THE JOURNAL OF PHYSICAL CHEMISTRY B
A JOURNAL OF THE AMERICAN CHEMICAL SOCIETY

pubs.acs.org/JPCB

Article

SCONES: Self-Consistent Neural Network for Protein Stability Prediction Upon Mutation

Published as part of The Journal of Physical Chemistry virtual special issue "Computational Advances in Protein Engineering and Enzyme Design".

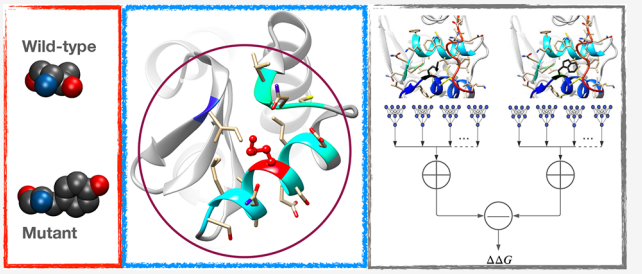
Yashas B. L. Samaga, Shampa Raghunathan, and U. Deva Priyakumar*

Cite This: https://doi.org/10.1021/acs.jpcb.1c04913

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Engineering proteins to have desired properties by mutating amino acids at specific sites is commonplace. Such engineered proteins must be stable to function. Experimental methods used to determine stability at throughputs required to scan the protein sequence space thoroughly are laborious. To this end, many machine learning based methods have been developed to predict thermodynamic stability changes upon mutation. These methods have been evaluated for symmetric consistency by testing with hypothetical reverse mutations. In this work, we propose transitive data augmentation, evaluating transitive consistency with our new $\mathbf{S}^{\text{transitive}}$ data set, and a new machine learning based method, the first of its kind, that incorporates both symmetric and transitive properties into the architecture. Our method, called SCONES, is an interpretable neural network that predicts small relative protein stability changes for missense mutations that do not significantly alter the structure. It estimates a residue's contributions toward protein stability (ΔG) in its local structural environment, and the difference between independently predicted contributions of the reference and mutant residues is reported as $\Delta\Delta G$. We show that this self-consistent machine learning architecture is immune to many common biases in data sets, relies less on data than existing methods, is robust to overfitting, and can explain a substantial portion of the variance in experimental data.



1. INTRODUCTION

Proteins are large biomolecules that naturally evolved to perform specific biological functions.[1] One of the exciting advances in recent decades has been the ability to design and develop new synthetic proteins with desired properties.[2−5] Synthetic proteins have a myriad of applications ranging from vaccine development[6] to tackling environmental problems.[5,7] Biotechnology and biomedical industries use engineered proteins that achieve superior functions.[8−12] For example, engineered enzymes react faster and are more efficient.[13,14] Protein engineering is a promising field and has vast academic and industrial interests.

One approach to protein engineering is redesigning existing proteins to have desired properties.[15,16] Natural proteins are not necessarily optimal;[17] moreover, physiological conditions in which proteins naturally evolved are often different from industrial and laboratory conditions.[15] Proteins are therefore mutated to optimize their properties, and one such property is protein stability.[15,18,19] Any protein must be stable in its environmental conditions for it to be useful.[20] Proteins are mutated to increase stability and often to compensate for destabilization arising from other functionally important mutations.[21] However, the protein sequence space is overwhelmingly large. Experimentally determining the stability of many variants is laborious and expensive,[13,22] and in silco simulations required (e.g., molecular dynamics free energy calculations) to accurately predict stability with high throughput are computationally prohibitive.[23,24] To narrow down the search space for experimental validation, fast computational methods have been developed to predict protein stability changes upon mutation.[25−38] These methods either classify a mutation as stabilizing or destabilizing, or predict a real-valued target indicating the extent of stabilization or destabilization. Popular real-valued targets are a change in denaturation midpoint temperature,[39,40] denoted by ΔT_m; a change in thermal deactivation temperature,[40] denoted by ΔT_{50}; and a change in free energy of folding,[25,27,28,33] denoted by $\Delta\Delta G$. T_m, T_{50}, and ΔG are state functions of proteins. ΔT_m, ΔT_{50}, and $\Delta\Delta G$ are computed by taking the difference

Received: June 4, 2021

between their respective state values in the reference and mutant proteins. This definition confers useful properties that can be exploited to evaluate and improve the performance of protein stability predictors. We present the properties for $\Delta\Delta G$, but they are also applicable to ΔT_m and ΔT_{50}.

A protein must exist in its folded form to function. The stability of a protein refers to its ability to exist in its folded form relative to its denatured state.[41,42] The thermodynamic stability of a protein can thus be quantified by the difference in the Gibbs free energy between the unfolded and folded state, denoted by ΔG:

$$\Delta G = G_{\text{unfolded}} - G_{\text{folded}}$$

The change in stability upon mutation can be quantified by the change in ΔG, denoted by $\Delta\Delta G$:

$$\Delta\Delta G = \Delta G_{\text{mutant}} - \Delta G_{\text{reference}}$$

One of the properties of $\Delta\Delta G$ is antisymmetry. The $\Delta\Delta G$ for a hypothetical reverse mutation must be equal in magnitude but opposite in sign to that of the corresponding forward mutation in identical experimental conditions.[43−46]

$$\Delta\Delta G_{\text{forward}} = \Delta G_2 - \Delta G_1 = -(\Delta G_1 - \Delta G_2) = -\Delta\Delta G_{\text{reverse}}$$

Another property that $\Delta\Delta G$ confers is transitivity: given three proteins X, Y, and Z, the $\Delta\Delta G$ of a mutation that goes from X → Z is the sum of the $\Delta\Delta G$'s of X → Y and Y → Z mutations provided the experimental conditions are kept the same throughout.

$$\begin{aligned}\Delta\Delta G_{X\to Z} &= \Delta G_Z - \Delta G_X \\ &= (\Delta G_Z - \Delta G_Y) + (\Delta G_Y - \Delta G_X) \\ &= \Delta\Delta G_{Y\to Z} + \Delta\Delta G_{X\to Y}\end{aligned}$$

The transitive property can be repeatedly applied to form long chains of mutations. The mutations in a chain can occur at different sites: we can start with two point-mutation samples and end up with a double-mutation sample. We can restrict the mutation site to a fixed position while constructing transitive pairs if we require the resulting combined mutation to be a point mutation.

A variety of computational methods have been developed to predict protein stability changes upon mutation. There are methods based on molecular dynamics,[23,24,47] physics-based force fields,[48] empirical and statistical potentials,[28,49] and more recently machine learning.[25,27,36,50,51] DDGun[28] and DDGun3D[28] are baseline methods that use linear regression on a set of hand-crafted features to predict $\Delta\Delta G$. ACDC-NN[51] and ThermoNet[27] use 3D convolutional neural networks (CNNs) on voxelized mutation site environments to predict $\Delta\Delta G$. PremPS[25] uses random forest (RF) regression on ten evolutionary and structure-based features to predict $\Delta\Delta G$. The use of machine learning to solve problems in fundamental sciences has surged in recent years.[13,25,27,50,52−57] While machine learning tools have powerful abilities to learn from data, their power also makes them prone to learning noise and spurious correlations, especially when training data is biased and limited. The amount of data available for $\Delta\Delta G$ prediction tasks in thermodynamic databases is limited. Moreover, the available data is biased toward specific mutation types. For example, mutations involving alanine are over-represented due to the abundance of alanine-scanning procedures in the literature (Figure S1 in Supporting Information). The distribution of $\Delta\Delta G$ values is also skewed toward destabilizing mutations (Figure S2 in the Supporting Information). Methods that predict protein stability changes upon mutation must exhibit symmetric and transitive properties to be consistent. Existing methods have been evaluated for symmetric consistency, and many older methods that were biased toward predicting destabilizing mutations were shown to lack consistency.[43−46,58] Newer methods resolved symmetric inconsistencies mainly by augmenting the training set with hypothetical reverse mutations.[25,27,38] Few methods incorporated symmetric consistency directly into their model architecture.[45,59] As far as we know, no machine learning based method has used the transitive property or checked for transitive consistency even though the transitive (cycle closure) property is widely used in free energy calculations.[60−62] We present an interpretable neural-network-based method called self-consistent neural network protein stability prediction (SCONES) that incorporates both symmetric and transitive properties of $\Delta\Delta G$ into the architecture. The neural network predicts the contributions of a residue toward the protein's ΔG. ΔG contributions are predicted for the reference and mutant residues of a missense mutation independently, and $\Delta\Delta G$ is calculated by taking the difference between the predicted ΔG values. We assume that the mutations do not alter the structure significantly and use the same structure for both reference and mutant proteins. The architectural constraints for self-consistency help regularize models at the cost of ease of learning. We demonstrate through a holistic evaluation scheme that this self-consistent architecture is robust to overfitting and can explain a substantial portion of the variance in experimental data. Besides, we use an improved sign-insensitive metric to evaluate symmetric consistency and reveal previously hidden biases in existing methods. Toward the end, we highlight the benefits of interpretable models, the importance of using balanced data sets and discuss ways to overcome the data set limitations.

2. METHODS

Our method is a neural network that estimates a residue's contributions toward protein stability in its local structural environment. For a missense mutation that does not significantly alter the structure, the difference between the independently predicted contributions of the reference and mutant residues is reported as $\Delta\Delta G$. Section 2.1 describes the architecture in detail. The method is trained using data sets derived from FireProtDB[63] and evaluated on S350,[31] $\mathbf{S}^{\text{sym}}$,[45] and S768[26] test sets. We introduce a new data set called $\mathbf{S}^{\text{transitive}}$ that is the transitive closure of $\mathbf{S}^{\text{sym}}$. We use it to evaluate methods for transitive consistency. Section 2.2 elaborates on training set construction and evaluation of data sets. We use an ensemble of 50 models obtained from ten rounds of 5-fold cross-validation for evaluation. Section 2.3 describes the training procedure in detail.

2.1. Architecture. Our method is a single neural network that estimates a residue's contributions to the protein's ΔG in its local structural environment. We predict $\Delta\Delta G$ for a missense mutation by taking the difference between the predicted ΔG contributions of the reference and mutant residues. This architecture incorporates both symmetric and transitive properties of $\Delta\Delta G$. The consistency properties only hold when both reference and mutant structures are available. We assume that the mutations do not significantly alter the structure and use the same structure for reference and mutant

https://doi.org/10.1021/acs.jpcb.1c04913
J. Phys. Chem. B XXXX, XXX, XXX−XXX

proteins. We believe that the architectural constraints are still good inductive biases.

