Accurate prediction of protein stability changes from single mutations using self-distillation and antisymmetric constraint strategies

Wenkang Wang[a,b]*, Yihang Zhou[b]*, Xiaoqiang Huang[c], Yifan Wu[a,d], Min Li[a†], Yang Zhang[b,d,e†].

[a] School of Computer Science and Engineering, Central South University, Changsha, 410083, China.

[b] Department of Computer Science, School of Computing, National University of Singapore, Singapore, 117417, Singapore.

[c] Department of Internal Medicine, University of Michigan, Ann Arbor, MI 48109, USA.

[d] Cancer Science Institute of Singapore, National University of Singapore, Singapore, 117599, Singapore.

[e] Department of Biochemistry, Yong Loo Lin School of Medicine, National University of Singapore, Singapore, 117596, Singapore.

[†]Min Li, Yang Zhang.

Email: limin@mail.csu.edu.cn (M.L.), zhang@nus.edu.sg (Y.Z.)

Author Contributions: M.L. and Y.Z. designed research; W.W performed research and analyzed data; Yi.Z., X.H., and Y.W. contributed to the discussion and experiment design; W.W., Yi.Z., H.X., Y.W., M.L. and Y.Z. wrote the paper. W.W. and Yi.Z. contributed equally to this work.

Competing Interest Statement: The authors declare no competing interest.

Classification: Biological Sciences; Biophysics and Computational Biology.

Keywords: protein stability, point mutation, large language model, self-distillation, deep learning.

This PDF file includes:

Main Text
Figures 1 to 5
Tables 1 to 2

1

Abstract

Computational approaches for accurately predicting protein stability changes upon residue mutations are crucial for protein engineering and design. Sequence-based methods are easier to apply to large-scale proteins since they do not rely on high-quantity structures. However, existing sequence-based approaches struggle to capture structural changes, resulting in lower performance compared to structure-based methods. In this study, we propose DPStab, a sequence-based deep learning solution that accurately predicts protein stability changes upon single residue mutations. DPStab transfers a protein large language model as a core component and incorporates a cross-attention mechanism to capture the contact changes around mutated positions for $\Delta\Delta G$ and ΔT_m prediction. To address data imbalance and the antisymmetric nature of mutation effects, DPStab employs a self-distillation inference strategy under the supervision of an antisymmetric constraint. Benchmarking demonstrates that DPStab achieves state-of-the-art performance in both $\Delta\Delta G$ and ΔT_m prediction. Practical evaluations confirm DPStab's capability in accurately ranking protein stability on large-scale datasets and effectively identifying critical structural contacts impacting stability. More experiments on extensive cDNA display proteolysis data demonstrate the significant contributions of self-distillation and antisymmetric constraint strategies.

Significance Statement

Single amino acid mutations significantly influence protein stability, thereby affecting biological function and potential therapeutic uses. Accurately predicting how mutations affect protein stability is fundamental to protein engineering and therapeutic design. However, current sequence-based computational methods fail to capture the structural context changes around mutated residues. To overcome this, we propose DPStab, a sequence-based deep learning approach that combines a protein language model and a cross-attention mechanism with self-distillation and antisymmetric strategies. DPStab effectively captures residue contact changes and predicts stability changes without structural data. Sufficient experiments demonstrate that DPStab significantly outperforms existing methods, providing a fast and practical tool for enhancing protein engineering and biomedical research.

Main Text

Introduction

Proteins exhibit a variety of functions in nature, including catalyzing chemical reactions, mediating intercellular signaling, and facilitating transcriptional transport, with their stability serving as a fundamental prerequisite for these processes (1, 2). Protein stability is highly correlated with its sequence and structure, which is easily altered by residue mutations (3, 4). Introducing specific residue mutations to enhance protein stability has become a widely used strategy, particularly in industrial applications (5, 6). However, a brute-force mutation strategy is both time-consuming and costly. Accurately predicting protein stability changes upon residue mutations can reveal crucial insights into the relationships between protein sequence, structure, folding, and function, serving as invaluable prior knowledge for guiding protein design (7).
Gibbs free energy (ΔG) is the difference between the folded and unfolded states, denoted as $\Delta G = G_{folded} - G_{unfolded}$, and T_m represents the melting temperature of proteins, which are two commonly used metrics to measure protein stability. Correspondingly, Gibbs free energy changes ($\Delta\Delta G$) and melting temperature changes (ΔT_m) reflect protein stability changes. These properties are antisymmetric, which means $\Delta\Delta G_{x\to y} = -\Delta\Delta G_{y\to x}$ and $\Delta T_{m,x\to y} = -\Delta T_{m,y\to x}$. In this study, both higher $\Delta\Delta G$ and ΔT_m denote the mutation more stable.

To accurately predict $\Delta\Delta G$ and ΔT_m, numerous predictors centered on the geometric potential energy of protein structures have been proposed, such as FoldX (8), EvoEF (9, 10), STRUM (11), and Korpm (12). These computational methods rely on protein structures, rendering them limited when high-quality experimental or predicted structures are not available. Additionally, machine learning methods, like DDGun (13, 14) and MutateEverything (15), have also been proposed for $\Delta\Delta G$ prediction from sequences. Notably, a few deep learning approaches (16, 17) integrate protein sequence and structure information to predict $\Delta\Delta G$ and ΔT_m, which achieve an improvement compared to other methods and provide new sights for incorporating sequence and structure information through deep learning. However, these methods require the native or predicted protein structures, which prevents them from performing end-to-end prediction directly from sequence. Meanwhile, the imbalance of protein stability data presents a significant challenge for existing methods, where the number of stabilizing mutations is considerably less than that of destabilizing mutations. This phenomenon can easily impact these approaches, leading to results with distinct preferences (18).

Protein large language models (pLLMs) (19–21), owing to their self-supervised training on vast amounts of unlabeled protein data, exhibit remarkable generalization capabilities and have achieved robust performance across numerous downstream tasks (2, 19). Recently, some efforts have attempted to use pLLMs to predict protein stability changes (22, 23). For instance, ThermoMPNN (23) extracts the learnable features from a structure-based pLLM, ProteinMPNN (24), and leverages transfer learning from the mega-scale data used in ProteinMPNN to the protein stability data. In contrast, PROSTATA (22) adopts a sequence-based pLLM, ESM2 (20), as the backbone and uses different regression heads to predict protein stability changes. However, most existing pLLMs and structure modeling methods take a single protein as input during their pre-training process, which makes it hard to distinguish the single mutation effects. Consequently, it is essential to capture the contact changes around the mutation sites, even though the structural backbone may not change significantly before or after the mutation. Additionally, distillation on existing LLMs shows great potential to further enhance the performance of LLMs for specific downstream tasks (25) and has made much progress in various areas (26), such as DeepSeek (27), providing new perspectives to transfer LLMs. Here, we propose DPStab, a protein sequence-based deep learning method for accurate protein stability change prediction. DPStab can capture residue features, infer contact maps, and incorporate neighbors around the mutated positions with the architecture of pLLM. Notably, we propose a self-distillation inference strategy and introduce an antisymmetric constraint to further improve its performance. Comprehensive benchmarks and analyses demonstrate DPStab to be the state-of-the-art method for both $\Delta\Delta G$ and ΔT_m prediction.

Results

DPStab consists of three modules (Figure 1a): (1) a sequential encoder transferred from ESM2 for extracting evolutionary information from sequence and conformational information from contact maps (Figure 1b). (2) a neighboring encoder that crosses the information between sequence and structure, with weighted attention to neighboring residues, which includes the long-range interaction (Figure 1c). (3) a feature transition module that incorporates the representations of the wild-type and mutated proteins, along with environmental factors (e.g., pH and temperature), to predict stability changes ($\Delta\Delta G$ or ΔT_m). Notably, temperature is used only for $\Delta\Delta G$ prediction (Figure 1d).

