Tsinghua Science and Technology
Doi: 10.26599/TST.2026.9010100

Research Article

Self-distillation and antisymmetric constraints enable accurate prediction of protein stability changes from single mutations

Wenkang Wang, Wentao Yu, Qiurong Yang, Ruiqing Zheng (✉), and Min Li (✉)

Abstract Accurately predicting protein stability changes upon residue-level mutations is essential for protein engineering and design. However, previous computational methods are typically designed for specific tasks and struggle to generalize across different stability systems, preventing the accurate guidance of protein mutagenesis. Sequence-based methods are easier to apply to large-scale protein datasets since they do not rely on high-quality 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 DeepTherm, a sequence-based deep learning solution to model changes in folding free energy ($\Delta\Delta G$) and melting temperature (ΔT_m) resulting from single-point mutations. DeepTherm transfers a protein large language model as a core component and incorporates a cross-attention mechanism to capture the contact-map changes around mutated positions. To address data imbalance and the antisymmetric nature of mutation effects, DeepTherm employs a self-distillation inference strategy under the supervision of an antisymmetric constraint. Benchmarking across multiple datasets shows that DeepTherm achieves state-of-the-art accuracy in both $\Delta\Delta G$ and ΔT_m predictions. Furthermore, practical evaluations confirm DeepTherm's capability in accurately ranking protein stability on large-scale datasets and effectively identifying critical structural contacts impacting stability. Detailed analysis on cDNA display proteolysis data further highlights the critical importance of self-distillation and antisymmetric constraints in enhancing predictive performance.

Keywords protein thermal stability, large language model, self-distillation, deep learning

1 Introduction

The thermal stability of a protein is essential for maintaining its biological functions in living organisms [1, 2]. Protein stability is closely linked to its amino acid sequence and three-dimensional structure, both of which can be significantly altered by single residue mutations, thereby disrupting the overall stability and potentially impairing functions [3, 4]. Consequently, introducing specific mutations to enhance protein stability has become a common strategy, particularly in industrial and therapeutic protein engineering [5, 6]. However, brute-force mutagenesis approaches are both time-consuming and costly. Therefore, developing computational methods for accurately predicting protein stability changes upon residue mutations can reveal crucial insights into the relationships between protein sequence, structure, stability, and function, offering valuable guidance for rational protein design [7].

In the context of protein stability change prediction, stability is typically quantified by the Gibbs free energy (ΔG) and melting temperature (T_m). Specifically, ΔG represents the free energy difference between the folded and unfolded states ($G_{unfolded} - G_{folded}$) and T_m denotes the midpoint of thermal denaturation. Correspondingly, stability changes upon mutation are commonly expressed as changes in Gibbs free energy ($\Delta\Delta G$) and melting temperature (ΔT_m), which are both antisymmetric: $\Delta\Delta G_{x\to y} = -\Delta\Delta G_{y\to x}$ and $\Delta T_{m,x\to y} = -\Delta T_{m,y\to x}$. In this study, higher $\Delta\Delta G$ and ΔT_m indicate stabilizing mutations.

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 Mutate Everything [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 insights 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

• Wenkang Wang, Wentao Yu, Qiurong Yang, Ruiqing Zheng, and Min Li are with the School of Computer Science and Engineering, Central South University, Changsha 410083, China. Email:rqzheng@csu.edu.cn, limin@mail.csu.edu.cn
✉To whom correspondence should be addressed.
Manuscript received: 2026-10-01; revised: 2026-10-23; accepted: 2026-10-26

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 massive 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] transfers a structure-based pLLM [24] from the mega-scale data to the protein stability data. In contrast, PROSTATA [22] adopts a sequence-based pLLM [20] as the backbone and uses different regression heads to predict protein stability changes. However, most existing pLLMs and structure modeling methods treat each protein in isolation during pre-training, which limits their ability to distinguish the single mutation effects [18]. Although the overall protein backbone may remain globally unchanged upon single mutations, the subtle yet crucial alterations in local residue contacts around mutation sites are often critical determinants of stability. These local impacts are challenging to capture for models that primarily rely on global protein information.

Recently, distillation on existing LLMs shows great potential to enhance the performance of LLMs for specific downstream tasks [25] and has made much progress in various areas [26], such as DeepSeek-R1 [27], providing new perspectives for LLM transfer. Here, we propose DeepTherm, a protein sequence-based deep learning method for accurate prediction of protein stability changes. A key strength of DeepTherm lies in its ability to extract residue-level features, infer contact patterns, and integrate local contextual information around mutation sites within the architecture of a pLLM. To further enhance predictive accuracy, we introduce a self-distillation inference strategy and an antisymmetric constraint tailored to the nature of mutation effects. Comprehensive benchmarks and data analyses demonstrate that DeepTherm achieves state-of-the-art accuracy for both $\Delta\Delta G$ and ΔT_m prediction.

2 Methods

The overall DeepTherm pipeline is illustrated in Fig. 1a. Starting from a pair of proteins (wild-type and mutant protein sequences), DeepTherm first leverages a set of transformer layers to extract comprehensive evolutionary and conformational information from both the wild-type and mutant protein sequences (Fig. 1b). These layers process each sequence separately, generating their respective residue representations and inferring their contact maps. DeepTherm then introduces a neighboring encoder to capture the local structural changes around the mutation site (Fig. 1c). It cross-incorporates information between the residue representations and inferred contact information, assigns weighted attention to neighboring residues, and adaptively integrates neighboring information. Finally, DeepTherm predicts thermal stability changes through a feature transition module (Fig. 1d), which compares the wild-type and mutant protein representations and incorporates external environment factors to predict $\Delta\Delta G$ or ΔT_m.

Additionally, at the inference stage, we propose a self-distillation strategy to enrich the training data (Fig. 1e) and incorporate a constraint to align DeepTherm with the antisymmetric nature of mutations. Specifically, multiple DeepTherm models are first trained separately on training data with different distributions. 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 an independent validation set. Meanwhile, a contrastive loss is applied to make DeepTherm produce opposite outputs for direct and inverse mutations.

