PROTEIN SEQUENCE DESIGN WITH A LEARNED POTENTIAL

Namrata Anand-Achim
Department of Bioengineering
Stanford University
namrataa@stanford.edu

Raphael R. Eguchi
Department of Biochemistry
Stanford University
reguchi@stanford.edu

Alexander Derry
Department of Biomedical Data Science
Stanford University
aderry@stanford.edu

Russ B. Altman
Departments of Bioengineering, Genetics, and Medicine
Stanford University
russ.altman@stanford.edu

Po-Ssu Huang
Department of Bioengineering
Stanford University
possu@stanford.edu

ABSTRACT

The primary challenge of fixed-backbone protein design is to find a distribution of sequences that fold to the backbone of interest. This task is central to nearly all protein engineering problems, as achieving a particular backbone conformation is often a prerequisite for hosting specific functions. In this study, we investigate the capability of a deep neural network to learn the requisite patterns needed to design sequences. The trained model serves as a potential function defined over the space of amino acid identities and rotamer states, conditioned on the local chemical environment at each residue. While most deep learning based methods for sequence design only produce amino acid sequences, our method generates full-atom structural models, which can be evaluated using established sequence quality metrics. Under these metrics we are able to produce realistic and variable designs with quality comparable to the state-of-the-art. Additionally, we experimentally test designs for a *de novo* TIM-barrel structure and find designs that fold, demonstrating the algorithm's generalizability to novel structures. Overall, our results demonstrate that a deep learning model can match state-of-the-art energy functions for guiding protein design.

Significance

Protein design tasks typically depend on carefully modeled and parameterized heuristic energy functions. In this study, we propose a novel machine learning method for fixed-backbone protein sequence design, using a learned neural network potential to not only design the sequence of amino acids but also select their side-chain configurations, or rotamers. Factoring through a structural representation of the protein, the network generates designs on par with the state-of-the-art, despite having been entirely learned from data. These results indicate an exciting future for protein design driven by machine learning.

Introduction

Computational protein design has emerged as a powerful tool to expand the space of known protein folds [1, 2, 3, 4, 5, 6], create variations on existing topologies [7, 8, 9, 10, 11], and access the vast space of sequences yet to be traversed by evolution [12]. These advances have enabled significant achievements in the engineering of therapeutics [13, 14, 15], biosensors [16, 17, 18], enzymes [19, 20, 21, 22], and more [23, 24, 25, 26, 27]. Key to such successes are robust sequence design methods that minimize the folded-state energy of a pre-specified backbone conformation, which can either be derived from existing structures or generated *de novo*. This difficult task is often described as the "inverse" of protein folding—given a protein backbone, design a sequence that folds into that conformation.

Current approaches for fixed-backbone sequence design commonly involve specifying an energy function and sampling sequence space to find a minimum-energy configuration. Methods for optimizing over residues and rotamers include variations on the dead-end elimination algorithm [28, 29, 30, 31], integer linear programming [32, 33], belief propagation [34, 35], and Markov Chain Monte Carlo (MCMC) with simulated annealing [36, 37, 38]. RosettaDesign [39, 40], an example of the latter, is the only method among these that has been broadly experimentally validated. [39, 40].

Most molecular mechanics force fields are highly sensitive to specific atom placement, and as a result the designed sequences can be convergent for a given starting backbone conformation. For most native proteins, however, the existence of many structural homologs with low sequence identity suggests that there are a distribution of viable sequences that fold into a given target structure. Access to these sequences can be important for function, as is evident in most evolutionary trajectories, both natural and *in vitro* [41, 42]. In practice, the generation of diverse sequences with Rosetta often requires manually manipulating amino acid identities at key positions [7, 9], adjusting the energy function [43, 44], or explicitly modeling perturbations of the protein backbone [45, 46, 47, 48]. Design methods that account for flexibility of the protein backbone often use "softer" potentials [44, 49], such as statistical potentials that are derived from data [50, 51, 50, 52, 53, 54], but these methods often do not perform as favorably as well-parameterized "hard" molecular mechanics force fields.

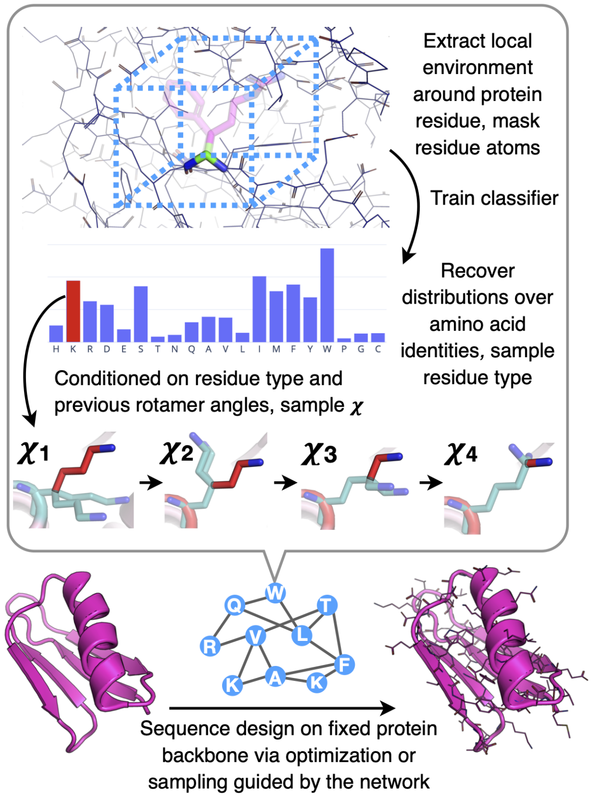
Extract local environment around protein residue, mask residue atoms
Train classifier
Recover distributions over amino acid identities, sample residue type
H K R D E S T N Q A V L I M F Y W P G C
Conditioned on residue type and previous rotamer angles, sample χ
χ1
χ2
χ3
χ4
Q W T R V L F K A K
Sequence design on fixed protein backbone via optimization or sampling guided by the network

Figure 1: Method Overview. We train a deep convolutional neural network (CNN) to predict amino acid identity and discretized rotamer angles given the local environment around the residue (box not to scale). Amino acid type and rotamer angles are sampled in an autoregressive fashion for each residue position. Sequences are designed onto fixed protein backbones by sampling or optimization of the pseudo-log-likelihood of the sequence under the model.

We sought to develop a method that could potentially produce diverse sequences from a fixed starting backbone while capturing accurate physical interactions to guarantee the viability of the sequences. Additionally, while conventional energy functions used in sequence design calculations are often composed of pairwise terms that model interatomic interactions, we hypothesized that a more comprehensive representation of the environment could better capture the multi-body nature of interacting amino acids (e.g., in hydrogen bonding networks).

With the emergence of deep learning systems and their ability to learn patterns from high-dimensional data, it is now possible to build models that learn complex functions of protein sequence and structure, including models for protein backbone generation [55] and protein structure prediction [56]. We anticipated that a deep learning approach conditioned on local chemical environments could capture higher order interactions relevant for sequence design, while still allowing for implicit modeling of backbone flexibility.

In this study, we explored an approach for fixed-backbone sequence design that uses a neural network model as a statistical potential learned directly from crystal structure data. This potential predicts a distribution over the amino acid identity and rotamers at a given residue position, conditioned on the surrounding chemical environment. To design sequences and rotameric states for a candidate protein backbone, we iteratively sample from these learned conditional distributions at each residue position. While conventional energy functions require careful parameterization and iterative refinement, our method learns directly from data and requires no modeling or additional fitting to data apart from training the network.

Not relying on any human-specified priors, rather learning directly from crystal structure data, our method produces realistic and variable sequences with quality on par with Rosetta designs under several key metrics. We computationally validate designed sequences via structure prediction and experimentally test the generalization of our method to a *de novo* TIM-barrel scaffold, showcasing its practical utility in design applications for which there are no closely-related native structures.

Results

Approach

We train a deep 3D convolutional neural network (CNN) to predict amino acid identity given the residue's local environment, consisting of all neighboring backbone and side-chain atoms within a fixed field-of-view, masking the side-chain atoms of the residue of interest. This **conditional model** defines a distribution over amino acid types and rotamer angles at each sequence position (Figure 1). We also train a **baseline model** that has only backbone atoms as an input, as a means to initialize the backbone with a starting sequence.

Given a backbone structure X for an n-residue protein, we are interested in sampling from the true distribution of sequences Y given X,

$$P(Y|X) = p(y_1, \ldots, y_n|X)$$

We assume that the identity and conformation (rotamer) of each side-chain y_i is entirely determined by its local context, namely the identities and rotamers of its neighboring residues $y_{NB(i)}$ and backbone atoms X.

Under this assumption, the input backbone X defines a graph structure, where nodes correspond to rotamers y_i and edges exist between pairs of residues (y_i, y_j) within some threshold distance of each other. We can intepret this graph as a Markov Random Field (MRF), where each rotamer y_i is independent of all residues conditioned on those in its Markov blanket, i.e.

$$p(y_i|y_{-i}, X) = p(y_i|y_{NB(i)}, X) \triangleq p(y_i|\text{env})$$

The conditional distribution at a single residue position i can be factorized as follows:

$$\begin{aligned} p(y_i|\text{env}_i) = p(r_i|\text{env})\, p(\chi_{1i}|r_i, \text{env}_i) p(\chi_{2i}|r_i, \chi_{1i}, \text{env}_i) \\ p(\chi_{3i}|r_i, \chi_{1i}, \chi_{2i}, \text{env}_i) p(\chi_{4i}|r_i, \chi_{1i}, \chi_{2i}, \chi_{3i}, \text{env}_i) \end{aligned}$$

where r_i is the amino acid type of residue i and χ_1-χ_4 are the torsion angles for the side-chain.

If we had access to the the true conditional distributions $p(y_i|\text{env}_i)$ for each residue, by doing Gibbs sampling over the graph, we would be able draw samples as desired from the joint distribution $P(Y|X)$. Gibbs sampling is an MCMC algorithm and involves repeatedly sampling values for each variable conditioned on the assignments of all other variables [57]. We train a network to *learn* these conditional distributions from data. For a residue position i, given backbone atoms X and neighboring rotamers $y_{NB(i)}$, our trained classifier outputs $p_\theta(y_i|\text{env})$, a discrete distribution over amino acid types and rotamers where $p_\theta(y_i|\text{env}) \approx p(y_i|\text{env})$.

The network is trained to learn this conditional distribution in an autoregressive manner. Conditioning on the local environment, the network predicts a distribution over residue types $p_\theta(r_i|\text{env}_i)$. Then, both the environment feature vector and the ground-truth residue identity $\hat{r}_i$ are used to predict a discrete distribution $p_\theta(\chi_{1i}|\hat{r}_i, \text{env}_i)$. Then, the environment feature vector, the ground-truth residue identity, and the discretized ground-truth angle $\hat{\chi_1}_i$ are used to predict a discrete distribution $p_\theta(\chi_{2i}|\hat{\chi_1}_i, \hat{r}_i, \text{env}_i)$. This autoregressive unroll proceeds up to χ_{4i}.

We approximate the joint probability of a sequence $P(Y|X)$ with the pseudo-log-likelihood (PLL) [58] of the sequence under the model,

$$PLL(Y|X) = \sum_i \log p_\theta(y_i|\text{env}_i)$$

Training the neural network with cross-entropy loss involves maximizing the log probability of the correct residue class given the residue environment, and the PLL is the sum of these log probabilities. We treat the negative PLL as a heuristic *model energy* for selecting sequences and in some experiments optimize the PLL via simulated annealing.

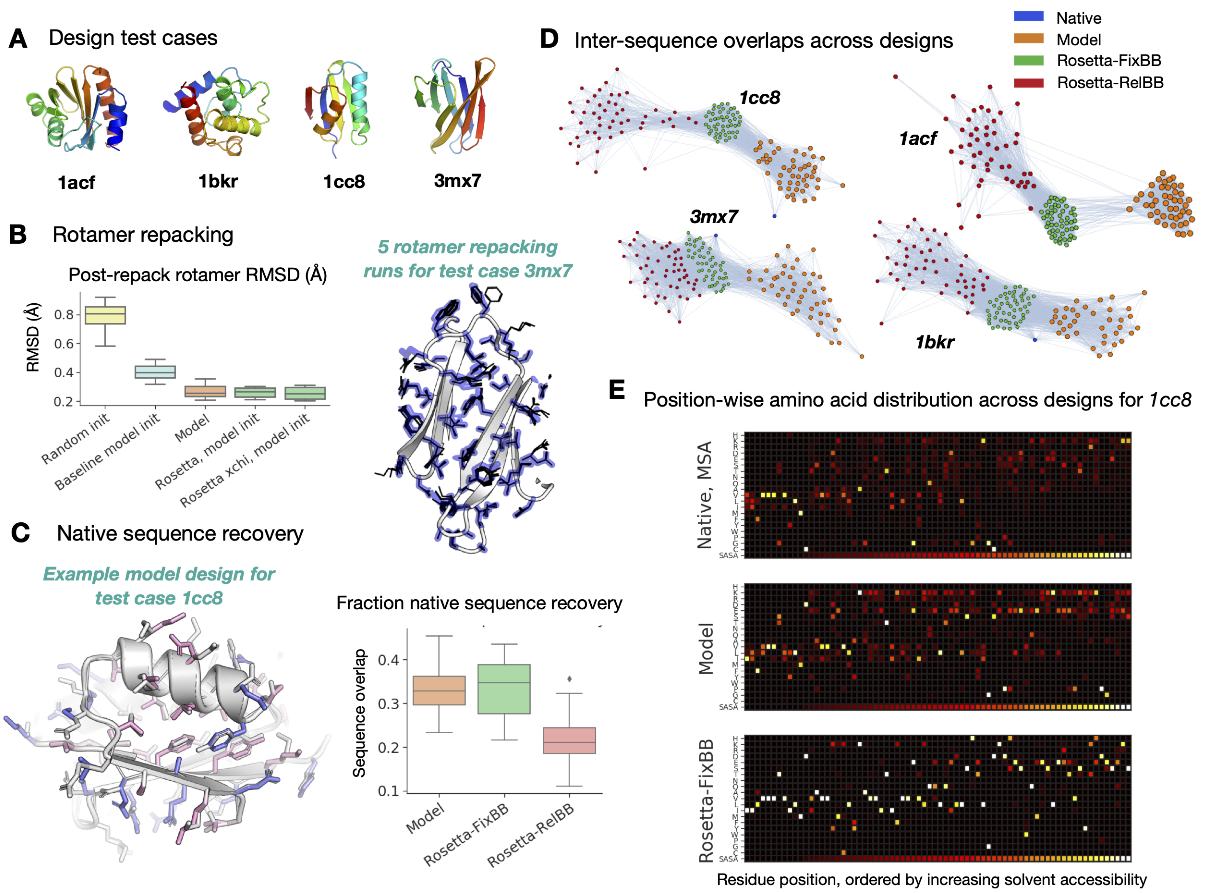

Figure 2: Model-based rotamer repacking and sequence design. A) Native test case backbones. **B-C) Post-design metrics:** B) Rotamer repacking results. (Left) Post-repack rotamer side-chain RMSD (Å) across all test cases, 5 runs each (Random initialization of rotamers, initialization with baseline model prediction from backbone atoms only, rotamer repacking with conditional model, Rosetta rotamer packing starting from baseline model initialization with and without extra χ_1, χ_2 sampling). (Right) Visualization of 5 rotamer repacking trajectories for test case *3mx7* (Black – model-designed rotamers. Blue – crystal structure side-chains). C) Native sequence recovery. (Left) Model design for test case *1cc8* (White – crystal structure. Pink/purple – model-designed non-mutations/mutations). (Right) Fraction native sequence recovery for model versus Rosetta baselines across all test cases($n = 50$). **D-E) Design variability:** D) Cluster plot representation of inter-sequence string distances across methods for test cases. Edges are between sequences with greater than 37.5% overlap with weights proportional to percent sequence overlap. Nodes are positioned using the Fruchterman-Reingold force-directed algorithm [59]. E) Position-wise amino acid distributions for test case *1cc8*. Columns are ordered by increasing solvent-accessible surface area (SASA) of native sequence residues from left to right. (Top) Native sequence and aligned homologous sequences from MSA ($n = 670$); (Middle) Model designs ($n = 50$); (Bottom) *Rosetta-FixBB* designs. ($n = 50$). MSAs obtained using PSI-BLAST v2.9, the NR50 database, BLOSUM62 with an E-value cutoff of 0.1.