Additionally, at the inference stage, we propose a self-distillation strategy to enrich the training data (Figure 1e) and incorporate a constraint to align DPStab with the antisymmetric nature of mutations. Specifically, multiple DPStab models are first trained on the direct and symmetrical training data. These trained models act as teachers to produce pseudo-labels for test data, which will be fed back to generate better student models with the original training set and independent validation set. Meanwhile, a contrastive loss is applied to make DPStab produce opposite outputs

for direct and inverse mutations. Further details of each module can be found in the "Methods" section.

DPStab outperforms existing methods overall in $\Delta\Delta G$ and ΔT_m prediction. To evaluate the overall performance of DPStab in $\Delta\Delta G$ prediction, we compare it against seven structure-based (8, 13, 14, 16, 23, 28, 29) and three sequence-based (13–15, 22) approaches. To ensure a fair comparison, we utilize a manually curated independent test dataset, S461 (12), containing proteins sharing a low sequence identity with those in the training set. S461 is also used as a benchmark set in previous studies (16, 17). Four regression metrics - Spearman correlation coefficient (SCC), Pearson's correlation coefficient (PCC), root mean square error (RMSE) and mean absolute error (MAE) - as well as three multi-class classification metrics: accuracy (ACC), Matthews correlation coefficient (MCC) and Cohen's kappa score (κ) are used to evaluate the performance of these methods. Higher SCC, PCC, ACC, MCC and κ values indicate superior performance, whereas lower RMSE and MAE values correspond to smaller prediction errors (see "Methods" for more details).

Table 1 presents the performance of various methods on the S461 dataset in these metrics. DPStab outperforms all other approaches. It achieves a smaller error range, with an RMSE below 0.93 kcal/mol and an MAE below 0.68 kcal/mol, while the best performance of other methods corresponds to an RMSE of 0.98 kcal/mol and an MAE of 0.72 kcal/mol, respectively (details can be seen in Supplementary Table S1). In terms of rank correlations, the predictions of DPStab exhibit greater consistency with experimental results. Compared to other methods, DPStab improves from 0.83 to 0.86 in SCC and from 0.82 to 0.84 in PCC. In Figure 2a, we further compare the absolute errors of individual mutations predicted by DPStab against several control methods. The median of absolute errors of DPStab is significantly lower than other sequence-based and structure-based approaches, where the corresponding P-values confirm the statistical significance of these improvements.

In certain scenarios such as soluble protein design, classifying mutations rather than predicting exact $\Delta\Delta G$ values is more practically useful (30–32). To establish a classification task, we categorize mutations into three classes: destabilizing ($\Delta\Delta G$ < -0.5 kcal/mol), stabilizing ($\Delta\Delta G$ > -0.5 kcal/mol) and neutral. DPStab achieves a significantly higher ACC than other methods (Table 1). Among the 922 mutations (comprising 461 direct and 461 inverse mutations), DPStab correctly predicts the mutation classes for 684 cases (Figure 2b). Additionally, DPStab achieves the highest MCC and κ (both > 0.61), whereas the second-best method, GeoDDG-3D, achieves only 0.51 (Figure 2c), highlighting its superior performance in mutation classification.

Furthermore, we compare DPStab with GeoDTm-AF2 and GeoDTm-3D (16) on ΔT_m prediction. Interestingly, DPStab achieves a comparable SCC to GeoDTm-AF2 and GeoDTm-3D while significantly outperforming them in terms of PCC, RMSE and MAE (Table 2 and Supplementary Table S2). For classification, we similarly categorized mutations into three groups: destabilizing (ΔT_m < -2 ℃), stabilizing (ΔT_m > 2 ℃), and neutral. DPStab achieves an ACC of 0.56, surpassing the 0.52 for GeoDTm-AF2 and 0.53 for GeoDTm-3D (Table 2). Moreover, the evaluation on Cohen's kappa score and MCC confirms the better performance of DPStab relative to the two GeoDTm models (Figure 2d). Overall, our data indicates that DPStab achieves a stronger or comparable performance for both $\Delta\Delta G$ and ΔT_m prediction.

DPStab provides accurate guidance on individual proteins. We further evaluate DPStab's performance on individual proteins to investigate its capability for protein engineering. We independently evaluate its performance on individual proteins. Accordingly, there are 388 mutations from 15 proteins used here; each protein has at least 6 mutations. DPStab exhibits substantial improvements in SCC, PCC, and ACC, compared to GeoDDG-3D (Figure 3a). Specifically, it achieves remarkable median improvements of 9.1% (0.88 versus 0.81) in SCC, 7.5% (0.86 versus 0.80) in PCC, and 19.4% (0.80 versus 0.67) in ACC. Regarding error ranges, DPStab surpasses GeoDDG-3D across most proteins (Figure 3b). A similar conclusion can also be obtained for ΔT_m prediction (Supplementary Figure S1). Collectively, these results highlight the

advantages of DPStab in $\Delta\Delta G$ and ΔT_m at protein-level, which provides strong guidance for protein engineering.

We further investigate protein 2PTL-A, which is a clear sample of DPStab outperforming GeoDDG-3D (Figure 3b). Figure 3c presents a head-to-head comparison of DPStab and GeoDDG-3D for $\Delta\Delta G$ prediction on all known mutations of 2PTL-A, which indicate that DPStab achieves higher R^2 correlation values with an improvement of 9% over GeoDDG-3D, as well as SCC (increase 11%) and PCC (increase 5%). These demonstrate that DPStab can provide a more reliable rank list for mutations of 2PTL-A. To explore the reason for the DPStab boost, we analyze the representation similarities of these mutations from our model and ESM2 that are also used in GeoDDG-3D. In the original ESM2 model, these mutations have high similarities and are hard to distinguish for $\Delta\Delta G$ prediction, while in DPStab, the correlation between representation similarities and $\Delta\Delta G$ is significantly improved (Figure 3d).

DPStab is not only able to identify positive mutations but also predict negative mutations. We provide two specific mutations here. The first one is 2PTL-A (Figure 3e) and its mutation A34G (Figure 3f), which is provided in the manually curated test dataset and leads to a decreased stability (-2.1 kcal/mol). Notably, DPStab accurately predicts this mutation (-1.2 kcal/mol), while GeoDDG-3D considers A34G a neutral mutation with a prediction of -0.3 kcal/mol. To investigate the underlying reasons for their stability changes, we analyze the structural alterations induced by this mutation via FoldX. Specifically, Glycine (Gly, G) lacks a C_β atom, which may decrease the Solvent Accessible Surface Area (SASA) and backbone flexibility, leading to local structural instability. Secondly, Gly has a smaller molecular volume compared to Alanine (Ala, A), resulting in fewer interactions with surrounding residues. Consequently, it loses interactions with Y50 and the hydrophobic residue F36, reducing hydrophobic interactions and further contributing to structural destabilization (see Figure 3f).

Similarly, Figure 3g and Figure 3h illustrate another mutation H51A of protein 1O6X-A. In wild-type protein, Histidine (His, H) at position 51 generates two hydrogen bonds with W40 and T49, respectively. However, after the mutation H51A, the volume of the residue A51 reduces significantly, which does not interact with E14 and K41, resulting in the area being hollow and unstable. More importantly, due to the disappearance of the phenyl ring, the hydrogen bond between A51 and T49 is lost, which is detrimental to structural stability. Consequently, the mutation H51A will reduce protein stability ($\Delta\Delta G$ = -0.7 kcal/mol), and DPStab detects it accurately with a prediction of -0.8 kcal/mol. In contrast, GeoDDG-3D considers this mutation stable with a prediction of 0.6 kcal/mol. All evidence demonstrates that DPStab can effectively predict protein stability changes upon single mutations.

