in Supporting Information Material S1.[38]
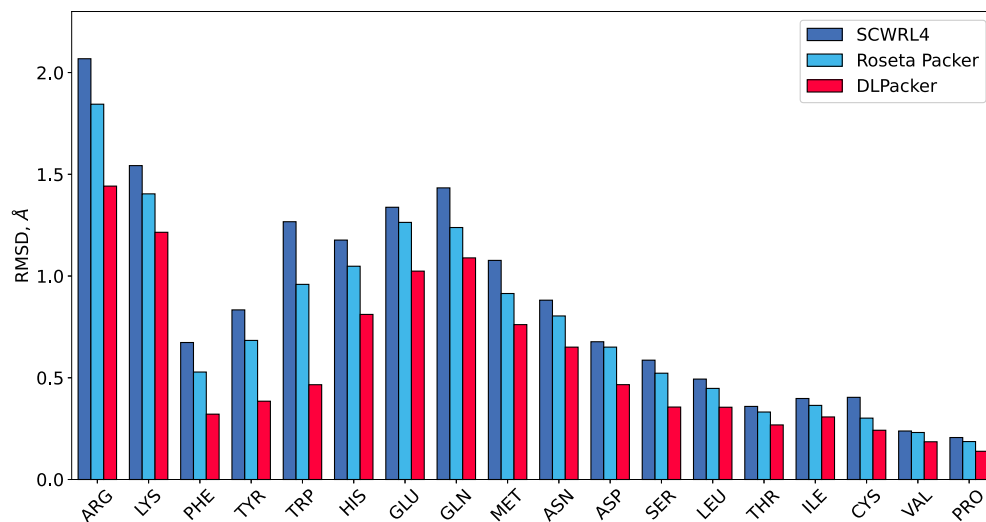
To evaluate the performance of DLPacker in more detail, we first analyzed RMSD histograms for bulky hydrophobic amino acids Phe, Tyr, Trp, and His (Figure 3). Histograms for all four amino acids show two major peaks: one peak in the region of low RMSDs (below 2 Å) and one in the region of high RMSDs (2 Å and above). The first peak represents correct prediction of χ_1 dihedral angle, which leads to small RMSD errors during reconstruction. Small RMSD errors equate to correct reconstruction of the side chain and that the entire side chain will be located in approximately the correct location. Here we say "approximately in the correct location" because it is important to take into account the existence of thermal fluctuations at normal temperatures, the limited resolution of experimentally determined structures, and other factors that lead to impossibility of precise reconstruction. The second peak, however, represents incorrect prediction of χ_1, which places side chains in the wrong place entirely. This scenario is problematic as it not only leads to an incorrect reconstruction of one amino acid, but can potentially lead to incorrect reconstruction of the entire surrounding microenvironment.[39,40] Based on this observation, we measured what we call error rates for all three algorithms and compare them in Table 1. The thresholds in Table 1 were chosen to lie between the low and the high RMSD maxima. The exact values of these thresholds, thus, will not affect the error rates in a significant way since the number of RMSD predictions lying around the chosen threshold values is very low, which makes this analysis robust. It can be seen that DLPacker significantly reduces error rates for all four bulky hydrophobic amino acids. For example, in the case of Phe, the error rate decreases from 4.8% for SCWRL4 and 3.6% for Rosetta Packer to 1.2% for DLPacker. Similar improvements are observed for Tyr and Trp: DLPacker demonstrates about a fourfold reduction in error rate compared to SCWRL4 and a threefold reduction compared to Rosetta Packer. The improvement for His is more moderate: about 1.8–2 times.

Figure 4 shows how the error rates depend on an amino acid's position in a protein. We use atom counts in an amino acid's microenvironment as an indicator of its position within a protein: higher atom counts correspond to the denser environment of a protein's core, while lower atom counts indicate either a very small protein or the surface of a protein. This is important since erroneous side chain predictions for bulky amino acids in dense environments are much more likely to cause incorrect packing of larger regions around them. As a result, large portions of protein's hydrophobic core have a chance of not being reconstructed accurately, which can be problematic for some downstream tasks. In contrast, an incorrect side chain conformation on the surface of a protein might not lead to significant disruptions. As Figure 4 illustrates, the error rate for all algorithms decreases significantly as the number of atoms in the microenvironment increases, which is expected as more crowded environment leads to more interactions with neighboring residues. DLPacker exhibits significantly smaller error rates across the whole range of atom counts and for all amino acids presented in Figure 4.

Next, we wanted to investigate the quality of side chain restoration for charged amino acids, like Arg, Lys, Asp, and Glu. Since these amino acids have multiple degrees of freedom (up to four dihedral χ angles), an analysis similar to that performed for bulky hydrophobic amino acids above is not possible, as the RMSD histograms do not show any useful structure. Instead, we decided to analyze the quality of reconstruction of salt bridges formed by charged amino acids since salt bridges play a significant role in protein structure formation and stabilization. Results of this analysis are presented in Table 2. We used a precision-recall metric to quantify the quality of salt bridge reconstruction. The precision value shows what fraction of salt bridges in reconstructed protein structure exist in the original PDB structure, while the recall value shows what fraction of salt bridges from the original PDB structures were reconstructed. The F_1 score is a harmonic mean of precision and recall

$$F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}},$$ (1)