Trained classifier recapitulates expected biochemical features

Our conditional model achieves a 57.3% test set accuracy for amino acid type prediction, and the baseline classifier achieves a 33.5% test set accuracy. Compared to other machine learning models for the same task, our model gives an improvement of 14.7% over [60] (42.5%), and similar performance to [61] (52.4%), [62] (56.4%) and [63] (58.0%). We note that we do not use the same train/test sets as these studies. The predictions of the network correspond well with biochemically justified substitutability of the amino acids (Fig. S1A and B). For example, the model often confuses large hydrophobic residues phenylalanine (F), tyrosine (Y), and tryptophan (W), and within this group, more often confuses Y for F which are similarly sized compared to the bulkier W. This type of learned degeneracy is necessary in order for the classifier to be useful for guiding residue selection in sequence design, as native proteins are often structurally robust to mutations at many residue positions.

Looking at residue-specific accuracies (Fig. S1C, top), on the whole, the conditional model is more certain about hydrophobic/non-polar residue prediction compared to hydrophilic/polar; this corresponds with the intuition that exposed regions in many cases allow a variety of polar residues, while buried core regions might be more constrained and therefore accommodate a limited set of hydrophobic residues. Both the conditional and baseline models do especially well at predicting glycine and proline residues, both of which are associated with distinct backbone torsion distributions (Fig. S1C).

The conditional model is trained to predict binned rotamer torsion (χ) angles in an autoregressive fashion. Across 24 bins, the hard test set accuracy of the model is 52.7%, 43.6%, 26.8%, and 30.6% for χ_1-χ_4, respectively. The hard accuracy represents accuracy to within less than 15 degrees. We also report the accuracy to within 30, 45, and 60 degrees (Fig. S1D).

The rotamer prediction module by construction is a backbone-dependent rotamer library that samples from the joint distribution $p(\chi_{1i}, \chi_{2i}, \chi_{3i}, \chi_{4i}|X, r_i)$, given backbone atoms X and residue identity r_i. We see that the network-learned factorized χ distributions match well with the empirical residue-specific rotamer distributions (Fig. S1D), which rotamer libraries typically seek to capture.

Design algorithm recovers native rotamers and sequence

We selected four native structures from the test set as test cases for evaluating our method: PDB entries *1acf*, *1bkr*, *1cc8*, and *3mx7* (Fig. 2A). We selected these test structures because they span the three major CATH [64, 65] protein structure classes (mostly alpha, alpha-beta, and mostly beta) and because their native sequences were recoverable via structure prediction with Rosetta AbInitio, ensuring they could serve as a positive control for later *in silico* folding experiments.

We assess the performance of our sequence design method by comparison with two Rosetta baselines. The first, *Rosetta-FixBB*, is a fixed-backbone Rosetta design protocol, and the second, *Rosetta-RelaxBB*, interleaves backbone relaxes between fixed-backbone design cycles, allowing the template backbone to move. The *Rosetta-FixBB* protocol provides the most relevant comparison to our method as both operate on fixed backbones. However, *Rosetta-RelaxBB* is the more commonly used mode of sequence design, generating solutions that are more variable than those of *Rosetta-FixBB*. We have therefore included designs from both methods for comparison.

We first evaluated how well the model can pack rotamers, given a fixed backbone and sequence – a benchmark normally requiring highly accurate energy functions to achieve good results (Movie S1). The model is capable of recovering rotamers with low side-chain RMSD (Å) and low l_2 χ recovery error to native (Fig. 2B and S3A and Dataset S2). Moreover, Rosetta is far more sensitive to the backbone than the model: for the test case *1acf*, Rosetta does well once the crystal structure backbone has been relaxed under the Rosetta energy function with the native sequence and rotamers in place; our model, however, is robust to the deviations in the backbone and performs similarly in both cases (Fig. S3A).

We next ran sequence and rotamer design trajectories for the native backbone test cases (Dataset S2 and Movie S2). On average, the model recapitulates the native sequence to a similar extent as *Rosetta-FixBB* and to a better extent than *Rosetta-RelBB* (Fig. 2C and S3B). In addition, model-designed sequences adhere to an amino acid distribution similar to the native sequence and its aligned homologs (Fig. S4A and SI Appendix 1A).

Model designs achieve greater sequence diversity than fixed-backbone Rosetta designs

While the *Rosetta-FixBB* designs are convergent, the model designs are more variable in terms of inter-sequence overlaps across designs (Fig. 2D and S4B).

We studied patterns of variability by position, specifically between solvent-exposed and buried regions. The model designs are more variable in solvent-exposed regions, as are sequences homologous to the native found through multiple sequence alignment (MSA); in comparison, the *Rosetta-FixBB* protocol is much more convergent (Fig. 2E and S4C). However, both the model designs and Rosetta designs converge to similar core (low solvent-accessible surface area) regions, indicating that perhaps the need for a well-packed core in combination with volume constraints determined by the backbone topology might limit the number of feasible core designs. In general, model designs have higher position-wise entropy (Figure S4D, bottom), while still retaining correct patterns of hydrophobic/non-polar and hydrophilic/polar residue positions at buried and solvent-exposed positions, respectively.

Model designs are comparable to Rosetta designs under a range of biochemically relevant metrics

We next assessed the validity of designed sequences en masse using metrics that are agnostic to any energy function.

Sequences designed onto the test case backbones should ideally have the following key biochemical properties [66, 67]: (1) core regions should be tightly packed with hydrophobic residues, a feature that is important for driving protein folding and maintaining fold stability [68, 69, 70]; (2) designs should have few exposed hydrophobic residues, as these

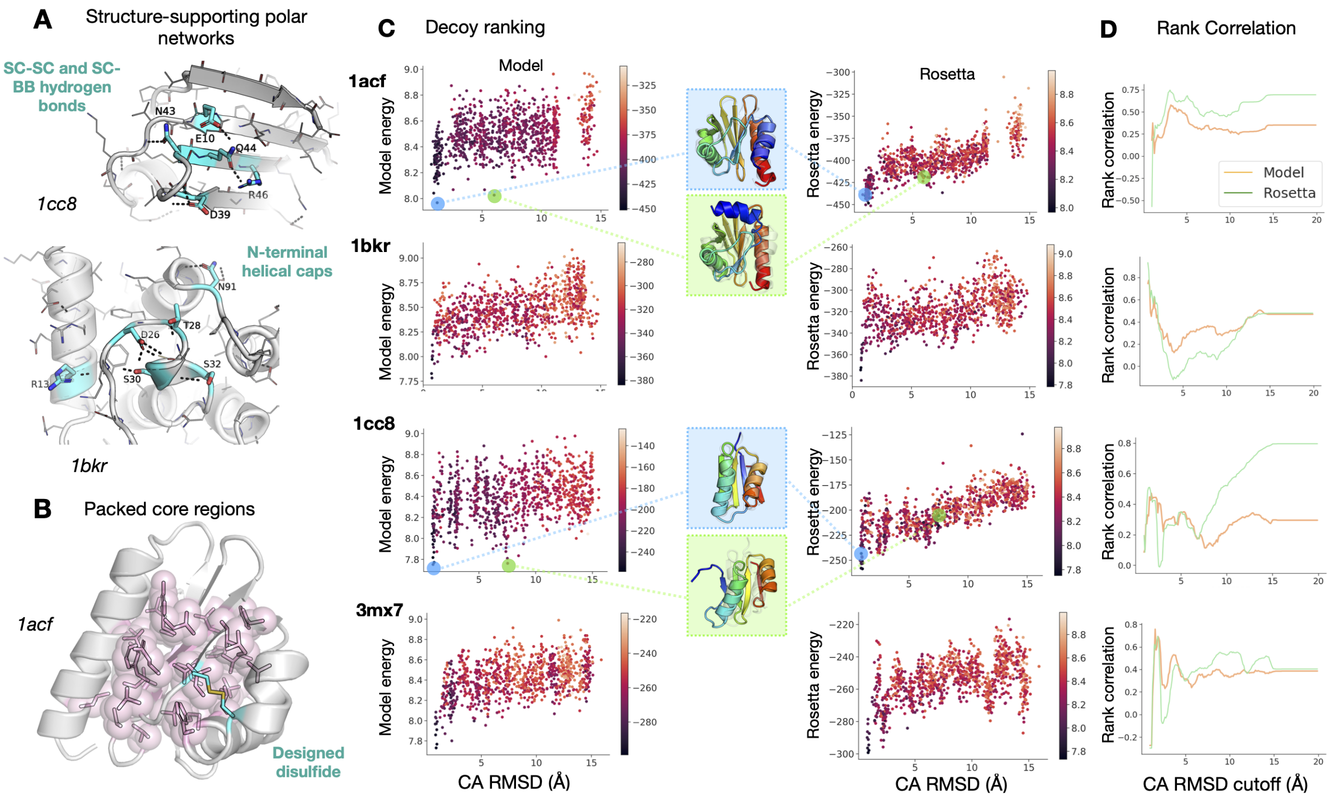

Figure 3: Model captures salient design features and general sequence-structure relationship. A-B) Designed structural features of interest. A) Structure supporting polar networks, inter-side-chain (SC-SC) and side-chain–backbone (SC-BB) contacts, and N-terminal helical caps. B) Well-packed hydrophobic core regions and model-designed disulfide. C-D) Decoy ranking for test cases. C) Decoy ranking with model and Rosetta energies. (Left) Model energy (negative PLL) vs. alpha-carbon RMSD (Å) for Rosetta AbInitio folded structures. Points are colored by structure Rosetta energy. (Right) Rosetta energy (negative PLL) vs. alpha carbon RMSD (Å) for folded structures. Points are colored by structure model energy. (Inset) Select structures rendered to visualize alternative minima under model ranking for *1acf* and *1cc8*. D) Spearman rank correlation between model/Rosetta energies and structure alpha-carbon RMSD (Å) as a function of increasing RMSD cutoff.

incur an entropic penalty as a result of solvent ordering under the hydrophobic effect and an enthalpic penalty from disrupting hydrogen bonding between polar solvent molecules [71, 72, 73], making it energetically favorable to place polar residues at solvent-accessible positions and apolar residues at buried positions; (3) If polar residues are designed in core regions, they should be supported by hydrogen bonding networks [74, 23].

Model designs tend to be well-packed, with core packing overall improving after relaxing the backbone under the Rosetta energy function (Fig. S5A,B). In general, the model-designed sequences do not have exposed hydrophobics in solvent-exposed positions, similar to the Rosetta designs (Fig. S5C). Finally, the model designs match the native structure in terms of numbers of side-chain and backbone buried unsatisfied hydrogen bond acceptors and donors (Fig. S5D-F); this indicates that over the course of model design, side-chains that are placed in buried regions are adequately supported by backbone hydrogen bonds or by design of other side-chains that support the buried residue (SI Appendix 1B).

To confirm that key sequence features are retained despite the increased variability seen, we calculated Psipred [75, 76, 77, 78, 79, 80] secondary structure prediction cross-entropy relative to DSSP assignments [81]. We found that the predictions for model-designed sequences from single sequences are comparable in accuracy to those for the native sequence and Rosetta designs (Fig. S6A), indicating that despite variation, the designed sequences retain local residue patterns that allow for accurate backbone secondary structure prediction.

A key structural feature is the placement of N-terminal helix capping residues—typically there will be an aspartic acid (D), asparagine (N), serine (S), or threonine (T) preceding the helix starting residue, as these amino acids can hydrogen bond to the second backbone nitrogen in the helix [82, 83]. The majority of both the model designs and the Rosetta designs successfully place these key residues at the native backbone capping positions (Fig. S6B).

By inspection, we also see a number of specific notable features across the test cases. These include placement of glycines at positive ϕ backbone positions (Fig. S6C), placement of prolines at cis-peptide positions (Fig. S6D), polar networks supporting loops and anchoring secondary structure elements (Fig. 3A), and well-packed core regions (Fig. 3B).

Designed sequences are corroborated by blind structure prediction

To determine whether the model-designed sequences could adopt structures with low Rosetta energy and low root-mean-square (RMS) deviation from the starting backbone, we performed blind structure prediction using the AbInitio application in Rosetta [84, 85, 86].

Top sequences for validation were selected based on quantitative design metrics (see *Methods*), and for each test case we ran AbInitio folding on the selected model design in addition to the best Rosetta designs and a 50% perturbed native sequence baseline under our selection criteria (Dataset S3).

The model designs had generally low sequence identity to the native sequence ($\leq 43.1\%$), but were able to closely recover the native backbones (Fig. S8). All of the model designs achieved significantly better recovery than the 50% randomly perturbed cases. These results suggest that (1) close recovery of the native backbone is due to features learned by the model and not simply due to sequence overlap with the native, and (2) that our method is able to generate solutions not accessible via Rosetta design, that differ from native sequences, and yet fold under the Rosetta energy function.

Model captures sequence-structure relationship

We next asked whether the model has learned to be robust to a larger range of input backbones beyond what is seen in the train set, as well as whether the model PLL can be used to find high likelihood backbones given a target sequence. In essence, we query the extent to which the classifier can capture the likelihood of *structure* given *sequence*.