DPStab achieves high ranking accuracy in large-scale DMS data. We further test the scalability of DPStab on deep mutational scanning (DMS) data (33) that has been used in GeoDDG-3D. As DMS data is not equivalent to $\Delta\Delta G$, we use Kendall rank correlation coefficient to measure the correlation between two ranks (34). In the self-distillation stage, we randomly extract 10% (2000) mutations to enrich our data to avoid introducing much noise. DPStab improves SCC and Kendall by 13% and 15%, respectively (Figure 4a). In terms of individual proteins, this advantage is more pronounced. DPStab gets better predictions for most proteins, improving median SCC (0.72 versus 0.64 in Figure 4b) and Kendall (0.53 versus 0.46 in Figure 4c) significantly on individual proteins. And the corresponding scatter plots in Figure 4b and 4c demonstrate that DPStab outperforms GeoDDG-3D on most proteins, indicating that DPStab can rank multiple associated mutations in a single protein in most cases.

Furthermore, for each protein, we measure the accuracy of top-k predicted mutations. Figure 4d shows the results for k from 1 to 60. DPStab is maintained and does not diminish as k increases, which represents that its predictions for top-ranked mutations are more reliable. On the other hand, we statistically compare the true $\Delta\Delta G$ values corresponding to the top-k ratio of mutations predicted by different methods, where higher $\Delta\Delta G$ values suggest that the methods are more likely to accurately provide higher $\Delta\Delta G$ values, i.e., stabilizing mutations (32). DPStab continues to excel across different ratios (Figure 4e), indicating that its top-ranked predictions generally

exhibit higher ground-truth $\Delta\Delta G$ values and therefore are more likely to correspond to stabilizing mutations.

Generalization of self-distillation inference and antisymmetric constraint in DPStab. The self-distillation inference strategy can enhance training data diversity and further improve the accuracy of existing trained models. After removing the inference stage, the performance will reduce in all metrics (Figure 5a). Specifically, for $\Delta\Delta G$ prediction, both SCC and PCC reduce by 3%, and RMSE and MAE correspondingly increase by 6% and 4%. This phenomenon also exists in ΔT_m (Figure 5b).

To investigate the role of self-distillation more thoughtfully, we collect more protein stability-related data, which is generated by cDNA display proteolysis for fast measuring folding stability (35). It infers ΔG of proteins and $\Delta\Delta G$ of mutations from sequencing counts with a Bayesian model. To validate the accuracy of inferred $\Delta\Delta G$ values, we extract the overlapping mutations in the inferred data and our training data. The correlation between them achieves 0.85 when considering $\Delta\Delta G$ between -2 and 1 kcal/mol (Figure 5c). Consequently, we split all mutations into six groups by protein length and randomly sample three proteins with 450 mutations in each group (Supplementary Figure S2), resulting in six sampled independent test datasets (Supplementary Table S3). Then, based on our trained teacher models, we first generate pseudo-labels for these test datasets and re-train corresponding DPStab models, respectively. As shown in Figure 5d, SCC and PCC values at the inference stage can be further improved in most cases, which demonstrates the effectiveness of self-distillation in our model.

Additionally, as can be deduced from the definitional formula of $\Delta\Delta G_{wt\rightarrow mut} = \Delta G_{wt} - \Delta G_{mut}$, direct and inverse mutations should produce exactly opposite $\Delta\Delta G$ values. Since DPStab does not employ a fully symmetrical network architecture (i.e., GeoDDG-AF2 and GeoDDG-3D), it may not produce inverse outputs for symmetric inputs (x_{wild}, x_{mut}) and (x_{mut}, x_{wild}). This limitation also exists in other methods, such as PROSTATA, ThermoMPNN, and Mutate Everything, which need to be addressed to make models more robust in practical cases. To mitigate this limitation, we introduce a constraint loss to guide our model in learning this antisymmetric property. The absolute value of the sum of predicted results for direct and inverse mutations is used here as an error evaluation metric $\langle\delta\rangle$. It can be observed that after introducing the antisymmetric constraint, DPStab exhibits a significant reduction in error and is nearly capable of predicting perfectly opposite $\Delta\Delta G$ values for direct and inverse mutations, making it more valuable for practical applications (Figure 5e). Above all, the self-distillation strategy can improve the robustness of our model, and the antisymmetric constraint makes the predictions more practical.

Discussion

Effective and efficient predicting protein stability changes ($\Delta\Delta G$ and ΔT_m) of mutations is crucial for protein design and protein engineering. Sequence-based predictors are more easily applicable to large-scale datasets compared to structure-based approaches. However, existing sequence-based methods fail to capture structural contact changes, resulting in inferior performance to structure-based methods. In this study, we develop a sequence-based deep learning method for predicting protein stability changes from single residue mutations, called DPStab. It leverages a pLLM to encode residue representations and predict residue contacts. Subsequently, a cross-attention mechanism is introduced to capture the contact changes around the mutated positions. Comprehensive and rigorous analyses highlight the significant improvements achieved by DPStab. Notably, DPStab not only demonstrates strong overall performance, but also shows its advantages on individual protein level, which is more efficient in practical cases.

Although fine-tuning pLLMs for predicting protein stability changes has already been explored in previous studies (22, 23), these models still face the limitations of $\Delta\Delta G$ training data, which is very unbalanced. This can lead to decreased performance of these models when predicting the independent test set with different data distributions (32). Consequently, we adopt a self-distillation strategy at inference stage to further improve the generalization of models. This

strategy can generate more data and enrich data distributions, which further improves the performance of existing models. On the other hand, traditional asymmetrical frameworks cannot generate inverse outputs for symmetric inputs, except when the wild-type and mutated proteins share the whole network. However, completely sharing model weights between wild-type and mutated proteins may make it difficult to learn the difference between these two proteins. To solve these limitations, we introduce a simple antisymmetric constraint to guide our model to learn antisymmetric data. Comprehensive ablation studies suggest that the self-distillation strategy substantially enhances the performance of DPStab, offering new perspectives into further improving existing protein large language models (pLLMs) for specific downstream tasks. And contrastive loss can also enable an asymmetric network architecture to better learn the constraints of antisymmetric data.

Overall, DPStab only takes protein sequences as input and does not rely on high-quality structural data. It can predict residue contact changes and potentially capture the key residues or regions involved in important interactions that maintain protein stability, such as $\pi - \pi$ interactions and hydrogen bonds. In the future, we will consider side-chain changes and introduce more geometric potential to help models capture structure changes during mutations. Additionally, in practical protein engineering, optimizing a protein for specific applications often requires multiple mutations. Consequently, how to design an effective framework to predict stability changes upon multiple residue mutations is another challenge to be addressed.

Materials and Methods

Datasets. The $\Delta\Delta G$ and ΔT_m training dataset used in this article is derived from the previous study (16), named S8754 and S4346, respectively, which are collected from the ProThermDB (36) and ThermoMutDB (37) databases. S8754 consists of 8754 single-point mutations across 301 proteins, along with environmental factors (pH and temperature) and $\Delta\Delta G$ values. S4346 includes 358 proteins with 4346 single residue mutations, along with pH factors and ΔT_m values. We randomly select 10% of the training data to serve as a validation set to assess whether the model training is sufficient. Additionally, we use an independent dataset to test the performance of models in $\Delta\Delta G$ prediction, named S461. This dataset maintains <25% sequence identity with the proteins in S8754 and each mutation has been manually corrected. Finally, there are 461 single mutations of 48 proteins used in our test dataset. For ΔT_m prediction, we use S571 as test data. It includes 39 proteins with 571 single mutations, which are all controlled by 25% sequence identities to the S4346.