2.1 Architecture of sequential encoder

To learn evolutionary and conformational information of protein sequence, DeepTherm adopts the same sequential encoder as ESM2 and utilizes the pre-trained weights of ESM2 as its initial parameters (Fig. 1b), 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 $p = \{r_1, r_2, ..., r_L\}$ with L residues, where $r_i \in \{A, C, ..., Y\}$ includes 20 standard amino acids. It is first fed into a token embedding layer:

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

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

where $Tokens_p$ denotes the token sequence representing p, $\langle CLS\rangle$ and $\langle EOS\rangle$ are two additional signs indicating the beginning and end of protein sequences, and d is the representation dimension. 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:

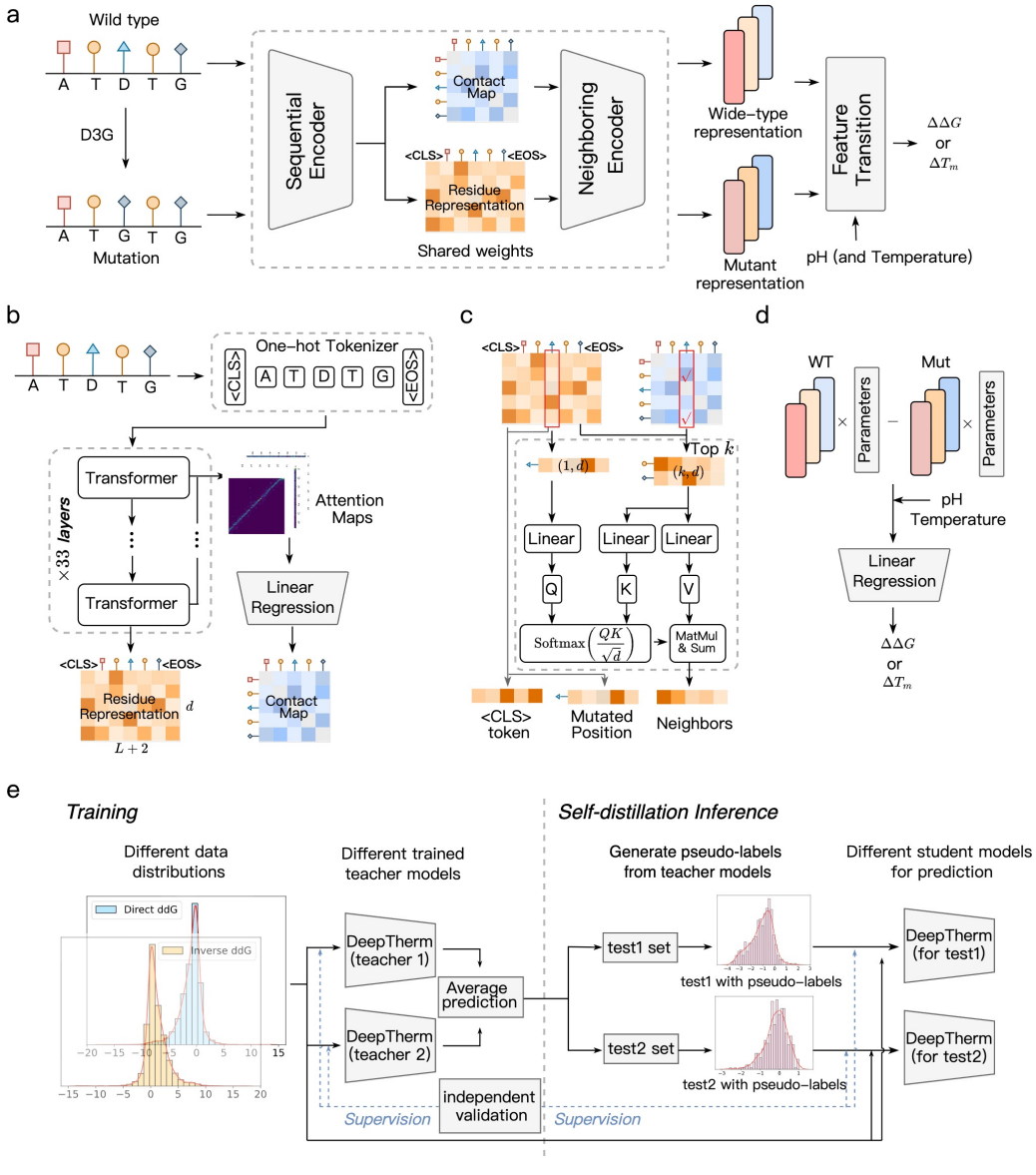


Fig. 1 Model architecture of DeepTherm. **a.** The overview of DeepTherm. It consists of three modules. Both the wild-type and mutant protein sequences are processed by the sequential and neighboring encoders to obtain representations of the whole protein, the mutation site, and its neighbors. Subsequently, 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 ($\Delta\Delta G$ 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.

$$H_{l+1} = MLP\left(SelfAttention\left(H_l\right) + H_l\right) \tag{3}$$

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

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

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

2.2 Architecture of neighboring encoder

Through sequential encoder, DeepTherm generates the residue representations H_n and corresponding contact map CM from the target protein sequence p. Based on these features, DeepTherm introduces a cross-attention module to extract the neighboring information of mutated site m ($1 \leqslant m \leqslant L$). As shown in Fig. 1c, in the contact map CM, the top-k neighboring residues with the highest contact probabilities with the mutation site m are selected. Subsequently, the representations of these k neighbors (denoted as $H_n\left[neighbors(m),:\right] \in \mathbb{R}^{k\times d}$), 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:

$$\begin{cases} Q = H_n[m,:] \times W_Q \\ K = H_n\left[neighbors(m),:\right] \times W_K \\ V = H_n\left[neighbors(m),:\right] \times W_V \end{cases} \tag{5}$$

$$A = \text{Softmax}\left(\frac{QK^{\mathsf{T}}}{\sqrt{d}}\right) \tag{6}$$

$$X_{\text{temp}} = AV \tag{7}$$

$$X_{\text{neighbors}} = MLP(X_{\text{temp}}) + X_{\text{temp}} \tag{8}$$

where $\left(K \in \mathbb{R}^{k\times d}, V \in \mathbb{R}^{k\times d}\right)$ is linearly transferred from the k neighbors and $Q \in \mathbb{R}^{1\times d}$ is from the mutated site. Through this cross-attention mechanism, DeepTherm 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 $LayerNorm(\cdot)$ are also employed here to prevent the loss of features.

2.3 Architecture of feature transition module

Finally, through two previous encoders, DeepTherm 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, as illustrated in Fig. 1d, is applied to learn the difference between the two proteins:

$$\begin{aligned} X_{wild} = Concat\big(&H_n^{wild}[0,:] \odot W_{wild}, \\ &H_n^{wild}[m,:] \odot W_{wild}, \\ &X_{neighbors}^{wild} \odot W_{wild}\big) \end{aligned} \tag{9}$$

$$\begin{aligned} X_{mut} = Concat\big(&H_n^{mut}[0,:] \odot W_{mut}, \\ &H_n^{mut}[m,:] \odot W_{mut}, \\ &X_{neighbors}^{mut} \odot W_{mut}\big) \end{aligned} \tag{10}$$

$$X_{diff} = MLP\left(X_{wild} - X_{mut}\right) \tag{11}$$

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 and T_m changes upon single-point mutation:

$$X_{env} = MLP\left([pH, temperature]\right) \tag{12}$$

$$\Delta\Delta G_{pred} = MLP\left(Concat\left(X_{diff}, X_{env}\right)\right) \tag{13}$$

$$\Delta T_m = MLP\left(Concat\left(X_{diff}, X_{env}\right)\right) \tag{14}$$

where pH and temperature are both required for $\Delta\Delta G$, while ΔT_m only needs pH.