We evaluated the model energy (negative PLL) and the Rosetta energy for a range of structures (decoys) folded with the Rosetta AbInitio structure prediction protocol (Fig. 3C). Although we do not expect the model to generalize to highly out-of-distribution backbones (for example, distended or unfolded backbones), we see that the model energy is low for the decoys with lowest RMS to the native backbone, and that the energy vs. RMS plots have a “funnel”-like shape. This indicates that for low RMS structures the model is robust to deviations in the input backbone and can be used to select structures closer to the native backbone. There are cases where the model assigns high PLL to high RMS backbones (Fig. 3C, inset); for example, for *1acf* an alternative N-terminal helix conformation and for *1cc8* an alternative pattern of beta strand pairing are assigned high PLLs.

Quantitatively, we see that in the lower RMS regime ($\sim< 10$Å from the native), the relationships between the model or Rosetta energies and the structure RMS are similarly monotonic as measured by the Spearman rank correlation (Fig. 3D), indicating that the model energy and Rosetta energy can rank order structures by RMS with similar accuracy.

Unlike an analytical energy function, the model energy is not expected to generalize to structures far from the training distribution. However, these results suggest that the model has to an extent learned features that capture sequence-structure energetics, and in particular are useful for identifying and ranking low RMS decoys.

Application – Model-based sequence design of a *de novo* TIM-barrel

Design of *de novo* proteins remains a challenging task as it requires robustness to backbones that lie near, but outside the distribution of known structures. Successful *de novo* protein designs often lack homology to any known sequences despite the fact that *de novo* structures can qualitatively resemble known folds [9, 7, 88]. For a design protocol to perform well on *de novo* backbones it must therefore avoid simple recapitulation of homologous sequences.

To assess whether our model could perform sequence design for *de novo* structures, we tested our method on a Rosetta-generated four-fold symmetric *de novo* TIM-barrel backbone.

The TIM-barrel backbone is a circularly permuted variant of the reported design *5bvl* [7], which was excluded from the training set. While we did not exclude native TIM-barrels from the training set, *5bvl* is distant in sequence and structure from any known protein, and contains local structural features that differ significantly from naturally occurring TIM-barrels [7], making it unlikely that the model would return memorized sequences found in the training set as solutions. The original design *5bvl* was designed using a combination of Rosetta protocols and manual specification of residues.

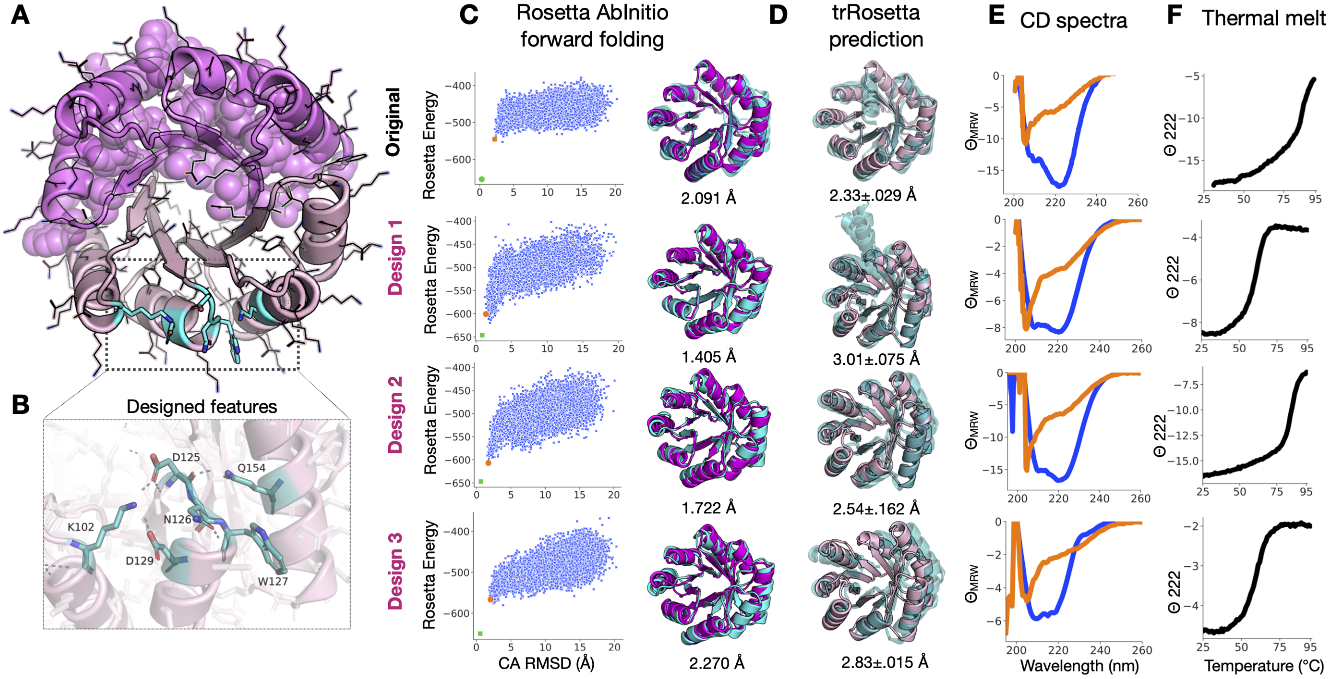

Figure 4: Model-guided sequence design of four-fold symmetric *de novo* TIM-barrels. A) Four-fold symmetric TIM-barrel design (Model design 2). Hydrophobic core packing shown for half of the protein (pink). B) Designed polar networks, as well as N-terminal helical caps and helix space-setting tryptophan placement. C) Rosetta AbInitio structure prediction for original TIM-barrel design [7] (top) and three model designs (bottom). (Left) Rosetta energy versus alpha-carbon RMSD (Å) to template for folded structures. Selected structure with best summed rank of template-RMSD and Rosetta energy is shown in orange. The design after RosettaRelax is shown in green. (Right) Selected decoy (blue) aligned to native backbone (pink). D) trRosetta recovered structures [87] for original and model designs. 5 structures shown (teal) aligned to native backbone (light pink) E) Far UV circular dichroism (CD) spectra taken at 20°C (blue) and 95 °C (orange). Raw data is normalized to show mean residue ellipticity Θ_{MRW} (deg cm^2 dmol^{-1}). F) Thermal melting curves showing normalized CD signal Θ_{MRW} at 222 nm.

We generated four-fold symmetric sequences for this template by sampling without annealing and using Rosetta for rotamer repacking in place of the model rotamer predictions. We found that designs exhibited dense hydrophobic packing (Fig. 4A), and had hydrogen-bonding networks and correctly placed helical capping residues (Fig. 4B).

We selected 8 four-fold symmetric designs to test experimentally (Dataset S4). These structures were predicted to fold with low deviation into the desired structure by Rosetta AbInitio structure prediction (Fig. 4C), as well as by the newer deep-learning based trRosetta method [87] (Fig. 4D). Of these 8 designs, 3 are cooperatively folded proteins, as indicated by circular dichroism (CD) wavelength scans (Fig. 4E) and by the presence of clear two-state transitions in thermal melt curves (Fig. 4F).

Overall, these results indicate that our method can produce folded designs even for protein backbones outside the space of known native structures.

Discussion

In this study, we demonstrate that a design algorithm guided by an entirely learned neural network potential can give a distribution of possible sequences for a fixed backbone. Our method is able to design sequences which are comparable to those designed by Rosetta protocols across a range of key biochemical metrics, and we show how our design method generalizes to *de novo* backbones with experimental evidence supporting designed sequences folding to stable proteins. Remarkably, the our design algorithm captures key behaviors of established design methods guided by energy functions, such as the ability to (1) accurately determine side-chain conformations, (2) differentiate the hydrophobic interior and polar exterior of the proteins, (3) design multi-body interactions as reflected by readily produced hydrogen-bonding networks, and (4) discriminate low RMS structures in a folding simulation trajectory.

Ours is the first deep learning method to tackle both sequence and rotamer design. Previous machine learning models for the task of residue prediction conditioned on chemical environment have only been applied to single-shot residue prediction or architecture class, secondary structure, or $\Delta\Delta G$ prediction [60, 89, 62, 63, 61]. Other deep learning approaches have been developed for sequence design or rotamer packing [90, 91, 92, 93, 94, 95], but most of these methods' designs have not been comprehensively validated by a range of biochemical metrics or by folding *in silico* or *in vitro*.

The primary limitations to using a deep neural network to guide design in this manner are the possibility of artifacts due to the model learning non-robust features from the data and the inability of the network to generalize to backbones far from the training distribution. We show that the model does in fact produce biochemically justifiable designs, and moreover generalizes to out-of-distribution backbones (Fig. 3C-F). Overall, our results show that a neural network can learn necessary patterns to match state-of-the-art energy functions for guiding protein design, even without any hand-engineered features. Our method is also flexible—the design protocol easily allows for adding position-specific constraints during design, and other neural network models could be used in place of the classifier network presented without fundamentally changing the method. Though we evaluated our method against Rosetta in this paper, in practice our method could be combined with Rosetta in a natural way. Rosetta design protocols run MCMC with a uniform proposal distribution over amino acid types and rotamers. Instead, the distribution provided by the classifier could be used as a proposal distribution for Rosetta design protocols, potentially speeding up convergence. The model energy, along with other metrics, could also be used to guide early stopping to avoid over-optimization of the Rosetta energy function.

Although we have shown in this paper how sampling with a trained deep neural network classifier can be used for sequence design, the framework we have outlined has many possible applications beyond fixed-backbone sequence design. We anticipate that the procedure we have established could be used to guide the design of interfaces, protein-nucleic acid complexes, and ligand binding sites. Because our model can design networks of polar interactions, we believe the method could potentially be extended to model water-mediated contacts during design.

Our results demonstrate both the practical applicability of an entirely learned method for protein design and also the capability of deep learning models to learn sequence-structure features, heretofore best captured by heuristic energy functions. We believe this study demonstrates the potential for machine learning methods to transform current methods in structure-based protein design.

Methods

Dataset S5 available for download at `https://drive.google.com/file/d/1yBQ42M4yRWlb4G6vF1QS8__IUfv5eJul/view?usp=sharing`

Code available at `https://github.com/nanand2/protein_seq_des`

Design algorithm

Given a candidate protein backbone for which a corresponding sequence is desired, we initialize a sequence and then iteratively sample amino acid identities and rotameric states from the conditional probability distribution defined by the classifier network in order to design sequences.

Initialization can be done arbitrarily. In this study, we initialize the sequence and the rotamers by sampling from the baseline model predicted distributions. The baseline model is only conditioned on backbone atoms.

Given a residue position i and a local environment env_i around that residue with either just backbone atoms (baseline model) or other residue side-chains as well (conditional model), the sampling procedure is as follows. First, sample a residue type $\hat{r_i}$ conditioned on the environment.

$$\hat{r_i} \sim p_\theta(r_i|\text{env}_i)$$

Then, conditioned on the environment and the sampled amino acid type, sample rotamer angles for that residue type in an autoregressive manner:

$$\hat{\chi_1}_i \sim p_\theta(\chi_{1i}|\hat{r}_i, \text{env}_i)$$
$$\hat{\chi_2}_i \sim p_\theta(\chi_{2i}|\hat{r}_i, \hat{\chi_1}_i, \text{env}_i)$$

$$\hat{\chi_3}_i \sim p_\theta(\chi_{3i}|\hat{r}_i, \hat{\chi_1}_i, \hat{\chi_2}_i, \text{env}_i)$$

$$\hat{\chi_4}_i \sim p_\theta(\chi_{4i}|\hat{r}_i, \hat{\chi_1}_i, \hat{\chi_2}_i, \hat{\chi_3}_i, \text{env}_i)$$

After a discrete rotamer bin has been sampled, rotamer angles are sampled uniformly within the selected bin.

As residues and rotamers are sampled at different positions along the protein backbone, we monitor the negative pseudo-log-likelihood of the sequence as a heuristic model energy. Note that most residues have fewer than four χ angles. At sampling time, for each residue only the corresponding χ angles are sampled. For residues which do not have a particular χ_j, an average value for the log probability of χ_j under the conidtional model across the train set and across the rotamer bins is used instead in the PLL calculation.

In this study we use a Gibbs sampling procedure, where we iteratively sample residue types and rotamer states from the learned conditional distributions. Although the learned distribution $p_\theta(y_i|\text{env})$ is only an approximation of the true conditional distribution, if the learned conditionals were to match the true data conditionals, then a Gibbs sampler run with the learned conditionals would have the same stationary distribution as a Gibbs sampler run with the true data conditionals [96]. In some cases, we do simulated annealing to optimize the PLL.

In order to speed up convergence, we do blocked sampling of residues. In practice, we draw edges in the graph between nodes where corresponding residues have C_β atoms that are less than 20 Å apart, guaranteeing that non-neighboring nodes correspond to residues that do not appear in each other's local environments. During sampling, we use greedy graph coloring to generate blocks of independent residues. We then sample over all residues in a block in parallel, repeating the graph coloring every several iterations.

The model is restricted from designing glycines at non-loop positions, based on DSSP assignment [97, 81].

Rotamer repacking. Five rounds of rotamer repacking were done on each of the four test case backbones. Repacking is done by fixing the native sequence and randomizing starting rotamers, or using baseline model predictions on backbone atoms only to initialize rotamers. Rotamer prediction at each step and each residue position conditions on the true native residue identity. Model negative PLL averaged by protein length was annealed for 2500 iterations with starting temperature 1 and annealing multiplicative factor 0.995. Rotamer repacking was evaluated under two metrics: the χ angle l_2 distance between the unit vectors corresponding to the rotamer angles, and residue side-chain RMSD (Å).

Native test case sequence design. Fifty rounds of sequence and rotamer design were done on each of the four test case backbones. Model negative PLL averaged by protein length was annealed for 2500 iterations with starting temperature 1 and annealing multiplicative factor 0.995. Top sequences for structure prediction were filtered by the following criteria: all helices capped at the N-terminal, packstat post-RosettaRelax ≥ 0.65, and alpha-carbon RMSD ≤ 1.2 Å. After filtering, sequences were selected by lowest model energy post-relax, or lowest Rosetta energy for the baselines and controls.

TIM-barrel design. Six rounds of residue sampling were run for 25K iterations each. Design was done on a TIM-barrel template prepared using RosettaRemodel [98, 7]. Rotamers were packed at each iteration using PyRosetta with a pack radius of 5 Å and extra χ_1, χ_2 sampling [99]. Residue and rotamer symmetry is enforced by averaging predicted logits across symmetric positions before normalizing to a discrete distribution and sampling. Cysteines and methionines were restricted during design. Top sequences were selected by finding local minima of the model residue negative pseudo-log-likelihood (model energy) over the course of sampling, specifying that the local minima must be below a 5% percentile cutoff of model energy and at least 50 steps apart. Then, the final designs for experimental validation were selected based on model energy and packstat score and by looking at qualitative design features such as core packing and polar network formation.

