FlowPacker: Protein side-chain packing with torsional flow matching

Jin Sub Lee
University of Toronto
jinsub.lee@mail.utoronto.ca

Philip M. Kim
University of Toronto
pm.kim@utoronto.ca

Abstract

Accurate prediction of protein side-chain conformations is necessary to understand protein folding, protein-protein interactions and facilitate de novo protein design. Here we apply torsional flow matching and equivariant graph attention to develop FlowPacker, a fast and performant model to predict protein side-chain conformations conditioned on the protein sequence and backbone. We show that FlowPacker outperforms previous state-of-the-art baselines across most metrics with improved runtime. We further show that FlowPacker can be used to inpaint missing side-chain coordinates and also for multimeric targets, and exhibits strong performance on a test set of antibody-antigen complexes. Code will be released shortly.

1 Introduction

A protein's three dimensional structure, determined by its primary amino acid sequence, is the main determinant of its function. The side-chain atoms heavily influences its folding and interaction with other proteins/ligands through various interatomic interactions and potentials. Therefore, to fully understand protein folding and identify protein-protein interactions, accurate models of protein side-chain packing must be developed.

Recent advances in artificial intelligence has made astounding progress in protein structure prediction [1, 2] and design [3, 4, 5]. Here we focus on the problem of side-chain packing, which seeks to predict the side-chain conformations given the amino acid sequence and backbone coordinates of a protein structure. Many prior efforts rely on physics-based modeling, which uses empirical scoring functions [6], discrete rotamer libraries [7], and/or MCMC-based sampling to identify plausible rotamers. However, these methods are often ineffective due to inefficient search algorithms and the reliance of often inaccurate scoring functions that converge to suboptimal local minima. Recent deep learning-based methods have shown significant improvements in runtime and efficacy to physics-based modeling in side-chain packing [8, 9, 10, 11], the most notable of which is DiffPack [10], a torsional diffusion model that autoregressively generates the four χ torsion angles that constitute the only degrees of freedom for side-chain conformations. DiffPack presents a number of innovations, such as autoregressive generation and confidence sampling, to attain state-of-the-art performance in protein side-chain packing. In this work, we approach the side-chain packing problem using flow matching and equivariant graph attention networks to attain state-of-the-art performance.

Flow matching [12] is a novel generative modeling paradigm that allows the training of continuous normalizing flows (CNFs) in a simulation-free manner, and have shown stronger performance, faster training convergence, and faster inference than standard diffusion models. In FlowPacker, we replace the torsional diffusion framework with torsional flow matching, derived from flow matching on Riemannian manifolds [13]. We also apply state-of-the-art equivariant graph attention models - namely, EquiformerV2 [14] - which has been shown to improve performance over invariant message passing networks due to increased expressivity and parameter efficiency.

Preprint. Under review.

We observe FlowPacker outperforms other side-chain packing baselines across most evaluated metrics while being considerably faster. We show that FlowPacker can be used for partial inpainting of side-chain conformations, a feature not readily available with other packing methods. We also show that FlowPacker can be extended to multimeric complexes and specifically with antibody-antigen complexes and show significant performance improvements in CDRH3 and full variable chain (Fv) side-chain packing. Therefore, it can be easily incorporated with existing backbone generative models and sequence design tools to generate accurate full-atom structures.

2 Background

Preliminaries. Proteins are macromolecules that adopt three-dimensional conformations based on its amino acid identity and resulting covalent and non-covalent interactions between atoms. The atoms that constitute a full-atom protein structure are divided into backbone atoms, which are the N, Ca, C, and O atoms that constitute the peptide backbone, and side-chain atoms, which are dependent on the amino acid identity and can contain up to 9 heavy atoms in total. Side-chain flexibility is usually limited to four degrees of freedom defined by the four χ torsion angles whose empirical distributions are highly constrained based on amino acid type and neighboring atomic forces. The problem of side-chain packing, therefore, can be reduced to finding $\chi_{1...4} \in [0, 2\pi)$ conditioned on the protein sequence $s \in \{0, ..., 19\}$ and backbone coordinates $X_{bb} \in \mathcal{R}^3$.

Related Work. Side-chain packing methods has traditionally relied on energy-based sampling such as Rosetta [6] or SCWRL [15], but has largely been replaced with the rise of performant deep learning methods. Notably, AttnPacker [9] is a model based on AlphaFold2 [1] that uses sparse triangular attention and invariant point attention to directly predict all-atom coordinates. DiffPack [10] proposes a diffusion-based approach that applies a relational graph convolution network on atom-level graphs, with several innovations such as autoregressive sampling to attain state-of-the-art performance. SidechainDiff [16] presents a sidechain packing diffusion model for the prediction of mutational effects using transfer learning. More recently, two peptide-specific torsional flow matching models have been published, where PPFlow [17] uses torsional flow matching on both backbone and side-chain torsion angles for peptide design, while PepFlow [18] uses a multi-modal approach to full-atom peptide design against protein pockets, using $S(E)3$ flow matching for backbone generation, simplex flow for amino acid sequences, and torsional flow matching for side-chain χ angles.

3 Methods

A schematic of FlowPacker is provided in Figure 1. Here we provide a brief overview of the theoretical background and model details.

3.1 Torsional Flow Matching

Continuous normalizing flows (CNFs) are a class of generative models that uses a learned vector field to transform a simple prior to the desired data distribution. However, the maximum likelihood simulation-based training of CNFs have limited its scalability and application to complex datasets. Recently, flow matching was proposed to train continuous normalizing flows (CNFs) in a simulation-free manner. Lipman et. al. [12] show that CNFs can be trained by regressing on a conditional probability path $p_t(x|x_1)$ to learn the unconditional path $p_t(x)$ that transforms a prior density $p(0)$ to the data distribution $p(1)$. Chen et al. [13] extends the flow matching framework to Riemannian manifolds including the high-dimensional torus, which is used in this work to define a torsional flow matching framework for side-chain conformation generation and briefly discussed below.