2.4 Self-distillation strategy and model training

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

$$MSELoss = \frac{1}{N}\sum_{i=1}^{N}\left(y_{pred}^{i} - y_{true}^{i}\right)^2 \tag{15}$$

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

$$AtSymmConstraint = \sum_{i=1}^{N} \left| y_{pred}^{i,wild\to mut} + y_{pred}^{i,mut\to wild} \right| \tag{16}$$

$$Loss = MSELoss + AtSymmConstraint \tag{17}$$

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 DeepTherm student model is trained to produce the final predictions. Throughout training and inference, the hidden dimension d is set to 1280. The model is optimized using the AdamW optimizer with a dynamically decaying learning rate. During the first two training epochs, the ESM2 Transformer layers are frozen. They are subsequently unfrozen, allowing the entire model to be jointly optimized in an end-to-end manner. DeepTherm will stop training if the loss value of validation data is stable or trending upwards.

All experiments of DeepTherm are carried out using one NVIDIA Tesla H100 GPU card with 94 GB of memory. The training process takes around 15 hours with a batch size of 4.

3 Experiments and results

3.1 Datasets

The $\Delta\Delta G$ and ΔT_m training datasets used in this article are derived from the previous study [16], named S8754 and S4346, respectively, which are collected from the ProThermDB [28] and ThermoMutDB [29] 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 [18]. 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 [16]. 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 $\Delta\Delta G_{wild\to mut} = -\Delta\Delta G_{mut\to wild}$, 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 17508 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 [30], 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 randomly select 150 mutations per protein to construct the test set. In total, six test sets were generated, each containing approximately 450 mutations (Fig. S3 in the ESM).

3.2 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 the predicted values Y_{pred} and true labels Y_{true}, SCC is defined as:

$$SCC(Y_{pred}, Y_{true}) = \frac{\mathrm{Cov}[Rank(Y_{pred}), Rank(Y_{true})]}{\sigma_{Rank(Y_{pred})}\sigma_{Rank(Y_{true})}} \tag{18}$$

where $Rank(\cdot)$ denotes the ranks of the elements in a given list, with average ranks assigned to tied values. Cov denotes the covariance between two lists, and σ represents the standard deviation. PCC is calculated similarly to SCC but directly measures the linear correlation between the original values:

$$PCC(Y_{pred}, Y_{true}) = \frac{\mathrm{Cov}[Y_{pred}, Y_{true}]}{\sigma_{Y_{pred}}\sigma_{Y_{true}}} \tag{19}$$

Both SCC and PCC range from -1 to 1, with higher values indicating stronger positive correlations between the predicted and true values. RMSE and MAE reflect the error distances between predictions and labels:

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

Table 1 Comparison of $\Delta\Delta G$ prediction on S461 in terms of SCC, PCC, RMSE, MAE, and ACC for Overall, Direct, and Inverse evaluations.

Method	Overall					Direct					Inverse				
	SCC	**PCC**	**RMSE**	**MAE**	**ACC**	**SCC**	**PCC**	**RMSE**	**MAE**	**ACC**	**SCC**	**PCC**	**RMSE**	**MAE**	**ACC**
Structure-based															
mCSM[a]	0.394	0.464	1.710	1.307	0.423	0.514	0.535	1.070	0.803	0.670	0.123	0.176	2.168	1.812	0.171
FoldX[a]	0.526	0.440	2.003	1.262	0.579	0.349	0.224	2.232	1.382	0.601	0.447	0.407	1.743	1.141	0.557
ThermoNet[a]	0.604	0.673	1.270	0.969	0.555	0.487	0.556	1.228	0.925	0.555	0.430	0.516	1.311	1.013	0.555
DDGun-3D[a]	0.722	0.755	1.138	0.826	0.629	0.583	0.636	1.101	0.800	0.629	0.551	0.587	1.174	0.851	0.629
ThermoMPNN	0.659	0.637	1.344	0.985	0.555	0.453	0.456	1.280	0.927	0.607	0.466	0.480	1.406	1.044	0.503
GeoDDG-AF2[b]	0.830	0.824	0.984	0.728	0.662	0.682	**0.690**	0.983	0.728	0.662	0.682	**0.690**	0.984	0.728	0.662
GeoDDG-3D[b]	0.831	0.805	1.008	0.722	0.672	0.669	0.668	1.008	0.722	0.672	0.669	0.668	1.008	0.722	0.672
Sequence-based															
PROSTATA	0.786	0.777	1.206	0.892	0.629	0.656	0.664	0.961	0.697	0.716	0.533	0.566	1.410	1.086	0.542
DDGun[a]	0.755	0.752	1.276	0.945	0.637	0.569	0.588	1.251	0.939	0.644	0.546	0.551	1.302	0.951	0.629
Mutate Everything	0.539	0.568	1.490	1.091	0.502	0.630	0.622	1.035	0.752	0.681	0.135	0.164	1.835	1.429	0.323
DeepTherm	**0.860**	**0.837**	**0.926**	**0.671**	**0.742**	**0.690**	0.680	**0.926**	**0.671**	**0.742**	**0.690**	0.679	**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 Comparison of ΔT_m prediction on S571 in terms of SCC, PCC, RMSE, MAE, and ACC for Overall, Direct, and Inverse evaluations.

Method	Overall					Direct					Inverse				
	SCC	**PCC**	**RMSE**	**MAE**	**ACC**	**SCC**	**PCC**	**RMSE**	**MAE**	**ACC**	**SCC**	**PCC**	**RMSE**	**MAE**	**ACC**
GeoDTm-AF2[a]	0.536	0.518	8.112	5.546	0.522	**0.513**	0.462	8.112	5.546	0.522	**0.513**	0.462	8.112	5.546	0.522
GeoDTm-3D[a]	0.547	0.535	8.034	5.308	0.525	0.510	0.467	8.034	5.308	0.525	0.510	0.467	8.034	5.308	0.525
DeepTherm	**0.549**	**0.592**	**7.643**	**5.050**	**0.561**	0.512	**0.526**	**7.642**	**5.050**	**0.562**	0.512	**0.526**	**7.643**	**5.050**	**0.560**

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

$$ACC = \sum_{c \in C} \frac{tp_c}{N} \tag{20}$$

$$MCC = \frac{\sum_c tp_c \times N - \sum_c p_c \times n_c}{\sqrt{(N^2 - \sum_c p_c^2)\,(N^2 - \sum_c n_c^2)}} \tag{21}$$

$$\kappa = \frac{N \sum_c tp_c - \sum_c p_c n_c}{N^2 - \sum_c p_c n_c} \tag{22}$$

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.

3.3 DeepTherm outperforms existing methods overall in $\Delta\Delta G$ and ΔT_m prediction.

To comprehensively evaluate the overall performance of DeepTherm in $\Delta\Delta G$ prediction, we benchmark DeepTherm against seven structure-based (FoldX [8], DDGun-3D [13, 14], GeoDDG-3D [16], GeoDDG-AF2 [16], ThermoMPNN [23], mCSM [31], ThermoNet [32]) and three sequence-based (DDGun [13, 14], Mutate Everything [15], PROSTATA [22]) approaches.