Runtime. The runtime for our method for sequence design is determined primarily by two steps: (1) sampling residues and rotamer angles and (2) computing the model energy (negative PLL). These times are determined by the speed of the forward pass of the neural network, which is a function of the batch size, the network architecture, and the GPU itself. Note that to compute the model energy, a forward pass of the network is done at each residue position where the environment around that residue has changed.

Annealing for 2500 steps takes between 1 to 2 hours for the native test cases on a computer with 32 GB RAM and on a single GeForce GTX TITAN X GPU, with other processes running on the machine (Dataset S2). *Rosetta-RelaxBB* takes 20-30 minutes per design, while *Rosetta-FixBB* takes 5-15 minutes per design. Compressing the network or modifying the architecture and parallelizing the sampling procedure across more GPUs would improve the overall runtime. In addition, a faster annealing schedule and early stopping of optimization would also reduce the runtime.

Data

To train our classifier, we used X-ray crystal structure data from the Protein Data Bank (PDB) [100], specifically training on CATH 4.2 S95 domains [64, 65]. We separated domains into train and test sets based on CATH topology classes, splitting classes into $\sim 95\%$ and 5%, respectively (1374 and 78 classes, 53414 and 4372 domains each, see Dataset S1). This ensured that sequence and structural redundancy between the datasets was largely eliminated. We first applied a resolution cutoff of 3.0 Å and eliminated NMR structures from the dataset. During training, we did not excise domains from their respective chains but instead retained the complete context around a domain. When a biological assembly was listed for a structure, we trained on the first provided assembly. This was so that we trained primarily on what are believed to be functional forms of the protein macromolecules, including in some cases hydrophobic protein-protein interfaces that would otherwise appear solvent-exposed.

The input data to our classifier is a $20 \times 20 \times 20$ $\mathring{A}^3$ box centered on the target residue, and the environment around the residue is discretized into voxels of volume 1 $\mathring{A}^3$. We keep all backbone atoms, including the C_α atom of the target residue, and eliminate the C_β atom of the target residue along with all of its other side-chain atoms. We center the box at an approximate C_β position rather than the true C_β position, based on the average offset between the C_α and C_β positions across the training data. For ease of data loading, we only render the closest 400 atoms to the center of the box.

We omit all hydrogen atoms and water molecules, as well as an array of small molecules and ions that are common in crystal structures and/or possible artifacts (Dataset S1). We train on nitrogen (N), carbon (C), oxygen (O), sulfur (S), and phosphorus (P) atoms only. Ligands are included, except those that contain atoms other than N, C, O, S, and P. Bound DNA and RNA are also included. Rarer selenomethionine residues are encoded as methionine residues. For the baseline model, we omit all side-chain atoms while training, so that the model conditions only on backbone atoms. For the conditional model, the input channels include: atom type (N, C, O, S, or P), indicator of backbone (1) or side-chain (0) atom, and one-hot encoded residue type (masked for backbone atoms for the center residue). For the baseline model, the input channels only encode atom type, since all atoms are backbone atoms and we assume no side-chain information is known.

We canonicalize each input residue environment in order to maximize invariance to rotation and translation of the atomic coordinates. For each target residue, we align the N-terminal backbone $N - C_\alpha$ bond to the x-axis. We then rotate the structure so that the normal of the $N - C_\alpha - C$ plane points in the direction of the positive z-axis. Finally, we center the structure at the effective C_β position. By using this strategy, we not only orient the side-chain atoms relative to the backbone in a consistent manner (in the positive z-direction), but also fix the rotation about the z-axis. We then discretize each input environment and one-hot encode the input by atom type.

Model training

Our model is a fully convolutional neural network, with 3D convolution layers followed by batch normalization [101] and LeakyReLU activation. We regularize with dropout layers with dropout probability of 10% and with L2 regularization with weight 5×10^{-6}. We train our model using the PyTorch framework, with orthogonal weight initialization [102]. We train with a batch size of 2048 parallelized synchronously across eight NVIDIA v100 GPUs. The momentum of our BatchNorm exponential moving average calculation is set to 0.99. We train the model using the Adam optimizer ($\beta_1 = 0.5$, $\beta_2 = 0.999$) with learning rate 7.5×10^{-5} [103] and L2 weight regularization 5×10^{-6}. We use the same model architecture and optimization parameters for both the baseline and conditional models.

Our final conditional classifier is an ensemble of four models corresponding to four concurrent checkpoints. Predictions are made by averaging the logits (unnormalized outputs) from each of the four networks.

Design baselines

The *Rosetta-FixBB* baseline uses the Rosetta packer [39], invoked via the *RosettaRemodel* [98] application, to perform sequence design on fixed backbones. This design protocol performs Monte Carlo optimization of the Rosetta energy function over the space of amino acid types and rotamers [39]. Between each design round, side-chains are repacked, while backbone torsions are kept fixed. Importantly, the Rosetta design protocol samples uniformly over residue identities and rotamers, while our method instead samples from a learned conditional distribution over residue identities. The *Rosetta-RelaxBB* protocol is highly similar to the *Rosetta-FixBB* protocol but performs energy minimization of the template backbone in addition to repacking between design cycles, allowing the template backbone to move. Starting templates for both baselines have all residues mutated to alanine, which helps eliminate early rejection of sampled residues due to clashes. The REF2015 Rosetta energy function was used for all experiments [104, 105].

Metrics

To assess biochemical properties of interest for the designed sequences we use the following three metrics: (1) packstat, a non-deterministic measure of tight core residue packing [106], (2) exposed hydrophobics, which calculates the solvent-accessible surface area (SASA) of hydrophobic residues [73], and (3) counts of buried unsatisfied backbone (BB) and side-chain (SC) atoms, which are the number of hydrogen bond donor and acceptor atoms on the backbone and side-chains, respectively, that are not supported by a hydrogen bond. We use PyRosetta implementations of these metrics. Backbone relaxes for designs were done with the RosettaRelax application [107, 108, 109, 110], with the ex1 and ex2 options for extra χ_1 and χ_2 rotamer sampling.

Rosetta AbInitio structure prediction

Rosetta AbInitio uses secondary structure probabilities obtained from Psipred [75, 76, 77, 78] to generate a set of candidate backbone fragments at each amino acid position in a protein. These fragments are sampled via the Metropolis-Hastings algorithm to construct realistic candidate structures (decoys) by minimizing Rosetta energy. Native test case runs used Psipred predictions from MSA features after UniRef90 [80] database alignment. TIM-barrel design runs used Psipred predictions directly from sequence. All designs were selected without using any external heuristics, manual filtering, or manual reassignments. We obtained Psipred predictions, picked 200 fragments per residue position [86], and ran 10^4 trajectories per design. Folding trajectories in Fig. 3 were seeded with native fragments.

Protein Purification

Proteins were produced as fusions to a N-terminal 6xHis tag followed by a Tobacco Etch Virus (TEV) cleavage sequence. Expression was performed in E. coli BL21(DE3) using the pET24a expression vector under an IPTG inducible promoter. Proteins were purified with Ni-Sepharose columns (GE Healthcare). Purity and monomeric state were confirmed using a Superdex 75 size-exclusion column (GE Healthcare) and SDS-PAGE gels. Protein concentration was using predicted extinction coefficient and 280 nm absorbance measured on a NanoDrop spectrometer (Thermo Scientific).

Circular Dichroism Spectroscopy

Circular Dichroism spectra were collected using a Jasco 815 spectropolarimeter with measurements taken in Phosphate Buffered Saline (PBS) using a 1.0mm pathlength cuvette. Wavelength scans were collected and averaged over 3 accumulations. Melting curves were collected monitoring CD signal at 222nm over a range of 25°C to 90°C at 1°C intervals, 1 minute equilibration time and 10 second integration time. Spectra are normalized to mean residue ellipticity (deg cm^2 dmol^{-1}) from millidegrees.

Contributions

N.A. and P.-S.H. conceived research. N.A. devised the model training strategy and design algorithm, built the codebase, trained models, ran the experiments, and analyzed the results. R.R.E. helped run Rosetta baselines, Psipred, and AbInitio experiments. A.D. helped run initial model training and AbInitio experiments. N.A. and R.R.E expressed and purified proteins for validation. R.R.E and A.D. contributed to discussion and analysis. P.-S.H. and R.B.A. supervised the research. All authors contributed to the writing and editing of the manuscript.

Acknowledgements

We thank Tudor Achim for detailed discussion and feedback. We thank Wen Torng, Sergey Ovchinnikov, Alex Chu, and Christian Choe for helpful discussion, and Frank DiMaio for providing a set of decoys from which we selected test case structures. We thank Rohan Kannan for help with running and analyzing rotamer repacking experiments. This project was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Scientific Discovery through Advanced Computing (SciDAC) program. Additionally, cloud computing credits were provided by Google Cloud. We acknowledge support from NIH GM102365, GM61374, and the Chan Zuckerberg Biohub. R.R.E acknowledges support from the Stanford ChEM-H Chemistry/Biology Interface Predoctoral Training Program and the National Institute of General Medical Sciences of the National Institutes of Health under Award Number T32GM120007. A.D. acknowledges support from the National Library of Medicine BD2K training grant LM012409.

References

[1] Brian Kuhlman, Gautam Dantas, Gregory C. Ireton, Gabriele Varani, Barry L. Stoddard, and David Baker. Design of a novel globular protein fold with atomic-level accuracy. Science, 302(5649):1364–1368, 2003.

[2] TJ Brunette, Fabio Parmeggiani, Po-Ssu Huang, Gira Bhabha, Damian C Ekiert, Susan E Tsutakawa, Greg L Hura, John A Tainer, and David Baker. Exploring the repeat protein universe through computational protein design. Nature, 528(7583):580, 2015.

[3] Lindsey Doyle, Jazmine Hallinan, Jill Bolduc, Fabio Parmeggiani, David Baker, Barry L Stoddard, and Philip Bradley. Rational design of α-helical tandem repeat proteins with closed architectures. Nature, 528(7583):585, 2015.

[4] Brian Koepnick, Jeff Flatten, Tamir Husain, Alex Ford, Daniel-Adriano Silva, Matthew J Bick, Aaron Bauer, Gaohua Liu, Yojiro Ishida, Alexander Boykov, et al. De novo protein design by citizen scientists. Nature, page 1, 2019.

[5] Gaurav Bhardwaj, Vikram Khipple Mulligan, Christopher D Bahl, Jason M Gilmore, Peta J Harvey, Olivier Cheneval, Garry W Buchko, Surya VSRK Pulavarti, Quentin Kaas, Alexander Eletsky, et al. Accurate de novo design of hyperstable constrained peptides. Nature, 538(7625):329, 2016.

[6] TM Jacobs, B Williams, T Williams, X Xu, A Eletsky, JF Federizon, T Szyperski, and B Kuhlman. Design of structurally distinct proteins using strategies inspired by evolution. Science, 352(6286):687–690, 2016.

[7] Po-Ssu Huang, Kaspar Feldmeier, Fabio Parmeggiani, D Alejandro Fernandez Velasco, Birte Höcker, and David Baker. De novo design of a four-fold symmetric tim-barrel protein with atomic-level accuracy. Nature chemical biology, 12(1):29, 2016.

[8] Fabio Parmeggiani, Po-Ssu Huang, Sergey Vorobiev, Rong Xiao, Keunwan Park, Silvia Caprari, Min Su, Jayaraman Seetharaman, Lei Mao, Haleema Janjua, et al. A general computational approach for repeat protein design. Journal of molecular biology, 427(2):563–575, 2015.

[9] Jiayi Dou, Anastassia A. Vorobieva, William Sheffler, Lindsey A. Doyle, Hahnbeom Park, Matthew J. Bick, Binchen Mao, Glenna W. Foight, Min Yen Lee, Lauren A. Gagnon, Lauren Carter, Banumathi Sankaran, Sergey Ovchinnikov, Enrique Marcos, Po-Ssu Huang, Joshua C. Vaughan, Barry L. Stoddard, and David Baker. De novo design of a fluorescence-activating b-barrel. Nature, 561(7724):485–491, 2018.

[10] Yu-Ru Lin, Nobuyasu Koga, Rie Tatsumi-Koga, Gaohua Liu, Amanda F Clouser, Gaetano T Montelione, and David Baker. Control over overall shape and size in de novo designed proteins. Proceedings of the National Academy of Sciences, 112(40):E5478–E5485, 2015.

[11] Enrique Marcos, Benjamin Basanta, Tamuka M Chidyausiku, Yuefeng Tang, Gustav Oberdorfer, Gaohua Liu, GVT Swapna, Rongjin Guan, Daniel-Adriano Silva, Jiayi Dou, et al. Principles for designing proteins with cavities formed by curved β sheets. Science, 355(6321):201–206, 2017.

[12] Po-Ssu Huang, Scott E Boyken, and David Baker. The coming of age of de novo protein design. Nature, 537(7620):320, 2016.

[13] Timothy A Whitehead, Aaron Chevalier, Yifan Song, Cyrille Dreyfus, Sarel J Fleishman, Cecilia De Mattos, Chris A Myers, Hetunandan Kamisetty, Patrick Blair, Ian A Wilson, et al. Optimization of affinity, specificity and function of designed influenza inhibitors using deep sequencing. Nature biotechnology, 30(6):543, 2012.

[14] Daniel-Adriano Silva, Shawn Yu, Umut Y Ulge, Jamie B Spangler, Kevin M Jude, Carlos Labão-Almeida, Lestat R Ali, Alfredo Quijano-Rubio, Mikel Ruterbusch, Isabel Leung, et al. De novo design of potent and selective mimics of il-2 and il-15. Nature, 565(7738):186, 2019.

[15] Bruno E Correia, John T Bates, Rebecca J Loomis, Gretchen Baneyx, Chris Carrico, Joseph G Jardine, Peter Rupert, Colin Correnti, Oleksandr Kalyuzhniy, Vinayak Vittal, et al. Proof of principle for epitope-focused vaccine design. Nature, 507(7491):201, 2014.

[16] Christine E Tinberg, Sagar D Khare, Jiayi Dou, Lindsey Doyle, Jorgen W Nelson, Alberto Schena, Wojciech Jankowski, Charalampos G Kalodimos, Kai Johnsson, Barry L Stoddard, et al. Computational design of ligand-binding proteins with high affinity and selectivity. Nature, 501(7466):212, 2013.

[17] Anum A Glasgow, Yao-Ming Huang, Daniel J Mandell, Michael Thompson, Ryan Ritterson, Amanda L Loshbaugh, Jenna Pellegrino, Cody Krivacic, Roland A Pache, Kyle A Barlow, et al. Computational design of a modular protein sense/response system. bioRxiv, page 648485, 2019.