which serves as an aggregated value showing the quality of prediction and combines precision and recall into a single value. We can see that DLPacker outperforms both SCWRL4 and Rosetta Packer leading to
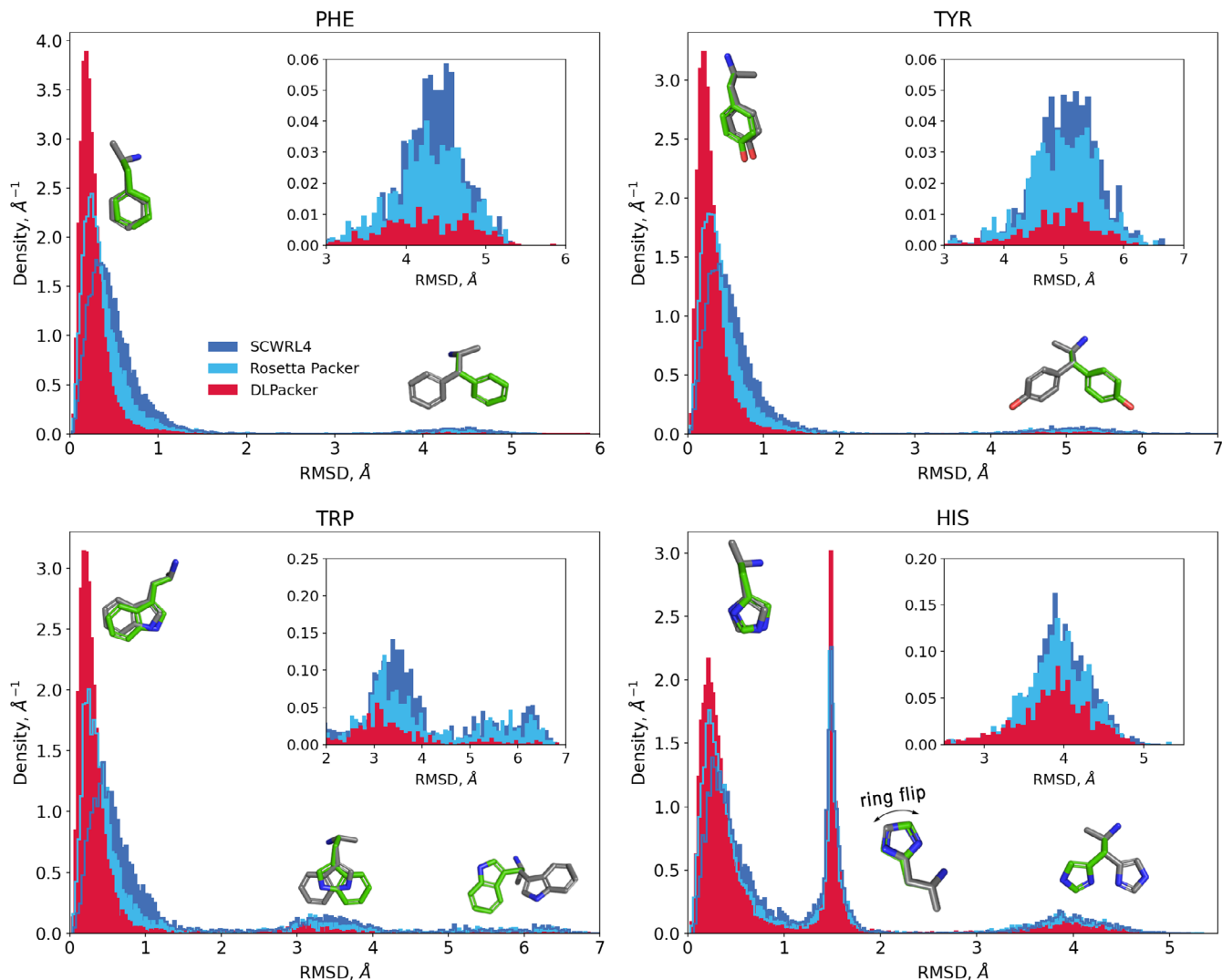
FIGURE 3 Side chain prediction RMSDs for four bulky amino acids: Phe, Tyr, Trp, and His. All histograms have similar structure and show two major peaks (three in the case of His). Insets show high RMSD peaks in more detail. Cartoon models of side chains (gray is the ground truth, green is prediction) show specific examples of side chain restoration errors that compose each peak. Our algorithm (DLPacker) yields significantly fewer predictions with high RMSD resulting in higher quality packing

TABLE 1 Side chain reconstruction error rates for bulky amino acids

AA name	Threshold, Å	Error rate, %		
		SCWRL4	Rosetta Packer	DLPacker
Phe	3.0	4.8	3.7	1.2
Tyr	3.0	6.2	5.0	1.4
Trp	2.0	20.3	14.6	5.1
His	2.5	11.1	10.8	6.1

Note: Values are measured on ∼1000 PDB structures from the test set. Errors are defined as predictions with high RMSDs, higher than the particular threshold. The motivation for this definition is illustrated in Figure 3 and addressed in more detail in the text. DLPacker demonstrated smaller error rates resulting in higher quality of reconstruction of side chains of bulky amino acids.

the highest F_1 score among the three: 0.62 versus 0.56 for Rosetta Packer and 0.50 for SCWRL4.