Table 1 presents the performance of various methods on the S461 dataset. DeepTherm achieved the lowest errors, with an RMSE of 0.926 and an MAE of 0.671, compared with the best values among the other methods (RMSE 0.984; MAE 0.722). It also achieved the highest SCC of 0.86 and PCC of 0.84. In Fig. 2a, we examine the absolute errors of individual mutations predicted by DeepTherm. The median of absolute errors of DeepTherm is significantly lower than other sequence-based and structure-based approaches, where the corresponding P-values confirm the statistical significance of these improvements. Additionally, we further evaluated the ability of these methods to identify stabilizing mutations. As shown in the Fig. S2 of the ESM, the top-ranked mutations identified by DeepTherm generally exhibit greater experimentally measured stability improvements, indicating that it can effectively predict stable mutations.

In certain scenarios such as soluble protein design, classifying mutations rather than predicting exact $\Delta\Delta G$ values is more practically useful [33–35]. To establish a classification task, we categorize mutations into three classes: destabilizing ($\Delta\Delta G \leqslant$ -0.5 kcal/mol), stabilizing ($\Delta\Delta G \geqslant$ 0.5 kcal/mol) and neutral. DeepTherm achieves a significantly

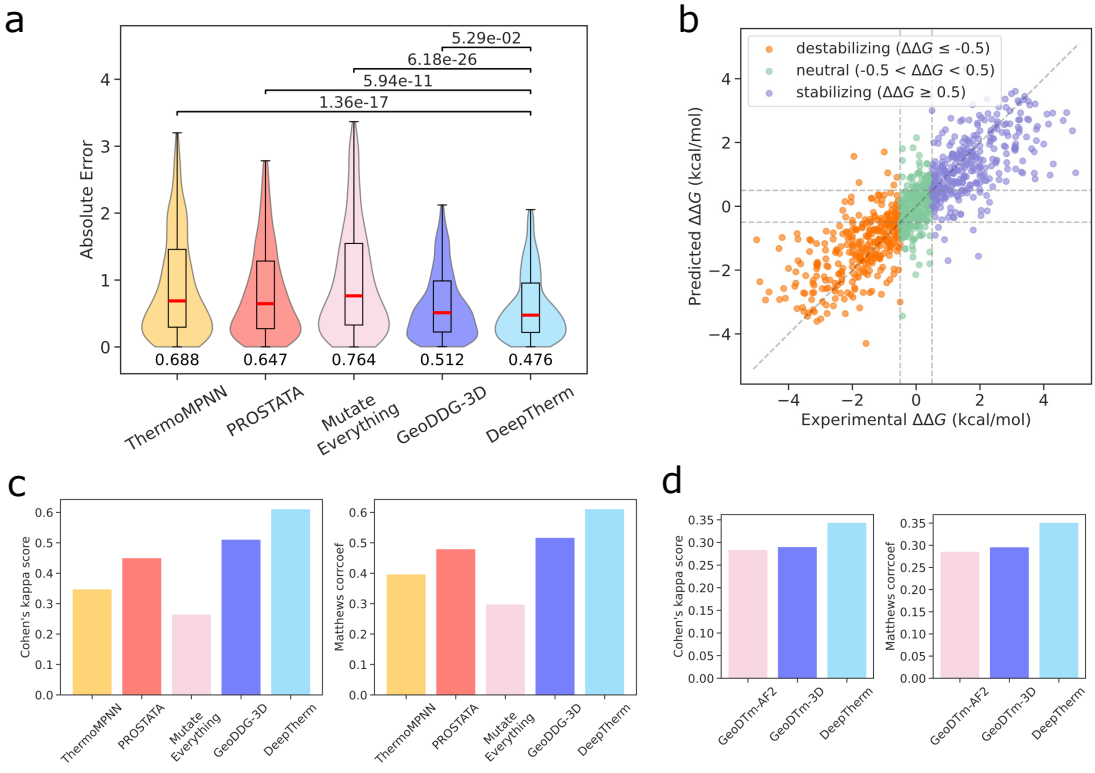


Fig. 2 Performance comparison of DeepTherm. **a.** The violin plot of DeepTherm and other representative methods in terms of absolute error. The median values are denoted by the red lines. The first and third quartiles are indicated by the bounds of the box. The whiskers represent the 0.8 interquartile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

higher ACC than other methods (Table 1). Among the 922 mutations (comprising 461 direct and 461 inverse mutations), DeepTherm correctly predicts the mutation classes for 684 cases (Fig. 2b). Additionally, DeepTherm achieves the highest MCC and κ (both > 0.61), whereas the second-best method, GeoDDG-3D, achieves only 0.51 (Fig. 2c), highlighting its superior performance in mutation classification.

Furthermore, we compare DeepTherm with GeoDTm-AF2 [16] and GeoDTm-3D [16] on ΔT_m prediction. Interestingly, DeepTherm achieves a comparable SCC to GeoDTm-AF2 and GeoDTm-3D while significantly outperforming them in terms of PCC, RMSE and MAE (Table 2). For classification, we similarly categorized mutations into three groups: destabilizing ($\Delta T_m \leqslant$ -2 ℃), stabilizing ($\Delta T_m \geqslant 2$ ℃), and neutral. DeepTherm achieved an ACC of 0.561, compared with GeoDTm-AF2 (0.522) and GeoDTm-3D (0.525) (Table 2). Moreover, the evaluation on Cohen's kappa score and MCC confirms the better performance of DeepTherm relative to the two GeoDTm models (Fig. 2d). Overall, our results indicate that DeepTherm achieves a stronger or comparable performance for both $\Delta\Delta G$ and ΔT_m prediction.

3.4 DeepTherm shows accurate ranking ability on individual proteins.

We evaluate DeepTherm's performance on individual proteins to investigate its capability for protein engineering. We independently evaluate its performance on individual proteins in S461. Accordingly, there are 388 mutations from 15 proteins used here; each protein has at least 6 mutations. DeepTherm exhibits substantial improvements in SCC, PCC, and ACC, compared to GeoDDG-3D (Fig. 3a). Specifically, it achieves remarkable median improvements of 8.6% (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, DeepTherm surpasses GeoDDG-3D across most proteins (Fig. 3b). A similar conclusion can also be obtained for ΔT_m prediction (Fig. S1 in the ESM). Collectively, these results highlight the advantages of DeepTherm in $\Delta\Delta G$ and ΔT_m at the protein level, which provides strong guidance for protein engineering.