The local environment of a residue (referred to as the central residue henceforth) is defined to consist of all neighboring residues whose C_β atom is within 8 Å distance from the C_β atom of the central residue (Figure 1). The local

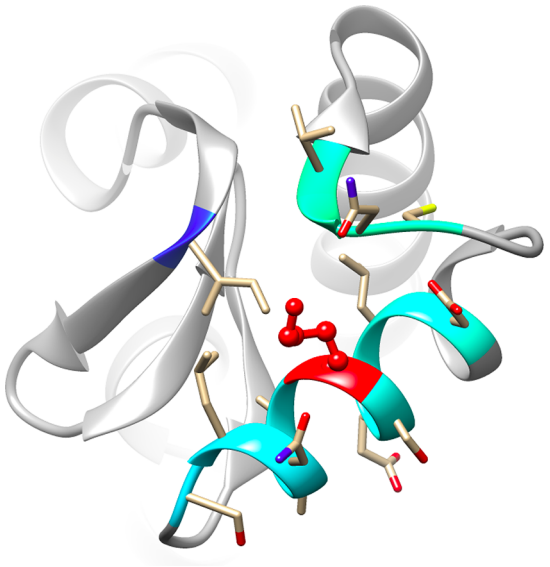


Figure 1. Residue local environment. The red-colored residue in ball and stick representation is the central residue. The highlighted residues with side-chain atoms shown with sticks are the neighboring residues.

environment can be thought of as a graph with residues as nodes and interactions as edges. Our neural network operates independently on each edge (residue−residue interaction) and estimates its contributions toward ΔG. The sum of all contributions of a residue's interactions with its neighbors is predicted as its total contribution toward protein stability (Figure 2 and Figure 3). The node features consist of a learnable amino-acid-embedding and solvent-accessible surface area. We initialize the embedding layer with amino acid properties (Table 1). The edge features consist of inter-residue distances and inter-residue orientations[57] (Figure 4). The C_β−C_β distance (referred to as d henceforth) of any two residues is the distance between the C_β atoms of the two residues. We use glycine's C_α atom in place of the C_β atom for computing d. d values raised to various powers and exponentials of d (which appear in theoretical formulations of pair potentials) are given as inter-residue distance features. The inter-residue orientation angles are defined in Figure 4. The sine and cosine of the angles are provided as inter-residue orientation features. Orientation features requiring glycine's C_β atom are set to zero. We use the reference protein's structure to compute the structure-based features for both reference and mutant proteins. This ensures that the same set of neighbors in the local environment is considered while calculating ΔG contributions of the central residue in the reference and mutant proteins. We do not use any features that involve both reference and mutant residues (such as the change in residue volume) to maintain the architecture's consistency properties. The two residues' node features and the edge features form the network's input features. See Figure S3 and Section 2 in the Supporting Information for detailed information on the architecture.

Predicting stability changes due to mutations that cause large structural disruptions and have distal effects is beyond the scope of this work; such mutations cause significant stability changes that are hard to predict. Therefore, our objective is to estimate the relative stabilities of proteins due to mutations that do not lead to significant structural changes. Methods must predict or use the lowest energy structures in computations to be fully symmetric and transitively consistent.[67] We assume that such corrections are small for mutations that do not cause significant structural alterations, as hinted by the good performance of sequence-only methods[28] and the low structural sensitivity of existing machine learning methods.[68] Therefore, we use the reference protein's structure to represent the mutant protein.

We hypothesize that all contributions to the ΔG of a protein can be accounted for by decomposing ΔG into a sum of inter-residue interactions and residue−solvent interactions. The interaction energies of residues structurally far away from the mutation site are assumed to change negligibly; their contributions to the overall ΔG are assumed equal in both reference and mutant structures and would cancel out when we calculate $\Delta\Delta G$. Hence, we only consider the local environment at the mutation site to estimate ΔG for the reference and mutant proteins. We also assume that most of the stability changes upon mutation can be captured by the change in interactions between the mutation site residue and the neighboring residues.

The self-consistency constraints on the architecture help regularize the models. A prediction for hypothetical reverse mutation would be equivalent to predicting and subtracting the ΔG estimates in the reverse order. This would yield an exactly negated prediction of $\Delta\Delta G$ of the forward mutation unless different structures are used. Regression loss functions generally are functions of the absolute difference between the predicted and target values. Therefore, losses for forward and corresponding reverse mutations are the same, and augmenting the training set with hypothetical reverse mutations is the same as duplicating the entire data set (unless different structures are used). Our architecture also captures the transitive property, but transitive samples can still aid in learning, unlike hypothetical reverse mutations. Losses for $\Delta\Delta G_{X\to Y}$, $\Delta\Delta G_{Y\to Z}$, and $\Delta\Delta G_{X\to Z}$ are different and not linear combinations of each other due to the nonlinearity of loss functions. Therefore, new transitive samples can give substantially different learning signals (Section 4.1 in Supporting Information). The network learns to predict ΔG indirectly from the $\Delta\Delta G$ reference data set. This makes learning difficult: the optimization process needs to learn to estimate the two ΔG values from the $\Delta\Delta G$ target and then further estimate individual contributions to each ΔG value (Figure 2). There are over ten neighbors in a typical sample; that would mean the optimization process needs to separate around 20 components of a sum from the $\Delta\Delta G$ target. We show that the model can still learn to predict protein stability changes without requiring explicit supervision at a finer level.

2.2. Data Set. FireProtDB[63] is a database of point mutations which have experimental thermostability data. We curated a new data set from FireProtDB according to the following criteria:

https://doi.org/10.1021/acs.jpcb.1c04913
J. Phys. Chem. B XXXX, XXX, XXX−XXX

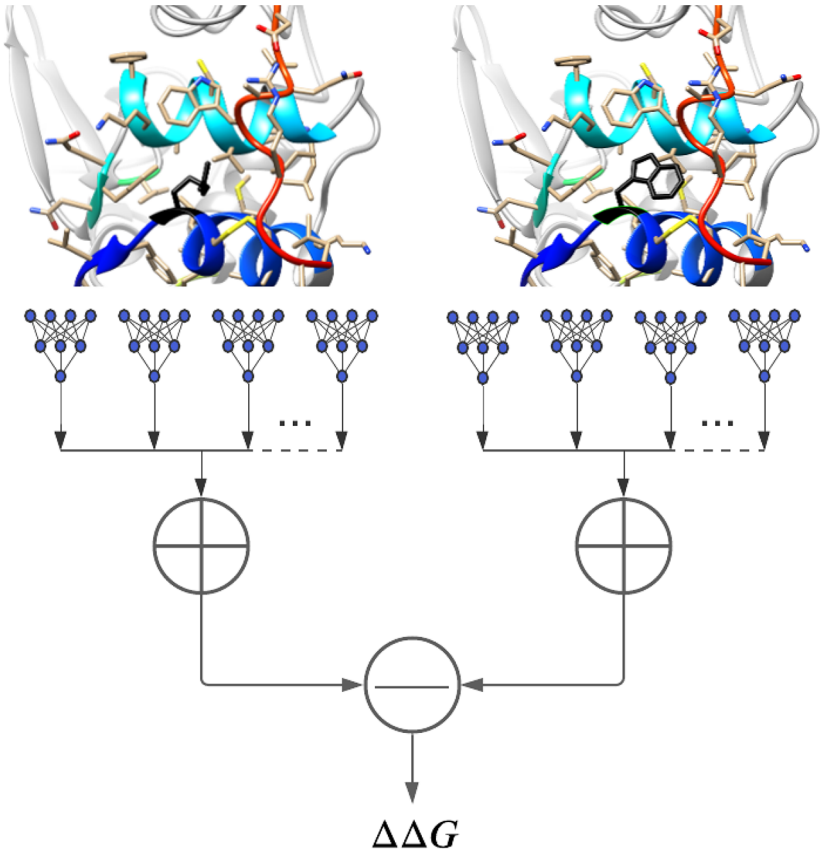


Figure 2. Self-consistent architecture. The neural network estimates a residue pair's contributions toward ΔG in the local environment. These estimates for all central-residue/neighbor-residue pairs are added to predict the central residue's total ΔG contribution. The same procedure is applied on both reference and mutant proteins to obtain two ΔG estimates. The difference between the two is the predicted $\Delta\Delta G$.

- Samples with structures where the mutated residue or its neighboring residues did not have N, C_α, and C_β atom coordinates (C_β atom coordinates are not required for glycine) were removed.
- Samples with known pH that lie outside 2 and 12 were removed. The samples with unknown pH were assumed to lie in the range under the assumption that the benefits of having more samples outweigh the cost of having a few outlier samples.
- Samples with $|\Delta\Delta G|$ greater than 8 kcal/mol were removed since such large stability changes suggest significant structural alterations.
- A sample is a duplicate of another if the reference and mutant proteins involved are identical in both samples and the absolute difference in the pH of the two samples is less than 0.1. $\Delta\Delta G$ values of duplicate samples were merged into one sample by averaging. The entire group of duplicates was discarded if the standard deviation of the $\Delta\Delta G$ values exceeded 0.5 kcal/mol.

The resulting curated data set contains 5989 samples.

The data available is scarce, and it may not be feasible to create a single test set that is balanced on all accounts. Such a hypothetical data set would be large and drastically decrease the size of the training set. Hence, we used multiple test sets, with each used to evaluate a different aspect of the method. We evaluated our method on S350,[31] $\mathbf{S}^{sym}$,[45] and S768[26] data sets. The S350 data set was generated by randomly sampling from a data set that reflects the characteristics of thermodynamic databases.[31] The distribution of $\Delta\Delta G$ measurements in this data set is heavily skewed toward destabilizing mutations (Figure 5). $\mathbf{S}^{sym}$ is a data set consisting of samples for which both reference and mutant structures are available. This test set is primarily used to evaluate the symmetric consistency of methods. The composition of S350 and $\mathbf{S}^{sym}$ test sets is not balanced across mutation types (Figure 6); both test sets are abundant in mutations involving hydrophobic residues like alanine and valine (Section 5 in Supporting Information) but lack sufficient samples in other categories. Therefore, we also evaluate on a category-wise balanced data set created by Frenz et al.[26] which we call S768.

Two training sets of varying difficulty were created for each test set from the curated data set by removing samples similar to the test samples. Two samples are similar if they satisfy all of the following conditions:

- The samples have the same mutation or the reverse mutation of each other.
- An 11-residue amino acid sequence centered around the mutating residue is extracted for both samples, and the sequence identity of the extracted sequences is greater than 50%.