It is also interesting to look at the comparison between MAE of χ angles produced by the algorithms in comparison (Table 3). DLPacker works much better than other methods when comparing MAEs of χ_1, but as we move to χ_2, χ_3, and χ_4, it starts to lose its advantage. Charged and polar amino acids with lots of degrees of freedom (Lys, Arg, Glu, Gln) seem to be the most challenging ones for our method: DLPacker is often worse at predicting χ_3 and χ_4 than SCWRL4 and/or Rosetta Packer. Apparently, the model struggles assigning conformations to amino acids with large number of degrees of freedom (the number of available conformations grows exponentially with the number of degrees of freedom, that is, χ angles in this case), while having fewer troubles dealing with bulky amino acids with smaller number of degrees of freedom. New training methods that might improve
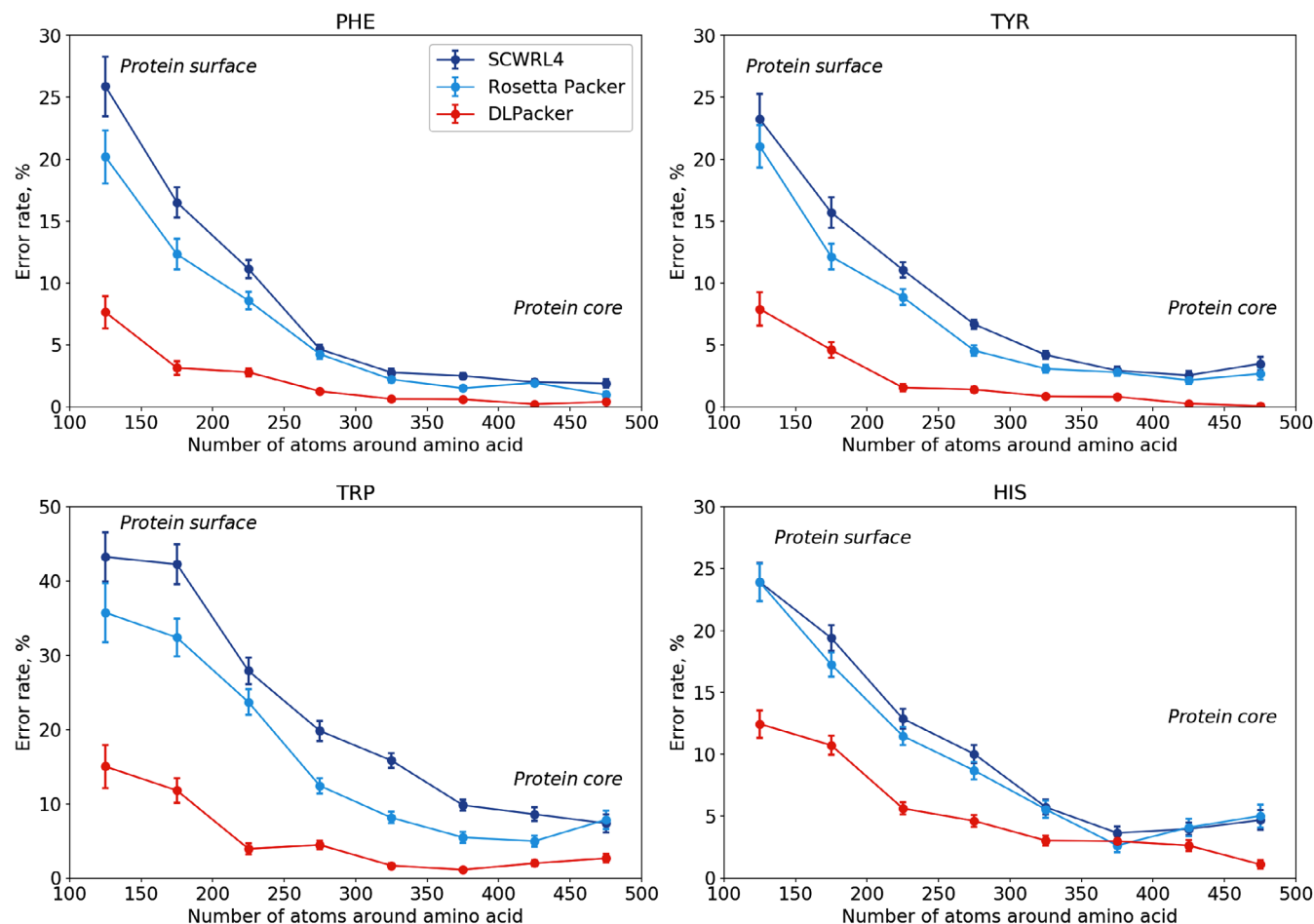
FIGURE 4 Side chain reconstruction error rates for bulky amino acids as a function of the number of atoms in their microenvironment. Values are measured on ~1000 PDB structures from the test set. Vertical lines show standard deviations. The definition of the error rate is defined in detail in the text. The number of atoms in the surrounding microenvironment is an indicator of an amino acid's position in a protein: higher atom counts indicate that an amino acid is located in the protein's core, while low atom counts indicate a location on or near the surface. DLPacker demonstrates smaller error rates at all atom counts resulting in higher quality of reconstruction of cores of proteins as well as their surfaces

TABLE 2 Salt bridge reconstruction metrics measured on ~1000 PDB structures from the test set

	Precision	Recall	F_1 score
SCWRL4	0.59	0.44	0.50
Rosetta Packer	0.59	0.54	0.56
DLPacker	0.70	0.56	0.62

Note: DLPacker outperforms SCWRL4 and Rosetta Packer by all three metrics indicating higher quality side chain reconstruction for charged amino acids. Details are in the text.

assignment of side chains for these amino acids is an area of further research interest.

3.3 | Optimization of restoration order

Since our approach implies the restoration of side chains structure in a sequential manner (one amino acid at a time), we need heuristic

arguments to choose the most optimal restoration order. In this study, we explored three different ordering strategies. First, we tried a naive strategy that restores side chains in the order of protein's sequence: from N-terminus to C-terminus. This strategy is both simple and effective since amino acids that are next to each other in the sequence will also be next to each other in the 3D structure. Next, we tried a strategy that orders amino acids according to the number of atoms in their microenvironment and restores side chains in the most crowded microenvironments first. This strategy assumes that it is easier to correctly predict side chain conformation in more crowded microenvironment because there are more interactions and a larger fraction of the volume is already occupied by backbone atoms. It roughly corresponds to first restoring the side chains in the protein's interior and then gradually moving to its exterior. The last strategy orders amino acids according to the prediction quality and is a two-stage process. First, predictions are made for each amino acid (without actually restoring the side chains) and then residues are sorted by their prediction score, normalized by the number of atoms in a side chain (see Section 2.4). The rationale is that the lower fitness score

TABLE 3 Comparison between mean absolute errors of χ angles for SCWRL4, Rosetta Packer, and DLPacker