In flow matching, we desire to learn the time-conditioned vector field $v_t(x)$ with $t \in [0, 1]$ that transforms a simple prior distribution $p(0)$ to the data distribution $p(1)$, where the learned vector field can then be used to integrate a prior sample from $t = 0$ to $t = 1$ for generative modeling. However, the probability path $p_t(x)$ is intractable to compute. Lipman et. al. [12] show that we can use the *conditional vector field* $v_t(x|x_1)$ that generates *conditional probability paths* $p_t(x|x_1)$ as regression targets, which converges to the same optima as the unconditional vector field $v_t(x)$ that produces the unconditional probability path $p_t(x)$. Given a conditional flow $\psi_t(x_0|x_1)$ that transforms the prior distribution $p_0(x|x_1)$ to $p_t(x|x_1)$, we observe that:

$$\frac{d}{dt}\psi_t(x) = v_t(\psi_t(x_0|x_1)|x_1) = \dot{x}_t \tag{1}$$

which defines a conditional vector field that can serve as the regression target for simulation-free training, resulting in the conditional flow matching (CFM) loss below:

$$\mathcal{L}_{CFM} = \mathbb{E}_{t,p_1(x_1),p_0(x_0)}\left[||v_t(x_t) - \dot{x}_t||^2\right] \tag{2}$$

Chen and Lipman [13] introduce Riemannian flow matching, which extends flow matching to manifolds and general geometries. They show that the conditional flows $\psi_t(x|x_1)$ can be constructed on simple manifolds using a linear scheduler $\kappa(t) = 1 - t$ with the geodesic distance as the *premetric* that concentrates all mass at x_1 for $t = 1$. The geodesic distance for simple manifolds, such as the high-dimensional torus considered in this work, can be computed in closed form with the logarithmic map between two points on the manifold, and mapped back to the manifold using the exponential map. Thus, the conditional flow is defined as:

$$\psi_t(x_t|x_1) = \exp_{x_0}(t \log_{x_0}(x_1)) \tag{3}$$

where the exponential and logarithmic map for high-dimensional tori are defined as:

$$\exp_{x_0}(x_1) = (x_0 + x_1) \ \% \ (2\pi) \tag{4}$$

$$\log_{x_0}(x_1) = \text{arctan2}(\sin(x_1 - x_0)\cos(x_1 - x_0)) \tag{5}$$

This corresponds to a flow along the geodesic distance between x_1 and x_0 that varies linearly with time. This can be used to interpolate x_t in (2) to define the torsional flow matching loss used in this work.

3.2 Equivariant graph attention using Equiformerv2

Equivariant neural networks have been shown to increase model performance and efficiency by directly integrating the symmetries of the data into the model architecture, without the need for data augmentation nor additional parameters [19] [1]. Equiformer [21] is a graph attention network that operates on equivariant irreducible representation (irreps) features and transfers information between various type-L vectors using tensor product operations, where different type-L vectors can be combined through the use of Clebsch-Gordan coefficients. Equiformer uses depthwise tensor product (DTP) blocks - where one type-L vector in the output irreps is only dependent on one type-L‘ vector in the input irreps - to reduce computational complexity of tensor product, but is still practically restricted to max $L = 3$ vectors due to efficiency. For a more detailed discussion on tensor product operations, we refer readers to [22] and [23].

Equiformerv2 [14] facilitates the scaling of the model to higher-degree features by replacing $SO(3)$ convolutions with eSCN convolutions [24], which intuitively aligns the relative edge direction to one axis and therefore greatly sparsifies the Clebsch-Gordon coefficient matrix and simplifies to $SO(2)$ convolutions, reducing computational complexity and allowing scaling to higher-degree tensors up to $L = 6$ or $L = 8$ for increased model expressivity. While the task of side-chain packing does not necessitate the use of equivariant networks since torsion angles are invariant features, in preliminary studies we observed stronger empirical performance of Equiformerv2 over invariant and equivariant baseline architectures.

3.3 FlowPacker specifications

Dataset curation. We train FlowPacker on the BC40 dataset (release date 2020-07-28, available at `https://drug.ai.tencent.com/protein/bc40/download.html`), which are representative

[1] Although the need for equivariance has been challenged by recent works, notably AlphaFold3 [2], since equivariant networks often are more expensive to train over optimized Transformer architectures [20].