[18] Matthew J Bick, Per J Greisen, Kevin J Morey, Mauricio S Antunes, David La, Banumathi Sankaran, Luc Reymond, Kai Johnsson, June I Medford, and David Baker. Computational design of environmental sensors for the potent opioid fentanyl. Elife, 6:e28909, 2017.

[19] Daniela Röthlisberger, Olga Khersonsky, Andrew M Wollacott, Lin Jiang, Jason DeChancie, Jamie Betker, Jasmine L Gallaher, Eric A Althoff, Alexandre Zanghellini, Orly Dym, et al. Kemp elimination catalysts by computational enzyme design. Nature, 453(7192):190, 2008.

[20] Lin Jiang, Eric A Althoff, Fernando R Clemente, Lindsey Doyle, Daniela Röthlisberger, Alexandre Zanghellini, Jasmine L Gallaher, Jamie L Betker, Fujie Tanaka, Carlos F Barbas, et al. De novo computational design of retro-aldol enzymes. science, 319(5868):1387–1391, 2008.

[21] Daniel N Bolon and Stephen L Mayo. Enzyme-like proteins by computational design. Proceedings of the National Academy of Sciences, 98(25):14274–14279, 2001.

[22] Justin B Siegel, Alexandre Zanghellini, Helena M Lovick, Gert Kiss, Abigail R Lambert, Jennifer L St Clair, Jasmine L Gallaher, Donald Hilvert, Michael H Gelb, Barry L Stoddard, et al. Computational design of an enzyme catalyst for a stereoselective bimolecular diels-alder reaction. Science, 329(5989):309–313, 2010.

[23] Scott E Boyken, Zibo Chen, Benjamin Groves, Robert A Langan, Gustav Oberdorfer, Alex Ford, Jason M Gilmore, Chunfu Xu, Frank DiMaio, Jose Henrique Pereira, et al. De novo design of protein homo-oligomers with modular hydrogen-bond network–mediated specificity. Science, 352(6286):680–687, 2016.

[24] Nathan H Joh, Tuo Wang, Manasi P Bhate, Rudresh Acharya, Yibing Wu, Michael Grabe, Mei Hong, Gevorg Grigoryan, and William F DeGrado. De novo design of a transmembrane zn2+-transporting four-helix bundle. Science, 346(6216):1520–1524, 2014.

[25] Gevorg Grigoryan, Yong Ho Kim, Rudresh Acharya, Kevin Axelrod, Rishabh M Jain, Lauren Willis, Marija Drndic, James M Kikkawa, and William F DeGrado. Computational design of virus-like protein assemblies on carbon nanotube surfaces. Science, 332(6033):1071–1076, 2011.

[26] Sandra M Malakauskas and Stephen L Mayo. Design, structure and stability of a hyperthermophilic protein variant. Nature structural biology, 5(6):470, 1998.

[27] Grant S Murphy, Jeffrey L Mills, Michael J Miley, Mischa Machius, Thomas Szyperski, and Brian Kuhlman. Increasing sequence diversity with flexible backbone protein design: the complete redesign of a protein hydrophobic core. Structure, 20(6):1086–1096, 2012.

[28] Johan Desmet, Marc De Maeyer, Bart Hazes, and Ignace Lasters. The dead-end elimination theorem and its use in protein side-chain positioning. Nature, 356(6369):539, 1992.

[29] Bassil I Dahiyat and Stephen L Mayo. De novo protein design: fully automated sequence selection. Science, 278(5335):82–87, 1997.

[30] Johan Desmet, Jan Spriet, and Ignace Lasters. Fast and accurate side-chain topology and energy refinement (faster) as a new method for protein structure optimization. Proteins: Structure, Function, and Bioinformatics, 48(1):31–43, 2002.

[31] Mark A Hallen, Daniel A Keedy, and Bruce R Donald. Dead-end elimination with perturbations (deeper): A provable protein design algorithm with continuous sidechain and backbone flexibility. Proteins: Structure, Function, and Bioinformatics, 81(1):18–39, 2013.

[32] Carleton L Kingsford, Bernard Chazelle, and Mona Singh. Solving and analyzing side-chain positioning problems using linear and integer programming. Bioinformatics, 21(7):1028–1039, 2004.

[33] Gevorg Grigoryan, Aaron W Reinke, and Amy E Keating. Design of protein-interaction specificity gives selective bzip-binding peptides. Nature, 458(7240):859, 2009.

[34] Chen Yanover and Yair Weiss. Approximate inference and protein-folding. In Advances in neural information processing systems, pages 1481–1488, 2003.

[35] Hetunandan Kamisetty, Eric P Xing, and Christopher J Langmead. Free energy estimates of all-atom protein structures using generalized belief propagation. Journal of Computational Biology, 15(7):755–766, 2008.

[36] Lisa Holm and Chris Sander. Fast and simple monte carlo algorithm for side chain optimization in proteins: application to model building by homology. Proteins: Structure, Function, and Bioinformatics, 14(2):213–223, 1992.

[37] Brian Kuhlman and David Baker. Native protein sequences are close to optimal for their structures. Proceedings of the National Academy of Sciences, 97(19):10383–10388, 2000.

[38] Carol A Rohl, Charlie EM Strauss, Kira MS Misura, and David Baker. Protein structure prediction using rosetta. In Methods in enzymology, volume 383, pages 66–93. Elsevier, 2004.

[39] Andrew Leaver-Fay, Michael Tyka, Steven M. Lewis, Oliver F. Lange, James Thompson, Ron Jacak, Kristian W. Kaufman, P. Douglas Renfrew, Colin A. Smith, Will Sheffler, Ian W. Davis, Seth Cooper, Adrien Treuille, Daniel J. Mandell, Florian Richter, Yih-En Andrew Ban, Sarel J. Fleishman, Jacob E. Corn, David E. Kim, Sergey Lyskov, Monica Berrondo, Stuart Mentzer, Zoran Popović, James J. Havranek, John Karanicolas, Rhiju Das, Jens Meiler, Tanja Kortemme, Jeffrey J. Gray, Brian Kuhlman, David Baker, and Philip Bradley. Chapter nineteen - rosetta3: An object-oriented software suite for the simulation and design of macromolecules. In Michael L. Johnson and Ludwig Brand, editors, Computer Methods, Part C, volume 487 of Methods in Enzymology, pages 545 – 574. Academic Press, 2011.

[40] Yi Liu and Brian Kuhlman. Rosettadesign server for protein design. Nucleic acids research, 34(suppl_2):W235–W238, 2006.

[41] Frances H Arnold. Design by directed evolution. Accounts of chemical research, 31(3):125–131, 1998.

[42] Kevin M Esvelt, Jacob C Carlson, and David R Liu. A system for the continuous directed evolution of biomolecules. Nature, 472(7344):499, 2011.

[43] Benjamin Borgo and James J Havranek. Automated selection of stabilizing mutations in designed and natural proteins. Proceedings of the National Academy of Sciences, 109(5):1494–1499, 2012.

[44] Ian W Davis and David Baker. Rosettaligand docking with full ligand and receptor flexibility. Journal of molecular biology, 385(2):381–392, 2009.

[45] Xiaozhen Hu, Huanchen Wang, Hengming Ke, and Brian Kuhlman. High-resolution design of a protein loop. Proceedings of the National Academy of Sciences, 104(45):17668–17673, 2007.

[46] Colin A Smith and Tanja Kortemme. Structure-based prediction of the peptide sequence space recognized by natural and synthetic pdz domains. Journal of molecular biology, 402(2):460–474, 2010.

[47] Colin A Smith and Tanja Kortemme. Predicting the tolerated sequences for proteins and protein interfaces using rosettabackrub flexible backbone design. PloS one, 6(7):e20451, 2011.

[48] Colin A Smith and Tanja Kortemme. Backrub-like backbone simulation recapitulates natural protein conformational variability and improves mutant side-chain prediction. Journal of molecular biology, 380(4):742–756, 2008.

[49] Gautam Dantas, Brian Kuhlman, David Callender, Michelle Wong, and David Baker. A large scale test of computational protein design: folding and stability of nine completely redesigned globular proteins. Journal of molecular biology, 332(2):449–460, 2003.

[50] Peng Xiong, Meng Wang, Xiaoqun Zhou, Tongchuan Zhang, Jiahai Zhang, Quan Chen, and Haiyan Liu. Protein design with a comprehensive statistical energy function and boosted by experimental selection for foldability. Nature communications, 5:5330, 2014.

[51] Christopher M Topham, Sophie Barbe, and Isabelle Andre. An atomistic statistically effective energy function for computational protein design. Journal of chemical theory and computation, 12(8):4146–4168, 2016.

[52] Xiaoqun Zhou, Peng Xiong, Meng Wang, Rongsheng Ma, Jiahai Zhang, Quan Chen, and Haiyan Liu. Proteins of well-defined structures can be designed without backbone readjustment by a statistical model. Journal of structural biology, 196(3):350–357, 2016.

[53] Jianfu Zhou, Alexandra E Panaitiu, and Gevorg Grigoryan. A general-purpose protein design framework based on mining sequence–structure relationships in known protein structures. Proceedings of the National Academy of Sciences, 2019.

[54] Manfred J Sippl. Knowledge-based potentials for proteins. Current opinion in structural biology, 5(2):229–235, 1995.

[55] Namrata Anand and Possu Huang. Generative modeling for protein structures. In Advances in Neural Information Processing Systems, pages 7494–7505, 2018.

[56] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Protein structure prediction using multiple deep neural networks in casp13. Proteins: Structure, Function, and Bioinformatics, 2019.

[57] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. IEEE Transactions on pattern analysis and machine intelligence, (6):721–741, 1984.

[58] Julian Besag. Efficiency of pseudolikelihood estimation for simple gaussian fields. Biometrika, pages 616–618, 1977.

[59] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[60] Wen Torng and Russ B Altman. 3d deep convolutional neural networks for amino acid environment similarity analysis. BMC bioinformatics, 18(1):302, 2017.

[61] Raghav Shroff, Austin W Cole, Barrett R Morrow, Daniel J Diaz, Isaac Donnell, Jimmy Gollihar, Andrew D Ellington, and Ross Thyer. A structure-based deep learning framework for protein engineering. bioRxiv, page 833905, 2019.

[62] Wouter Boomsma and Jes Frellsen. Spherical convolutions and their application in molecular modelling. In Advances in Neural Information Processing Systems, pages 3433–3443, 2017.

[63] Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. In Advances in Neural Information Processing Systems, pages 10381–10392, 2018.

[64] Tony E Lewis, Ian Sillitoe, Natalie Dawson, Su Datt Lam, Tristan Clarke, David Lee, Christine Orengo, and Jonathan Lees. Gene3d: extensive prediction of globular domains in proteins. Nucleic acids research, 46(D1):D435–D439, 2017.

[65] Natalie L Dawson, Tony E Lewis, Sayoni Das, Jonathan G Lees, David Lee, Paul Ashford, Christine A Orengo, and Ian Sillitoe. Cath: an expanded resource to predict protein function through structure and sequence. Nucleic acids research, 45(D1):D289–D295, 2016.

[66] Robert L Baldwin. Energetics of protein folding. Journal of molecular biology, 371(2):283–301, 2007.

[67] Ken A Dill. Additivity principles in biochemistry. Journal of Biological Chemistry, 272(2):701–704, 1997.

[68] James T Kellis Jr, Kerstin Nyberg, and Alan R Fersht. Energetics of complementary side chain packing in a protein hydrophobic core. Biochemistry, 28(11):4914–4922, 1989.

[69] A Elisabeth Eriksson, Walter A Baase, Xue-Jun Zhang, Dirk W Heinz, MPBE Blaber, Enoch P Baldwin, and Brian W Matthews. Response of a protein structure to cavity-creating mutations and its relation to the hydrophobic effect. Science, 255(5041):178–183, 1992.

[70] AE Eriksson, WA Baase, JA Wozniak, and BW Matthews. A cavity-containing mutant of t4 lysozyme is stabilized by buried benzene. Nature, 355(6358):371, 1992.

[71] Walter Kauzmann. Some factors in the interpretation of protein denaturation. In Advances in protein chemistry, volume 14, pages 1–63. Elsevier, 1959.

[72] David Eisenberg and Andrew D McLachlan. Solvation energy in protein folding and binding. Nature, 319(6050):199, 1986.

[73] Scott M Le Grand and Kenneth M Merz Jr. Rapid approximation to molecular surface area via the use of boolean logic and look-up tables. Journal of Computational Chemistry, 14(3):349–352, 1993.

[74] Patrick J Fleming and George D Rose. Do all backbone polar groups in proteins form hydrogen bonds? Protein Science, 14(7):1911–1917, 2005.

[75] Daniel WA Buchan and David T Jones. The psipred protein analysis workbench: 20 years on. Nucleic acids research, 47(W1):W402–W407, 2019.

[76] David T Jones. Protein secondary structure prediction based on position-specific scoring matrices. Journal of molecular biology, 292(2):195–202, 1999.

[77] Stephen F Altschul, John C Wootton, E Michael Gertz, Richa Agarwala, Aleksandr Morgulis, Alejandro A Schäffer, and Yi-Kuo Yu. Protein database searches using compositionally adjusted substitution matrices. The FEBS journal, 272(20):5101–5109, 2005.

[78] Alejandro A Schäffer, L Aravind, Thomas L Madden, Sergei Shavirin, John L Spouge, Yuri I Wolf, Eugene V Koonin, and Stephen F Altschul. Improving the accuracy of psi-blast protein database searches with composition-based statistics and other refinements. Nucleic acids research, 29(14):2994–3005, 2001.

[79] Stephen F Altschul, Thomas L Madden, Alejandro A Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. Nucleic acids research, 25(17):3389–3402, 1997.

[80] Baris E Suzek, Yuqi Wang, Hongzhan Huang, Peter B McGarvey, Cathy H Wu, and UniProt Consortium. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. Bioinformatics, 31(6):926–932, 2014.

[81] Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers: Original Research on Biomolecules, 22(12):2577–2637, 1983.

[82] Leonard G Presta and George D Rose. Helix signals in proteins. Science, 240(4859):1632–1641, 1988.

[83] Rajeev Aurora and George D. Rosee. Helix capping. Protein Science, 7(1):21–38, 1998.

[84] Kim T Simons, Rich Bonneau, Ingo Ruczinski, and David Baker. Ab initio protein structure prediction of casp iii targets using rosetta. Proteins: Structure, Function, and Bioinformatics, 37(S3):171–176, 1999.

[85] Richard Bonneau, Jerry Tsai, Ingo Ruczinski, Dylan Chivian, Carol Rohl, Charlie EM Strauss, and David Baker. Rosetta in casp4: progress in ab initio protein structure prediction. Proteins: Structure, Function, and Bioinformatics, 45(S5):119–126, 2001.