| AA name | Software | MAE, ° | | | | AA name | Software | MAE, ° | |
		χ_1	χ_2	χ_3	χ_4			χ_1	χ_2
	SCWRL4	28.4	28.3	58.0	59.1		SCWRL4	22.3	45.0
Arg	Rosetta Packer	24.3	26.9	54.4	**55.3**	Asn	Rosetta Packer	20.6	41.2
	DLPacker	**18.6**	**24.6**	**48.5**	57.0		DLPacker	**15.1**	**38.2**
	SCWRL4	29.5	30.5	**36.9**	53.2		SCWRL4	22.0	20.9
Lys	Rosetta Packer	25.6	30.0	37.9	**51.9**	Asp	Rosetta Packer	21.3	19.9
	DLPacker	**19.1**	**28.8**	43.4	65.8		DLPacker	**14.2**	**16.4**
	SCWRL4	34.8	38.8	31.4			SCWRL4	13.2	19.5
Glu	Rosetta Packer	32.7	37.4	**30.3**		Leu	Rosetta Packer	10.8	19.0
	DLPacker	**25.0**	**31.7**	31.3			DLPacker	**7.9**	**15.6**
	SCWRL4	27.6	43.1	54.2			SCWRL4	10.4	22.1
Gln	Rosetta Packer	24.2	35.2	**49.1**		Ile	Rosetta Packer	9.3	**19.5**
	DLPacker	**21.3**	**33.2**	50.2			DLPacker	**7.6**	19.8
	SCWRL4	21.4	31.4	53.1			SCWRL4	18.5	
Met	Rosetta Packer	19.0	23.8	48.4		Thr	Rosetta Packer	16.9	
	DLPacker	**13.8**	**20.4**	**44.9**			DLPacker	**12.8**	
	SCWRL4	12.2	12.3				SCWRL4	41.2	
Phe	Rosetta Packer	9.2	10.0			Ser	Rosetta Packer	36.6	
	DLPacker	**5.3**	**7.7**				DLPacker	**24.0**	
	SCWRL4	13.7	12.5				SCWRL4	20.5	
Tyr	Rosetta Packer	10.8	10.5			Cys	Rosetta Packer	14.0	
	DLPacker	**5.5**	**8.2**				DLPacker	**11.4**	
	SCWRL4	15.1	32.7				SCWRL4	15.0	
Trp	Rosetta Packer	12.8	24.5			Val	Rosetta Packer	13.5	
	DLPacker	**5.6**	**12.3**				DLPacker	**10.1**	
	SCWRL4	19.2	67.5						
His	Rosetta Packer	17.7	58.8						
	DLPacker	**12.1**	**55.6**						

Note: The values were obtained by averaging MAE values for ~1000 PDB structures from the test set. Bold values show minimum among three packages in comparison.

corresponds to the smaller prediction uncertainty and more likely to lead to correct predictions. Side chains are then restored in the second pass.

All three methods yield similar results, with the first strategy performing slightly worse than the other two for most amino acids (Figure 5). The best performing strategy is the third one (by the prediction score) with the only drawback being that it uses twice as much time and computational resources since we need to pass through the whole protein's sequence twice. The second strategy (by the number of atoms in the microenvironment) seems to be a perfect compromise between speed and quality.

3.4 | Performance on raw PDB structures

Since our algorithm was trained and tested on PDB structures refined by the PDB Redo server, we wanted to test how well it generalizes to

raw PDB structures.[25] The results are shown in Figure 5. The resulting RMSD values are only slightly larger in the case of Raw PDB structures. It is not clear, however, if this difference is due to PDB Redo structures having higher quality (better resolution) or due to the Raw PDB structures just being slightly different from the data our algorithm was trained on.

3.5 | Significance of partial charges as input

To investigate the significance of information about the partial charges of atoms on the accuracy of our predictions, we performed an additional test by removing information about charges from the input (by setting all charge values to zero) and evaluating our model's performance. The results are presented in Figure S1. Removal of charges only moderately affects accuracy of predictions, mostly affecting charged and polar amino acids, as expected. From these results, we
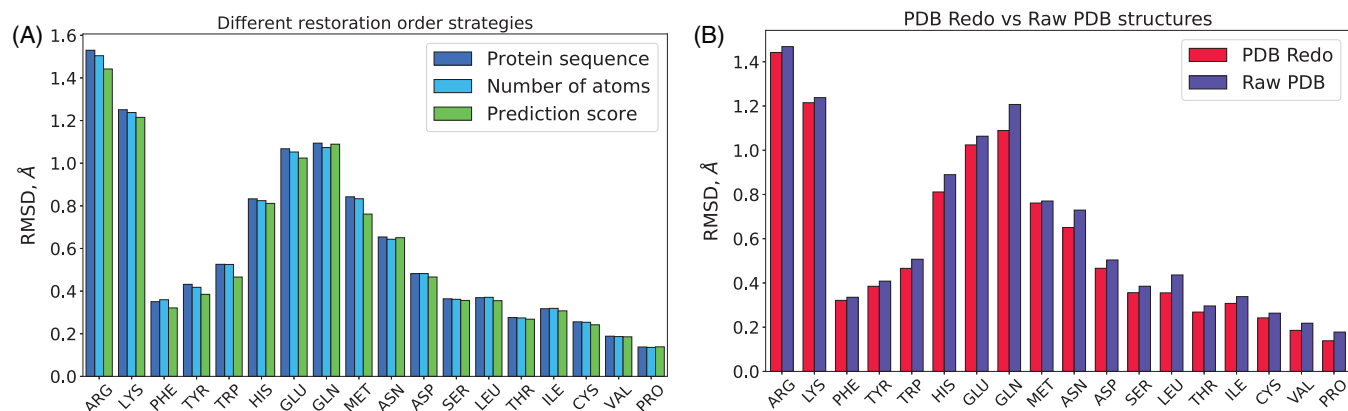
FIGURE 5 (A) Effect of different restoration order on reconstruction RMSD. All three ordering strategies—by sequence, by number of atoms in the microenvironment, and by quality of prediction (prediction score)—yield approximately the same results, but latter is the best. (B) Comparison between performance on PDB Redo structures and raw PDB structures. Performance on raw PDB structures is only slightly worse. More details are in the text

can conclude that the information about the charges is not a key driver of DLPacker's performance and that use of different force fields is unlikely to yield major improvements. This finding suggests that the most crucial information for correct packing is provided by atom coordinates and atom types.

4 | DISCUSSION

We developed and comprehensively evaluated a new deep learning approach for amino acid side chain prediction in proteins. Besides much better performance compared to other commonly used tools, our approach has a number of other advantages. It can be used on any protein structure, including structures of protein complexes, structures containing DNA, RNA, or any small molecules. Our approach can also be utilized in a combination with other physics-based approaches, for example, as one component of a scoring function. The output from the NN is a 3D density map, in which values in every voxel are proportional to probability of finding an amino acid's side chain atom at that location (Figure 1C), which can be used to score every possible rotamer of an amino acid. This score can then be applied, for example, as a Context Independent One Body Energy term and added to the default Rosetta scoring function. In our experiments, however, this did not lead to further improvement compared to our algorithm (data not shown). There are several other directions that can be explored by connecting our method with existing approaches in order to optimize the whole process of amino acid side chain structure prediction in proteins.