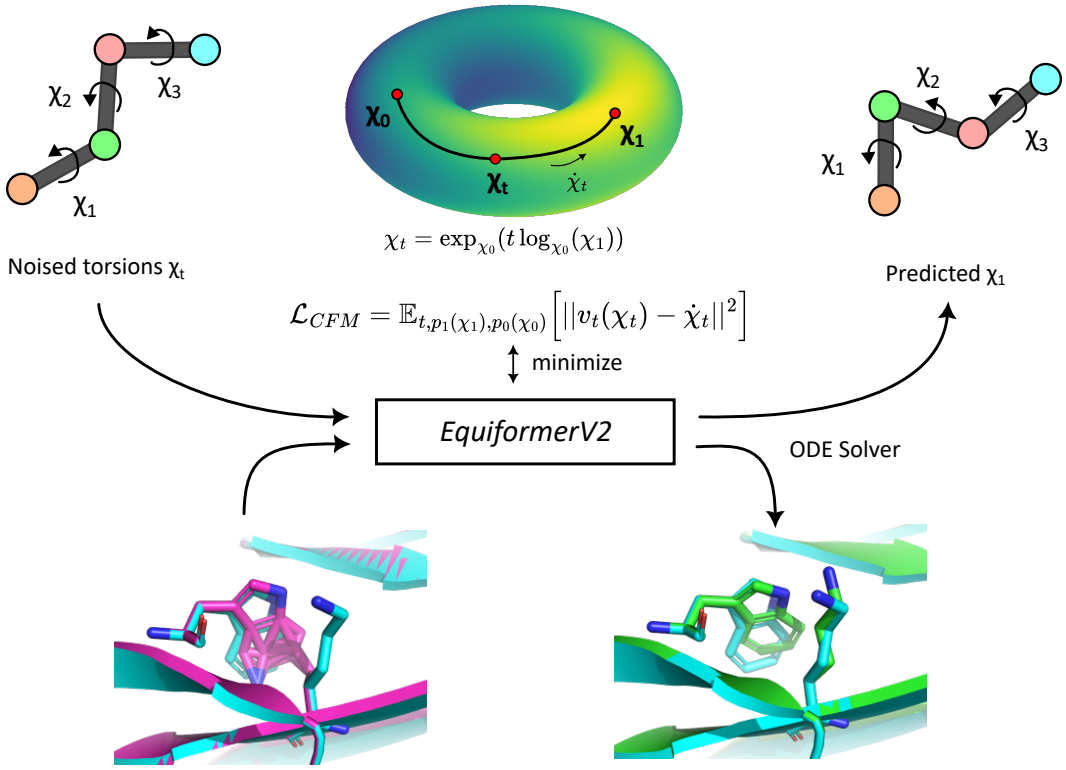

Figure 1: FlowPacker Overview. FlowPacker is an equivariant graph attention network that generates side-chain conformations of a given protein structure and sequence using torsional flow matching. The model predicts the vector field of the conditional flow along the hypertorus between a prior angle χ_0 and the ground-truth angle χ_1, which is used with an ODE Solver (ex. Euler's method) to generate a sample from the data distribution. Note we show a 'virtual' side-chain conformation with three χ angles, but each amino acid can contain up to 4 χ angles that affect the positions of up to 9 atoms.

PDB structures clustered by MMseqs2 [25] at 40% sequence identity, and is used by previous side-chain packing models (DiffPack and AttnPacker) for model training. We use the CASP13, 14, and 15 targets for testing (available at `https://predictioncenter.org/download_area/`). We reduce data redundancy in the training set by removing any structures with >40% sequence similarity to any of the CASP13/14/15 targets using MMseqs2's `easy-search` workflow. We discard structures of which more than 25% of residues are unknown, and remove residues with missing backbone coordinates or overlapping alpha carbon positions. We also filter for proteins with at least 40 residues. This results in 36,451 training examples and 99 test structures, with 20, 34, and 45 targets in CASP13, 14, and 15, respectively.

Model architecture. FlowPacker uses the Equiformerv2 model architecture as described previously. We use $L_{max} = 3$, channel dimension of 256, and 4 blocks, resulting in a total of 18.0M trainable parameters. Interestingly, we did not observe a change in performance from scaling L_{max} up to 6, suggesting that higher-order features are not informative for side-chain packing.

Loss functions. The model is trained to predict the conditional vector field as defined in Section 3.1:

$$\mathcal{L}_{CFM} = \mathbb{E}_{t,p_1(\chi_1),p_0(\chi_0)}\left[\frac{1}{(1-t)^2}||v_t(\chi_t) - \dot{\chi}_t||^2\right] \tag{6}$$

where $v_t(\chi_t)$ corresponds to the model output, $\dot{\chi}_t$ is the exact conditional vector field efficiently computed with autograd, and $\frac{1}{(1-t^2)}$ is a weighting factor.

Dataset	Method	Angle MAE ° ↓				Angle Accuracy % ↑				Atom RMSD Å ↓		
		χ_1	χ_2	χ_3	χ_4	χ_1	χ_2	χ_3	χ_4	All	Core	Surface
CASP13	Rosetta	24.84	30.96	45.35	58.28	78.36%	67.75%	47.92%	41.23%	0.822	0.502	1.025
	AttnPacker	17.82	33.41	67.31	**48.89**	83.32%	64.96%	32.76%	41.67%	0.745	0.531	0.874
	AttnPacker-pp	16.33	26.00	51.18	49.40	83.53%	70.05%	41.83%	41.10%	0.676	0.456	**0.815**
	DiffPack-fix	22.14	28.80	45.20	52.28	81.63%	71.57%	50.98%	48.89%	0.789	0.432	1.005
	FlowPacker	16.19	23.82	41.07	51.33	85.53%	75.56%	51.01%	**50.66%**	0.683	**0.399**	0.874
	+ confidence	**15.96**	**22.93**	**40.66**	52.12	**85.86%**	**76.48%**	**51.41%**	50.30%	**0.675**	0.412	0.865
CASP14	Rosetta	32.32	35.47	49.19	54.27	67.62%	58.48%	39.52%	42.15%	1.001	0.697	1.199
	AttnPacker	27.29	39.26	67.94	49.99	70.77%	55.22%	27.44%	37.12%	0.955	0.675	1.130
	AttnPacker-pp	26.06	32.75	55.06	50.59	71.03%	59.57%	34.53%	36.87%	0.900	0.607	1.085
	DiffPack-fix	31.00	34.43	51.72	57.50	69.59%	61.32%	38.69%	39.35%	0.994	0.639	1.225
	FlowPacker	22.66	28.59	**44.57**	51.01	76.78%	66.87%	**45.70%**	45.75%	0.844	**0.522**	1.060
	+ confidence	**22.33**	**28.54**	44.70	**49.79**	**77.04%**	**66.95%**	45.62%	**46.58%**	**0.837**	0.525	**1.047**
CASP15	Rosetta	32.37	35.22	45.29	58.92	70.09%	61.81%	43.91%	41.93%	0.938	0.611	1.158
	AttnPacker	28.16	41.90	69.90	**53.22**	71.99%	55.15%	30.24%	36.70%	0.925	0.696	1.099
	AttnPacker-pp	26.80	32.74	56.78	53.98	72.29%	61.65%	36.04%	36.70%	0.851	0.549	1.052
	DiffPack-fix	31.86	34.46	48.06	61.01	71.08%	62.93%	44.11%	41.27%	0.921	0.524	1.145
	FlowPacker	23.78	28.69	**40.91**	54.40	77.79%	69.01%	**49.48%**	**46.88%**	0.779	0.415	0.995
	+ confidence	**23.60**	**28.53**	41.04	54.37	**77.98%**	**69.34%**	49.19%	45.72%	**0.774**	**0.411**	**0.991**

Table 1: Performance evaluation of FlowPacker and other methods on CASP targets. The top performing model per test dataset is highlighted in **bold**. DIFFPACK-FIX shows results for DiffPack using structures with idealized bond lengths and angles as input. We display the average values of 16 generations for FlowPacker, and use default settings for DiffPack (4 samples with confidence model selection). AttnPacker-pp corresponds to samples with post-processing applied as denoted in [9].