Due to the antisymmetric nature of mutations, where $\Delta\Delta G_{wild\rightarrow mut} = -\Delta\Delta G_{mut\rightarrow wild}$, it is necessary to measure the prediction results of models in antisymmetric consistency. Consequently, we generate corresponding inverse mutation data ('mutation'→'wild type') from the direct mutation data ('wild type'→'mutation'), resulting in a maximum of 17235 samples available for training and 922 samples for test. Similarly, there are 8692 samples for ΔT_m training and 1142 samples for test.

Moreover, we also collect an inferred $\Delta\Delta G$ dataset generated by cDNA display proteolysis (35), consisting of 389068 single residue mutations across 412 proteins. Based on the length of the wild-type protein sequences, we randomly select three proteins within each length interval of 7 residues, and further randomly selected 150 mutations per protein to construct the test set. Totally, there are 6 generated test sets, each containing approximately 450 mutations. Additionally, to compare DPStab with GeoDDG-3D fairly, another sampled DMS dataset is used, including 20000 mutations from 73 proteins, where all these proteins have native structures (16).

Architecture of sequential encoder. To learn evolutionary and conformational information of protein sequence, DPStab adopts the same sequential encoder as ESM2 and utilizes the pre-trained weights of ESM2 as its initial parameters, including one token embedding layer, 33 stacked transformer layers, and one linear regression layer for contact map prediction. Specifically, the input is a wild type or mutated protein sequence $p = \{r_1, r_2, \ldots, r_L\}$ with L residues,

where $r_i \in \{A, B, \ldots, Y, Z\}$ are the different residue types. It is first fed into a token embedding layer:

$$Tokens_p = \{\langle CLS \rangle, r_1, r_2, \ldots, r_L, \langle EOS \rangle\} \#(1)$$

$$R_p = \mathrm{Embedding}\left(Tokens_p\right) \in \mathbb{R}^{(L+2)\times d} \#(2)$$

where $Tokens_p$ is the tokens that represent the protein sequence p, $\langle CLS \rangle$ and $\langle EOS \rangle$ are two additional signs indicating the beginning and end of protein sequences. Then, a trainable embedding matrix is used to generate corresponding representations for each token, denoted as R_p, which is fed into later transformer layers:

$$H_{l+1} = \mathrm{MLP}(\mathrm{SelfAttention}(H_l) + H_l) \#(3)$$

where $H_l \in \mathbb{R}^{(L+2)\times d}$ is the input of $(l+1)-th$ transformer layer $(l = 0, \ldots, n-1)$, and $H_0 = R_p$ initially. $\mathrm{SelfAttention}(\cdot)$ is traditional attention mechanism that generates (Q, K, V) items, calculates attention scores between different tokens and updates their representations. $\mathrm{MLP}(\cdot)$ is composed of several linear layers and corresponding activate functions. Through transformer layers, different attention maps can be obtained from different heads, which are concatenated to predict contact probabilities between residues:

$$CM = \mathrm{Sigmoid}\left(\mathrm{Concat}\left(a_{1,1}, \ldots, a_{i,j}, \ldots, a_{n,h}\right) \times W_{lr} + b_{lr}\right) \#(4)$$

$$\mathrm{Sigmoid}(x) = \frac{1}{1+e^{-x}} \#(5)$$

where $\alpha_{i,j}$ is the attention map from $j-th$ attention heads in $i-th$ transformer layer, $\left(W_{lr} \in \mathbb{R}n\times h,1,blr\in\mathbb{R}1\right.$ is a linear layer, and n,h represent the number of transformer layers and attention heads of each layer, respectively.

Architecture of neighboring encoder. Through sequential encoder, DPStab generates the residue representations H_n and corresponding contact map CM from the target protein sequence p. Based on these features, DPStab introduces a cross-attention module to extract the neighboring information of mutated site m $(1 \le m \le L)$. Specifically, in the contact map CM, the top-k neighboring residues with the highest concat probabilities with the mutation site m are selected:

$$neighbors(m) = \mathrm{topk}(CM[m,:]).index \#(6)$$

Subsequently, the representations of these k neighbors (denoted as $H_n[neighbors(m),:] \in \mathbb{R}^{k\times d}$), along with the mutation site ($H_n[m,:] \in \mathbb{R}^{1\times d}$), are then extracted and used as inputs for the cross-attention module to integrate the neighboring information:

$$\begin{cases} Q = H_n[neighbors(m),:] \times W_Q \\ K = H_n[m,:] \times W_K \\ V = H_n[neighbors(m),:] \times W_V \end{cases} \#(7)$$

$$X_{temp} = \mathrm{Softmax}\left(\frac{Q \times K^T}{\sqrt{d}}\right)^T \times V + H_n[neighbors(m),:] \#(8)$$

$$X_{output} = \mathrm{MLP}\left(X_{temp}\right) + X_{temp} \#(9)$$

$$X_{neighbors} = \sum_{i=1}^{k} X_{output}[i,:] \#(10)$$

where $(Q \in \mathbb{R}^{k\times d}, V \in \mathbb{R}^{k\times d})$ is linearly transferred from the k neighbors and $V \in \mathbb{R}^{1\times d}$ is from the mutated site. Through this cross-attention mechanism, DPStab can learn the importance of neighbors and integrate their representations adaptively, denoted as $X_{neighbors} \in \mathbb{R}^{1\times d}$. Additionally, residual operation and $\mathrm{LayerNorm}(\cdot)$ are also employed here to prevent the loss of features.

Architecture of feature transition module. Finally, by two previous encoders, DPStab generates the features of the target protein, mutated site, and corresponding neighbors, denoted as $H_n[0,:] \in \mathbb{R}^{1\times d}$ ($\langle CLS \rangle$ token), $H_n[m,:] \in \mathbb{R}^{1\times d}$, and $X_{neighbors} \in \mathbb{R}^{1\times d}$, respectively. Hence, the features of wild-type protein ($H_n^{wild}[0,:], H_n^{wild}[m,:], X_{neighbors}^{wild}$) and mutated protein

$(H_n^{mut}[0,:], H_n^{mut}[m,:], X_{neighbors}^{mut})$ can be obtained separately. Based on these paired features, a simple feature transition module is applied to learn the difference between the two proteins:

$$X_{wild} = \mathrm{Concat}\left(H_n^{wild}[0,:] \odot W_{wild}, H_n^{wild}[m,:] \odot W_{wild}, X_{neighbors}^{wild} \odot W_{wild}\right) \#(11)$$

$$X_{mut} = \mathrm{Concat}\left(H_n^{mut}[0,:] \odot W_{mut}, H_n^{mut}[m,:] \odot W_{mut}, X_{neighbors}^{mut} \odot W_{mut}\right) \#(12)$$

$$X_{diff} = \mathrm{MLP}(X_{wild} - X_{mut}) \#(13)$$

where $W_{wild} \in \mathbb{R}^{1\times d}$ and $W_{mut} \in \mathbb{R}^{1\times d}$ are two learnable parameters. After obtaining the difference between wild-type and mutated proteins, environmental parameters are combined to predict ΔG changes ($\Delta\Delta G$) upon single-point mutation:

$$X_{env} = \mathrm{MLP}([ph, temperature]) \#(14)$$

$$\Delta\Delta G_{pred} = \mathrm{MLP}\left(\mathrm{Concat}\left(X_{diff}, X_{env}\right)\right) \#(15)$$