We investigate protein 2PTL-A, which is a clear sample

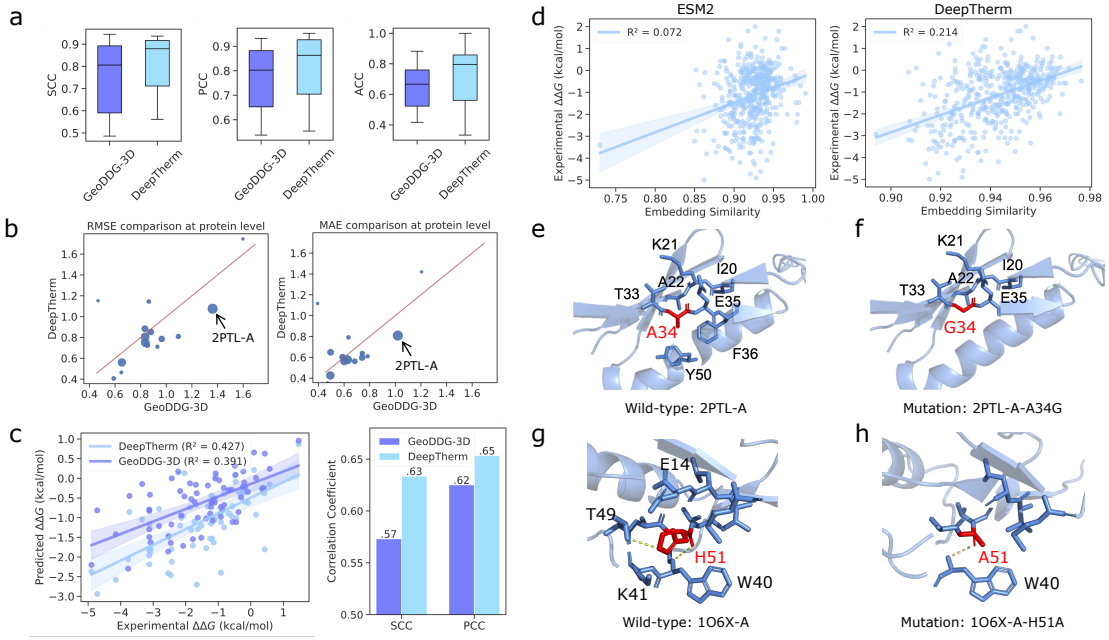


Fig. 3 Performance comparison and analysis at the protein level. **a.** The performance comparison of DeepTherm and other methods on 15 individual proteins in terms of SCC, PCC and ACC. **b.** The performance comparison of DeepTherm and other methods on 15 individual proteins in terms of RMSE and MAE, where each point represents a protein, and the point size indicates the number of mutations. **c.** Comparison of detailed predictions of 2PTL-A by DeepTherm and other methods in terms of correlation. **d.** The correlation between representation similarity and experimental $\Delta\Delta G$ in original ESM2 and DeepTherm, 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 1O6X-A mutation (H51A) near the mutation site Ala51, where the yellow lines indicate the hydrogen bonds, and the residues with annotations are associated with changes before and after the mutation H51A.

of DeepTherm outperforming GeoDDG-3D (Fig. 3b). Fig. 3c presents a head-to-head comparison of DeepTherm and GeoDDG-3D for $\Delta\Delta G$ prediction on all known mutations of 2PTL-A, which indicate that DeepTherm achieves higher R^2 values with an improvement of 9% over GeoDDG-3D, as well as SCC (increase 11%) and PCC (increase 5%). These demonstrate that DeepTherm can provide a more reliable rank list for mutations of 2PTL-A. To explore the reason for the DeepTherm 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 DeepTherm, the correlation between representation similarities and $\Delta\Delta G$ is significantly improved (Fig. 3d).

DeepTherm is not only able to identify positive mutations but also has high accuracy to predict negative mutations, while current structure-based predictors often fail to reliably identify destabilizing mutations. For instance, GeoDDG-3D misclassifies destabilizing variants such as A34G in 2PTL-A and H51A in 1O6X-A as neutral or stabilizing. In contrast, DeepTherm accurately captures both cases, demonstrating its higher resolution in identifying negative mutations. As shown in Fig. 3e and 3f, A34G is a curated destabilizing mutation ($\Delta\Delta G$ = -2.1 kcal/mol), which GeoDDG-3D predicts as nearly neutral ($\Delta\Delta G$ = -0.3 kcal/mol), whereas DeepTherm outputs a substantially more accurate prediction ($\Delta\Delta G$ = -1.2 kcal/mol). Structural analysis via FoldX reveals that replacing alanine with glycine at position 34 removes the C_β atom, reducing solvent accessible surface area (SASA) and local backbone flexibility. Furthermore, the smaller molecular volume of glycine weakens contacts with neighboring residues, particularly Y50 and the hydrophobic residue F36, thereby diminishing hydrophobic interactions and reducing local stability (Fig. 3f).

A similar pattern is observed for the H51A mutation in protein 1O6X-A (Fig. 3g-h). While DeepTherm accurately predicts its destabilizing effect ($\Delta\Delta G$ = -0.8 kcal/mol), GeoDDG-3D incorrectly classifies it as stabilizing ($\Delta\Delta G$ = 0.6 kcal/mol). In the wild-type structure, H51 forms hydrogen