The first training set was created based on the above similarity criteria. This training set contains samples at the same mutation site but with a different mutation as some samples in the test sets. The effects of different mutations in the same local environment may not be independent. There can be a mutation site where mutations always cause minor changes in protein stability irrespective of the mutation type. In such a situation, the model can learn to output a near-zero stability

https://doi.org/10.1021/acs.jpcb.1c04913
J. Phys. Chem. B XXXX, XXX, XXX−XXX

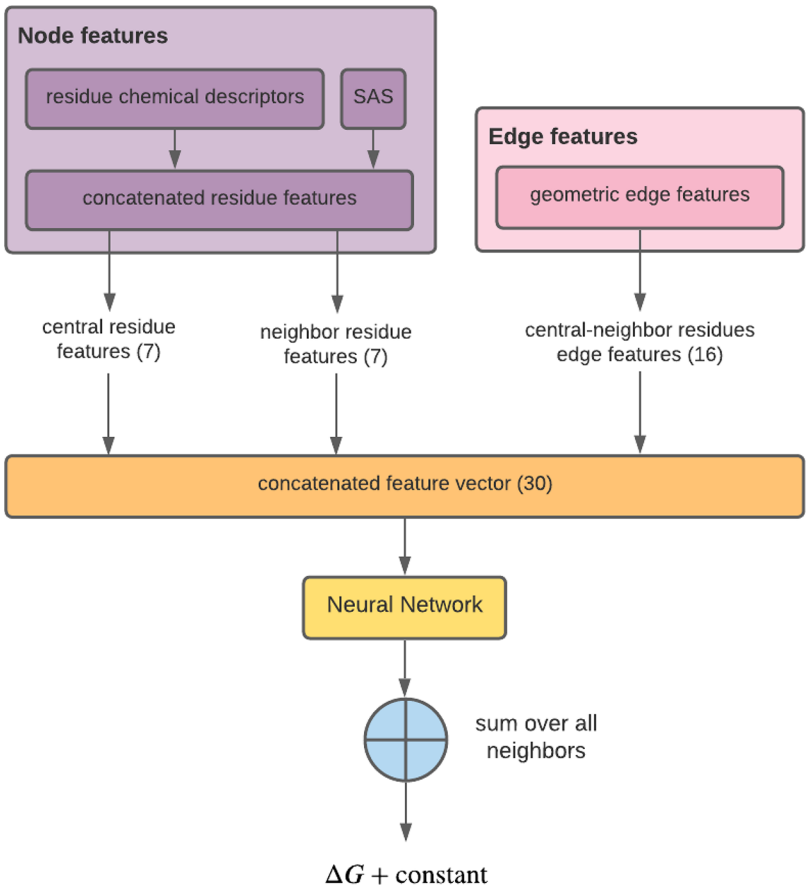


Figure 3. Overview of the model. The neural network estimates the contributions of a residue pair's interactions toward ΔG. The two residues' node features and edge features are concatenated to form the network's input feature vector. For a given local environment, feature vectors are constructed for all central-residue/neighbor-residue pairs. The network predictions for all the neighbors are then summed to predict the central residue's ΔG contributions. The constant factor contains contributions to the protein's ΔG that are not accounted for in our calculations. We assume such contributions to be identical in both reference and mutant proteins. It is fixed for a given protein and local environment and cancels out while calculating $\Delta\Delta G$. SAS stands for solvent-accessible surface area.

Table 1. List of Node Features

feature	description or AAindex[a]
formal charge[b]	Table S1 in Supporting Information
normalized van der Waals volume[b]	FAUJ880103
hydropathy index[b]	KYTJ820101
steric parameter[b]	CHAM810101
polarity[b]	GRAR740102
residue-accessible surface area in tripeptide[b]	CHOC760101
solvent-accessible surface area	calculated using DSSP[64,65]

[a]AAindex[66] is a database of numerical indices for physicochemical and biochemical properties of amino acids and pairs of amino acids. The code in the column is the AAindex database ID of the property. [b]These features are constants for a given amino acid. They are used to initialize the amino acid embedding layer.

change for that site irrespective of the mutation type. Test data can leak into the training set this way which would result in overestimation of performance. Therefore, we created a more challenging training set by filtering with the second condition alone. We refer to the first kind of training sets as "easy" training sets and the second kind as "hard" training sets. We omit "easy" in future references for the first kind of training sets, and hard training sets will have "hard" explicitly mentioned.

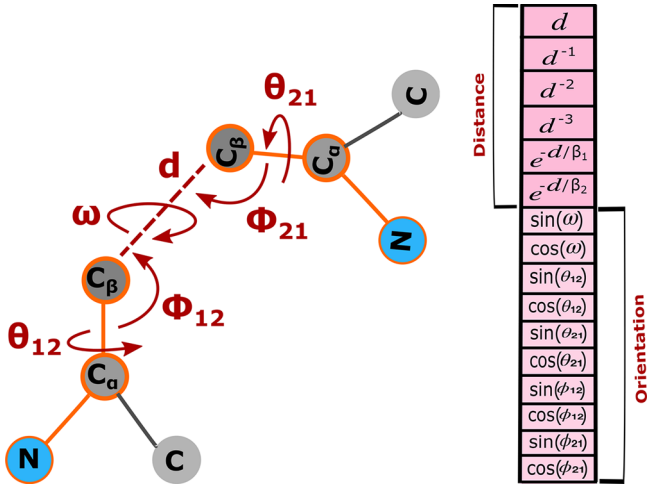
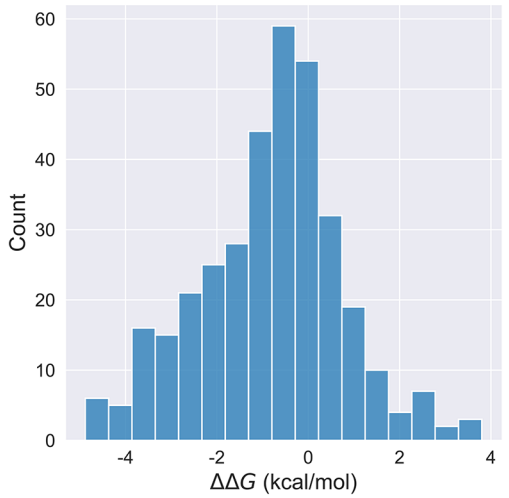
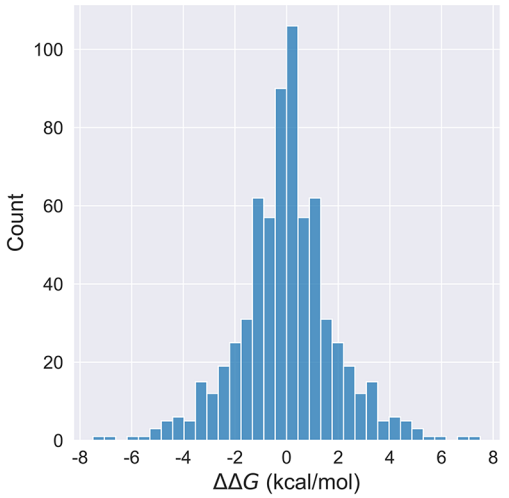


Figure 4. Visualization of geometric edge features between two residues. θ and ϕ depend on the order of the residues d and ω are the same for both residues irrespective of the residue order. The two planar angles (ϕ_{12} and ϕ_{21}), three dihedral angles (ω, θ_{12}, and θ_{21}), and d fully describe the relative positions of the backbone atoms.[57] All the distance-derived features are normalized to have zero mean and unit standard deviation. Note that these features capture the backbone conformation but not the side-chain conformation.

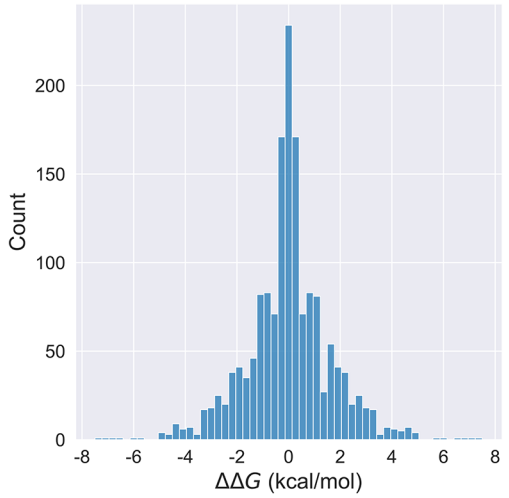
https://doi.org/10.1021/acs.jpcb.1c04913
J. Phys. Chem. B XXXX, XXX, XXX−XXX



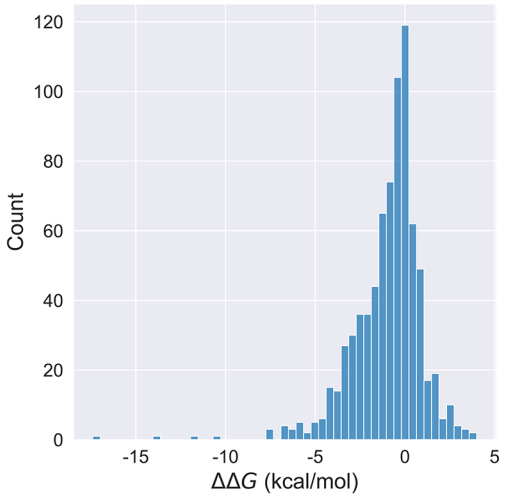
(a) Distribution of $\Delta\Delta G$ values in S350.



(b) Distribution of $\Delta\Delta G$ values in $\mathbf{S}^{\mathrm{sym}}$.



(c) Distribution of $\Delta\Delta G$ values in $\mathbf{S}^{\mathrm{transitive}}$.



(d) Distribution of $\Delta\Delta G$ values in S768.

Figure 5. Distribution of ΔΔG in the test sets. We consider samples as stabilizing if $\Delta\Delta G > 0.5$ kcal/mol, neutral if $|\Delta\Delta G| \leq 0.5$ kcal/mol, and destabilizing if $\Delta\Delta G < -0.5$ kcal/mol. S350 largely inherits the biases of the thermodynamic databases. The distribution is heavily skewed toward destabilizing mutations. Out of the 350 samples, 54 are stabilizing, 105 neutral, and 191 destabilizing. $\mathbf{S}^{\mathrm{sym}}$ consists of 342 pairs of forward and reverse mutations; this makes the distribution symmetric about zero. S768 is well balanced across mutation types by construction; however, it is biased toward destabilizing mutations.

The training set for $\mathbf{S}^{\mathrm{sym}}$ might contain transitive samples that can be obtained from $\mathbf{S}^{\mathrm{sym}}$ or samples that can be used to create transitive samples which are a part of $\mathbf{S}^{\mathrm{sym}}$. Hence, for an unbiased evaluation, we also evaluate the performance on the $\mathbf{S}^{\mathrm{sym}}$ data set after filtering the training set against the transitive closure of $\mathbf{S}^{\mathrm{sym}}$ (which we call $\mathbf{S}^{\mathrm{transitive}}$ in future references). The filtered training sets for both $\mathbf{S}^{\mathrm{sym}}$ and $\mathbf{S}^{\mathrm{transitive}}$ are almost the same. The hard training sets are identical, and the easy training sets are similar (Tables 2 and 3); this suggests that our filtering protocol is good. We also use the transitive pairs in $\mathbf{S}^{\mathrm{transitive}}$ to evaluate the transitive consistency of our method.

2.3. Training. Bioinformatics-related processing was carried out using BioPython.[71] The amino acid chemical descriptors were obtained from the AAindex database[66] using the quantiprot[72] python package. The neural network was trained using PyTorch,[73] and evaluation metrics were computed using scikit-learn.[74] PyTorch and NumPy[75] were used for general array processing.