Traditional methods of packing of amino acid side chains rely heavily on hand-crafted scoring functions, which try to take into account all known interactions in proteins.[41] This approach, however, is fundamentally limited by our knowledge of aforementioned interactions. The list of currently known interactions in proteins is quite long. In addition to well-known and studied ones, like Van der Waals forces, electrostatic interactions, and hydrogen bonds, there is also a list of

less studied interactions, which include $\pi-\pi$ stacking, π–cation, π–anion, π–sulfur, NH–π, CH–π, C–halogen–π interactions.[42–52] To make things more complicated, while all the interactions listed above are usually viewed and characterized as two-body interactions, that is just a simplification of the real picture and many interactions can be highly affected by multi-body effects. Other still unknown two-body and more complicated multi-body interactions are also likely to exist in proteins.[53–56] All these factors enormously complicate the development of scoring functions for protein packing and make it extremely difficult to estimate the strengths of all interactions and their dependence on geometry and nature of the microenvironment.

Deep learning approaches, on the other hand, have no need for explicitly defined scoring functions and do not require any assumptions on strength and nature of any existing interactions.[9,57–60] Studies suggest that current number of structures in PDB allows for careful and precise estimation of interactions between amino acids, including multi-body interactions.[54,56] Trying to elucidate the roots of DLPacker's performance, we looked at some of the structures where it yielded correct prediction, while Rosetta Packer (the better performing of two algorithms we compared to DLPacker in this study) did not. Results are presented in Figure 6. Visual inspection shows that the most frequent reason of Rosetta Packer's failure was positioning side chains facing outwards (into solution) instead of burying into a protein. This is true for the most structures shown in Figure 6— instead of putting the target side chain in position to interact with other side chains shown, Rosetta Packer put it facing outwards and often formed no strong interactions at all. This may be a result of either underestimation of the strengths of some interactions or drawing from a limited number of initial rotamers which do encompass the correct configuration of side chain atoms. In our experiments, however, further increase of the number of initial rotamers only marginally improved RMSD, while drastically increasing computational time.

We have also investigated instances where DLPacker predicted the wrong conformation, while Rosetta Packer predicted the correct one. The pool of such cases is quite small due to DLPacker's
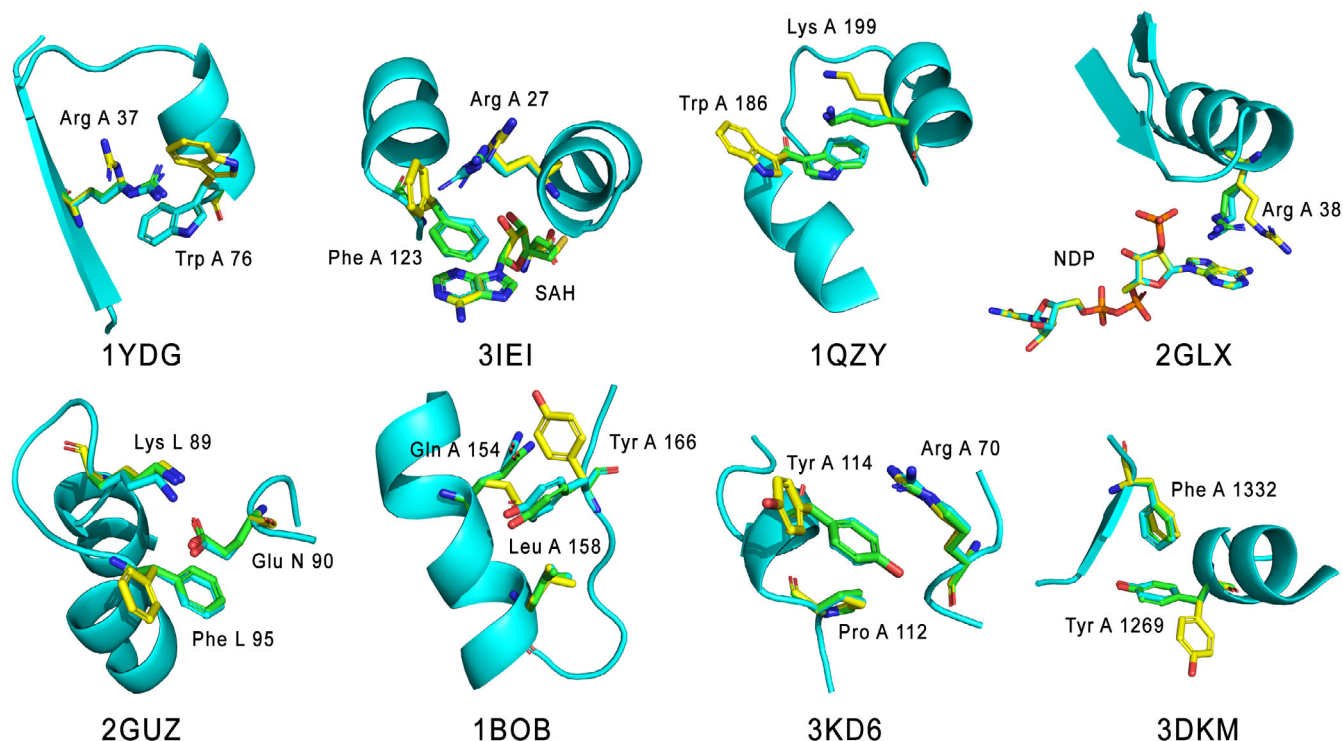
FIGURE 6   Examples of amino acids for which DLPacker yielded correct result, but Rosetta Packer did not. Backbone is shown in cartoon representation (cyan) and selected side chains are shown using stick representation. Ground truth side chains are shown in green, DLPacker's output in cyan, and Rosetta Packer's in yellow. PDB codes of structures and amino acid names are shown as well

performance, but we identified a few characteristic cases of failure, which are shown in Figure 7. In some cases, our model's prediction is just wrong, as exemplified in Figure 7A, despite being seemingly reasonable and confident. This is most often observed for aromatic amino acids (Trp, Tyr, Phe). In the cases of large amino acids with larger number of degrees of freedom (Lys, Arg, Met) we often see predictions like the one depicted in Figure 7B using Lys as an example: the model only outputs confident prediction for a part of the side chain closest to the backbone and does not output good prediction for the rest of the side chain (in Panel (B), this corresponds to green volume, showing NNs prediction, not covering part of the predicted side chain). We hypothesize that this is the consequence of large number of degrees of freedom and hence large number of available conformations leading to model not being able to find a specific conformation to output. This is what most likely leads to growing MAEs for χ angles as we move to χ_3 and χ_4 (Table 3). More powerful models possibly combined with active learning techniques might help to achieve a better performance for these amino acids. Another frequent situation we observed is depicted in Figure 7C–F. In all these cases, our model predicts multiple possible conformations for the target side chain. In many such cases, we see that the correct conformation is also encapsulated into prediction, so it was probably "outpaced" by the incorrect one by a little bit. These cases make us think that there is a potential of combining models similar to ours with physics-based algorithms, where our method can serve as an additional component of a force field.