We experimented with multiple variations of the loss function including using the vector field approximation $\log_{\chi_t}(\chi_1)/(1-t)$, which showed negligible differences in training dynamics and performance. We also tried using a reparameterized direct χ_1 prediction as with rotation matrices in FrameFlow [4], which can be used to approximate the vector field during sampling. However, this caused unstable training dynamics, most likely due to the degeneracy of χ angles.

Handling symmetry issues. Certain χ angles exhibit π-symmetry that may adversely affect model efficiency. We handle these cases by taking `mod` π for π-symmetric χ angles and modifying equation (4) to:

$$\exp_{\chi_0}(\chi_1) = (\chi_0 + \chi_1) \ \% \ (c\pi), \tag{7}$$

where $c = 1$ for π-symmetric χ angles and $c = 2$ otherwise, restricting π-symmetric angles to $[0, \pi)$.

Training details. In FlowPacker, we use an residue-level graph, where each residue represents a node with the position given by its idealized Cβ atom. At each training step, we sample timesteps t and generate noised torsions χ_t, which are used to reconstruct noised atomic coordinates X_t using idealized coordinates. The node features consist of the amino acid identity, backbone torsion angles, and sinusoidal embeddings of the current timestep, while the edge features are relative positional encoding clamped at $[-32, 32]$, and the Euclidean distance between all idealized coordinates (using the atom14 representation) between two connected residues. The edges are defined by a k-nn graph with $k = 30$. The model is trained on 4 40GB NVIDIA A100s, effective batch size of 16, and context length of 512 residues (cropped when length is greater than 512) for 300 epochs, or approximately 6 days. We use the AdamW optimizer with learning rate $1.0e^{-4}$ and gradient clipping at a norm value of 1.0.

Inference strategies. Though the model was trained using a conditional flow with a linear scheduler $\kappa(t) = 1 - t$, we observe a similar phenomenon as described in FrameFlow where the generated samples are poor using the same linear schedule during sampling. Instead, we adopt the exponential schedule $v_t = c(\log_{x_t}(x_1))$, and empirically find that c = 5 works well. We use a uniform distribution on $SO(2)$ as the base distribution and Euler solver with t = 10 for all sampling steps, as we observed no significant increase in performance with additional timesteps. We experimented with using higher-order solvers such as Heun's method and RK45 solvers, but did not observe a noticeable performance improvement at the cost of increased runtime.

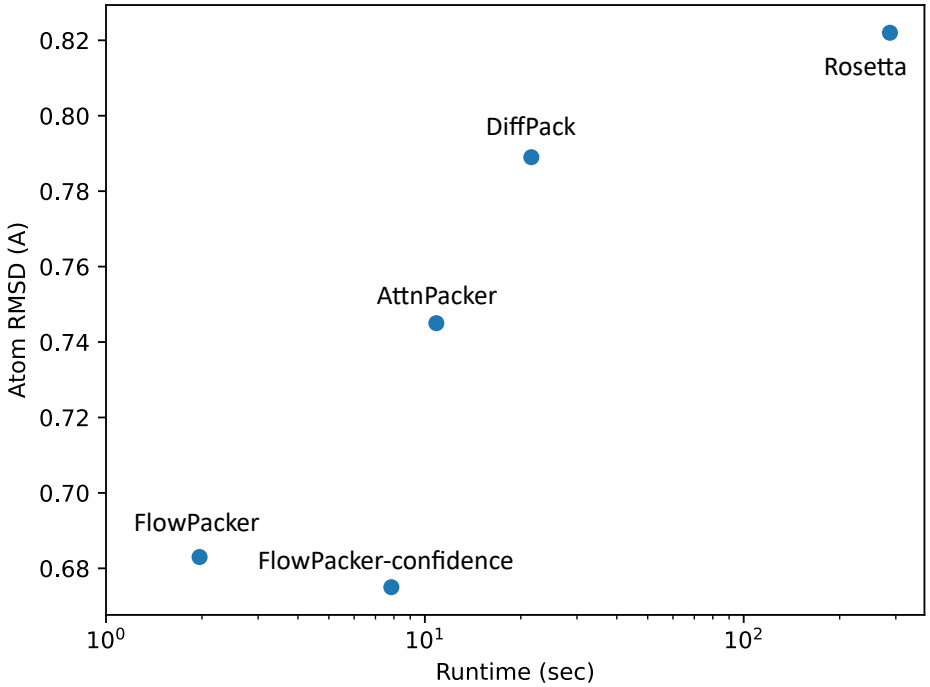

Figure 2: Runtime and performance analysis of various methods. We use the CASP13 dataset to analyze the total runtime (averaged per protein) and Atom RMSD across the tested methods. FlowPacker exhibits the best RMSD-runtime tradeoff, while FlowPacker with the confidence model generates conformations with the lowest RMSD. We report the metrics of DIFFPACK-FIX in Table 1 here, and all runtime metrics are averaged over three independent seeds for each method. All models were tested on AMD Ryzen 7 5800 (Rosetta) or NVIDIA RTX 3060 (others). Note that runtime is plotted in log scale.

Confidence model. The confidence model is trained in a similar manner to the main model, except we regress on the residue-level sidechain RMSDs rather than χ-level vector fields. We identify the best sample by simply taking the mean predicted RMSD across all residues and selecting the sample with the lowest predicted RMSD. Unless otherwise noted, we generate 4 samples per test case and select the highest confidence sample. The confidence model trained in this work contains 2.2M trainable parameters with max $l = 2$ and hidden dimension of 64 - we did not perform extensive hyperparameter tuning on this module.

4 Results

We compare FlowPacker to physics-based (Rosetta [6]) and deep learning-based (AttnPacker [9] and DiffPack [10]) methods across three test datasets - CASP13, CASP14, and CASP15 - in Table 1. As with previous studies [9, 10], we report three metrics: 1. ANGLE MAE corresponds to the angular mean absolute error, which returns $\min(\chi, \chi \bmod 2\pi)$, 2. ANGLE ACCURACY refers to the percentage of angles within $20°$ of the ground-truth, and 3. ATOM RMSD refers to the average root-mean squared deviation of side-chain atoms per residue, where core residues are defined as residues with at least 20 $C\beta$ atoms within $10\dot{A}$, and surface residues are those with at most 15 $C\beta$ within $10\dot{A}$.