The loss function (denoted by $\mathcal{L}$) consists of general adaptive robust loss function by Barron[76] for the ΔΔG prediction (denoted by $\mathcal{L}_{\text{adaptive robust loss}}$) and a symmetry loss term (denoted by $\mathcal{L}_{\text{symmetry loss}}$). The symmetry loss enforces the requirement that the interaction energy between two residues must be identical irrespective of the order of the residues in the feature vector. Feature vectors for both residue orders are generated during training, and the mean value is returned as the ΔG contribution (Figure S4 in Supporting Information). The $\mathcal{L}_{\text{symmetry loss}}$ ensures that the predictions from both feature vectors are equal.

$$\mathcal{L} = \mathcal{L}_{\text{adaptive robust loss}} + \alpha \mathcal{L}_{\text{symmetry loss}}$$

https://doi.org/10.1021/acs.jpcb.1c04913
J. Phys. Chem. B XXXX, XXX, XXX−XXX

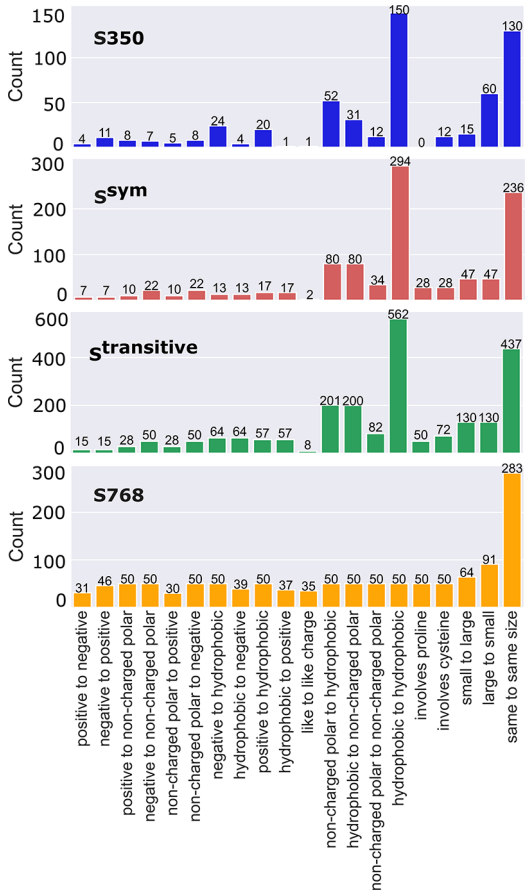


Figure 6. Composition of mutations in the test sets. We used the classification scheme proposed by Frenz et al.[26] to classify the mutations in the test sets. Mutations involving hydrophobic residues are over-represented in S350, $\mathbf{S}^{sym}$, and $\mathbf{S}^{transitive}$. S768 by construction tried to be as balanced as possible by incorporating 50 samples in each category; however, some categories that had an insufficient number of samples in the ProTherm[69] database have fewer than 50 samples.

$$\mathcal{L}_{\text{symmetry loss}} = \frac{1}{N}\sum_{i}\sum_{j}(E_{cj} - E_{jc})^2$$

where α is a scalar hyperparameter; c is the index of the central residue; j iterates over the indices of the neighboring residues; E_{cj} and E_{jc} are the energies predicted for residue c and j in the two possible orders; i iterates over the minibatch; and N is the minibatch size.

The Adam optimizer[77] was used with a weight decay of 0.004 and default parameters for the rest. The initial learning rate was set to 0.005 and decreased by a factor of 10 every time the learning plateaued (Section 2 in the Supporting Information for the detailed training procedure). The model was trained for 40 epochs with a minibatch size of 32. The embedding layer was initialized with the features listed in Table 1. We freeze the embedding layer weights for the first ten epochs. We do not unfreeze the weights after ten epochs while training with the S768 (hard) training set. The models were trained using 5-fold cross-validation with the training set. The model with the minimum validation loss was saved as the final model for each training instance. The final prediction is the ensemble average of predictions from the models generated in ten rounds of cross-validation (50 models in total). This process is repeated for each training set.

Table 3. List of Training Sets

data set[a]	size
unfiltered	5989
S350	5532
S350 (hard)	2735
$\mathbf{S}^{sym}$	5290
$\mathbf{S}^{sym}$ (hard)	2654
$\mathbf{S}^{transitive}$	5267
$\mathbf{S}^{transitive}$ (hard)	2654
S768	4857
S768 (hard)	1176

[a]The name indicates the test set based on which the training set was filtered. The filtering protocol is specified in parentheses for the "hard" protocol and omitted for the "easy" protocol.

3. RESULTS

We evaluated the performance of our method on S350, $\mathbf{S}^{sym}$, and S768 test sets. The characteristics of the three test sets have been visualized in Figure 5 and Figure 6. We created two separate training sets of varying difficulty for each test set after removing similar samples (Section 2.2). For each test set, the performance metrics were computed from the average ensemble predictions of all the models from ten rounds of cross-validation. Table 4 reports the performance of our method on all the test sets. Our method is able to learn and explain a substantial portion of the variance in the experimental

Table 2. List of Test Sets

data set	actual size[a]	filtered size[b]	description
S350	350	345	This test set was created by randomly sampling mutations from S2648.[31,70]
S87	87	86	This is a subset of S350 for which |ΔΔG| exceeds 2 kcal/mol.
$\mathbf{S}^{sym}$[c]	684	683	This test set contains 342 pairs of forward and reverse mutations. All the samples have experimentally determined reference structures.
$\mathbf{S}^{transitive}$[d]	1601	1596	This test set is the transitive closure of $\mathbf{S}^{sym}$. 917 transitive pairs can be formed from the samples. All the samples have a reference structure.
S768	768	745	This data set is a mutation category-wise balanced data set created to evaluate performance on different mutation types and identify potential biases in methods.

[a]The size of the originally published data set. [b]The size of the data set after removing samples that had incomplete backbone structural information for residues in the mutation site's local environment. [c]Note that the forward and reverse mutations are different data points for our method as they use different reference structures. [d]$\mathbf{S}^{transitive}$ is a new test set introduced in this work. The actual size indicates the number of samples after computing the transitive closure on $\mathbf{S}^{sym}$. The filtered size is the number of samples that have complete backbone structural information for the residues in the mutation site's local environment.

https://doi.org/10.1021/acs.jpcb.1c04913
J. Phys. Chem. B XXXX, XXX, XXX−XXX

Table 4. Performance on Different Test Sets[a]

					stabilization		destabilization	
training set	test set	PCC	MAE[b]	RMSE[b]	AUC[c]	PCC[d]	AUC[c]	PCC[d]
S350	S87	0.65	1.82	2.21	0.95	0.22	0.95	0.20
S350 (hard)	S87	0.67	1.89	2.23	0.95	0.17	0.95	0.23
S350	S350	0.55	1.13	1.53	0.80	0.63	0.73	0.43
S350 (hard)	S350	0.54	1.16	1.52	0.78	0.57	0.73	0.44
$\mathbf{S}^{sym}$	$\mathbf{S}^{sym}$	0.56	1.07	1.54	0.76	0.40	0.76	0.45
$\mathbf{S}^{transitive}$	$\mathbf{S}^{sym}$	0.56	1.07	1.54	0.76	0.40	0.77	0.45
$\mathbf{S}^{sym}$ (hard)	$\mathbf{S}^{sym}$	0.50	1.14	1.61	0.72	0.36	0.73	0.40
S768	S768	0.43	1.28	1.94	0.70	0.36	0.68	0.38
S768 (hard)	S768	0.37	1.37	2.03	0.60	0.10	0.64	0.37

[a]PCC stands for Pearson's correlation coefficient; AUC stands for area under ROC curve; RMSE stands for root mean squared error; and MAE stands for mean absolute error. [b]RMSE and MAE are in kcal/mol. [c]AUC is calculated for strongly affecting mutations only ($|\Delta\Delta G| \geq 1$ kcal/mol). [d]PCC is calculated using all the samples.

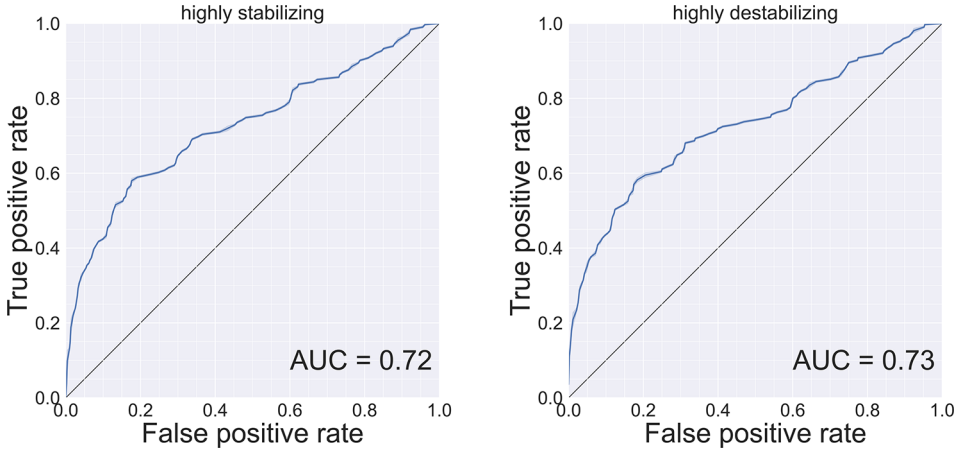


Figure 7. ROC plots for classifying strongly stabilizing and destabilizing mutations in $\mathbf{S}^{sym}$ with predictions obtained from the ensemble average of the models trained using $\mathbf{S}^{sym}$ (hard). Left: ROC plot for strongly stabilizing mutations ($\Delta\Delta G > 1$ kcal/mol). Right: ROC plot for strongly destabilizing mutations ($\Delta\Delta G < -1$ kcal/mol).

data. It shows balanced performance for both stabilizing and destabilizing mutations. It classifies significant mutations ($|\Delta\Delta G| \geq 1$ kcal/mol) fairly well with nearly all the samples in S87 classified correctly. The performance of the method decreases as we move from the smallest test set S350 to the largest test set S768.

Figure 7 shows the receiver operating characteristic (ROC) curve for strongly stabilizing and destabilizing mutations computed from ensemble predictions for the $\mathbf{S}^{sym}$ test set. The $\mathbf{S}^{sym}$ test set consists of 342 pairs of forward and reverse mutations. All the 684 samples have known experimental structures. Our method is symmetrically consistent by construction; however, the forward and reverse mutations are different data points for our method as they use different reference structures. The ROC curves are almost identical and hint that that our method is capable of identifying significantly stabilizing and destabilizing mutations equally well.

Table 5 compares the performance of our method on S350 with existing methods. Many methods that are biased toward predicting destabilizing mutations perform well on S350 but fare poorly on balanced test sets like $\mathbf{S}^{sym}$ (Table 6). Among the methods compared in both Table 5 and Table 6, only SCONES and PremPS show balanced performance on both S350 and $\mathbf{S}^{sym}$ test sets; all other methods show poor symmetric consistency with $\mathbf{S}^{sym}$.