[86] Dominik Gront, Daniel W Kulp, Robert M Vernon, Charlie EM Strauss, and David Baker. Generalized fragment picking in rosetta: design, protocols and applications. PloS one, 6(8):e23294, 2011.

[87] Jianyi Yang, Ivan Anishchenko, Hahnbeom Park, Zhenling Peng, Sergey Ovchinnikov, and David Baker. Improved protein structure prediction using predicted interresidue orientations. Proceedings of the National Academy of Sciences, 117(3):1496–1503, 2020.

[88] Arnout R. D. Voet, Hiroki Noguchi, Christine Addy, David Simoncini, Daiki Terada, Satoru Unzai, Sam-Yong Park, Kam Y. J. Zhang, and Jeremy R. H. Tame. Computational design of a self-assembling symmetrical -propeller protein. Proceedings of the National Academy of Sciences, 111(42):15102–15107, 2014.

[89] Yuan Zhang, Yang Chen, Chenran Wang, Chun-Chao Lo, Xiuwen Liu, Wei Wu, and Jinfeng Zhang. Prodconn: Protein design using a convolutional neural network. Proteins: Structure, Function, and Bioinformatics, 2020.

[90] John Ingraham, Vikas K Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. 2019.

[91] Yilun Du Du, Joshua Meier, Jerry Ma, Rob Fergus, and Alexander Rives. Energy-based models for atomic-resolution protein conformations. International Conference on Learning Representations 2020, 2020.

[92] Jingxue Wang, Huali Cao, John ZH Zhang, and Yifei Qi. Computational protein design with deep learning neural networks. Scientific reports, 8(1):6349, 2018.

[93] James O'Connell, Zhixiu Li, Jack Hanson, Rhys Heffernan, James Lyons, Kuldip Paliwal, Abdollah Dehzangi, Yuedong Yang, and Yaoqi Zhou. Spin2: Predicting sequence profiles from protein structures using deep neural networks. Proteins: Structure, Function, and Bioinformatics, 86(6):629–633, 2018.

[94] Joe G Greener, Lewis Moffat, and David T Jones. Design of metalloproteins and novel protein folds using variational autoencoders. Scientific reports, 8(1):16189, 2018.

[95] Alexey Strokach, David Becerra, Carles Corbi-Verge, Albert Perez-Riba, and Philip M Kim. Designing real novel proteins using deep graph neural networks. bioRxiv, page 868935, 2019.

[96] David A Levin and Yuval Peres. Markov chains and mixing times, volume 107. American Mathematical Soc., 2017.

[97] Robbie P Joosten, Tim AH Te Beek, Elmar Krieger, Maarten L Hekkelman, Rob WW Hooft, Reinhard Schneider, Chris Sander, and Gert Vriend. A series of pdb related databases for everyday needs. Nucleic acids research, 39(suppl_1):D411–D419, 2010.

[98] Po-Ssu Huang, Yih-En Andrew Ban, Florian Richter, Ingemar Andre, Robert Vernon, William R Schief, and David Baker. Rosettaremodel: a generalized framework for flexible backbone protein design. PloS one, 6(8):e24109, 2011.

[99] Sidhartha Chaudhury, Sergey Lyskov, and Jeffrey J Gray. Pyrosetta: a script-based interface for implementing molecular modeling algorithms using rosetta. Bioinformatics, 26(5):689–691, 2010.

[100] J Berman. Hm and westbrook, z. feng, g. gilliland, tn bhat, h. weissig, in shindyalov, and pe bourne. the protein data bank. Nucleic Acids Research, 106:16972–16977, 2000.

[101] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

[102] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[103] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[104] Rebecca F Alford, Andrew Leaver-Fay, Jeliazko R Jeliazkov, Matthew J O'Meara, Frank P DiMaio, Hahnbeom Park, Maxim V Shapovalov, P Douglas Renfrew, Vikram K Mulligan, Kalli Kappel, et al. The rosetta all-atom energy function for macromolecular modeling and design. Journal of chemical theory and computation, 13(6):3031–3048, 2017.

[105] Hahnbeom Park, Philip Bradley, Per Greisen Jr, Yuan Liu, Vikram Khipple Mulligan, David E Kim, David Baker, and Frank DiMaio. Simultaneous optimization of biomolecular energy functions on features from small molecules and macromolecules. Journal of chemical theory and computation, 12(12):6201–6212, 2016.

[106] Will Sheffler and David Baker. Rosettaholes: rapid assessment of protein core packing for structure prediction, refinement, design, and validation. Protein Science, 18(1):229–239, 2009.

[107] Lucas Gregorio Nivón, Rocco Moretti, and David Baker. A pareto-optimal refinement method for protein design scaffolds. PloS one, 8(4):e59004, 2013.

[108] Patrick Conway, Michael D Tyka, Frank DiMaio, David E Konerding, and David Baker. Relaxation of backbone bond geometry improves protein energy landscape modeling. Protein Science, 23(1):47–55, 2014.

[109] Firas Khatib, Seth Cooper, Michael D Tyka, Kefan Xu, Ilya Makedon, Zoran Popović, and David Baker. Algorithm discovery by protein folding game players. Proceedings of the National Academy of Sciences, 108(47):18949–18953, 2011.

[110] Michael D Tyka, Daniel A Keedy, Ingemar André, Frank DiMaio, Yifan Song, David C Richardson, Jane S Richardson, and David Baker. Alternate states of proteins revealed by detailed energy landscape mapping. Journal of molecular biology, 405(2):607–618, 2011.

Movie S1. Examples of model-guided rotamer repacking on native backbone test cases.

Movie S2. Examples of model-guided sequence and rotamer design via annealing on native backbone test cases and *de novo* TIM-barrel scaffold.

SI Dataset S1 (`dataset_S1.xlsx`)

Training information, including excluded ions/ligands, train/test CATH topology classes, and train/test CATH domains

SI Dataset S2 (`dataset_S2.xlsx`)

Native test case analysis data including rotamer recovery data, sequence design metrics, amino acid distribution data post-design, decoy ranking, and runtime raw data

SI Dataset S3 (`dataset_S3.xlsx`)

AbInitio structure prediction raw data for native test cases.

SI Dataset S4 (`dataset_S4.xlsx`)

TIM-barrel design data, including details on experimentally tested four-fold symmetric designs, forward-folding data, CD (Circular dichroism) wavelength scans and thermal melt data, and metrics across design sampling runs.

SI Dataset S5 (`dataset_S5.zip`)

Other data including design scaffolds, rotamer repacking final models, designed sequence models, positionwise amino acid distributions, AbInitio fragment data, and trRosetta results. Dataset available for download at `https://drive.google.com/file/d/1yBQ42M4yRWlb4G6vF1QS8__IUfv5eJul/view?usp=sharing`

Code for running sequence design algorithm available at `https://github.com/nanand2/protein_seq_des`.

Additional analysis of model designs

Native test case structures belong to the beta-lactamase (CATH:3.30.450), T-fimbrin (CATH:1.10.418), alpha-beta plait (CATH:3.30.70), and lipocalin (CATH:2.40.128) topology classes, respectively.

Amino acid distribution post-design

Model-designed sequences adhere to an amino acid distribution similar to the native sequence and its aligned homologs (Fig. S4A), with the most pronounced difference being an apparent overuse of lysine (K) and glutamate (E) and under-use of rarer residues (H, N, Q, W, C) relative to the homolog sequences; *Rosetta-FixBB* designs also overuse K and E and under-use H, W, N, and C relative to the native MSA distribution. Like the model, Rosetta protocols match the native distribution well, with a tendency to overuse small hydrophobic residues, in particular alanine (A), valine(V), and isoleucine (I). Since the homologous sequences discovered by MSA likely capture amino acid patterns that can anchor and support the given backbones, the similarity of the distributions suggests that the model designs retain viable sequence patterns.

Biochemical metrics of interest

We compare our designs to the *Rosetta-FixBB* and *Rosetta-RelaxBB* protocol designs, as well as the native idealized structure, across key metrics (Fig. S5). Since the designs have less than 50% sequence overlap with the native on average, we also include data for the native sequence with either 50% or 75% of the residues randomly mutated; these perturbed sequences serve as a negative control and indicate expected performance for likely non-optimal deviations from the native sequence.

We also report performance across these metrics for designed sequences relaxed with the *RosettaRelax* protocol. This procedure allows the backbone and rotamers to move to adopt a lower Rosetta energy conformation. Relaxing the designs allows us to assess whether alternate rotamers and deviations of the starting protein backbone can lead to an improvement in performance across the metrics considered.

(1) *Packstat.* Model designs tend to have a lower packstat score compared to to the native sequence and the Rosetta designs (Fig. S5A); however, on average the packstat scores are still higher relative to random perturbations of the native sequence. While this trend seems to indicate that the model designs are less optimal than the Rosetta designs, when we look at the model designs post-relax (Fig. S5B), the packstat scores improve and better match those of the native sequence and Rosetta designs, while the perturbed sequence packstat scores remain low. At the same time, the post-relax alpha-carbon backbone RMSDs between design methods are also comparable (Fig. S7E). These results suggest that the designed sequences do tightly pack core regions, as slight movements of the backbone and repacked rotamers for model designs give well-packed structures.

(2) *Exposed hydrophobics.* Model designs in general do not place exposed hydrophobics in solvent-exposed positions, similar to the Rosetta designs (Fig. S5C). This trend in the model designs is largely due to the relative abundance of cytosolic proteins in the PDB compared to membrane proteins, which would in general have hydrophobic regions exposed to the membrane lipid environment. The native sequence for test case *3mx7* has many exposed hydrophobic residues, suggesting that the native protein might bind to a target, forming a hydrophobic interface.

(3) *Buried unsatisfied backbone atoms.* For all of the test cases except *1bkr*, model designs have similar or fewer unsatisfied backbone polar atoms compared to the native sequence (Fig. S5D,E). For *1bkr*, although the average number of unsatisfied backbone polar atoms is greater than that of the native sequence or Rosetta designs, the distribution is fairly wide, indicating that there are many designs that have fewer unsatisfied backbone polar atoms compared to the native sequence. However, some of the 50% perturbed sequences have fewer unsatisfied backbone polar atoms than the native, suggesting that this metric alone is not sufficient for selecting or rejecting designs.

(4) *Buried unsatisfied side-chain atoms.* Model designs across test cases have few unsatisfied buried side-chain polar atoms, similar to the native sequences and Rosetta designs (Fig. S5F,G). This indicates that over the course of sampling, side-chains that are placed in buried regions are adequately supported by backbone hydrogen bonds or by design of other side-chains that support the buried residue.

Heuristic structure energy

Although the Rosetta energy is not optimized during our design procedure, many of the designs have low Rosetta energy relative to perturbed sequence baselines (Fig. S7A). After relaxing the model designs, their Rosetta energy decreases to match the native backbone, with relatively low subsequent backbone deviation; in contrast, sequences that are 50% similar to native, with random perturbations, have a high Rosetta energy, even after relax (Fig. S7B) The model energy too is similarly low for the model designs and the Rosetta designs, indicating that the model energy as a heuristic roughly matches Rosetta in differentiating low-energy vs. high-energy designs (Fig. S7D). Overall, the alpha-carbon RMSD distributions post-RosettaRelax for the model designs correspond with the native backbone relaxed distributions (Fig. S7E).

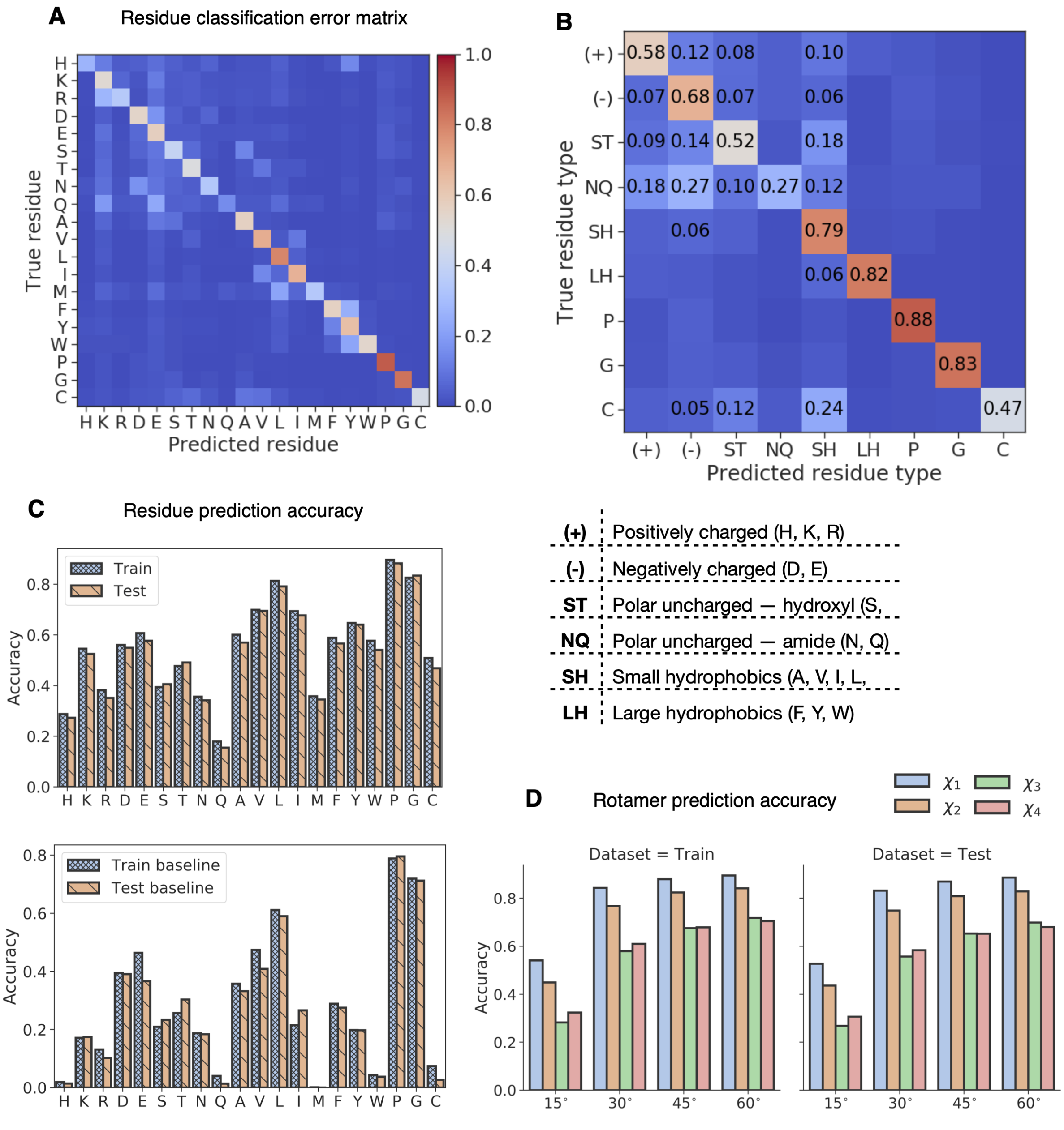

Figure S1: Classifier performance and learned rotamer distributions. A) Classifier error matrix for individual residue prediction on test set data and for B) prediction of groups of biochemically similar residues C) Residue-specific prediction accuracy for train and test set data for the conditional model (top) and baseline model (bottom). D) Rotamer prediction accuracy to within 15, 30, 45, and 60 degrees across train and test set.