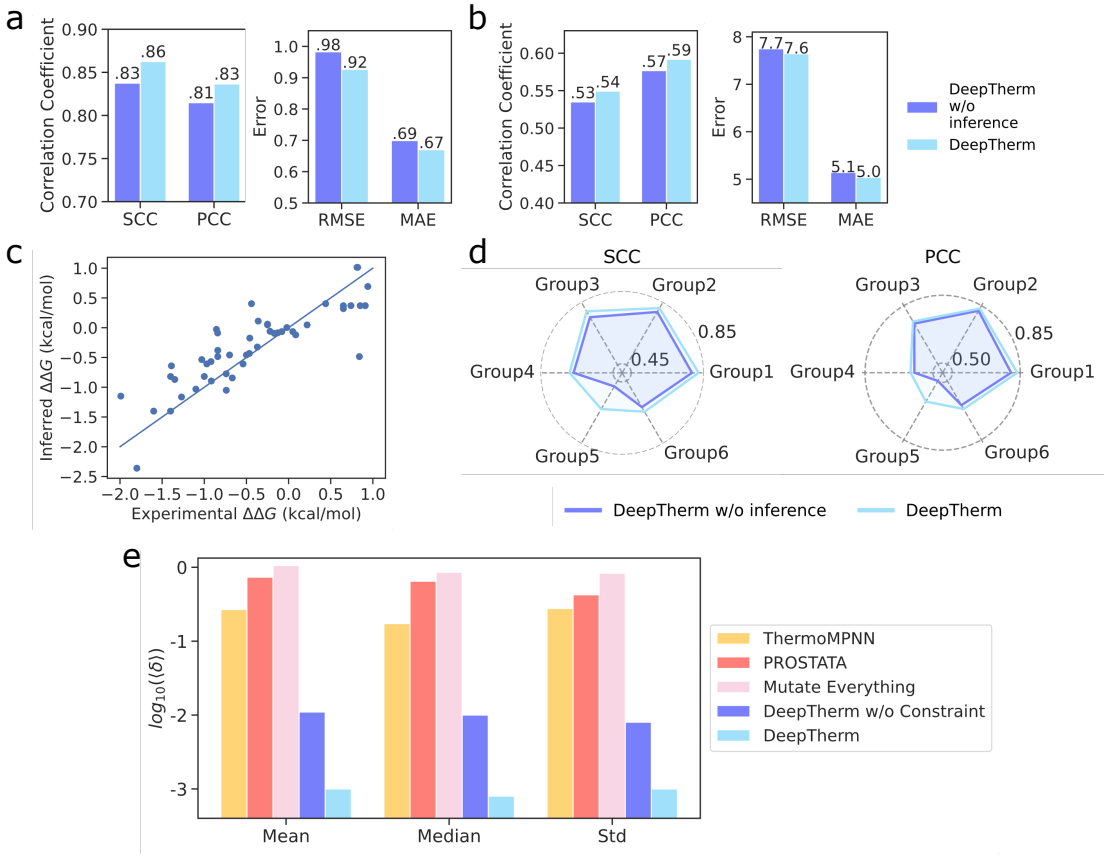


Fig. 4 The role analysis of self-distillation strategies. **a.** The performance comparison of DeepTherm and DeepTherm without self-distillation strategies (inference stage), in correlation coefficient (left panel, including SCC and PCC), error ranges (right panel, RMSE and MAE) for $\Delta\Delta G$ prediction. **b.** The performance comparison of DeepTherm and DeepTherm without self-distillation strategies (inference stage), in correlation coefficient (left panel, including SCC and PCC), error ranges (right panel, RMSE and MAE) for ΔT_m prediction. **c.** The correlation between experimental $\Delta\Delta G$ and inferred $\Delta\Delta G$ from cDNA display proteolysis. **d.** The performance comparison of DeepTherm and DeepTherm 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.

bonds with W40 and T49. The substitution with alanine significantly reduces side-chain volume, abolishes the hydrogen bond with T49, and disrupts packing interactions with E14 and K41. The loss of the aromatic ring further destabilizes the local structure by creating a cavity. These structural features, which underlie the decrease in stability (experimental $\Delta\Delta G$ = -0.7 kcal/mol), are effectively captured by DeepTherm but overlooked by GeoDDG-3D.

3.5 Generalization of self-distillation inference and antisymmetric constraint in DeepTherm.

The self-distillation inference strategy can enhance training data diversity and improve the accuracy of existing trained models. After removing the inference stage, the performance will reduce in all metrics (Fig. 4a). 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 (Fig. 4b).

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 [30]. 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 (Fig. 4c). Consequently, we split all mutations into six groups by protein length and randomly sample three proteins with 450 mutations in each group (Fig. S3 in the ESM), resulting in six sampled independent test datasets (Table S3 in the ESM). Then, based on our trained teacher models, we first generate pseudo-labels for these test datasets and re-train corresponding DeepTherm models, respectively. As shown in Fig. 4d,

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 $\Delta\Delta G_{wt\to mut} = \Delta G_{mut} - \Delta G_{wt}$, direct and inverse mutations should produce exactly opposite $\Delta\Delta G$ values. Since DeepTherm does not employ a fully symmetrical network architecture, 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, DeepTherm 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 (Fig. 4e). Above all, the self-distillation strategy can improve the robustness of our model, and the antisymmetric constraint makes the predictions more practical.

4 Discussion

Accurately 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 DeepTherm. 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 DeepTherm. Notably, DeepTherm not only demonstrates strong overall performance, but also shows its advantages at the 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 [35]. Consequently, we adopt a self-distillation strategy at the inference stage to 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 DeepTherm, offering new perspectives into improving existing protein large language models (pLLMs) for specific downstream tasks. Contrastive loss can also enable an asymmetric network architecture to better learn the constraints of antisymmetric data.

Overall, DeepTherm only takes protein sequences as input, does not rely on high-quality structural data, and outperforms existing methods. However, it still has limitations. For example, it requires retraining based on the test dataset, which belongs to transductive learning and is time-consuming. Furthermore, another challenge that needs to be addressed is how to account for side-chain changes and incorporate additional geometric potential energy to help the model capture structural changes during the mutation process.

Acknowledgements

This work is supported in part by the National Key Research and Development Program of China (2023YFF1205500 to R.Z.), the National Natural Science Foundation of China (No. 625B2184 to W.W. and No. 62225209 to M.L.), and the High Performance Computing Center of Central South University.

Author contributions

Wenkang Wang and Min Li conceived and designed the research. Wenkang Wang designed and developed the model and performed the experiments. Qiurong Yang helped analyze the data. Wentao Yu and Ruiqing Zheng contributed to the discussion and experimental design. Wenkang Wang, Wentao Yu, Ruiqing Zheng, and Min Li wrote the manuscript.

Competing interests

The authors have no competing interests to declare that are relevant to the content of this article.

Electronic Supplementary Material (ESM)

The ESM can be seen from DeepTherm_SI.

Reference

[1] D. Eisenberg, E. M. Marcotte, I. Xenarios, and T. O. Yeates, Protein function in the post-genomic era, *Nature*, vol. 405, no. 6788, pp. 823–826, 2000. [Online]. Available: https://doi.org/10.1038/35015694

[2] W. Wang, Y. Shuai, M. Zeng, W. Fan, and M. Li, DPFunc: accurately predicting protein function via deep learning with domain-guided structure information, *Nature Communications*, vol. 16, no. 1, p. 70, 2025. [Online]. Available: https://doi.org/10.1038/s41467-024-54816-8

[3] N. Tokuriki and D. S. Tawfik, Stability effects of mutations and protein evolvability, *Current Opinion in Structural Biology*, vol. 19, no. 5, pp. 596–604, 2009. [Online]. Available: https://doi.org/10.1016/j.sbi.2009.08.003

[4] R. A. Studer, B. H. Dessailly, and C. A. Orengo, Residue mutations and their impact on protein structure and function: detecting beneficial and pathogenic changes, *Biochemical Journal*, vol. 449, no. 3, pp. 581–594, 2013. [Online]. Available: https://doi.org/10.1042/bj20121221

[5] K. L. Morley and R. J. Kazlauskas, Improving enzyme properties: when are closer mutations better? *Trends in Biotechnology*, vol. 23, no. 5, pp. 231–237, 2005. [Online]. Available: https://doi.org/10.1016/j.tibtech.2005.03.005

[6] N. G. Nezhad, R. N. Z. R. A. Rahman, Y. M. Normi, S. N. Oslan, F. M. Shariff, and T. C. Leow, Thermostability engineering of industrial enzymes through structure modification, *Applied Microbiology and Biotechnology*, vol. 106, no. 13–16, pp. 4845–4866, 2022. [Online]. Available: https://doi.org/10.1007/s00253-022-12067-x

[7] Y. Ming, W. Wang, R. Yin, M. Zeng, L. Tang, S. Tang, and M. Li, A review of enzyme design in catalytic stability by artificial intelligence, *Briefings in Bioinformatics*, vol. 24, no. 3, p. bbad065, 2023. [Online]. Available: https://doi.org/10.1093/bib/bbad065