Table 5. Comparison of Performance on S350[a]

	S350		S87	
method	PCC	RMSE[b]	PCC	RMSE[b]
SCONES[c]	0.55	1.53	0.65	2.21
SCONES (hard)[d]	0.54	1.52	0.67	2.23
PremPS[M25]	0.72	1.09	0.81	1.52
PremPS[P25]	0.58	1.28	0.60	1.94
mCSM[36]	0.73	1.08	0.82	1.48
MAESTRO[34]	0.70	1.13	0.76	1.67
PoPMuSiC v2.0[31]	0.67	1.16	0.71	1.67
PoPMUSiC v1.0[78]	0.62	1.24	0.70	1.66
SDM2[49]	0.61	1.29	0.69	1.71
SDM[35]	0.52	1.80	0.63	2.11
Dmutant	0.48	1.81	0.57	2.31
AUTOMUTE[79]	0.46	1.43	0.45	1.99
CUPSAT[80]	0.37	1.91	0.50	2.14

[a]Our method only accepted 345 samples out of 350. Performance data were taken from refs 25, 31, 34, 36, and 49. The training sets of all methods did not have samples from S350. PremPS[P] did not have homologous proteins in the training set. [b]RMSE is in kcal/mol. [c]Trained using the S350 (easy) training set. [d]Trained using the S350 (hard) training set.

The prediction for forward mutations and their corresponding reverse mutations in identical experimental conditions

https://doi.org/10.1021/acs.jpcb.1c04913
J. Phys. Chem. B XXXX, XXX, XXX−XXX

Table 6. Comparison of Performance on S^{sym}[a]

	forward		reverse				
method	PCC	RMSE[b]	PCC	RMSE[b]	R_{FR}	$\langle\delta\rangle$[b]	Δ_S[b]
SCONES[c]	0.51	1.53	0.49	1.55	−0.97	0.04	0.24
SCONES ($S^{transitive}$)[d]	0.51	1.53	0.48	1.55	−0.98	0.04	0.23
SCONES (hard)[e]	0.45	1.61	0.43	1.61	−0.99	0.01	0.17
$PremPS^{M}$[25]	0.64	1.21	0.56	1.30	−0.91	0.03	0.34
$PremPS^{P}$[25]	0.56	1.32	0.50	1.37	−0.89	0.04	0.33
ThermoNet*[27]	0.58	1.38	0.59	1.38	−0.95	−0.05	-
ThermoNet[27]	0.47	1.56	0.47	1.55	−0.96	−0.01	0.23
$PoPMuSiC^{sym}$[45]	0.48	1.58	0.48	1.62	−0.77	0.03	-
PremPS[25]	0.81	0.96	0.74	1.12	−0.93	0.03	0.38
DDGun3D[28]	0.56	1.42	0.53	1.46	−0.99	−0.02	0.14
INPS[32]	0.51	1.42	0.50	1.44	−0.99	−0.04	-
DDGun[28]	0.48	1.47	0.48	1.50	−0.99	−0.01	0.04
INPS3D[33]	0.59	1.29	0.44	1.64	−0.86	−0.55	0.70
Rosetta[81]	0.69	2.31	0.43	2.61	−0.41	−0.69	-
FoldX[30]	0.63	1.56	0.39	2.13	−0.38	−0.47	-
MAESTRO[34]	0.52	1.36	0.32	2.09	−0.34	−0.58	1.44
SDM[35]	0.51	1.74	0.32	2.28	−0.75	−0.32	-
PoPMuSiC v2.1[29]	0.63	1.21	0.25	2.18	−0.29	−0.71	-
mCSM[36]	0.61	1.23	0.14	2.43	−0.26	−0.91	2.08
DUET[37]	0.63	1.20	0.13	2.38	−0.21	−0.84	1.91
MUPRO[82]	0.79	0.94	0.07	2.51	−0.02	−0.97	-

[a]SCONES, $PremPS^{M}$ and $PremPS^{P}$, ThermoNet, ThermoNet*, and $PoPMuSiC^{sym}$ did not have overlapping samples in their training set. ThermoNet and $PremPS^{P}$ also removed homologous proteins from their training set. All other methods were evaluated directly on the test set. Performance data were taken from Chen et al., Li et al., and Pucci et al.[25,27,45] [b]RMSE, $\langle\delta\rangle$, and Δ are in kcal/mol. [c]Trained using the S^{sym} (easy) training set. [d]Trained using the $S^{transitive}$ (easy) training set. [e]Trained using the S^{sym} (hard) training set.

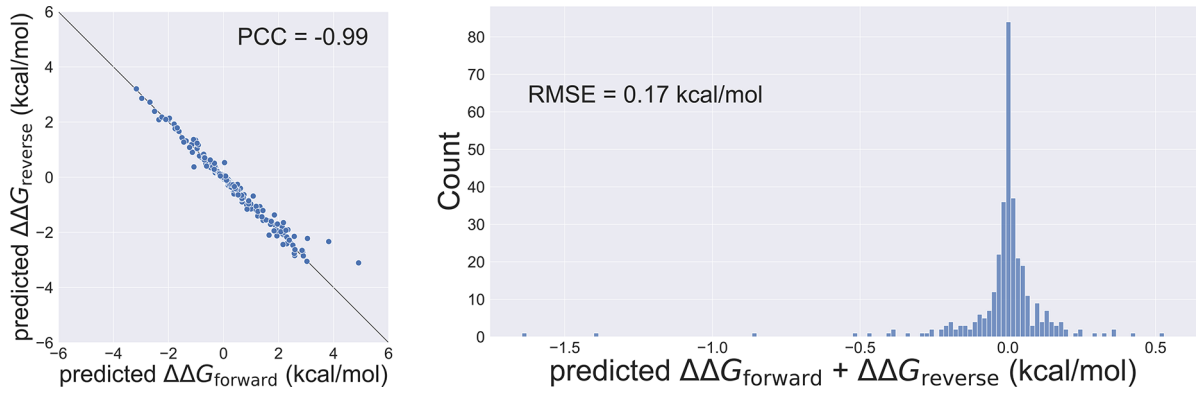


Figure 8. Mean predictions of an ensemble consisting of all the models from all the rounds of cross-validation trained using the S^{sym} (hard) training set are used. Left: plot of predicted values of $\Delta\Delta G$ for forward and reverse mutations. Right: distribution of prediction bias ($\Delta\Delta G_{forward}$ + $\Delta\Delta G_{reverse}$).

must be strongly negatively correlated. The S^{sym} data set consists of mutations for which both reference and mutant structures are available and is used to evaluate the symmetric consistency in methods. The symmetric consistency has been traditionally measured using a correlation coefficient R_{FR} and mean prediction bias $\langle\delta\rangle$.[46] R_{FR} is the correlation coefficient between the predictions for forward mutations and their corresponding reverse mutations. The δ for a sample is defined as $\Delta\Delta G_{forward}$ + $\Delta\Delta G_{reverse}$, and the average δ over all samples (denoted by $\langle\delta\rangle$) is used as a measure of overall consistency of the method. However, this sign-sensitive metric is prone to hiding symmetric biases in both directions: negative errors can cancel out positive errors.[27] It effectively measures the asymmetry in the bias but does not accurately inform about the presence of any bias. Therefore, we computed the root mean squared error of the prediction biases ($\sqrt{\langle\delta^2\rangle}$, denoted by Δ_S). A method with perfect symmetric consistency will have an R_{FR} value of the negative one, $\langle\delta\rangle$ value of zero, and Δ_S value of zero. Table 6 compares the performance of our method on S^{sym} with existing methods. Figure 8 plots our predictions for forward and reverse mutations and the distribution of the prediction bias. Table 6 and Figure 8 show that our method has has a near-perfect R_{FR} value and a low Δ_S value. We computed the Δ_S values for many existing methods and report them in Table 6. This new metric reveals a previously hidden bias in existing methods that had a near-perfect $\langle\delta\rangle$.

Table 7 presents transitive consistency metrics computed using $S^{transitive}$. $S^{transitive}$ was created by computing the transitive closure of S^{sym}. All the samples in $S^{transitive}$ have experimentally

https://doi.org/10.1021/acs.jpcb.1c04913
J. Phys. Chem. B XXXX, XXX, XXX−XXX

Table 7. Evaluating Transitive Consistency on $S^{transitive}$

method	R_T	Δ_T[a]
SCONES (S^{sym})	0.99	0.16
SCONES ($S^{transitive}$)	0.99	0.17
SCONES (S^{sym} hard)	0.99	0.13
DDGun3D[28]	0.99	0.13
ACDCNN[51] (with ref 3D structure)	0.94	0.23
ACDCNN[51] (sequence only)	0.93	0.24
MAESTRO[34]	0.90	0.58
INPS3D[33]	0.89	0.39
DUET[37]	0.82	0.89
mCSM[36]	0.81	1.02
SDM2[49]	0.78	0.86
PremPS[25]	0.77	0.54
DynaMut2[38]	0.73	1.05

[a] Δ_T is in kcal/mol.

determined structures. There are 917 transitive pairs among the 1601 samples in $S^{transitive}$. We introduce two new metrics to evaluate transitive consistency: correlation coefficient R_T and norm Δ_T. R_T is the correlation between $\Delta\Delta G_{X\to Y} + \Delta\Delta G_{Y\to Z}$ and $\Delta\Delta G_{X\to Z}$, and Δ_T is the root mean squared error of $\Delta\Delta G_{X\to Y} + \Delta\Delta G_{Y\to Z} - \Delta\Delta G_{X\to Z}$. A method with perfect transitive consistency must have an R_T of one and Δ_T of zero. Table 7 and Table 6 suggest that methods that have good symmetric consistency also tend to be transitively consistent. However, the converse is not true. Our method shows transitive consistency that is comparable to or better than the current state of the art methods. Figure 9 plots our predictions for the transitive tuples in $S^{transitive}$ and the distribution of the prediction bias. Table 7 and Figure 9 show that our method has very high transitive consistency with a near-perfect value of R_T and a low Δ_T.

Figure 10 shows the variation in performance of the trained models in our ensembles. We note that there is significant variance. The average performance on a test set, however, does not vary significantly across different training sets (Table 4).

The nature of computations in our method is hierarchical. At the lowest level of the hierarchy, our neural network predicts ΔG contributions of individual residue−residue interactions. These predictions can provide valuable insights into causes of stabilization or destabilization. An interaction map can summarize these intermediate predictions. Such interaction maps have been previously used to chemically interpret solvation free energies of druglike molecules in organic solvents.[52,83] Here, the interaction map consists of two columns representing the reference and mutant residues and several rows representing neighboring residues. Each cell contains the ΔG contributions predicted for the interactions between the central residue (column) and the neighboring residue (row). We used the ensemble average of the intermediate predictions to plot the interaction map in Figure 11.