During testing, we observed a data leakage issue with DiffPack, where the generation quality was dependent on the input side-chain coordinates, everything else held constant. We noticed that during training and sampling, DiffPack rotates the ground-truth side-chain coordinates given χ_t, maintaining

	ANGLE MAE ° ↓				RMSD Å ↓
Masking	χ_1	χ_2	χ_3	χ_4	All
5%	20.99	26.73	30.60	46.75	0.681
10%	20.68	28.67	35.20	58.82	0.696
25%	22.95	26.20	39.50	55.22	0.751
50%	22.63	27.94	39.23	54.97	0.761
75%	23.52	29.38	41.00	54.58	0.787
100%	23.78	28.69	40.91	54.40	0.779

Table 2: Performance on CASP15 targets with varying levels of masking. We mask varying proportions of residues randomly and calculate Angle MAE and Atom RMSD on the masked residues.

the ground-truth bond angles and bond lengths. Moreover, we discovered that if we idealize the side-chain atoms (as in AlphaFold2 [1] and FlowPacker) and provide the idealized structures (note we still use the non-idealized ground-truth backbone coordinates) to DiffPack, the generation quality is significantly worse (see DIFFPACK-FIX in Table 1). We hypothesize that although the $\chi_{1...4}$ angles are the primary degrees of freedom in side-chain packing, the atomic coordinates do exhibit minor deviations from various interatomic forces and potentials [26] that may provide clues into the ground-truth side-chain conformation. Moreover, we observed that using ground-truth bond angles and lengths provides a >0.25A RMSD advantage over using idealized ones in terms of side-chain RMSD. For further discussion, please refer to Supplementary Information 1.1. Since side-chain packing methods should require no *a priori* knowledge of ground-truth side-chain positions, we exclude DiffPack from ranking across metrics. We note that the numbers denoted under DIFFPACK-FIX may not accurately reflect the true performance of DiffPack since the model is not explicitly trained with idealized coordinates - however, this requires further investigation that is outside of the scope of this work.

We observe that FlowPacker outperforms all baselines across most metrics, suggesting the superiority of flow matching and equivariant networks over diffusion and invariant models, respectively. Using a confidence model to select the lowest predicted RMSD sample marginally improved performance across all test datasets. However, the < 0.1 RMSD decrease suggests that for high-throughput screening, the confidence model may not be necessary since it requires multiple generations (4 used here), each with an extra forward pass through the confidence model that results in a > 4X increase in runtime. We observe that χ_1 accuracy is generally higher than the subsequent χ angles, as seen with previous methods, due to the lever effect where the errors accumulate through each χ angle and the increased flexibility of side-chain positions at longer side-chains (ex. arginines and leucines). We also analyze runtime and performance across all methods in Figure 2, and observe that FlowPacker exhibits the best runtime and performance of all tested baselines. As shown in Supplementary Figure 2, FlowPacker can recapitulate the χ distributions with decent accuracy, including π-symmetric ones, which lie in the interval $[0, 180]$ due to the parameterization in Equation 7. An analysis of residue-level atom RMSDs for each unique amino acid reveals that longer and aromatic sidechains are often harder to predict accurately over shorter ones Supplementary Figure 3.

To test FlowPacker's ability to inpaint missing side-chain coordinates, we mask 5-75% randomly selected residues on the CASP15 test set and report the metrics in Table 2. As expected, we observe lower RMSDs as we provide more structural context, suggesting the utility of FlowPacker for conditional design. We expect this to be useful in common protein design cases such as motif-scaffolding or interface design, where protein backbone generative models design the backbone coordinates, and FlowPacker can be used in conjunction with sequence design tools to generate full-atom coordinates.

Although FlowPacker was specifically trained on single-chain proteins, we tested whether the model can be used to generate side-chain conformations of antigen-antibody complexes. We extract antibody-antigen complexes from SabDab [27] and filter for structures released after 2021-01-01. We also remove complexes with at least 40% antigen sequence similarity with MMseqs [25] to any complex structure published prior to 2021-01-01 to remove redundancy, and crop antigens to 256 residues

Target	Method	ANGLE MAE ° ↓				RMSD Å ↓
		χ_1	χ_2	χ_3	χ_4	All
CDRH3	Rosetta	28.05	26.04	**38.16**	58.64	0.886
	FlowPacker	**20.85**	**25.06**	42.16	**51.67**	**0.671**
Full Fv	Rosetta	30.28	30.72	48.50	54.83	0.790
	FlowPacker	**22.79**	**24.32**	**41.71**	**51.03**	**0.649**

Table 3: Test case on side-chain packing on antibody-antigen complexes.

proximal to the antibody. This results in 104 clusters, from which we select a representative member (provided by MMseqs) from each cluster for testing. For inference on multi-chain proteins, we simply fix the relative positional encodings for edges between residues of different chains to 32 and all other input features are kept the same as in single-chain inference. In Table 3, we report the metrics for two common design tasks: CDRH3 design and full Fv design. For all cases, we provide the antigen structure as conditional inputs and pack the sidechains of the CDRH3 or full Fv region (or heavy chain only if light chain data is not available). We only use Rosetta as the baseline, as the other baseline methods are not adapted for multimeric inputs. FlowPacker outperforms Rosetta in both tasks and across most metrics despite not being explicitly trained on multimeric complexes. Therefore, FlowPacker can be used to generate more accurate full-atom structures of both monomeric and multimeric proteins, not limited to antibody-antigen complexes.

5 Discussion

In this work, we present FlowPacker, a torsional flow matching model that exhibits superior performance and runtime efficiency over other baselines in side-chain packing. We show that FlowPacker can be used to inpaint partial residues and multi-chain inference, showcasing a test case with antibody-antigen CDR side-chain packing, outperforming other methods in angle accuracy and atom RMSD.