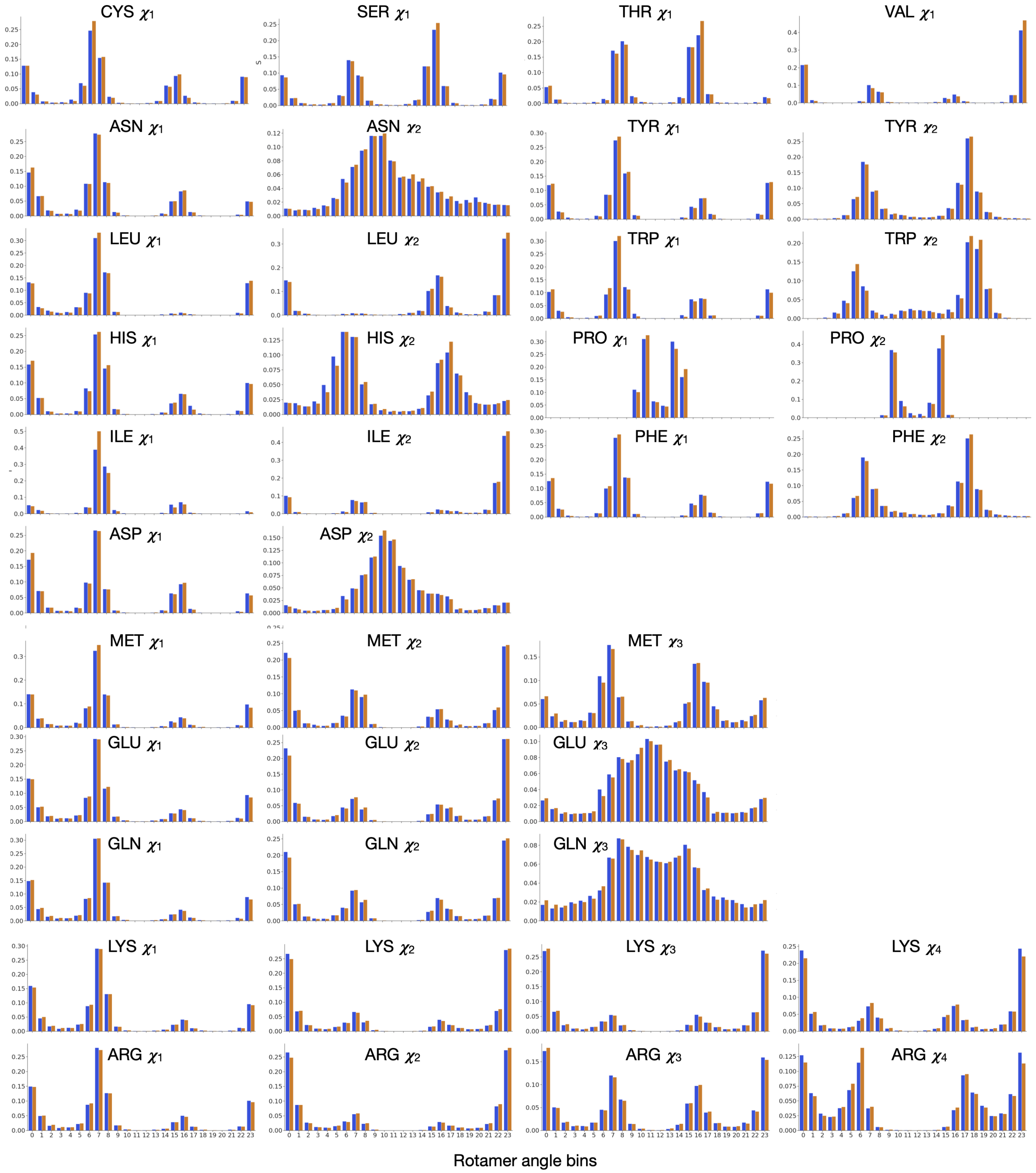

Figure S2: Learned residue-specific rotamer distributions. Binned rotamer distributions for the native test set domains (blue) versus the model predicted distributions (orange). Native distributions are the normalized empirical rotamer distributions for each residue across the test set. Model distributions are the average of the network-predicted distributions across the test set examples.

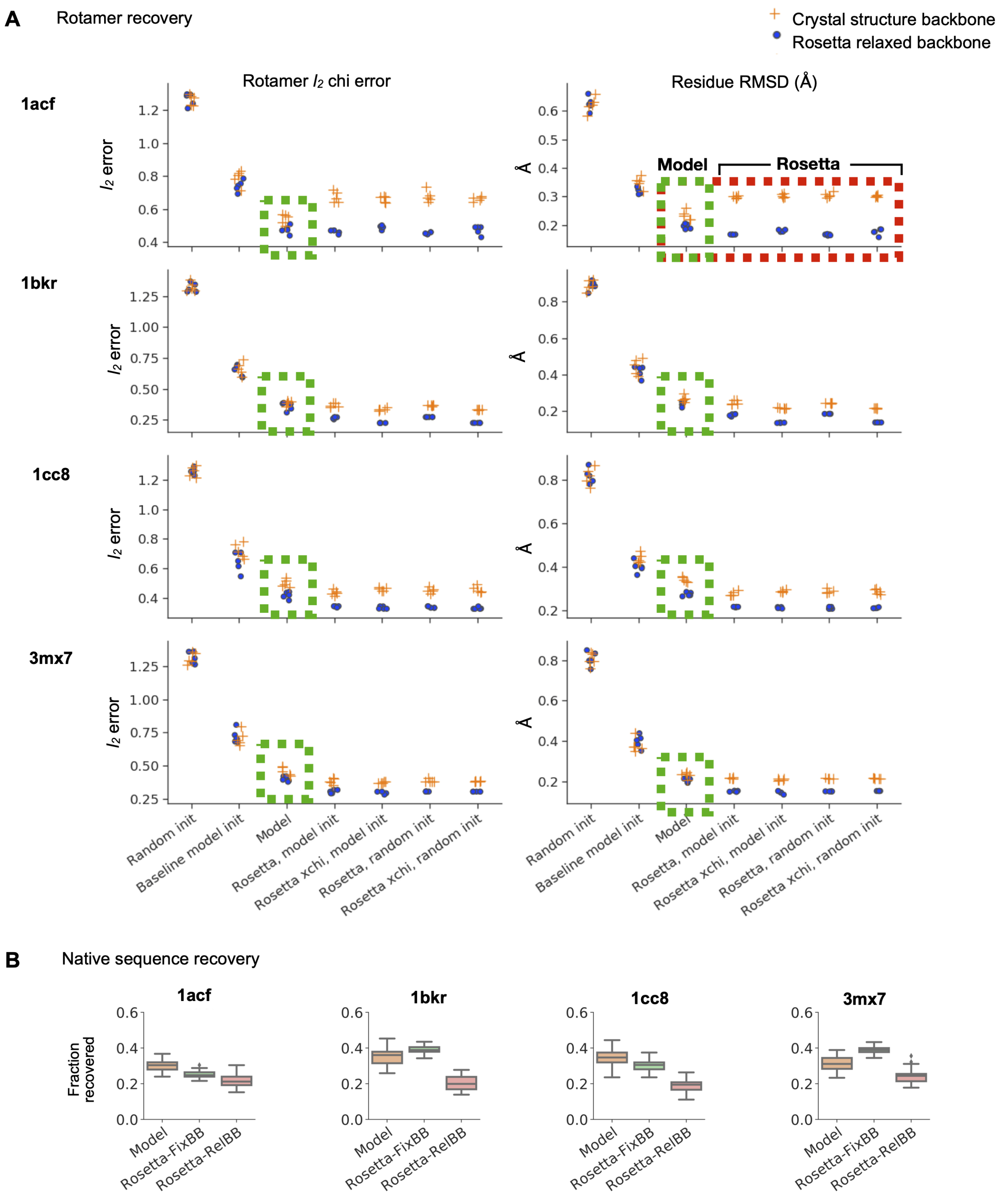

Figure S3: Rotamer and native sequence recovery. A) Rotamer recovery. $n = 5$ for all methods. Data shown for rotamer recovery on crystal structure backbone, as well as backbone relaxed under the Rosetta energy function with constraints to adhere to original backbone atom placement. Random init – random initialization of chi angles. Baseline model init – initialization with rotamer prediction by baseline classifier model from backbone alone. Model – 2500 iterations of rotamer sampling and annealing with conditional model. Rosetta protocols were run either with or without extra chi_1 and chi_2 sampling (xchi). (Left) Average l_2 chi angle error across design methods. (Right) Average residue side-chain RMSD (Å). B) Native sequence recovery across test case crystal structures ($n = 50$, each).

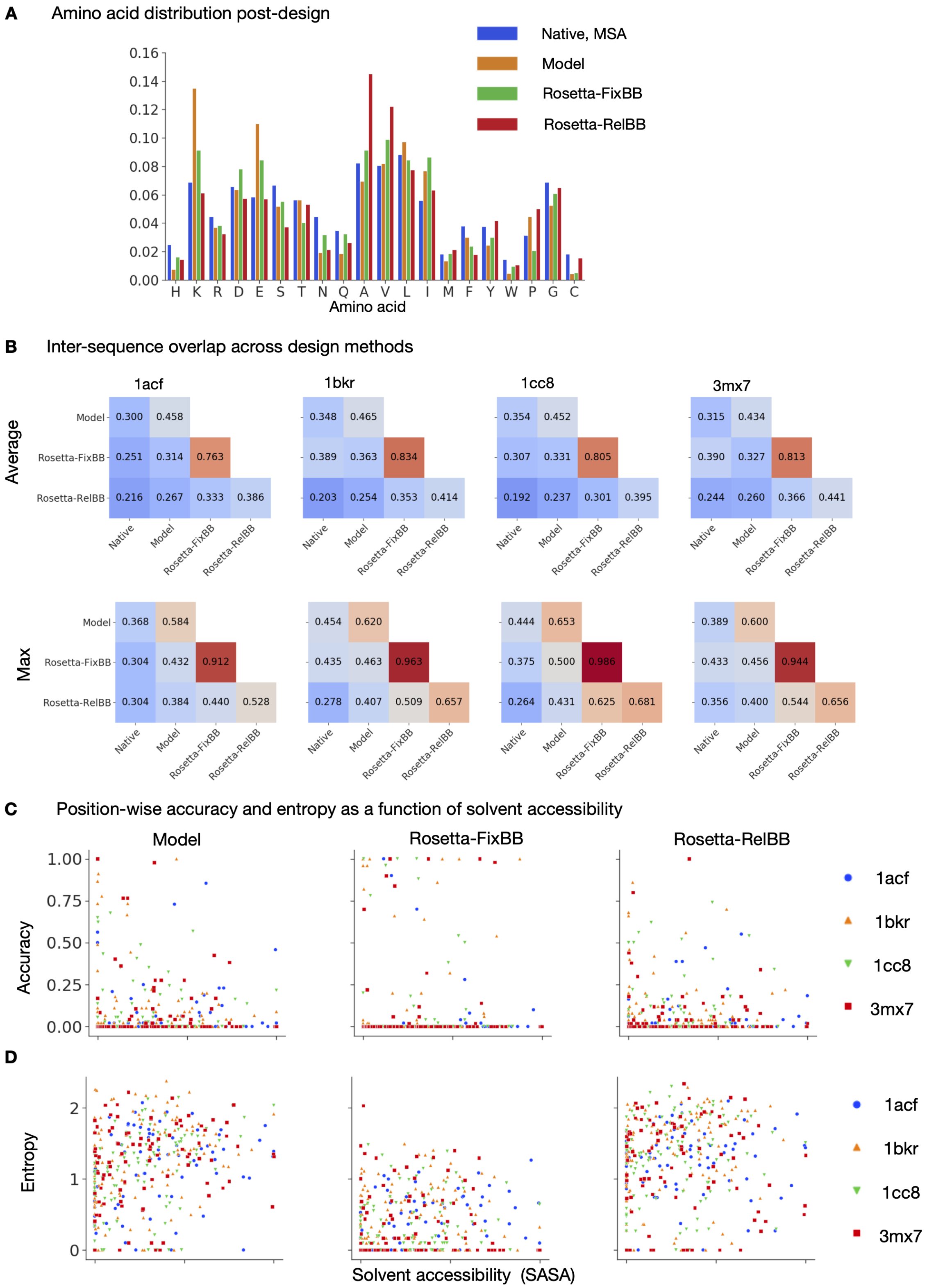

Figure S4: Additional metrics for assessing model designs. A) Amino acid distribution post-design relative to native sequence and its aligned homologs (max. 500 hits per sequence). MSAs obtained using PSI-BLAST v2.9, the NR50 database, BLOSUM62 with an E-value cutoff of 0.1. B) Inter-sequence overlap. Average (top) and maximum (bottom) inter-sequence fractional overlap within and across top sequences from each design method. C) Position-wise accuracy vs. solvent accessibility (SASA) across design methods for four test cases ($n = 50$, each). Accuracy with respect to native sequence. Solvent-accessible surface area (SASA) calculated for native residues [73]. D) Position-wise entropy vs. solvent accessibility (SASA) across design methods for four test cases ($n = 50$, each). Solvent-accessible surface area (SASA) calculated for native residues.