We hypothesize that the source of the DLPacker's high performance is the ability to carefully infer strengths of multiple two-body and multi-body interactions existing in proteins. We note that DLPacker demonstrates the largest improvement for aromatic amino acids (Phe, Tyr, and Trp). These amino acids are known to form the largest number of various X–π interactions, which involve their aromatic systems. These interactions are also likely to be highly influenced by the amino acid's microenvironment due to high polarizability of aromatic electrons, potentially leading to complicated dependencies on geometry and nature of its microenvironment.

As was already discussed, our algorithm outperforms other existing methods such as SCWRL4 and Rosetta Packer. It should be noted, however, that currently this comes at the cost of the increased computational time and resources (GPU, see Table 4). Despite the fact that SCWRL4 shows poorer metrics, it is highly optimized for speed and is notably faster than our approach. In our experiments, SCWRL4 was about seven times faster than out algorithm. Rosetta Packer, on the other hand, was about two times slower than the DLPacker (if run with the maximum number of rotamers, as described in Section 2, to achieve the performance we report in this study). If run with default parameters, Rosetta Packer becomes significantly faster at the cost of worse metrics, which are close to that of SCWRL4. It should be also noted, however, that SCWRL4 and Rosetta Packer are highly optimized and compiled C codes, while DLPacker runs as Python code, and if better performance is required, our method can be re-implemented in C for additional performance gain. In our current
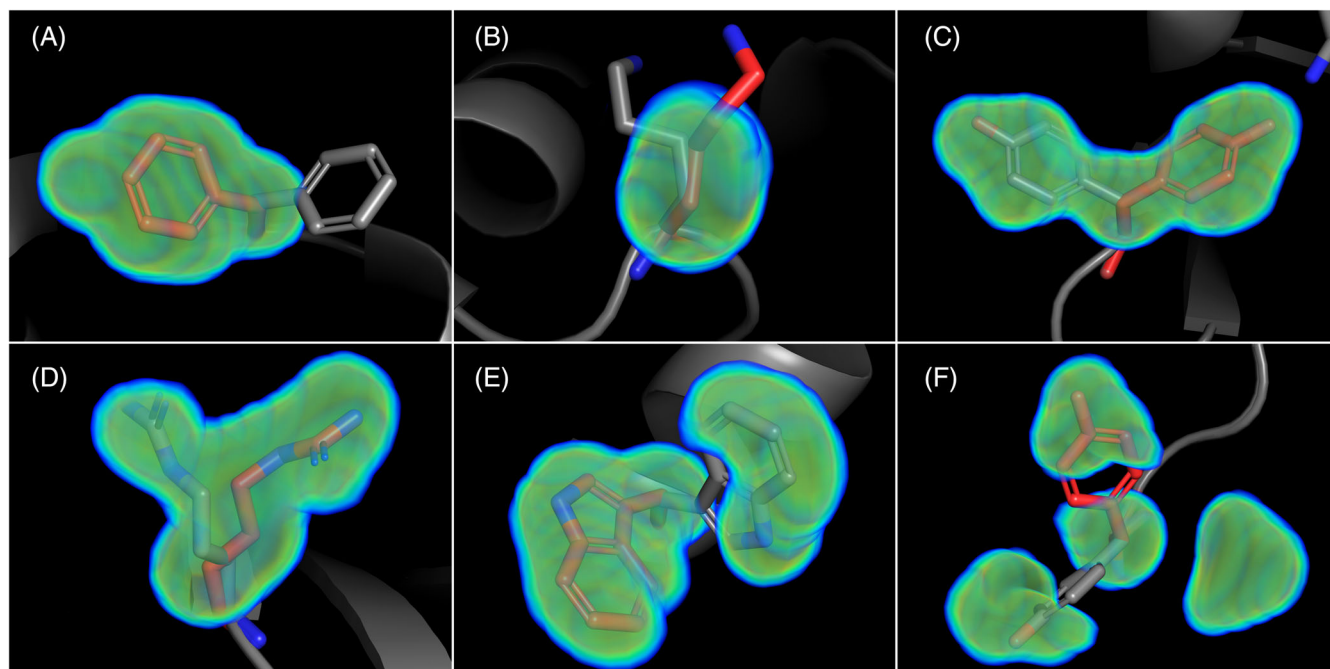
FIGURE 7 Examples of incorrect predictions by DLPacker. Ground truth is shown in gray with the target side chain depicted as sticks. Green volume shows DLPacker's model output and the final chosen side chain conformation is shown in red. In some cases, predictions are wrong (A) or incomplete (B), but in some cases, the model can output multiple possible conformations (C–F), which encapsulate the correct conformation as well as incorrect ones

TABLE 4 Comparison between average running times between different algorithms in comparison in this study

	Hardware	Average time per structure, s
SCWRL4	CPU	7.2
Rosetta Packer	CPU	91
DLPacker	CPU + GPU	43

Note: Performance was measured on 1000 PDB structures from our test dataset using Ubuntu 18.04 workstation with i9 CPU and RTX3090 GPU.

algorithm, only a fraction of the computational resources are used to generate predictions using the NN; a lot of the time is spent on generating input data (aligning the protein structure to properly position each target amino acid in the input box and then putting microenvironment atoms onto the 3D grid) and converting the output from density into valid side chain conformations. More efficient methods of performing these steps are expected to significantly accelerate DLPacker.