We see various promising avenues for future work, such as improved prediction of mutational effects using unsupervised [28] or supervised [16] learning, or alignment of generative models using preference data [29, 30] such as empirical force fields to increase biophysical plausibility. We also believe that FlowPacker's performance can be improved - we did not test autoregressive sampling as in DiffPack, which may help performance. We show that the confidence model modestly improves performance, but there are other applications such as resampling of low-confidence conformations and uncertainty analysis as with pLDDT in AlphaFold that may be further explored. We also encourage the exploration of novel representations of side-chain conformations, as explicit representation in 3D space [31, 2] may outperform implicit representations since the atom RMSD using a χ angle parameterization is often largely dependent on the accuracy of the χ_1 prediction. Finally, we envision accurate side-chain packing models to be increasingly necessary with the development of more powerful protein backbone generative models to provide a pipeline for end-to-end full-atom protein design.

6 Acknowledgements

We thank the many open-source codebases that this work was based on, including PyTorch, OpenFold, and e3nn. We also would like to thank Alliance Canada for computing resources.

References

[1] J. Jumper *et al.*, "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021.

[2] J. Abramson *et al.*, "Accurate structure prediction of biomolecular interactions with alphafold 3," *Nature*, pp. 1–3, 2024.

[3] J. L. Watson *et al.*, "De novo design of protein structure and function with rfdiffusion," *Nature*, vol. 620, no. 7976, pp. 1089–1100, 2023.

[4] J. Yim *et al.*, "Fast protein backbone generation with se (3) flow matching," *arXiv preprint arXiv:2310.05297*, 2023.

[5] J. S. Lee, J. Kim, and P. M. Kim, "Score-based generative modeling for de novo protein design," *Nature Computational Science*, vol. 3, no. 5, pp. 382–392, 2023.

[6] R. F. Alford *et al.*, "The rosetta all-atom energy function for macromolecular modeling and design," *Journal of chemical theory and computation*, vol. 13, no. 6, pp. 3031–3048, 2017.

[7] R. L. Dunbrack Jr and M. Karplus, "Backbone-dependent rotamer library for proteins application to side-chain prediction," *Journal of molecular biology*, vol. 230, no. 2, pp. 543–574, 1993.

[8] M. Misiura, R. Shroff, R. Thyer, and A. B. Kolomeisky, "Dlpacker: Deep learning for prediction of amino acid side chain conformations in proteins," *Proteins: Structure, Function, and Bioinformatics*, vol. 90, no. 6, pp. 1278–1290, 2022.

[9] M. McPartlon and J. Xu, "An end-to-end deep learning method for protein side-chain packing and inverse folding," *Proceedings of the National Academy of Sciences*, vol. 120, no. 23, e2216438120, 2023.

[10] Y. Zhang, Z. Zhang, B. Zhong, S. Misra, and J. Tang, "Diffpack: A torsional diffusion model for autoregressive protein side-chain packing," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[11] N. Z. Randolph and B. Kuhlman, "Invariant point message passing for protein side chain packing," *Proteins: Structure, Function, and Bioinformatics*, 2024.

[12] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," *arXiv preprint arXiv:2210.02747*, 2022.

[13] R. T. Chen and Y. Lipman, "Flow matching on general geometries," in *The Twelfth International Conference on Learning Representations*, 2024.

[14] Y.-L. Liao, B. Wood, A. Das, and T. Smidt, "Equiformerv2: Improved equivariant transformer for scaling to higher-degree representations," *arXiv preprint arXiv:2306.12059*, 2023.

[15] Q. Wang, A. A. Canutescu, and R. L. Dunbrack Jr, "Scwrl and molide: Computer programs for side-chain conformation prediction and homology modeling," *Nature protocols*, vol. 3, no. 12, pp. 1832–1847, 2008.

[16] S. Liu, T. Zhu, M. Ren, C. Yu, D. Bu, and H. Zhang, "Predicting mutational effects on protein-protein binding via a side-chain diffusion probabilistic model," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[17] H. Lin *et al.*, "Ppflow: Target-aware peptide design with torsional flow matching," *bioRxiv*, pp. 2024–03, 2024.

[18] J. Li *et al.*, "Full-atom peptide design based on multi-modal flow matching," *arXiv preprint arXiv:2406.00735*, 2024.

[19] C. K. Joshi, C. Bodnar, S. V. Mathis, T. Cohen, and P. Lio, "On the expressive power of geometric graph neural networks," in *International conference on machine learning*, PMLR, 2023, pp. 15 330–15 355.

[20] T. Dao, D. Fu, S. Ermon, A. Rudra, and C. Ré, "Flashattention: Fast and memory-efficient exact attention with io-awareness," *Advances in Neural Information Processing Systems*, vol. 35, pp. 16 344–16 359, 2022.

[21] Y.-L. Liao and T. Smidt, "Equiformer: Equivariant graph attention transformer for 3d atomistic graphs," *arXiv preprint arXiv:2206.11990*, 2022.

[22] M. Geiger and T. Smidt, "E3nn: Euclidean neural networks," *arXiv preprint arXiv:2207.09453*, 2022.

[23] F. Fuchs, D. Worrall, V. Fischer, and M. Welling, "Se (3)-transformers: 3d roto-translation equivariant attention networks," *Advances in neural information processing systems*, vol. 33, pp. 1970–1981, 2020.

[24] S. Passaro and C. L. Zitnick, "Reducing so (3) convolutions to so (2) for efficient equivariant gnns," in *International Conference on Machine Learning*, PMLR, 2023, pp. 27 420–27 438.

[25] M. Steinegger and J. Söding, "Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets," *Nature biotechnology*, vol. 35, no. 11, pp. 1026–1028, 2017.

[26] R. Engh and R. Huber, "Structure quality and target parameters," 2012.

[27] J. Dunbar *et al.*, "Sabdab: The structural antibody database," *Nucleic acids research*, vol. 42, no. D1, pp. D1140–D1146, 2014.

[28] S. Luo, Y. Su, Z. Wu, C. Su, J. Peng, and J. Ma, "Rotamer density estimator is an unsupervised learner of the effect of mutations on protein-protein interaction," *bioRxiv*, pp. 2023–02, 2023.

[29] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[30] X. Zhou, D. Xue, R. Chen, Z. Zheng, L. Wang, and Q. Gu, "Antigen-specific antibody design via direct energy-based preference optimization," *arXiv preprint arXiv:2403.16576*, 2024.

[31] J. H. Lee *et al.*, "Equifold: Protein structure prediction with a novel coarse-grained structure representation," *Biorxiv*, pp. 2022–10, 2022.