4. DISCUSSION

The data in thermodynamic databases for this task are biased and limited. The distribution of $\Delta\Delta G$ experimental values is skewed toward destabilizing mutations (Figure S1 in Supporting Information). The available data are not balanced across all mutation categories (Figure S2 in Supporting Information). Some mutations do not have any data at all (Figure S3 in Supporting Information). These biases pose a severe problem for machine learning based methods. Many methods until recently were trained on training sets directly derived from thermodynamic databases without accounting for the biases. These methods were shown to lack symmetric consistency by subsequent studies.[43−46,58,67] Recent methods augment data sets with hypothetical reverse mutations[25,27,38] or incorporate the symmetry into their architecture.[45,59] Few newer methods still show significant variance in performance across different training sets (Table 6) which may indicate overfitting. We proposed utilizing the transitive property to evaluate methods and improve performance by constraining methods to be transitively consistent. With that goal, we designed a method that incorporates both symmetric and transitive properties into the architecture.

Our method consists of a single neural network that estimates inter-residue interaction energies. We calculate a residue's contribution toward protein stability by summing up all the interaction energies with its neighbors. $\Delta\Delta G$ is predicted by taking the difference between the predicted energies of the reference and mutant residues. The network sees features for one residue−residue interaction at a time during training. We do not use any features that span more than two residues or involve both reference and mutant proteins. This, along with the independence in calculations of each ΔG contribution, makes it very robust toward overfitting. Moreover, nearly all of the 400 possible residue−residue pairs have hundreds of samples in the training set (Figure S9 in the Supporting Information). The network has seen all possible

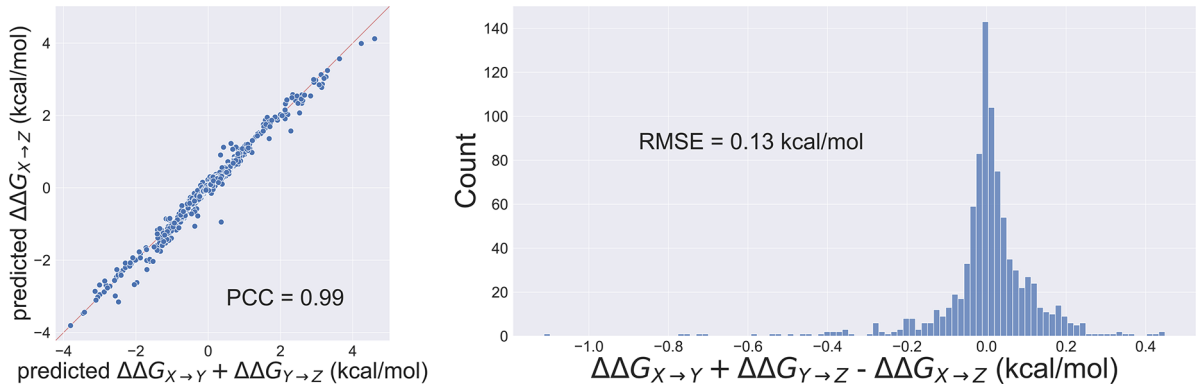


Figure 9. Mean predictions of an ensemble consisting of all the models from all the rounds of cross-validation trained with the S^{sym} (hard) training set is used. Left: plot of $\Delta\Delta G_{X\to Y} + \Delta\Delta G_{Y\to Z}$ vs $\Delta\Delta G_{X\to Z}$. Right: distribution of prediction bias ($\Delta\Delta G_{X\to Y} + \Delta\Delta G_{Y\to Z} - \Delta\Delta G_{X\to Z}$).

https://doi.org/10.1021/acs.jpcb.1c04913
J. Phys. Chem. B XXXX, XXX, XXX−XXX

Figure 10. Variance in model performance. Pearson correlation coefficient (PCC) was computed for each of the 50 models from ten rounds of cross-validation.

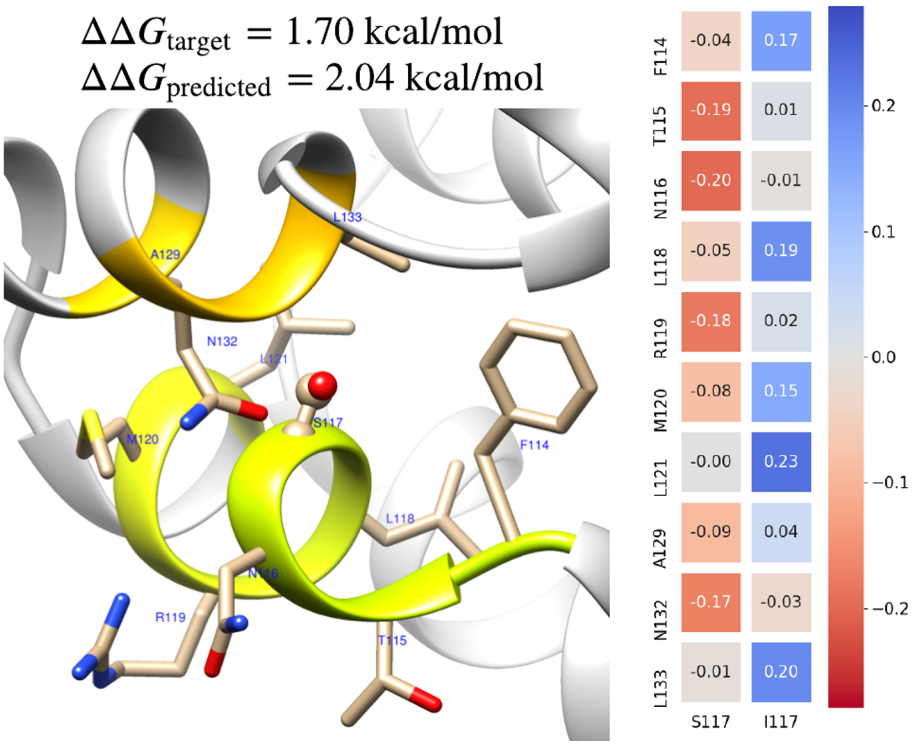


Figure 11. Interaction map for serine to isoleucine mutation in bacteriophage T4 lysozyme (pdb id: 2lzm). Isoleucine provides new beneficial hydrophobic interactions that lead to improved protein stability. The ensemble average of the intermediate predictions from the models trained using the S768 (hard) data set was used to plot the interaction map.

inter-residue interactions during training and therefore will not be befuddled by completely new samples at test time. It may generalize to mutations that do not occur or have very few samples in the training set.

Despite the tight constraints on the architecture, we showed that our method could capture a substantial portion of the variance in experimental data (Table 4). Our method provides balanced performance for both stabilizing and destabilizing mutations across all test sets (Table 4 and Figure 7). The variation in performance across test sets can be attributed to their vastly different compositions and sizes. In S350, one-third of the mutations involve alanine, and about half of the mutations involve alanine or valine. S^{sym} and S768 also suffer from similar biases but to a lesser extent (Section 5 in Supporting Information). Furthermore, the larger test sets lead to smaller filtered training sets as more samples overlap. The overall performance numbers in Table 4 correlate with the test data set size. Our method performs poorly while training with S768 (hard) if we do not freeze the embedding layer (Section 8 in Supporting Information). Freezing the weights of the embedding layer reduces the size of the hypothesis space of the network. The improved performance with the smaller hypothesis space case suggests that the training data set is small enough that additional flexibility offered by an embedding layer leads to worse performance.

To more accurately determine the predictive power of methods, we performed a holistic evaluation using multiple test sets to estimate a model's performance reliably. We checked

https://doi.org/10.1021/acs.jpcb.1c04913
J. Phys. Chem. B XXXX, XXX, XXX−XXX

for symmetric consistency, transitive consistency, and performance across different mutation categories. Many older methods show poor symmetric consistency.[43−46,58,67] The poor performance is largely due to the methods inheriting the biases of unbalanced training sets that contained far more destabilizing mutations than stabilizing mutations. Our method shows excellent symmetric and transitive consistency that is comparable to the state of the art methods (Table 6 and Table 7). We investigated the performance of our method on S768 in different mutation categories (Section 6 in Supporting Information). Our method performs poorly on surface mutations and mutations involving proline. We believe that the mutations involving proline tend to unusually alter the structure and break our assumptions. Existing literature has noted that predicting stability changes on surface mutations is generally more difficult than buried mutations.[25,29] The performance of surface mutations may also be due to our method not explicitly modeling residue−solvent interactions. Our method also performs badly on small-to-large mutations. Small-to-large mutations often cause steric clashes that lead to structural rearrangements.[84] Existing methods have dealt with this issue by modeling the mutant structures.[25,27,30,81] We did not perform any such procedure.

The poor performance of models trained on the S768 (hard) training set with learnable embeddings demonstrates the importance of using a balanced validation set. We conducted an experiment where we evaluated the performance of the models saved after every epoch on the S768 test set. We observed that the model achieves near peak performance just before epoch 11 and crashes soon after the embedding layer weights are unfreezed. In principle, a good validation set should save a model from the first ten epochs as the best model. Our validation sets generated by a random split from the training set choose a model that performed nearly 25% worse on the test set. This shows that validation sets that are not balanced across mutation categories fail to choose a good model.

The models trained with the hard training set show better consistency properties but lower prediction performance than those trained with the easy training set. This may indicate a potential trade-off between consistency and prediction performance. The easy unbalanced training sets may lead to overfitting, resulting in high prediction performance but poor consistency properties. Conversely, a method that always outputs a constant zero will have perfect consistency properties but would not give helpful predictions. Therefore, it is essential to conduct a holistic evaluation using many test sets and evaluation metrics.

Figure 10 indicates a fair amount of variance in the performance of the predictors in the ensemble. Our highly constrained architecture is challenging to train directly from limited data. It is also sensitive to weight initialization. We believe that improved sampling procedures for training sets, transfer learning, and auxiliary losses[85−87] can reduce the intra-ensemble variance. Our architecture is hierarchical and similar to force field computations. This allows easy integration of auxiliary losses from force fields or statistical potentials to guide the optimization at a finer level. The variance of the ensemble performance across different training data sets is low (Table 4) (Section 9 in Supporting Information). This shows that our method as a whole is relatively robust to overfitting.

The architecture of our method provides the ability to peak into the intermediate computations that lead to the final prediction. Explainable machine learning methods can provide valuable insights into the underlying mechanisms behind the predictive power of an otherwise black-box model. It also allows the method to be verified by human chemical intuition. For example, the interaction map in Figure 11 suggests that the mutation leads to better interactions with hydrophobic residues, which result in extra stabilization. This agrees with our intuition that hydrophobic residues like to be with other hydrophobic residues. We believe that such verification is vital given that the training data are limited and biased, and models have been shown to overfit easily.