[8] J. Delgado, L. G. Radusky, D. Cianferoni, and L. Serrano, FoldX 5.0: working with RNA, small molecules and a new graphical interface, *Bioinformatics*, vol. 35, no. 20, pp. 4168–4169, 2019. [Online]. Available: https://doi.org/10.1093/bioinformatics/btz184

[9] R. Pearce, X. Huang, D. Setiawan, and Y. Zhang, EvoDesign: Designing Protein–Protein Binding Interactions Using Evolutionary Interface Profiles in Conjunction with an Optimized Physical Energy Function, *Journal of Molecular Biology*, vol. 431, no. 13, pp. 2467–2476, 2019. [Online]. Available: https://doi.org/10.1016/j.jmb.2019.02.028

[10] X. Huang, R. Pearce, and Y. Zhang, EvoEF2: accurate and fast energy function for computational protein design, *Bioinformatics*, vol. 36, no. 4, pp. 1135–1142, 2020. [Online]. Available: https://doi.org/10.1093/bioinformatics/btz740

[11] L. Quan, Q. Lv, and Y. Zhang, STRUM: structure-based prediction of protein stability changes upon single-point mutation, *Bioinformatics*, vol. 32, no. 19, pp. 2936–2946, 2016. [Online]. Available: https://doi.org/10.1093/bioinformatics/btw361

[12] I. M. Hernández, Y. Dehouck, U. Bastolla, J. R. López-Blanco, and P. Chacón, Predicting protein stability changes upon mutation using a simple orientational potential, *Bioinformatics*, vol. 39, no. 1, p. btad011, 2023. [Online]. Available: https://doi.org/10.1093/bioinformatics/btad011

[13] L. Montanucci, E. Capriotti, Y. Frank, N. Ben-Tal, and P. Fariselli, DDGun: an untrained method for the prediction of protein stability changes upon single and multiple point variations, *BMC Bioinformatics*, vol. 20, no. Suppl 14, p. 335, 2019. [Online]. Available: https://doi.org/10.1186/s12859-019-2923-1

[14] L. Montanucci, E. Capriotti, G. Birolo, S. Benevenuta, C. Pancotti, D. Lal, and P. Fariselli, DDGun: an untrained predictor of protein stability changes upon amino acid variants, *Nucleic Acids Research*, vol. 50, no. W1, pp. W222–W227, 2022. [Online]. Available: https://doi.org/10.1093/nar/gkac325

[15] J. Ouyang-Zhang, D. Diaz, A. Klivans, and P. Kraehenbuehl, Predicting a Protein's Stability under a Million Mutations, in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 76 229–76 247. [Online]. Available: https://papers.neurips.cc/paper_files/paper/2023/hash/f03cb785864596fa5901f1359d23fd81-Abstract-Conference.html

[16] Y. Xu, D. Liu, and H. Gong, Improving the prediction of protein stability changes upon mutations by geometric learning and a pre-training strategy, *Nature Computational Science*, vol. 4, no. 11, pp. 840–850, 2024. [Online]. Available: https://doi.org/10.1038/s43588-024-00716-2

[17] Y. Chen, Y. Xu, D. Liu, Y. Xing, and H. Gong, An end-to-end framework for the prediction of protein structure and fitness from single sequence, *Nature Communications*, vol. 15, no. 1, p. 7400, 2024. [Online]. Available: https://doi.org/10.1038/s41467-024-51776-x

[18] C. Pancotti, S. Benevenuta, G. Birolo, V. Alberini, V. Repetto, T. Sanavia, E. Capriotti, and P. Fariselli, Predicting protein stability changes upon single-point mutation: a thorough comparison of the available tools on a new dataset, *Briefings in Bioinformatics*, vol. 23, no. 2, p. bbab555, 2022. [Online]. Available: https://doi.org/10.1093/bib/bbab555

[19] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, J. Ma, and R. Fergus, Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences, *Proceedings of the National Academy of Sciences*, vol. 118, no. 15, p. e2016239118, 2021. [Online]. Available:

https://doi.org/10.1073/pnas.2016239118

[20] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, and A. Rives, Evolutionary-scale prediction of atomic-level protein structure with a language model, *Science*, vol. 379, no. 6637, pp. 1123–1130, 2023. [Online]. Available: https://doi.org/10.1126/science.ade2574

[21] T. Hayes, R. Rao, H. Akin, N. J. Sofroniew, D. Oktay, Z. Lin, R. Verkuil, V. Q. Tran, J. Deaton, M. Wiggert, R. Badkundri, I. Shafkat, J. Gong, A. Derry, R. S. Molina, N. Thomas, Y. A. Khan, C. Mishra, C. Kim, L. J. Bartie, M. Nemeth, P. D. Hsu, T. Sercu, S. Candido, and A. Rives, Simulating 500 million years of evolution with a language model, *Science*, vol. 387, no. 6736, pp. 850–858, 2025. [Online]. Available: https://doi.org/10.1126/science.ads0018

[22] D. Umerenkov, F. Nikolaev, T. I. Shashkova, P. V. Strashnov, M. Sindeeva, A. Shevtsov, N. V. Ivanisenko, and O. L. Kardymon, PROSTATA: a framework for protein stability assessment using transformers, *Bioinformatics*, vol. 39, no. 11, p. btad671, 2023. [Online]. Available: https://doi.org/10.1093/bioinformatics/btad671

[23] H. Dieckhaus, M. Brocidiacono, N. Z. Randolph, and B. Kuhlman, Transfer learning to leverage larger datasets for improved prediction of protein stability changes, *Proceedings of the National Academy of Sciences*, vol. 121, no. 6, p. e2314853121, 2024. [Online]. Available: https://doi.org/10.1073/pnas.2314853121

[24] J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, and D. Baker, Robust deep learning–based protein sequence design using ProteinMPNN, *Science*, vol. 378, no. 6615, pp. 49–56, 2022. [Online]. Available: https://doi.org/10.1126/science.add2187

[25] X. Zhu, J. Li, Y. Liu, C. Ma, and W. Wang, A Survey on Model Compression for Large Language Models, *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 1556–1577, 2024. [Online]. Available: https://aclanthology.org/2024.tacl-1.85/

[26] C.-Y. Hsieh, C.-L. Li, C.-k. Yeh, H. Nakhost, Y. Fujii, A. Ratner, R. Krishna, C.-Y. Lee, and T. Pfister, Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes, in *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 8003–8017. [Online]. Available: https://aclanthology.org/2023.findings-acl.507/