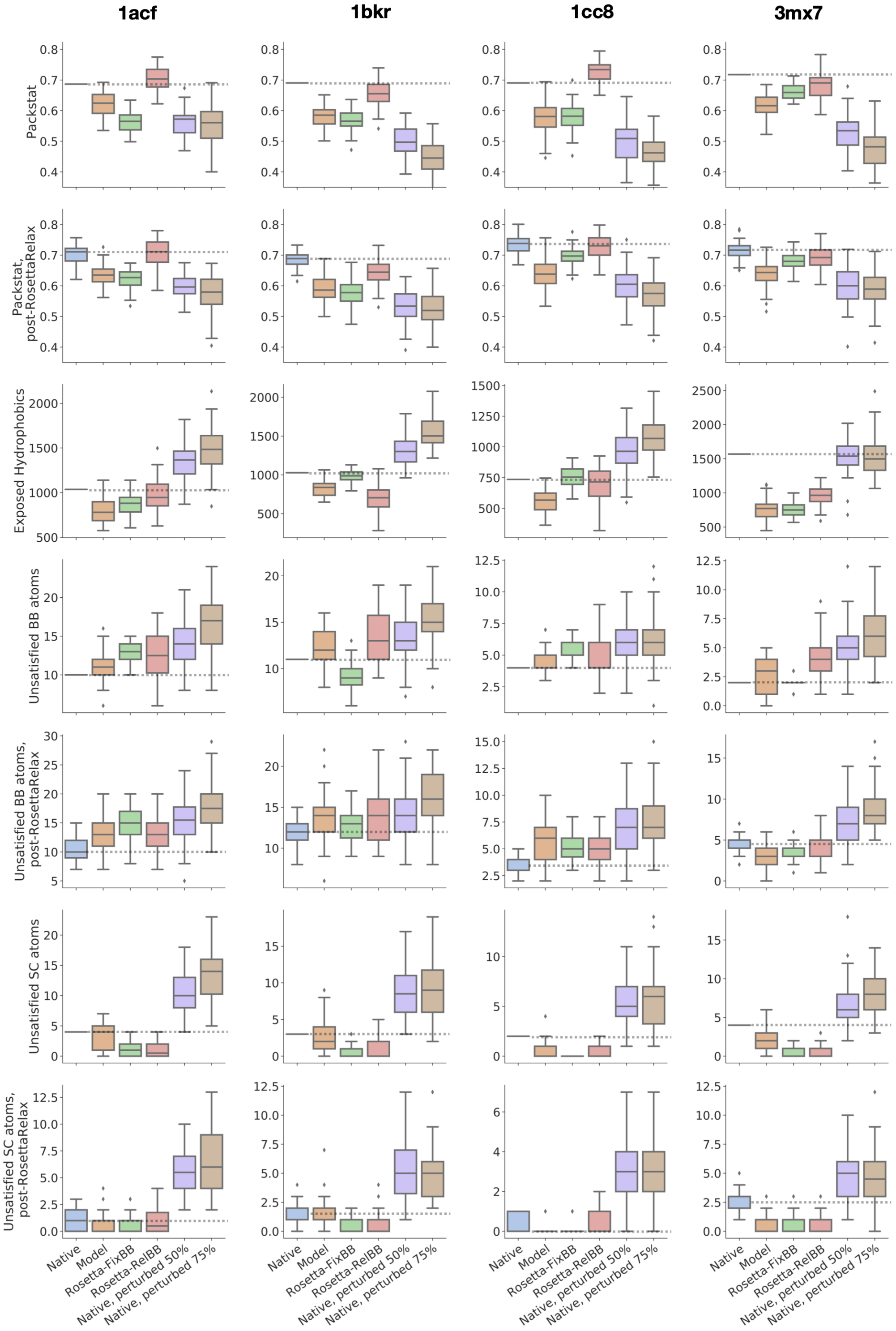

Figure S5: Biochemical metrics of interest for model designs. Designs ($n = 50$) compared to native idealized structure ($n = 1$) or distribution of relaxed native structures ($n = 50$). 50% and 75% mutated native sequences included as negative controls. A-B) Packstat, a measure of core residue packing [106] pre- (A) and post- (B) RosettaRelax [107, 108, 109, 110]. C) Total solvent-accessible surface area (SASA) of exposed hydrophobic residues [73]. D-E) Number of buried unsatisfied polar backbone (BB) atoms pre- (D) and post- (E) relax. E-F) Number of buried unsatisfied polar side-chain (SC) atoms pre- (E) and post- (F) relax.

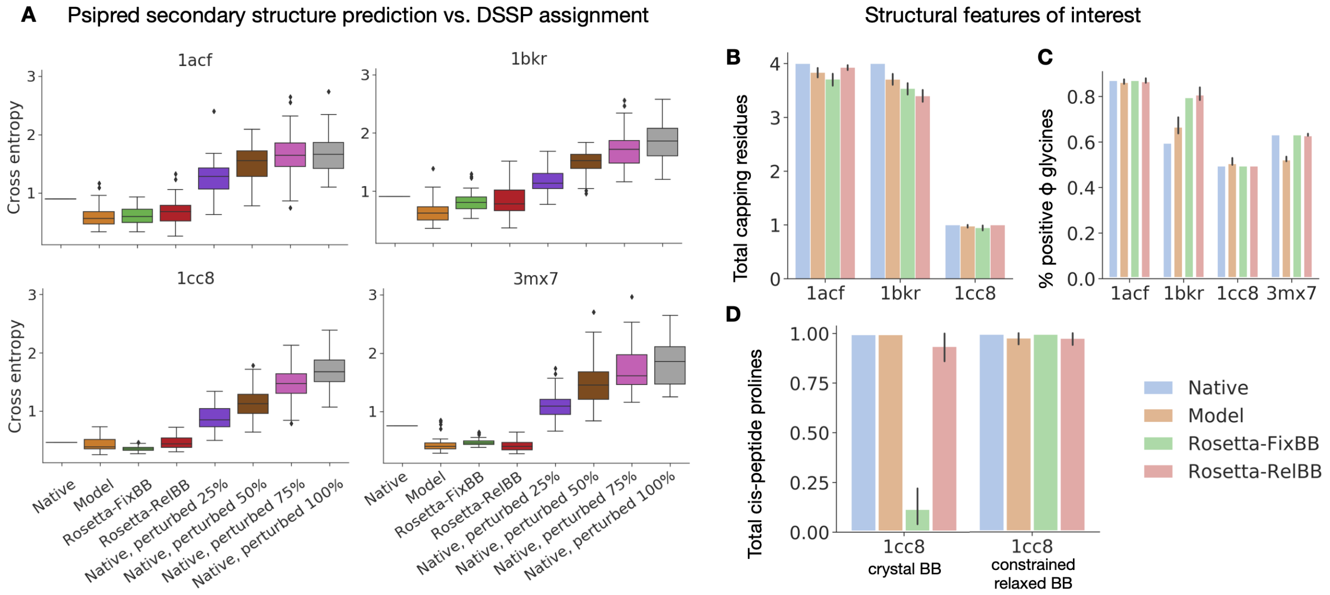

Figure S6: Additional design metrics across native test cases. $n = 50$ for all methods. A) Psipred secondary structure prediction for designed sequences. Cross-entropy of Psipred prediction (helix, loop, sheet) from sequence alone with respect to DSSP assignments [75, 76, 77, 78, 79, 81]. B) Capping residue placement. Average number of N-terminal helical capping residues across designs for test cases with capping positions. C) Percent glycines at positive ϕ backbone positions across test cases. D) Total number of cis-peptide prolines ($\omega_{i-1} < 15$) for *1cc8* for designs on crystal structure backbone vs. Rosetta constrained relaxed backbone.

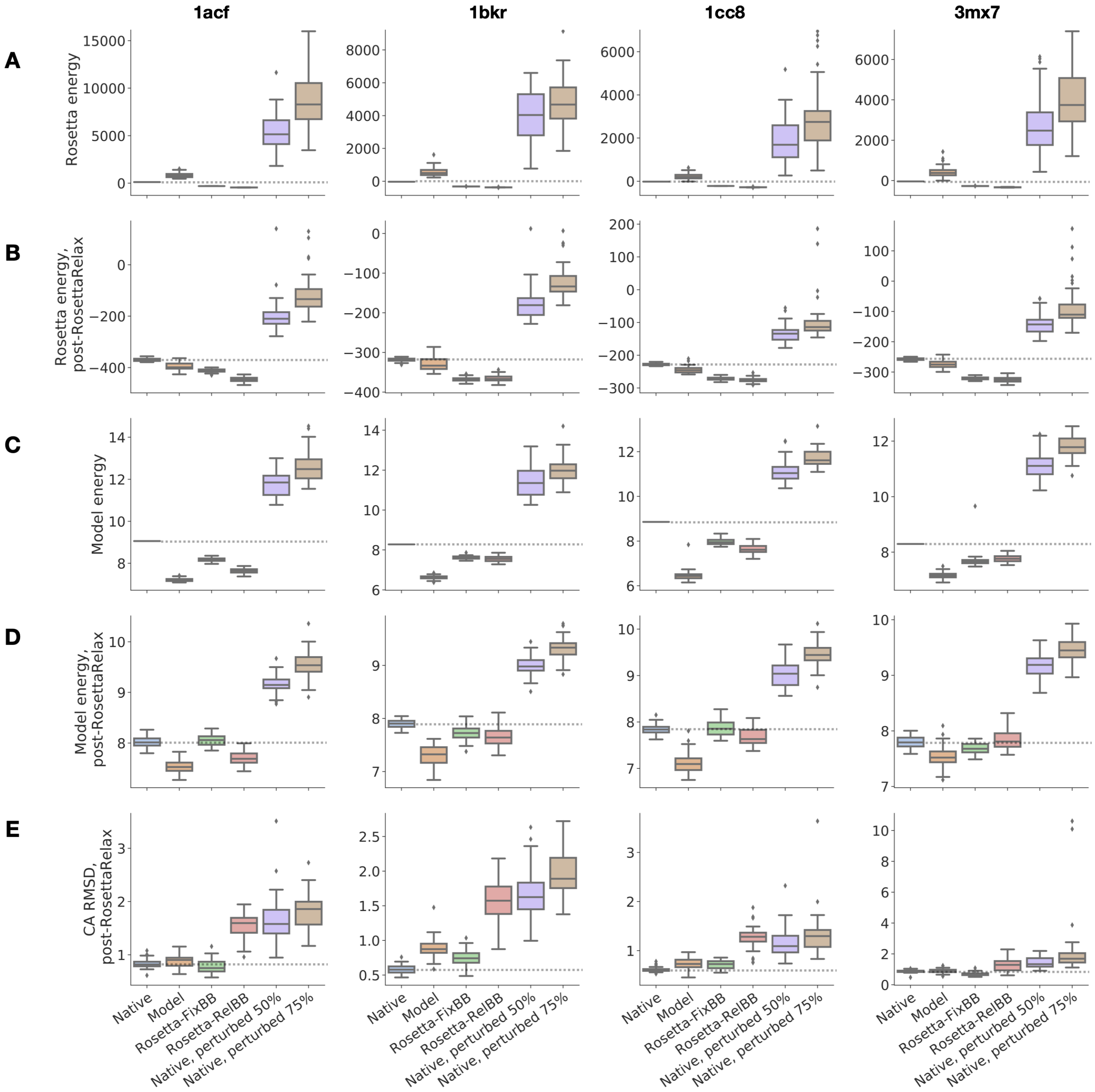

Figure S7: Energy under the model and under the Rosetta energy function pre- and post-RosettaRelax. 50% and 75% mutated native sequences included as negative controls. Designs ($n = 50$) compared to native idealized structure ($n = 1$) or distribution of relaxed native structures ($n = 50$). A-B) Rosetta energy pre- (A) and post- (B) relax. C-D) Model energy (negative pseudo-log-likelihood, normalized by protein length) pre- (C) and post- (D) relax. E) Alpha-carbon RMSD (Å) post-RosettaRelax.

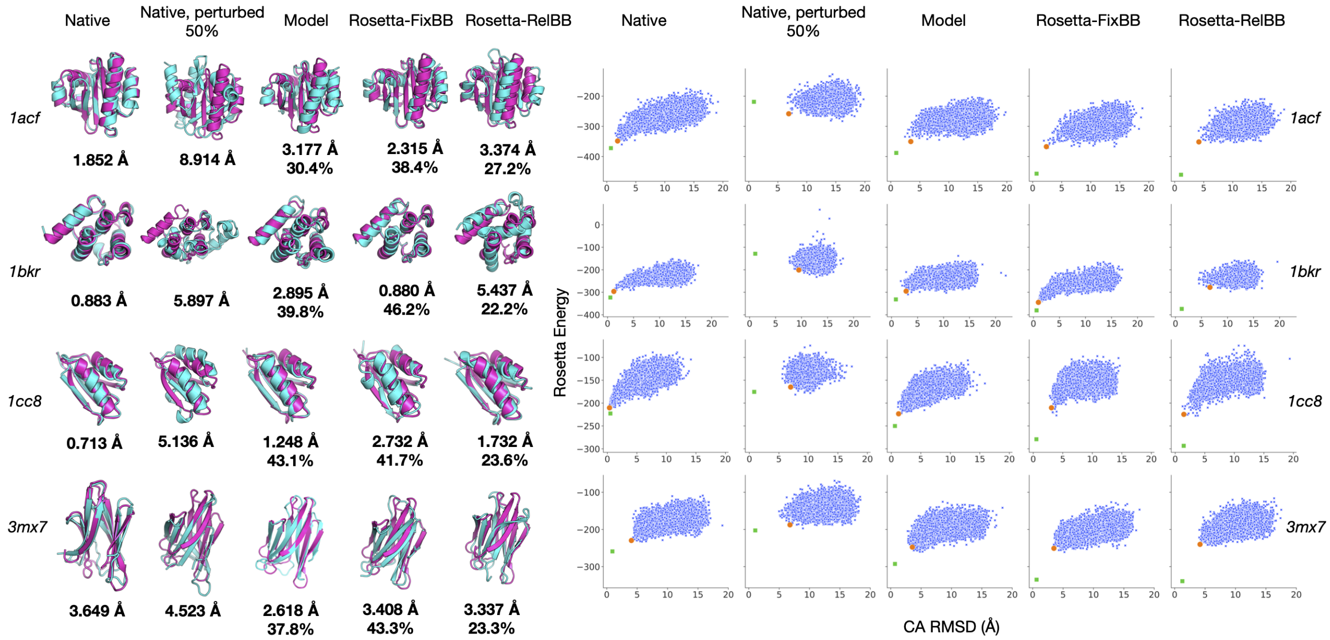

Figure S8: Blind structure prediction of designed sequences. Native sequences and selected designs were forward folded using Rosetta AbInitio for four design test cases. 50% randomly perturbed native sequences included as a negative control. (Left) Folded structure with best summed rank of template-RMSD and Rosetta energy across 10^4 folding trajectories. Decoys (yellow) are aligned to the idealized native backbones (blue). Sequence identity and RMSD (Å) compared to native are reported below the structures. For *Rosetta-RelaxBB*, RMSD of selected design to relaxed structure post-design is given in parentheses. (Right) Rosetta energy vs. RMSD (Å) to native funnel plots. RMSD is calculated with respect to the idealized and relaxed native crystal structure. For *Rosetta-RelaxBB*, RMSD in funnel plot is with respect to relaxed structure post-design. Selected structure with best summed rank of template-RMSD and Rosetta energy is shown in orange. The design after RosettaRelax is shown in green.