5. CONCLUSION

Protein engineering is a rapidly growing field with vast potential for biotechnological and biomedical applications. Protein stability is an important property that has to be tuned in protein engineering. Additionally, optimization of other properties of proteins should be done such that the proteins are still stable. Many computational methods have been developed to identify potential stability-altering mutations. The amount of data available to train protein stability predictors upon mutation is small. Incorporating domain knowledge directly into the method can help overcome data set limitations. Commonly used regression targets confer many properties which have not been fully exploited; the transitive property has been feebly used in existing methods. Here, we have proposed improving performance by incorporating the transitive property into model architectures and augmenting training sets with transitive samples. We have presented a method that incorporates both symmetric and transitive properties of $\Delta\Delta G$ into the architecture. It consists of a neural network that predicts a residue's contributions toward the protein's ΔG. The difference in the independently predicted ΔG contributions for the reference and mutant residues in a missense mutation is returned as $\Delta\Delta G$. This method relies less on the data to learn self-consistency properties and is immune to many common problems arising from unbalanced and undersampled data sets. Our method is simple but belongs to a class of methods that has not been explored thoroughly. The method can be extended to use more complex architectures such as graph neural networks and incorporate more domain knowledge using the method's hierarchical architecture. We hope that further exploration in this direction and the use of ideas presented in this work will help improve protein stability predictors.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jpcb.1c04913.

Biases in data sets, detailed neural network architecture, training procedure, formal charge table, statistics on transitive data augmentation, the composition of test sets, mutation category-wise performance statistics on S768, matrix of residue−residue interaction counts, and ablation studies (PDF)

■ AUTHOR INFORMATION

Corresponding Author

U. Deva Priyakumar − *Center for Computational Natural Sciences and Bioinformatics, International Institute of*

https://doi.org/10.1021/acs.jpcb.1c04913
J. Phys. Chem. B XXXX, XXX, XXX−XXX

Information Technology, Hyderabad 500 032, India; orcid.org/0000-0001-7114-3955; Email: deva@iiit.ac.in

Authors

Yashas B. L. Samaga − *Center for Computational Natural Sciences and Bioinformatics, International Institute of Information Technology, Hyderabad 500 032, India*

Shampa Raghunathan − *Center for Computational Natural Sciences and Bioinformatics, International Institute of Information Technology, Hyderabad 500 032, India*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jpcb.1c04913

Notes

The authors declare no competing financial interest.

We have published our documented code and data sets on GitHub: https://github.com/devalab/SCONES.

■ ACKNOWLEDGMENTS

We thank IHub-Data, IIIT Hyderabad, and DST/WOS-A (grant no. SR/WOS-A/CS-19/2018 (G)) for the financial support. We acknowledge Devdarshni Priyakumar for useful suggestions. We thank Gadila Shashank Reddy for providing the template code for web scraping.

■ REFERENCES

(1) Serpente, N. Beyond a pedagogical tool: 30 years of Molecular Biology of the Cell. *Nat. Rev. Mol. Cell Biol.* **2013**, *14*, 120−125.

(2) Brannigan, J. A.; Wilkinson, A. J. Protein engineering 20 years on. *Nat. Rev. Mol. Cell Biol.* **2002**, *3*, 964−970.

(3) Shaw, W. V. Protein engineering. The design, synthesis and characterization of factitious proteins. *Biochem. J.* **1987**, *246*, 1−17.

(4) Zanghellini, A. de novo computational enzyme design. *Curr. Opin. Biotechnol.* **2014**, *29*, 132−138. Cell and Pathway Engineering.

(5) Erb, T. J.; Jones, P. R.; Bar-Even, A. Synthetic metabolism: metabolic engineering meets enzyme design. *Curr. Opin. Chem. Biol.* **2017**, *37*, 56−62. Biocatalysis and biotransformation, Bioinorganic Chemistry.

(6) Koellhoffer, J. F.; Higgins, C. D.; Lai, J. R. Protein engineering strategies for the development of viral vaccines and immunotherapeutics. *FEBS Lett.* **2014**, *588*, 298−307.

(7) Austin, H. P.; et al. Characterization and engineering of a plastic-degrading aromatic polyesterase. *Proc. Natl. Acad. Sci. U. S. A.* **2018**, *115*, E4350−E4357.

(8) Poluri, K. M.; Gulati, K. *Protein Engineering Techniques: Gateways to Synthetic Protein Universe*; Springer Singapore: Singapore, 2017; pp 103−134.

(9) Griss, R.; Schena, A.; Reymond, L.; Patiny, L.; Werner, D.; Tinberg, C. E.; Baker, D.; Johnsson, K. Bioluminescent sensor proteins for point-of-care therapeutic drug monitoring. *Nat. Chem. Biol.* **2014**, *10*, 598−603.

(10) Zhou, L.; Bosscher, M.; Zhang, C.; Özçubukçu, S.; Zhang, L.; Zhang, W.; Li, C. J.; Liu, J.; Jensen, M. P.; Lai, L.; et al. A protein engineered to bind uranyl selectively and with femtomolar affinity. *Nat. Chem.* **2014**, *6*, 236−241.

(11) Raghunathan, G.; Sokalingam, S.; Soundrarajan, N.; Madan, B.; Munussami, G.; Lee, S.-G. Modulation of protein stability and aggregation properties by surface charge engineering. *Mol. BioSyst.* **2013**, *9*, 2379−2389.

(12) Coelho, M. B.; Ascher, D. B.; Gooding, C.; Lang, E.; Maude, H.; Turner, D.; Llorian, M.; Pires, D. E.; Attig, J.; Smith, C. W. Functional interactions between polypyrimidine tract binding protein and PRI peptide ligand containing proteins. *Biochem. Soc. Trans.* **2016**, *44*, 1058−1065.

(13) Siedhoff, N. E.; Schwaneberg, U.; Davari, M. D. In *Enzyme Engineering and Evolution: General Methods*; Tawfik, D. S., Ed.; Methods in Enzymology; Academic Press, 2020; Vol. *643*; pp 281−315.

(14) Basheer, S. M.; Chellappan, S. *Bioresources and bioprocess in biotechnology*; Springer, 2017; pp 151−168.

(15) Yang, H.; Liu, L.; Li, J.; Chen, J.; Du, G. Rational Design to Improve Protein Thermostability: Recent Advances and Prospects. *ChemBioEng Rev.* **2015**, *2*, 87−94.

(16) van den Berg, B. A.; Reinders, M. J.; van der Laan, J.-M.; Roubos, J. A.; de Ridder, D. Protein redesign by learning from data. *Protein Eng., Des. Sel.* **2014**, *27*, 281−288.

(17) Newton, M. S.; Arcus, V. L.; Gerth, M. L.; Patrick, W. M. Enzyme evolution: innovation is easy, optimization is complicated. *Curr. Opin. Struct. Biol.* **2018**, *48*, 110−116. Folding and binding in silico, in vitro, and in cellular Proteins: An Evolutionary Perspective.

(18) Robertson, A. D.; Murphy, K. P. Protein Structure and the Energetics of Protein Stability. *Chem. Rev.* **1997**, *97*, 1251−1268.

(19) Modarres, H. P.; Mofrad, M. R.; Sanati-Nezhad, A. Protein thermostability engineering. *RSC Adv.* **2016**, *6*, 115252−115270.

(20) Goldenzweig, A.; Fleishman, S. J. Principles of Protein Stability and Their Application in Computational Design. *Annu. Rev. Biochem.* **2018**, *87*, 105−129.

(21) Tokuriki, N.; Stricher, F.; Serrano, L.; Tawfik, D. S. How Protein Stability and New Functions Trade Off. *PLoS Comput. Biol.* **2008**, *4*, 1−7.

(22) Wang, Y.; Yu, X.; Zhao, H. Biosystems design by directed evolution. *AIChE J.* **2020**, *66*, e16716.

(23) Gill, M.; McCully, M. E. Molecular dynamics simulations suggest stabilizing mutations in a de novo designed α/β protein. *Protein Eng., Des. Sel.* **2019**, *32*, 317−329.

(24) Chen, Q.; Xiao, Y.; Shakhnovich, E. I.; Zhang, W.; Mu, W. Semi-rational design and molecular dynamics simulations study of the thermostability enhancement of cellobiose 2-epimerases. *Int. J. Biol. Macromol.* **2020**, *154*, 1356−1365.

(25) Chen, Y.; Lu, H.; Zhang, N.; Zhu, Z.; Wang, S.; Li, M. PremPS: Predicting the impact of missense mutations on protein stability. *PLoS Comput. Biol.* **2020**, *16*, 1−22.

(26) Frenz, B.; Lewis, S. M.; King, I.; DiMaio, F.; Park, H.; Song, Y. Prediction of Protein Mutational Free Energy: Benchmark and Sampling Improvements Increase Classification Accuracy. *Front. Bioeng. Biotechnol.* **2020**, *8*, 1175.

(27) Li, B.; Yang, Y. T.; Capra, J. A.; Gerstein, M. B. Predicting changes in protein thermodynamic stability upon point mutation with deep 3D convolutional neural networks. *PLoS Comput. Biol.* **2020**, *16*, e1008291−e1008291.

(28) Montanucci, L.; Capriotti, E.; Frank, Y.; Ben-Tal, N.; Fariselli, P. DDGun: an untrained method for the prediction of protein stability changes upon single and multiple point variations. *BMC Bioinf.* **2019**, *20*, 335.

(29) Dehouck, Y.; Kwasigroch, J. M.; Gilis, D.; Rooman, M. PoPMuSiC 2.1: a web server for the estimation of protein stability changes upon mutation and sequence optimality. *BMC Bioinf.* **2011**, *12*, 151.

(30) Schymkowitz, J.; Borg, J.; Stricher, F.; Nys, R.; Rousseau, F.; Serrano, L. The FoldX web server: an online force field. *Nucleic Acids Res.* **2005**, *33*, W382−W388.

(31) Dehouck, Y.; Grosfils, A.; Folch, B.; Gilis, D.; Bogaerts, P.; Rooman, M. Fast and accurate predictions of protein stability changes upon mutations using statistical potentials and neural networks: PoPMuSiC-2.0. *Bioinformatics* **2009**, *25*, 2537−2543.

(32) Fariselli, P.; Martelli, P. L.; Savojardo, C.; Casadio, R. INPS: predicting the impact of non-synonymous variations on protein stability from sequence. *Bioinformatics* **2015**, *31*, 2816−2821.

(33) Savojardo, C.; Fariselli, P.; Martelli, P. L.; Casadio, R. INPS-MD: a web server to predict stability of protein variants from sequence and structure. *Bioinformatics* **2016**, *32*, 2542−2544.

(34) Laimer, J.; Hofer, H.; Fritz, M.; Wegenkittl, S.; Lackner, P. MAESTRO - multi agent stability prediction upon point mutations. *BMC Bioinf.* **2015**, *16*, 116.