Self-distillation strategy and model training. As shown in Figure 1e, the whole process of DPStab can be divided into two stages, including training and inference. Specifically, the first stage generates two different trained DPStab models, based on the direct mutations and symmetrical mutations. To effectively learn the naturally imbalanced distribution dominated by destabilizing mutations, the first DPStab model is trained on direct training data and optimized using mean squared error (MSE) loss function:

$$MSELoss = \frac{1}{N}\left(\sum_{i=1}^{N} y_{pred}^{i} - y_{true}^{i}\right)^2 \#(16)$$

where N is the number of training samples and $(y_{pred}^{i}, y_{true}^{i})$ are the predicted and true $\Delta\Delta G$ values, respectively. Then, we train another DPStab model on symmetrical data with MSE loss and an additional antisymmetric constraint:

$$AtSymmConstraint = \sum_{i=1}^{N} \left| y_{pred}^{i,wild\to mut} + y_{pred}^{i,mut\to wild} \right| \#(17)$$

$$Loss = MSELoss + AtSymmConstraint \#(18)$$

where $y_{pred}^{i,wild\to mut}$ is the predicted result of $i-th$ training sample from wild-type protein to mutated protein, and $y_{pred}^{i,mut\to wild}$ corresponds to the predicted result for the inverse mutation. This loss function ensures that the model's predictions are consistent with the constraint as much as possible. Based on these two teacher models, at the inference stage, we first average their predicted results for test data as pseudo-labels. After incorporating training data and test data with pseudo-labels, a single dependent DPStab is trained to produce the final predictions. In the entire training and inference process, the hidden dimension size is set to 1280, AdamW optimizer is used to optimize models with dynamic decaying learning rates, and the weights of transformer layers from ESM2 are fixed in the first two epochs. DPStab will stop training if the loss value of validation data is stable or trending upwards.

All experiments of DPStab are carried out using one NVIDIA Tesla H100 GPU card with 96 GB of memory. The training process takes around 15 hours with a batch size of 4.

Evaluation metrics. In this study, seven metrics are used to evaluate the performance of models in regression and classification, including Spearman correlation coefficient (SCC), Pearson's correlation coefficient (PCC), root mean square error (RMSE), and mean absolute error (MAE), accuracy (ACC), Matthews correlation coefficient (MCC) and Cohen's kappa score. Specifically, given predicted results Y_{pred} and true labels Y_{true}, SCC is defined as:

$$SCC\left(Y_{pred}, Y_{true}\right) = \frac{Cov\left[Rank\left(Y_{pred}\right), Rank(Y_{true})\right]}{\sigma_{Y_{pred}}\sigma_{Y_{true}}} \#(19)$$

where $Rank(\cdot)$ is the rank positions of elements in the sorted order of the original list. Cov denotes the covariance between two lists, and σ is the standard deviation. PCC is similar to SCC but directly calculates the correlation between the original lists:

$$PCC(Y_{pred}, Y_{true}) = \frac{Cov[Y_{pred}, Y_{true}]}{\sigma_{Y_{pred}}\sigma_{Y_{true}}} \#(20)$$

where lager SCC and PCC indicate better predictions. RMSE and MAE reflect the error distances between predictions and labels:

$$RMSE(Y_{pred}, Y_{true}) = \sqrt{\frac{1}{N}\left(\sum_{i=1}^{N} y_{pred}^{i} - y_{true}^{i}\right)^{2}} \#(21)$$

$$MAE(Y_{pred}, Y_{true}) = \frac{1}{N}\sum_{i=1}^{N}\left|y_{pred}^{i} - y_{true}^{i}\right| \#(22)$$

Both smaller RMSE and MAE represent better performance. Additional three metrics, ACC, MCC, and Cohen's kappa score (κ) evaluate the multi-class classification abilities of models:

$$ACC = \sum_{c \in C} \frac{tp_c}{N} \#(23)$$

$$MCC = \frac{\sum_c tp_c \times N - \sum_c p_c \times n_c}{\sqrt{(N^2 - \sum_c p_c^2)(N^2 - \sum_c n_c^2)}} \#(24)$$

$$\kappa = 1 - \frac{1 - \sum_c tp_c}{1 - \sum_c p_c \times n_c} \#(25)$$

where tp_c denotes the number of true positives for category c, and N represents the number of samples. p_c and n_c represent the number of samples predicted as class c and the number of samples belonging to class c, respectively. Both higher ACC, MCC, and κ values show better performance in multi-class classification.

Data and Software Availability. The data and source code can be downloaded from https://github.com/CSUBioGroup/DPStab.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grant (No.62225209 to M.L.), Hunan Postgraduate Research and Innovation Project (CX20240018 to W.W.), and the High Performance Computing Center of Central South University.

References

1. D. Eisenberg, E. M. Marcotte, I. Xenarios, T. O. Yeates, Protein function in the post-genomic era. *Nature* **405**, 823–826 (2000).
2. W. Wang, Y. Shuai, M. Zeng, W. Fan, M. Li, DPFunc: accurately predicting protein function via deep learning with domain-guided structure information. *Nat. Commun.* **16**, 70 (2025).
3. N. Tokuriki, D. S. Tawfik, Stability effects of mutations and protein evolvability. *Curr. Opin. Struct. Biol.* **19**, 596–604 (2009).
4. R. A. Studer, B. H. Dessailly, C. A. Orengo, Residue mutations and their impact on protein structure and function: detecting beneficial and pathogenic changes. *Biochem. J.* **449**, 581–594 (2013).
5. K. L. Morley, R. J. Kazlauskas, Improving enzyme properties: when are closer mutations better? *Trends Biotechnol.* **23**, 231–237 (2005).
6. N. G. Nezhad, *et al.*, Thermostability engineering of industrial enzymes through structure modification. *Appl. Microbiol. Biotechnol.* **106**, 4845–4866 (2022).
7. Y. Ming, *et al.*, A review of enzyme design in catalytic stability by artificial intelligence. *Brief. Bioinform.* **24**, bbad065 (2023).