Supplementary Information

1.1 Performance issues with DiffPack.

The discrepancy in performance of DiffPack - as discussed in the main text - can be attributed to the way that it handles noising and denoising of the side-chain χ angles. We observed that DiffPack uses the ground-truth bond angles and bond lengths without idealization during training and inference [2], which is an issue due to two main reasons: 1. inference is not possible when the ground-truth side-chain coordinates are not known (ex. samples from backbone generative models), and 2. *a priori* information of input side-chain conformations should not impact predictive performance. The first reason is a trivial issue since any random initialization of torsion angles can be easily applied to backbone-only structures to serve as a starting point for side-chain packing tools. However, we discovered that for structures where ground-truth bond angles and bond lengths are unknown (i.e. using idealized structures), DiffPack performs notably worse, limiting applicability to side-chain packing for structures without access to ground-truth sidechain conformations.

First, we verify that the ground-truth bond angles and bond lengths are unchanged before and after DiffPack in Supplementary Figure 1. We analyze both Ca-Cβ bond length and N-Ca-Cβ bond angles for DiffPack and FlowPacker, and observe that DiffPack perfectly recapitulates the ground-truth distributions of both bond angles and bond lengths, while FlowPacker does not since it uses idealized bond angles and lengths.

To assess the performance discrepancy of using varying structures that **only differ in the side-chain coordinates**, we input various structures to both DiffPack and AttnPacker, one of the side-chain packing baselines used in this paper, and report the results in Supplementary Table 1. Note that in all the different input structures, the backbone coordinates are unchanged. First, we report that idealization of bond lengths and angles results in a $> 0.25A$ increase in RMSD, which provides significant advantage to DiffPack given that side-chain packing tools attains sub-Angstrom accuracy. Moreover, we observe that DiffPack shows worse performance when we use idealized, Rosetta-packed, and AttnPacker-packed structures as input, suggesting that the ground-truth bond angles and lengths contribute significantly to DiffPack's performance. On the other hand, AttnPacker shows the same performance across all inputs, since it is agnostic to the input sidechain coordinates. Therefore, we conclude that DiffPack does exhibit data leakage to some extent, since side-chain packing models should only be dependent on the backbone coordinates and not the input structure's side-chain coordinates. The current version of DiffPack may be useful when searching sidechain conformational states of known protein structures, but its applicability to structures without knowledge of the ground-truth side-chain conformations may be limited. We also note that DiffPack's performance may improve when retrained with idealized bond angles and lengths. We have contacted the authors of DiffPack to notify them of our findings.

[2] see `rotate_side_chain` function in `https://github.com/DeepGraphLearning/DiffPack/blob/main/diffpack/rotamer.py`

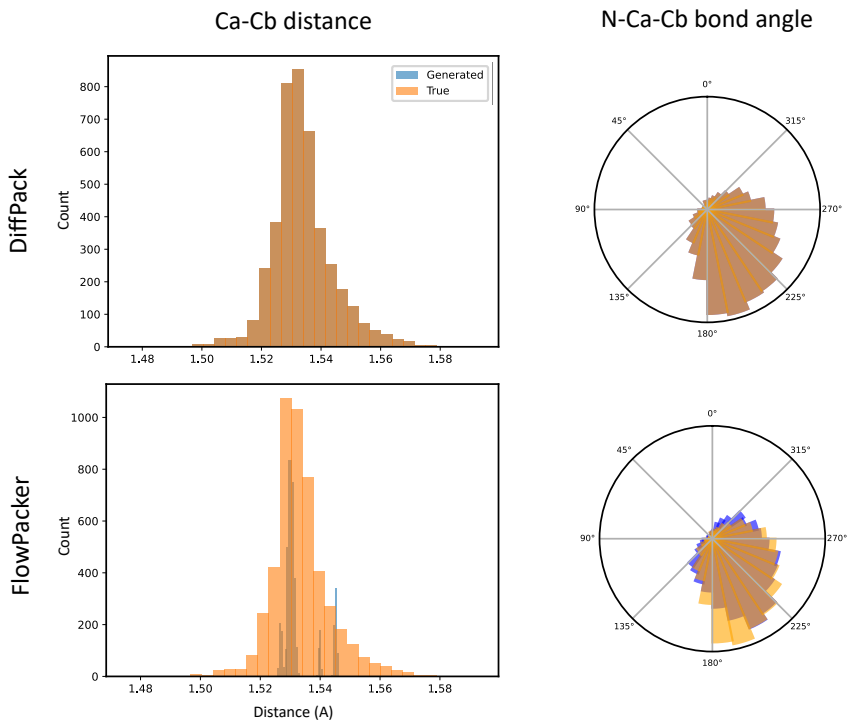

Figure S1: Analysis of N-Ca-Cβ bond angle and Ca-Cβ bond lengths before and after DiffPack and FlowPacker. We observe that DiffPack perfectly recapitulates the input structure's bond angle and lengths, which we believe may be a source of data leakage since it is not fully agnostic to the ground-truth sidechain. FlowPacker uses idealized coordinates given the noised torsions, which results in different distributions from the input ground-truth structures. Note that the variations in idealized distances of FlowPacker is a result of using ground-truth Ca coordinates with idealized Cβ ones. The plots were generated by using the CASP13 test set with the respective models.

		ANGLE MAE ° ↓				RMSD Å ↓
Model	**Input**	χ_1	χ_2	χ_3	χ_4	All
Idealizer	Ground Truth	1.33	0.02	0.03	0.03	0.254
DiffPack	Ground Truth	14.07	23.20	36.79	48.13	0.571
	Idealized	22.26	29.96	46.61	53.47	0.778
	Rosetta	21.43	29.90	46.20	54.25	0.771
	AttnPacker-pp	21.98	29.22	45.35	54.77	0.780
AttnPacker	Ground Truth	16.33	27.46	50.42	49.40	0.677
	Idealized	16.33	27.46	50.42	49.40	0.677
	Rosetta	16.33	27.46	50.42	49.40	0.677
	AttnPacker-pp	16.33	27.46	50.42	49.40	0.677

Table S1: Performance metrics of DiffPack and AttnPacker on various input structures on the CASP13 test set. We observe that idealization of bond angles and lengths results in a >0.25A RMSD from the ground-truth, suggesting models that directly use the ground-truth bond angles and lengths significantly benefits atom RMSD performance over idealized or agnostic packing tools.