[27] D. Guo, D. Yang, H. Zhang, J. Song, P. Wang, Q. Zhu, R. Xu, R. Zhang, S. Ma, X. Bi, X. Zhang, X. Yu, Y. Wu, Z. F. Wu, Z. Gou, Z. Shao, Z. Li, Z. Gao, A. Liu, B. Xue, B. Wang, B. Wu, B. Feng, C. Lu, C. Zhao, C. Deng, C. Ruan, D. Dai, D. Chen, D. Ji, E. Li, F. Lin, F. Dai, F. Luo, G. Hao, G. Chen, G. Li, H. Zhang, H. Xu, H. Ding, H. Gao, H. Qu, H. Li, J. Guo, J. Li, J. Chen, J. Yuan, J. Tu, J. Qiu, J. Li, J. L. Cai, J. Ni, J. Liang, J. Chen, K. Dong, K. Hu, K. You, K. Gao, K. Guan, K. Huang, K. Yu, L. Wang, L. Zhang, L. Zhao, L. Wang, L. Zhang, L. Xu, L. Xia, M. Zhang, M. Zhang, M. Tang, M. Zhou, M. Li, M. Wang, M. Li, N. Tian, P. Huang, P. Zhang, Q. Wang, Q. Chen, Q. Du, R. Ge, R. Zhang, R. Pan, R. Wang, R. J. Chen, R. L. Jin, R. Chen, S. Lu, S. Zhou, S. Chen, S. Ye, S. Wang, S. Yu, S. Zhou, S. Pan, S. S. Li, S. Zhou, S. Wu, T. Yun, T. Pei, T. Sun, T. Wang, W. Zeng, W. Liu, W. Liang, W. Gao, W. Yu, W. Zhang, W. L. Xiao, W. An, X. Liu, X. Wang, X. Chen, X. Nie, X. Cheng, X. Liu, X. Xie, X. Liu, X. Yang, X. Li, X. Su, X. Lin, X. Q. Li, X. Jin, X. Shen, X. Chen, X. Sun, X. Wang, X. Song, X. Zhou, X. Wang, X. Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. Zhang, Y. Xu, Y. Li, Y. Zhao, Y. Sun, Y. Wang, Y. Yu, Y. Zhang, Y. Shi, Y. Xiong, Y. He, Y. Piao, Y. Wang, Y. Tan, Y. Ma, Y. Liu, Y. Guo, Y. Ou, Y. Wang, Y. Gong, Y. Zou, Y. He, Y. Xiong, Y. Luo, Y. You, Y. Liu, Y. Zhou, Y. X. Zhu, Y. Huang, Y. Li, Y. Zheng, Y. Zhu, Y. Ma, Y. Tang, Y. Zha, Y. Yan, Z. Z. Ren, Z. Ren, Z. Sha, Z. Fu, Z. Xu, Z. Xie, Z. Zhang, Z. Hao, Z. Ma, Z. Yan, Z. Wu, Z. Gu, Z. Zhu, Z. Liu, Z. Li, Z. Xie, Z. Song, Z. Pan, Z. Huang, Z. Xu, Z. Zhang, and Z. Zhang, DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning, *Nature*, vol. 645, no. 8081, pp. 633–638, 2025. [Online]. Available: https://doi.org/10.1038/s41586-025-09422-z

[28] R. Nikam, A. Kulandaisamy, K. Harini, D. Sharma, and M. M. Gromiha, ProThermDB: thermodynamic database for proteins and mutants revisited after 15 years, *Nucleic Acids Research*, vol. 49, no. D1, pp. D420–D424, 2021. [Online]. Available: https://doi.org/10.1093/nar/gkaa1035

[29] J. S. Xavier, T.-B. Nguyen, M. Karmarkar, S. Portelli, P. M. Rezende, J. P. L. Velloso, D. B. Ascher, and D. E. V. Pires, ThermoMutDB: a thermodynamic database for missense mutations, *Nucleic Acids Research*, vol. 49, no. D1, pp. D475–D479, 2021. [Online]. Available: https://doi.org/10.1093/nar/gkaa925

[30] K. Tsuboyama, J. Dauparas, J. Chen, E. Laine, Y. Mohseni Behbahani, J. J. Weinstein, N. M. Mangan, S. Ovchinnikov, and G. J. Rocklin, Mega-scale experimental analysis of protein folding stability in biology and design, *Nature*, vol. 620, no. 7973, pp. 434–444, 2023. [Online]. Available: https://doi.org/10.1038/s41586-023-06328-6

[31] D. E. V. Pires, D. B. Ascher, and T. L. Blundell, mCSM: predicting the effects of mutations in proteins using graph-based signatures, *Bioinformatics*, vol. 30, no. 3, pp. 335–342, 2014. [Online]. Available: https://doi.org/10.1093/bioinformatics/btt691

[32] B. Li, Y. T. Yang, J. A. Capra, and M. B. Gerstein, Predicting changes in protein thermodynamic stability upon point mutation with deep 3D convolutional neural networks, *PLOS Computational Biology*, vol. 16, no. 11, p. e1008291, 2020. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1008291

[33] A. Broom, K. Trainor, Z. Jacobi, and E. M. Meiering, Computational Modeling of Protein Stability: Quantitative Analysis Reveals Solutions to Pervasive Problems, *Structure*, vol. 28, no. 6, pp. 717–726.e3, 2020. [Online]. Available: https://doi.org/10.1016/j.str.2020.04.003

[34] A. Broom, Z. Jacobi, K. Trainor, and E. M. Meiering, Computational tools help improve protein stability but with a solubility tradeoff, *Journal of Biological Chemistry*, vol. 292, no. 35, pp. 14 349–14 361, 2017. [Online]. Available: https://doi.org/10.1074/jbc.m117.784165

[35] S. Reeves and S. Kalyaanamoorthy, Zero-shot transfer of protein sequence likelihood models to thermostability prediction, *Nature Machine Intelligence*, vol. 6, no. 9, pp. 1063–1076, 2024. [Online]. Available: https://doi.org/10.1038/s42256-024-00887-7

Author biography



Wenkang Wang received the B.S. degree from Central South University, China, in 2020. He is working toward the Ph.D. degree in computer science and engineering at Central South University, China. His research interests include bioinformatics, complex network analysis, machine learning and deep learning.



Wentao Yu received the B.S. degree from Central South University, China, in 2025. He is currently pursuing the M.S. degree in computer science and technology at Central South University, China. His research interests include bioinformatics, protein classification and functional annotation.



Qiurong Yang received the B.S. degree from Central South University, China, in 2024. She is working toward the Ph.D. degree in electronic information at Central South University, China. Her current research interests include computational bioinformatics, protein function annotation and deep learning.



Ruiqing Zheng received his B.S., M.S., and Ph.D. degrees in Computer Science from Central South University in 2013, 2016, and 2021, respectively. He is currently a lecturer at the School of Computer Science and Engineering, Central South University. His main research interests include gene regulatory network reconstruction and single-cell data analysis.



Min Li received the Ph.D. degree in computer science from Central South University, China, in 2008. Currently, she is a professor and dean of the School of Computer Science and Engineering, Central South University, China. Her main research interests include bioinformatics and systems biology. She has published more than 100 technical papers in refereed journals such as Cell, Nature Communications, Genome Research, Bioinformatics, Briefings in Bioinformatics, IEEE/ACM Transactions on Computational Biology and Bioinformatics, and conference proceedings such as BIBM, GIW and ISBRA. According to Google Scholar, her publications have received more than 20,000 citations, and her h-index is 70.