8. J. Delgado, L. G. Radusky, D. Cianferoni, L. Serrano, FoldX 5.0: working with RNA, small molecules and a new graphical interface. *Bioinformatics* **35**, 4168–4169 (2019).
9. R. Pearce, X. Huang, D. Setiawan, Y. Zhang, EvoDesign: designing protein–protein binding interactions using evolutionary interface profiles in conjunction with an optimized physical energy function. *J. Mol. Biol.* **431**, 2467–2476 (2019).
10. X. Huang, R. Pearce, Y. Zhang, EvoEF2: accurate and fast energy function for computational protein design. *Bioinformatics* **36**, 1135–1142 (2020).
11. L. Quan, Q. Lv, Y. Zhang, STRUM: structure-based prediction of protein stability changes upon single-point mutation. *Bioinformatics* **32**, 2936–2946 (2016).
12. I. M. Hernández, Y. Dehouck, U. Bastolla, J. R. López-Blanco, P. Chacón, Predicting protein stability changes upon mutation using a simple orientational potential. *Bioinformatics* **39**, btad011 (2023).
13. L. Montanucci, E. Capriotti, Y. Frank, N. Ben-Tal, P. Fariselli, DDGun: an untrained method for the prediction of protein stability changes upon single and multiple point variations. *BMC Bioinformatics* **20**, 1–10 (2019).
14. L. Montanucci, *et al.*, DDGun: an untrained predictor of protein stability changes upon amino acid variants. *Nucleic Acids Res.* **50**, W222–W227 (2022).
15. J. Ouyang-Zhang, D. Diaz, A. Klivans, P. Krähenbühl, Predicting a protein's stability under a million mutations. *Adv. Neural Inf. Process. Syst.* **36** (2024).
16. Y. Xu, D. Liu, H. Gong, Improving the prediction of protein stability changes upon mutations by geometric learning and a pre-training strategy. *Nat. Comput. Sci.* 1–11 (2024).
17. Y. Chen, Y. Xu, D. Liu, Y. Xing, H. Gong, An end-to-end framework for the prediction of protein structure and fitness from single sequence. *Nat. Commun.* **15**, 7400 (2024).
18. C. Pancotti, *et al.*, Predicting protein stability changes upon single-point mutation: a thorough comparison of the available tools on a new dataset. *Brief. Bioinform.* **23**, bbab555 (2022).
19. A. Rives, *et al.*, Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci.* **118**, e2016239118 (2021).
20. Z. Lin, *et al.*, Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379**, 1123–1130 (2023).
21. T. Hayes, *et al.*, Simulating 500 million years of evolution with a language model. *Science* eads0018 (2025).
22. D. Umerenkov, *et al.*, PROSTATA: a framework for protein stability assessment using transformers. *Bioinformatics* **39**, btad671 (2023).
23. H. Dieckhaus, M. Brocidiacono, N. Z. Randolph, B. Kuhlman, Transfer learning to leverage larger datasets for improved prediction of protein stability changes. *Proc. Natl. Acad. Sci.* **121**, e2314853121 (2024).
24. J. Dauparas, *et al.*, Robust deep learning–based protein sequence design using ProteinMPNN. *Science* **378**, 49–56 (2022).
25. X. Xu, *et al.*, A survey on knowledge distillation of large language models. *ArXiv Prepr. ArXiv240213116* (2024).
26. C.-Y. Hsieh, *et al.*, Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *ArXiv Prepr. ArXiv230502301* (2023).
27. D. Guo, *et al.*, DeepSeek-Coder: When the Large Language Model Meets Programming–The Rise of Code Intelligence. *ArXiv Prepr. ArXiv240114196* (2024).
28. D. E. Pires, D. B. Ascher, T. L. Blundell, mCSM: predicting the effects of mutations in proteins using graph-based signatures. *Bioinformatics* **30**, 335–342 (2014).
29. B. Li, Y. T. Yang, J. A. Capra, M. B. Gerstein, Predicting changes in protein thermodynamic stability upon point mutation with deep 3D convolutional neural networks. *PLoS Comput. Biol.* **16**, e1008291 (2020).
30. A. Broom, K. Trainor, Z. Jacobi, E. M. Meiering, Computational modeling of protein stability: quantitative analysis reveals solutions to pervasive problems. *Structure* **28**, 717–726 (2020).
31. A. Broom, Z. Jacobi, K. Trainor, E. M. Meiering, Computational tools help improve protein stability but with a solubility tradeoff. *J. Biol. Chem.* **292**, 14349–14361 (2017).

32. S. Reeves, S. Kalyaanamoorthy, Zero-shot transfer of protein sequence likelihood models to thermostability prediction. *Nat. Mach. Intell.* **6**, 1063–1076 (2024).
33. D. M. Fowler, S. Fields, Deep mutational scanning: a new style of protein science. *Nat. Methods* **11**, 801–807 (2014).
34. X. Huang, W. Zheng, R. Pearce, Y. Zhang, SSIPe: accurately estimating protein–protein binding affinity change upon mutations using evolutionary profiles in combination with an optimized physical energy function. *Bioinformatics* **36**, 2429–2437 (2020).
35. K. Tsuboyama, *et al.*, Mega-scale experimental analysis of protein folding stability in biology and design. *Nature* **620**, 434–444 (2023).
36. R. Nikam, A. Kulandaisamy, K. Harini, D. Sharma, M. M. Gromiha, ProThermDB: thermodynamic database for proteins and mutants revisited after 15 years. *Nucleic Acids Res.* **49**, D420–D424 (2021).
37. J. S. Xavier, *et al.*, ThermoMutDB: a thermodynamic database for missense mutations. *Nucleic Acids Res.* **49**, D475–D479 (2021).

Figures and Tables

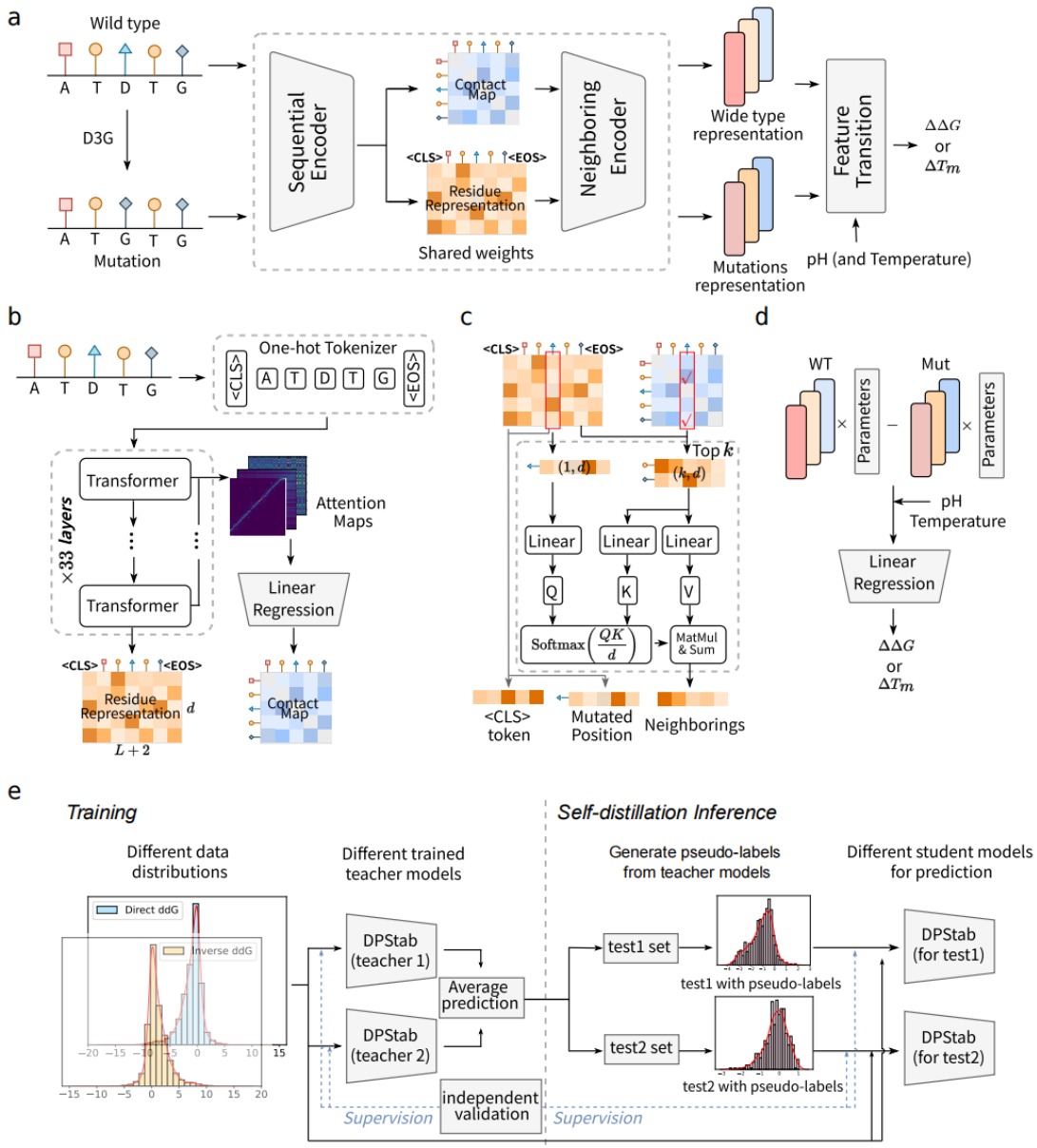

Figure 1. Model architecture of DPStab. **a.** The overview of DPStab. It consists of three modules. The wild-type protein sequence and mutated protein sequence are fed into the first two modules, respectively, to generate the features of the whole protein, mutated position, and its neighboring information. Subsequently, the features of wild-type and mutated proteins are combined in a feature transition module, with environmental parameters (pH and temperature), to predict the difference of stability (ΔG or ΔT_m) between two proteins. **b.** The architecture of the sequential encoder module. **c.** The architecture of the neighboring encoder module. **d.** The details of the feature transition module. **e.** The details of self-distillation for different test datasets.