Others have already used ML/DL techniques for side chain packing. SIDEpro uses an ensemble of NNs that deliver packing RMSDs on par with SCWRL4 while being significantly faster.[13] OPUS-RotaNN uses a set of deep NNs (convolutional NN, recurrent NN, and transformer) on a set of hand-crafted features, including evolutionary features, and delivers moderate improvement of packing quality on some datasets.[19] Our method, in contrast, does not rely on evolutionary information and uses only information derived from target protein's

backbone conformation and sequence. We think that our new method can be effectively utilized for situations that require highly precise side chain localization and where the computational times and resources are not a limiting factor. It will be important to develop our approach further to make it faster and work in conjunction with other available models as part of an integrated workflow for structural biology and protein engineering applications.

5 | CONCLUSIONS

Structural prediction of side chain orientation necessitates effectively filling space while simultaneously avoiding steric clashes. Here, we employed a 3D convolution network to better capture the underlying three-dimensional spatial relationships and show our novel approach toward amino acid side chain packing outperforms classical methods. While DLPacker translates atomic coordinates to predict a spatial density before restoring the coordinates of the improved side chains, we envision further refinements to our method could remove the necessary spatial translation and simply predict the side chain coordinates from only a list of input 3D coordinates, which might be achieved using highly specialized NN architectures.

PEER REVIEW

The peer review history for this article is available at https://publons.com/publon/10.1002/prot.26311.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available in the supplementary material of this article. The code for DLPacker with some usage examples is available at https://github.com/nekitmm/DLPacker.

REFERENCES

1. Krivov GG, Shapovalov MV, Dunbrack Jr RL. Improved prediction of protein side-chain conformations with SCWRL4. *Proteins: Struct, Funct, and Bioinform*. 2009;77(4):778-795.
2. Gaillard T, Panel N, Simonson T. Protein side chain conformation predictions with an MMGBSA energy function. *Proteins: Struct, Funct, and Bioinform*. 2016;84(6):803-819.
3. Badaczewska-Dawid AE, Kolinski A, Kmiecik S. Computational reconstruction of atomistic protein structures from coarse-grained models. *Comput Struct Biotechnol J*. 2020;18:162-176.
4. Francis-Lyon P, Koehl P. Protein side-chain modeling with a protein-dependent optimized rotamer library. *Proteins: Struct, Funct, and Bioinform*. 2014;82(9):2000-2017.
5. Jumper JM, Faruk NF, Freed KF, Sosnick TR. Accurate calculation of side chain packing and free energy with applications to protein molecular dynamics. *PLoS Comput Biol*. 2018;14(12):e1006342.
6. Shapovalov MV, Dunbrack Jr RL. A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions. *Structure*. 2011;19(6):844-858.
7. Song H, Bremer BJ, Hinds EC, Raskutti G, Romero PA. Inferring protein sequence-function relationships with large-scale positive-unlabeled learning. *Cell Systems*. 2021;12(1):92-101.
8. Wallach I, Dzamba M & Heifets A Atomnet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. arXiv:1510.02855, 2015.
9. Senior AW, Evans R, Jumper J, et al. Improved protein structure prediction using potentials from deep learning. *Nature*. 2020;577(7792):706-710.
10. Torng W, Altman RB. 3D deep convolutional neural networks for amino acid environment similarity analysis. *BMC Bioinform*. 2017;18(1):1-23.
11. Cunningham JM, Koytiger G, Sorger PK, AlQuraishi M. Biophysical prediction of protein–peptide interactions and signaling networks using machine learning. *Nat Methods*. 2020;17(2):175-183.
12. Alley EC, Khimulya G, Biswas S, AlQuraishi M, Church GM. Unified rational protein engineering with sequence-based deep representation learning. *Nat Methods*. 2019;16(12):1315-1322.
13. Nagata K, Randall A, Baldi P. Sidepro: a novel machine learning approach for the fast and accurate prediction of side-chain conformations. *Proteins: Struct, Funct, and Bioinform*. 2012;80(1):142-153.
14. Maziarka Ł, Danel T, Mucha S, Rataj K, Tabor J & Jastrzębski S Molecule attention transformer. arXiv:2002.08264, 2020.
15. Lei Y, Li S, Liu Z, Wan F, Tian T, Li S, Zhao D, and Zeng J. Camp: a convolutional attention-based neural network for multifaceted peptide-protein interaction prediction. bioRxiv, 2020.
16. Rives A, Meier J, Sercu T, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc Natl Acad Sci USA*. 2021;118(15):e2016239118.
17. Biswas S, Khimulya G, Alley EC, Esvelt KM, Church GM. Low-n protein engineering with data-efficient deep learning. *Nat Methods*. 2021;18(4):389-396.
18. Jumper J, Evans R, Pritzel A, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*. 2021;596:583-589.
19. Gang X, Wang Q, Ma J. Opus-rota3: improving protein side-chain modeling by deep neural networks and ensemble methods. *J Chem Inf Model*. 2020;60(12):6691-6697.
20. Liu K, Sun X & Ma J et al. Prediction of amino acid side chain conformation using a deep neural network. arXiv:1707.08381, 2017.
21. Liu K, Ni Z & Zhou Z et al. Molecular modeling with machine-learned universal potential functions. arXiv:2103.04162, 2021.
22. Yanover C, Schueler-Furman O, Weiss Y. Minimizing and learning energy functions for side-chain prediction. *J Comput Biol*. 2008;15(7):899-911.
23. Morcos F, Pagnani A, Lunt B, et al. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proc Nat Acad Sci USA*. 2011;108(49):E1293-E1301.
24. Shroff R, Cole AW, Diaz DJ, et al. Discovery of novel gain-of-function mutations guided by structure-based deep learning. *ACS Synth Biol*. 2020;9(11):2927-2935.
25. Joosten RP, Salzemann J, Bloch V, et al. PDB_REDO: automated re-refinement of x-ray structure models in the PDB. *J Appl Crystall*. 2009;42(3):376-384.
26. Iglovikov V & Shvets A Ternausnet: U-Net with vgg11 encoder pre-trained on imagenet for image segmentation. arXiv:1801.05746, 2018.
27. Zongwei Zhou M, Siddiquee MR, Tajbakhsh N, Liang J. UNet++: a nested U-Net architecture for medical image segmentation. *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer; 2018:3-11.
28. Ronneberger O, Fischer P, and Brox T. U-Net: convolutional networks for biomedical image segmentation, International Conference on Medical Image Computing and Computer-Assisted Intervention, 234–241, 2015.
29. Çiçek Ö, Abdulkadir A, Lienkamp SS, Brox T, and Ronneberger O. 3d u-net: learning dense volumetric segmentation from sparse annotation, International Conference on Medical Image Computing and Computer-Assisted Intervention, 424–432, 2016.
30. Alom MZ, Yakopcic C, Hasan M, Taha TM, Asari VK. Recurrent residual u-net for medical image segmentation. *J Med Imag*. 2019;6(1):014006.
31. Alford RF, Leaver-Fay A, Jeliazkov JR, et al. The rosetta all-atom energy function for macromolecular modeling and design. *J Chem Theory Comput*. 2017;13(6):3031-3048.
32. Park H, Bradley P, Per Greisen YL Jr, et al. Simultaneous optimization of biomolecular energy functions on features from small molecules and macromolecules. *J Chem Theory Comput*. 2016;12(12):6201-6212.
33. Leaver-Fay A, Snoeyink J, and Kuhlman B. On-the-fly rotamer pair energy evaluation in protein design. International Symposium on Bioinformatics Research and Applications, pages 343–354, 2008.
34. Leaver-Fay A, Kuhlman B, Snoeyink J. An adaptive dynamic programming algorithm for the side chain placement problem. *Biocomputing 2005*. World Scientific; 2005:16-27.
35. Leaver-Fay A, Kuhlman B, and Snoeyink J. Rotamer-pair energy calculations using a trie data structure. International Workshop on Algorithms in Bioinformatics, 389–400. 2005.
36. Chaudhury S, Lyskov S, Gray JJ. Pyrosetta: a script-based interface for implementing molecular modeling algorithms using rosetta. *Bioinformatics*. 2010;26(5):689-691.
37. Peng J, Hosur R, Berger B & Xu J itreepack: protein complex side-chain packing by dual decomposition. arXiv:1504.05467, 2015.
38. Wang G, Dunbrack Jr RL. Pisces: a protein sequence culling server. *Bioinformatics*. 2003;19(12):1589-1591.
39. Moghadasi M, Mirzaei H, Mamonov A, et al. The impact of side-chain packing on protein docking refinement. *J Chem Inf Model*. 2015;55(4):872-881.
40. Olson MA, Lee MS. Structure refinement of protein model decoys requires accurate side-chain placement. *Proteins: Struct, Funct, and Bioinform*. 2013;81(3):469-478.