(35) Worth, C. L.; Preissner, R.; Blundell, T. L. SDM-a server for predicting effects of mutations on protein stability and malfunction. *Nucleic Acids Res.* **2011**, *39*, W215−W222.
(36) Pires, D. E. V.; Ascher, D. B.; Blundell, T. L. mCSM: predicting the effects of mutations in proteins using graph-based signatures. *Bioinformatics* **2014**, *30*, 335−342.
(37) Pires, D. E.; Ascher, D. B.; Blundell, T. L. DUET: a server for predicting effects of mutations on protein stability using an integrated computational approach. *Nucleic Acids Res.* **2014**, *42*, W314−W319.
(38) Rodrigues, C. H.; Pires, D. E.; Ascher, D. B. DynaMut2: Assessing changes in stability and flexibility upon single and multiple point missense mutations. *Protein Sci.* **2021**, *30*, 60−69.
(39) Kulandaisamy, A.; Zaucha, J.; Frishman, D.; Gromiha, M. M. MPTherm-pred: Analysis and Prediction of Thermal Stability Changes upon Mutations in Transmembrane Proteins. *J. Mol. Biol.* **2021**, *433*, 166646.
(40) Huang, P.; Chu, S. K. S.; Frizzo, H. N.; Connolly, M. P.; Caster, R. W.; Siegel, J. B. Evaluating Protein Engineering Thermostability Prediction Tools Using an Independently Generated Dataset. *ACS Omega* **2020**, *5*, 6487−6493.
(41) Wales, D. *Energy landscapes: Applications to clusters, biomolecules and glasses*; Cambridge University Press, 2003.
(42) Raghunathan, S.; Jaganade, T.; Priyakumar, U. D. Urea-aromatic interactions in biology. *Biophys. Rev.* **2020**, *12*, 65−84.
(43) Sanavia, T.; Birolo, G.; Montanucci, L.; Turina, P.; Capriotti, E.; Fariselli, P. Limitations and challenges in protein stability prediction upon genome variations: towards future applications in precision medicine. *Comput. Struct. Biotechnol. J.* **2020**, *18*, 1968−1979.
(44) Fang, J. A critical review of five machine learning-based algorithms for predicting protein stability changes upon mutation. *Briefings in Bioinf.* **2020**, *21*, 1285−1292.
(45) Pucci, F.; Bernaerts, K. V.; Kwasigroch, J. M.; Rooman, M. Quantification of biases in predictions of protein stability changes upon mutations. *Bioinformatics* **2018**, *34*, 3659−3665.
(46) Thiltgen, G.; Goldstein, R. A. Assessing Predictors of Changes in Protein Stability upon Mutation Using Self-Consistency. *PLoS One* **2012**, *7*, 1−6.
(47) Seeliger, D.; de Groot, B. L. Protein thermostability calculations using alchemical free energy simulations. *Biophys. J.* **2010**, *98*, 2309−2316.
(48) Benedix, A.; Becker, C. M.; de Groot, B. L.; Caflisch, A.; Böckmann, R. A. Predicting free energy changes using structural ensembles. *Nat. Methods* **2009**, *6*, 3−4.
(49) Pandurangan, A. P.; Ochoa-Montaño, B.; Ascher, D. B.; Blundell, T. L. SDM: a server for predicting effects of mutations on protein stability. *Nucleic Acids Res.* **2017**, *45*, W229−W235.
(50) Cao, H.; Wang, J.; He, L.; Qi, Y.; Zhang, J. Z. DeepDDG: Predicting the Stability Change of Protein Point Mutations Using Neural Networks. *J. Chem. Inf. Model.* **2019**, *59*, 1508−1514.
(51) Benevenuta, S.; Pancotti, C.; Fariselli, P.; Birolo, G.; Sanavia, T. An antisymmetric neural network to predict free energy changes in protein variants. *J. Phys. D: Appl. Phys.* **2021**, *54*, 245403.
(52) Pathak, Y.; Mehta, S.; Priyakumar, U. D. Learning Atomic Interactions through Solvation Free Energy Prediction Using Graph Neural Networks. *J. Chem. Inf. Model.* **2021**, *61*, 689−698.
(53) Laghuvarapu, S.; Pathak, Y.; Priyakumar, U. D. BAND NN: ADeep Learning Framework for Energy Prediction and Geometry Optimization of Organic Small Molecules. *J. Comput. Chem.* **2020**, *41*, 790−799.
(54) Aggarwal, R.; Gupta, A.; Chelur, V.; Jawahar, C. V.; Priyakumar, U. D. *DeepPocket: Ligand Binding Site Detection and Segmentation using 3D Convolutional Neural Networks*. 2021; https://www.doi.org/10.26434/chemrxiv.14611146.v1, DOI: 10.1021/acs.jcim.1c00799.
(55) Senior, A. W.; et al. Improved protein structure prediction using potentials from deep learning. *Nature* **2020**, *577*, 706−710.
(56) Bagal, V.; Aggarwal, R.; Vinod, P. K.; Priyakumar, U. D. *LigGPT: Molecular Generation using a Transformer-Decoder Model*. 2021; DOI: 10.26434/chemrxiv.14561901.v1.
(57) Yang, J.; Anishchenko, I.; Park, H.; Peng, Z.; Ovchinnikov, S.; Baker, D. Improved protein structure prediction using predicted interresidue orientations. *Proc. Natl. Acad. Sci. U. S. A.* **2020**, *117*, 1496−1503.
(58) Caldararu, O.; Mehra, R.; Blundell, T. L.; Kepp, K. P. Systematic Investigation of the Data Set Dependency of Protein Stability Predictors. *J. Chem. Inf. Model.* **2020**, *60*, 4772−4784.
(59) Pucci, F.; Bernaerts, K.; Teheux, F.; Gilis, D.; Rooman, M. Symmetry Principles in Optimization Problems: an application to Protein Stability Prediction. *IFAC-PapersOnLine* **2015**, *48*, 458−463. Eighth Vienna International Conference on Mathematical Modelling.
(60) Funahashi, J.; Sugita, Y.; Kitao, A.; Yutani, K. How can free energy component analysis explain the difference in protein stability caused by amino acid substitutions? Effect of three hydrophobic mutations at the 56th residue on the stability of human lysozyme. *Protein Eng., Des. Sel.* **2003**, *16*, 665−671.
(61) Mobley, D. L.; Klimovich, P. V. Perspective: Alchemical free energy calculations for drug discovery. *J. Chem. Phys.* **2012**, *137*, 230901−230901.
(62) Du, X.; Li, Y.; Xia, Y.-L.; Ai, S.-M.; Liang, J.; Sang, P.; Ji, X.-L.; Liu, S.-Q. Insights into Protein-Ligand Interactions: Mechanisms, Models, and Methods. *Int. J. Mol. Sci.* **2016**, *17*, 144.
(63) Stourac, J.; Dubrava, J.; Musil, M.; Horackova, J.; Damborsky, J.; Mazurenko, S.; Bednar, D. FireProtDB: database of manually curated protein stability data. *Nucleic Acids Res.* **2021**, *49*, D319−D324.
(64) Kabsch, W.; Sander, C. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* **1983**, *22*, 2577−2637.
(65) Touw, W. G.; Baakman, C.; Black, J.; te Beek, T. A.; Krieger, E.; Joosten, R. P.; Vriend, G. A series of PDB-related databanks for everyday needs. *Nucleic Acids Res.* **2015**, *43*, D364−D368.
(66) Kawashima, S.; Kanehisa, M. AAindex: Amino Acid index database. *Nucleic Acids Res.* **2000**, *28*, 374−374.
(67) Usmanova, D. R.; Bogatyreva, N. S.; Ariño Bernad, J.; Eremina, A. A.; Gorshkova, A. A.; Kanevskiy, G. M.; Lonishin, L. R.; Meister, A. V.; Yakupova, A. G.; Kondrashov, F. A.; Ivankov, D. N. Self-consistency test reveals systematic bias in programs for prediction change of stability upon mutation. *Bioinformatics* **2018**, *34*, 3653−3658.
(68) Caldararu, O.; Blundell, T. L.; Kepp, K. P. A base measure of precision for protein stability predictors: structural sensitivity. *BMC Bioinf.* **2021**, *22*, 88.
(69) Gromiha, M. M.; Uedaira, H.; An, J.; Selvaraj, S.; Prabakaran, P.; Sarai, A. ProTherm, Thermodynamic Database for Proteins and Mutants: developments in version 3.0. *Nucleic Acids Res.* **2002**, *30*, 301−302.
(70) Mazurenko, S. Predicting protein stability and solubility changes upon mutations: data perspective. *ChemCatChem* **2020**, *12*, 5590−5598.
(71) Cock, P. J. A.; Antao, T.; Chang, J. T.; Chapman, B. A.; Cox, C. J.; Dalke, A.; Friedberg, I.; Hamelryck, T.; Kauff, F.; Wilczynski, B.; de Hoon, M. J. L. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **2009**, *25*, 1422−1423.
(72) Konopka, B. M.; Marciniak, M.; Dyrka, W. Quantiprot - a Python package for quantitative analysis of protein sequences. *BMC Bioinf.* **2017**, *18*, 339.
(73) Paszke, A. et al. In *Advances in Neural Information Processing Systems 32*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc., 2019; pp 8024−8035.
(74) Pedregosa, F.; et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825−2830.
(75) Harris, C. R.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357−362.
(76) Barron, J. T. A More General Robust Loss Function *CoRR* **2017**, *abs/1701.03077*.

(77) Kingma, D. P.; Ba, J. *Adam: A Method for Stochastic Optimization*; arXiv:1412.6980, 2017.
(78) Kwasigroch, J. M.; Gilis, D.; Dehouck, Y.; Rooman, M. PoPMuSiC, rationally designing point mutations in protein structures. *Bioinformatics* **2002**, *18*, 1701−1702.
(79) Masso, M.; Vaisman, I. I. AUTO-MUTE: web-based tools for predicting stability changes in proteins due to single amino acid replacements. *Protein Eng., Des. Sel.* **2010**, *23*, 683−687.
(80) Parthiban, V.; Gromiha, M. M.; Schomburg, D. CUPSAT: prediction of protein stability upon point mutations. *Nucleic Acids Res.* **2006**, *34*, W239−W242.
(81) Kellogg, E. H.; Leaver-Fay, A.; Baker, D. Role of conformational sampling in computing mutation-induced changes in protein structure and stability. *Proteins: Struct., Funct., Genet.* **2011**, *79*, 830−838.
(82) Cheng, J.; Randall, A.; Baldi, P. Prediction of protein stability changes for single-site mutations using support vector machines. *Proteins: Struct., Funct., Genet.* **2006**, *62*, 1125−1132.
(83) Pathak, Y.; Laghuvarapu, S.; Mehta, S.; Priyakumar, U. D. Chemically Interpretable Graph Interaction Network for Prediction of Pharmacokinetic Properties of Drug-Like Molecules. *Proceedings of the AAAI Conference on Artificial Intelligence* **2020**, *34*, 873−880.
(84) Yin, S.; Ding, F.; Dokholyan, N. V. Modeling Backbone Flexibility Improves Protein Stability Estimation. *Structure* **2007**, *15*, 1567−1576.
(85) Ruder, S. *An Overview of Multi-Task Learning in Deep Neural Networks*. arXiv:1706.05098, 2017.
(86) Liebel, L.; Körner, M. *Auxiliary Tasks in Multi-task Learning*. arXiv:1805.06334, 2018.
(87) Vafaeikia, P.; Namdar, K.; Khalvati, F. *A Brief Review of Deep Multi-task Learning and Auxiliary Task Learning*. arXiv:2007.01126, 2020.

https://doi.org/10.1021/acs.jpcb.1c04913
J. Phys. Chem. B XXXX, XXX, XXX−XXX