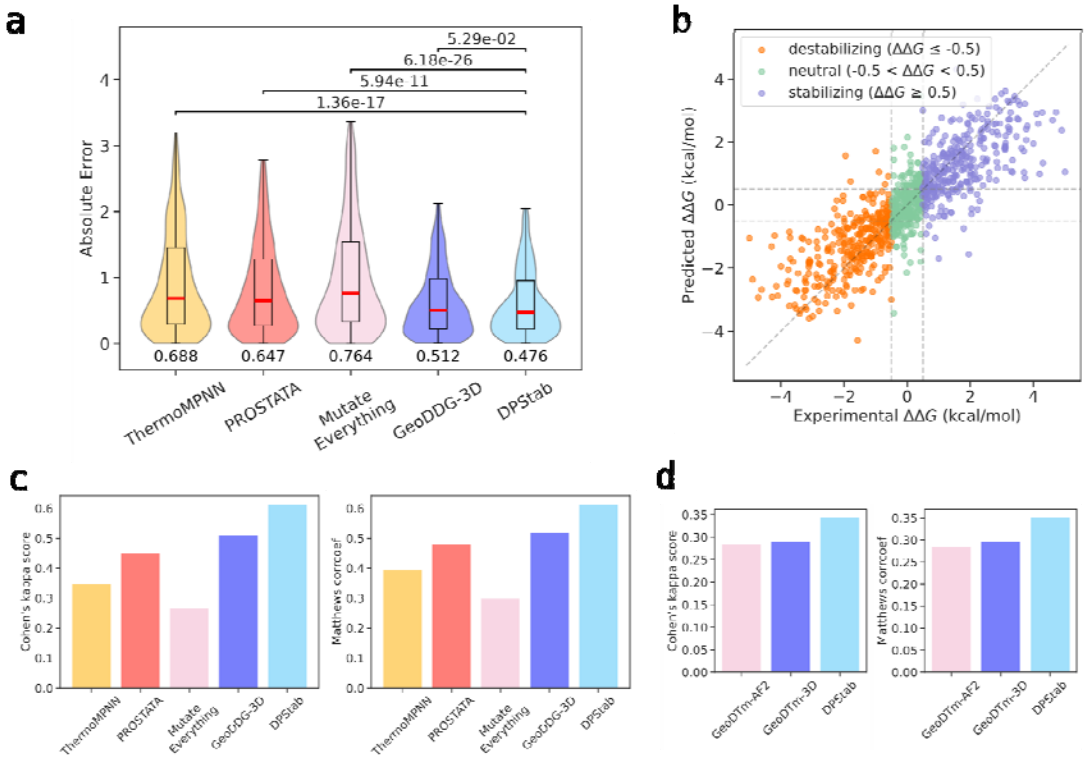

Figure 2. Performance comparison of DPStab. **a.** The violin plot of DPStab and other representative methods in terms of absolute error. The median values are denoted by the red lines. The first and third quartiles are indicated by the bounds of the box. The whiskers represent the 0.8 inter quartile range (IQR). Two-side paired t-tests are conducted between DPStab and other methods, and the P-values are annotated at the top of the violins. **b.** The detailed predictions of DPStab. The mutations are divided into three categories with different thresholds. **c.** The performance of DPStab and other representative methods in terms of MCC and κ score for prediction. **d.** The performance of DPStab and other representative methods in terms of MCC and *κ* score for prediction.

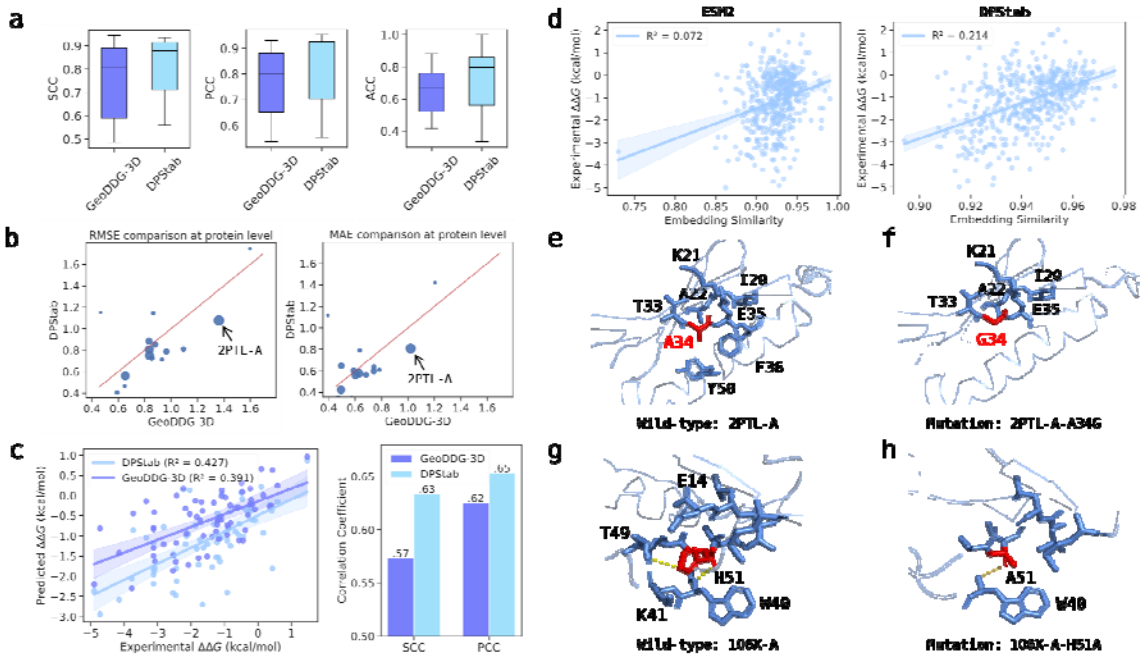

Figure 3. Performance comparison and analysis at protein-level. **a.** The performance comparison of DPStab and other methods on 15 individual proteins in terms of SCC, PCC and ACC. **b.** The performance comparison of DPStab and other methods on 15 individual proteins in terms of RMSE and MAE, where each point represents a protein, and the point size indicates the number of mutations. **c.** Comparison of detailed predictions of 2PTL-A by DPStab and other methods in terms of correlation. **d.** The correlation between representation similarity and experimental in original ESM2 and DPStab, where the similarity is calculated by single mutations and wild-type protein 2PTL-A. **e.** Local structure of 2PTL-A near the mutation site Ala34, where the residues with annotations are contacted with the wild-type residue A34. **f.** Local structure of 2PTL-A mutation (A34G) near the mutation site Gly34, where the residues with annotations are contacted with the mutated residue G34. **g.** Local structure of 1O6X-A near the mutation site H51, where the yellow lines indicate the hydrogen bonds, and the residues with annotations are associated with changes before and after the mutation H51A. **h.** Local structure of 2PTL-A mutation (A34V) near the mutation site Val34, where the yellow lines indicate the hydrogen bonds, and the residues with annotations are associated with changes before and after the mutation H51A.