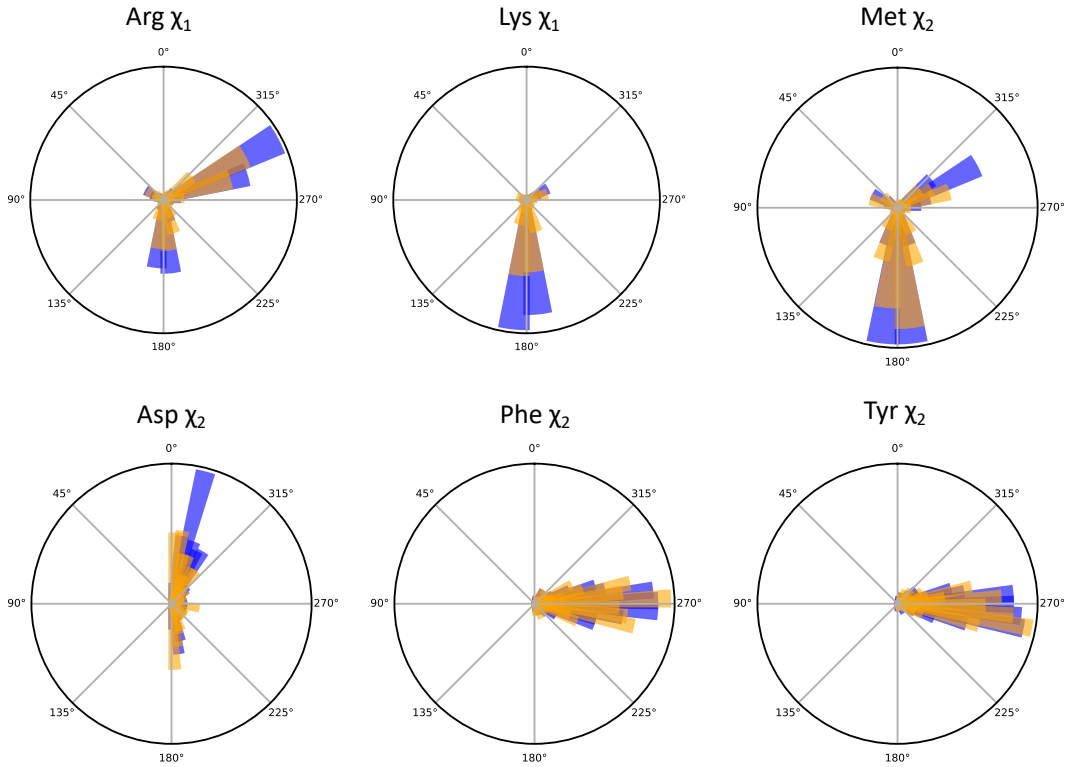

Figure S2: Generated vs. true χ distributions. Three randomly selected χ distributions are listed on the top row, while the bottom row contains three π-symmetric χ angles that lie in the interval $[0, 180]$ due to the parameterization in FlowPacker. Generated χ angles are depicted in orange, while ground-truth χ angles are in blue. The distributions are generated using the CASP13 test set.

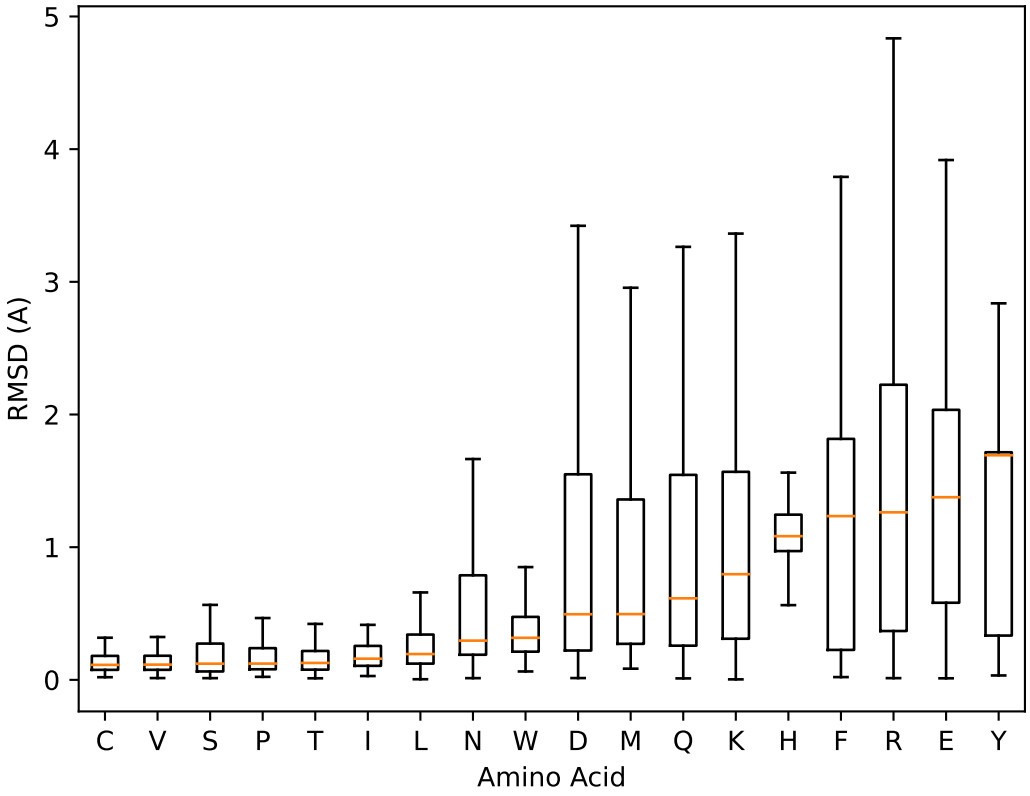

Figure S3: RMSD distributions per amino-acid identity, ranked from lowest to highest median atom RMSD. We observe that aromatic (Y, F, H) and long linear (E, R, K) sidechains exhibit the highest median RMSD.