41. Combs SA, Mueller BK, Meiler J. Holistic approach to partial covalent interactions in protein structure prediction and design with Rosetta. *J Chem Inf Model*. 2018;58(5):1021-1036.

42. Zacharias N, Dougherty DA. Cation–π interactions in ligand recognition and catalysis. *Trends Pharmacol Sci*. 2002;23(6):281-287.

43. Gallivan JP, Dougherty DA. Cation-π interactions in structural biology. *Proc Nat Acad Sci*. 1999;96(17):9459-9464.

44. Brandl M, Weiss MS, Jabs A, Sühnel J, Hilgenfeld R. Ch-π-interactions in proteins. *J Mol Biol*. 2001;307(1):357-377.

45. Burley SK, Petsko GA. Amino-aromatic interactions in proteins. *FEBS Lett*. 1986;203(2):139-143.

46. Kumar S, Nussinov R. Close-range electrostatic interactions in proteins. *Chembiochem*. 2002;3(7):604-617.

47. Tina KG. Rana Bhadra, and Narayanaswamy Srinivasan. Pic: protein interactions calculator. *Nucl Acids Res*. 2007;35(suppl_2):W473-W476.

48. Lucas X, Bauzá A, Frontera A, Quinonero D. A thorough anion–π interaction study in biomolecules: on the importance of cooperativity effects. *Chem Sci*. 2016;7(2):1038-1050.

49. Ishu Saraogi VG, Vijay SD, Sekar K, Guru Row TN. C–halogen... π interactions in proteins: a database study. *Cryst Eng*. 2003;6(2):69-77.

50. Auffinger P, Hays FA, Westhof E, Ho PS. Halogen bonds in biological molecules. *Proc Nat Acad Sci USA*. 2004;101(48):16789-16794.

51. Vargas R, Garza J, Dixon DA, Hay BP. How strong is the cα-h...o$=$c hydrogen bond? *J Am Chem Soc*. 2000;122(19):4750-4755.

52. Philip V, Harris J, Adams R, et al. A survey of aspartate- phenylalanine and glutamate- phenylalanine interactions in the protein data bank: searching for anion-π pairs. *Biochemistry*. 2011;50(14):2939-2950.

53. Pinheiro S, Soteras I, Gelpí JL, et al. Cation–π–cation interactions in structural biology. 2nd BSC Doctoral Symposium, 2015 103–105.

54. Johansson KE, Hamelryck T. A simple probabilistic model of multi-body interactions in proteins. *Proteins: Struct, Funct, and Bioinform*. 2013;81(8):1340-1350.

55. Moal IH, Moretti R, Baker D, Fernández-Recio J. Scoring functions for protein–protein interactions. *Curr Opin Struct Biol*. 2013;23(6):862-867.

56. Li X, Liang J. Geometric cooperativity and anticooperativity of three-body interactions in native proteins. *Proteins: Struct, Funct, and Bioinform*. 2005;60(1):46-65.

57. Pereira JC, Caffarena ER, Santos CND. Boosting docking-based virtual screening with deep learning. *J Chem Inf Model*. 2016;56(12):2495-2506.

58. Shen C, Ding J, Wang Z, Cao D, Ding X, Hou T. From machine learning to deep learning: advances in scoring functions for protein–ligand docking. *Wiley Interdiscip Rev: Comput Mol Sci*. 2020;10(1):e1429.

59. Pearce R, Zhang Y. Deep learning techniques have significantly impacted protein structure prediction and protein design. *Curr Opin Struct Biol*. 2021;68:194-207.

60. Guedes IA, Barreto AMS, Marinho D, et al. New machine learning and physics-based scoring functions for drug discovery. *Sci Rep*. 2021;11(1):1-19.

SUPPORTING INFORMATION

Additional supporting information may be found in the online version of the article at the publisher's website.