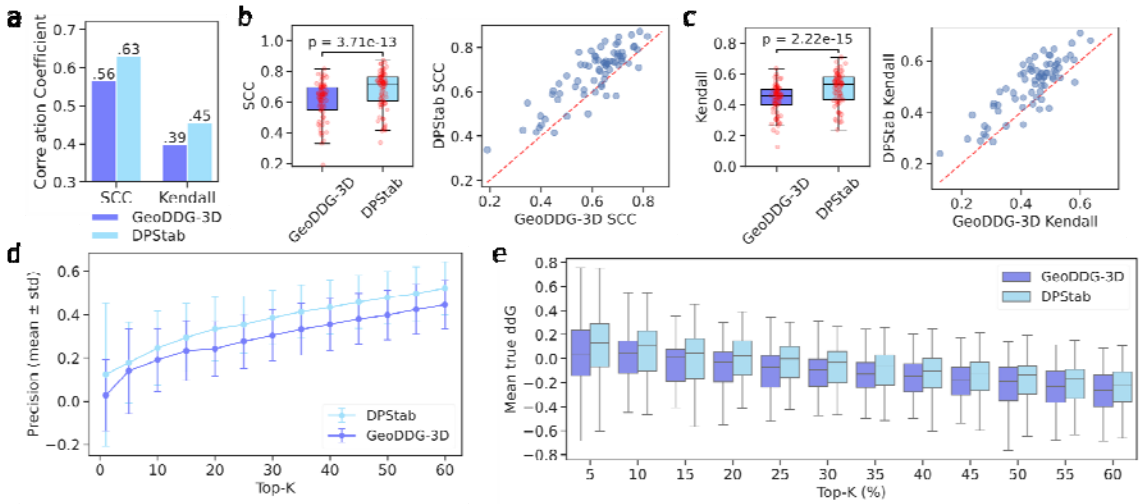

Figure 4. Benchmarking on large-scale DMS data. **a.** The overall performance comparison on 20000 inferred mutations of DPStab and other tools. **b.** The performance comparison on 73 individual proteins of DPStab and other methods in terms of SCC, where each point in the box plot or scatter plot represents one protein with corresponding several mutations. **c.** The performance comparison on 73 individual proteins of DPStab and other methods in terms of Kendall rank correlation coefficient, where each point in the box plot or scatter plot represents one protein with corresponding several mutations. **d.** The accuracy of top-k predicted mutations on 73 individual proteins, where the point indicates the mean accuracy across these proteins and error bars correspond to its standard deviation (std). **e.** The box plot of true values corresponding to the prediction results of top-k proportions, where higher values represent the predicted results more stabilizing.

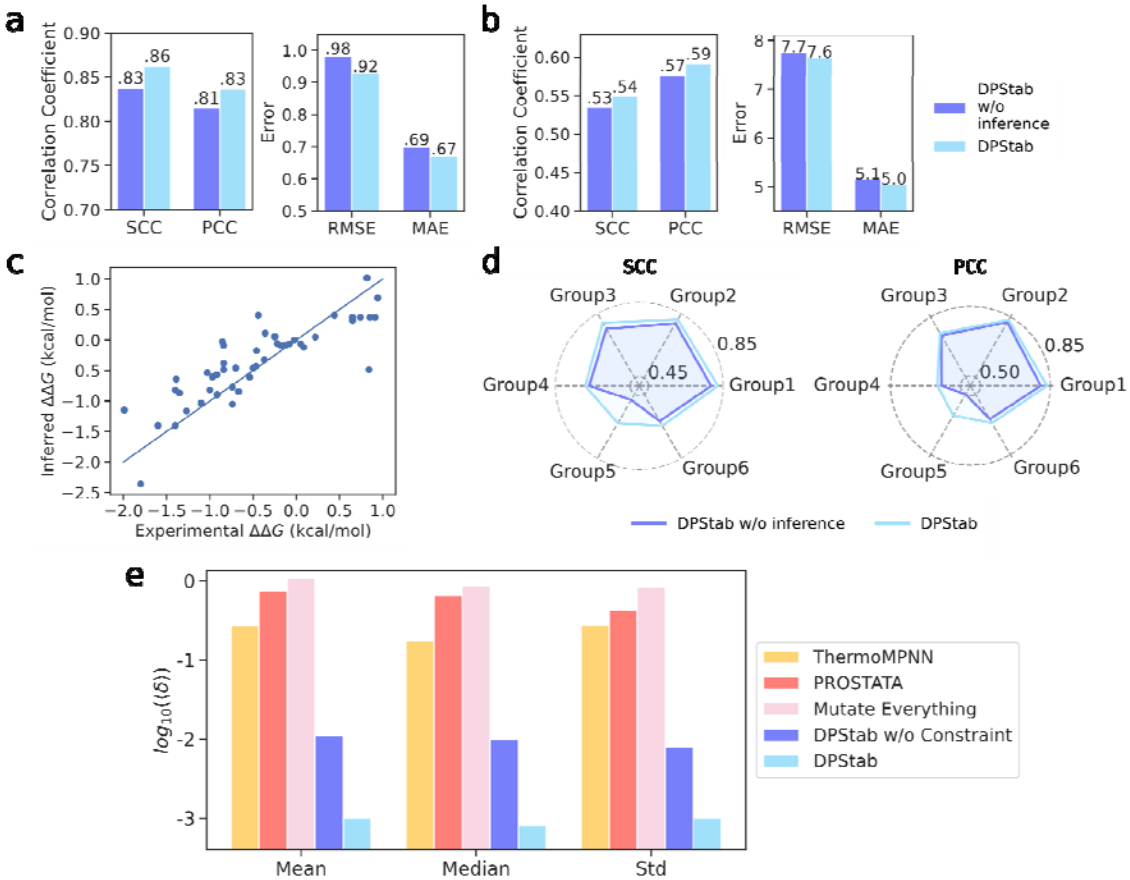

Figure 5. The role analysis of self-distillation strategies. **a.** The performance comparison of DPStab and DPStab without self-distillation strategies (inference stage), in correlation coefficient (left panel, including SCC and PCC), error ranges (right panel, RMSE and MAE) for prediction. **b.** The performance comparison of DPStab and DPStab without self-distillation strategies (inference stage), in correlation coefficient (left panel, including SCC and PCC), error ranges (right panel, RMSE and MAE) for prediction. **c.** The correlation between experiment and inferred from cDNA display proteolysis. **d.** The performance comparison of DPStab and DPStab without self-distillation strategies (inference stage) on different test groups from cDNA display proteolysis data. The left panel and right panel indicate SCC and PCC, respectively. **e.** The antisymmetric error comparison for direct and inverse mutations.

Table 1. Overall comparison of $\Delta\Delta G$ prediction on S461 in terms of SCC, PCC, RMSE, MAE, and ACC.

Method	SCC	PCC	RMSE	MAE	ACC
Structure-based					
mCSM[a]	0.394	0.464	1.710	1.307	0.423
FoldX[a]	0.526	0.440	2.003	1.262	0.579
ThermoNet[a]	0.604	0.673	1.270	0.969	0.555
DDGun-3D[a]	0.722	0.755	1.138	0.826	0.629
ThermoMPNN	0.659	0.637	1.344	0.985	0.555
GeoDDG-AF2[b]	0.830	0.824	0.984	0.728	0.662
GeoDDG-3D[b]	0.831	0.805	1.008	0.722	0.672
Sequence-based					
PROSTATA	0.786	0.777	1.206	0.892	0.629
DDGun[a]	0.755	0.752	1.276	0.945	0.637
Mutate Everything	0.539	0.568	1.490	1.091	0.502
DPStab	**0.860**	**0.837**	**0.926**	**0.671**	**0.742**

[a] indicates that the raw results are taken from a prior benchmark. [b] refers that the raw results are taken from its original paper.

Table 2. Overall comparison of ΔT_m prediction on S571 in terms of SCC, PCC, RMSE, MAE, and ACC.

Method	SCC	PCC	RMSE	MAE	ACC
GeoDTm-AF2[a]	0.536	0.518	8.112	5.546	0.522
GeoDTm-3D[a]	0.547	0.535	8.034	5.308	0.525
DPStab	**0.549**	**0.592**	**7.643**	**5.050**	**0.561**

[a] refers that the raw results are taken from its original paper.