Briefings in Bioinformatics, 2025, **26(2)**, bbaf120
https://doi.org/10.1093/bib/bbaf120
Problem Solving Protocol

𝒮able: bridging the gap in protein structure understanding with an empowering and versatile pre-training paradigm

Jiashan Li[1], Xi Chen[2], He Huang[2], Mingliang Zeng[2], Jingcheng Yu[2], Xinqi Gong[1,*], Qiwei Ye[2,*]

[1]Institute for Mathematical Sciences, Renmin University of China, 59 Zhongguancun Street, Beijing 100872, China
[2]Bio Computing Center, Beijing Academy of Artificial Intelligence, 150 Chengfu Road, Beijing 100084, China

[*]Corresponding authors. Xinqi Gong, Institute for Mathematical Sciences, Renmin University of China, 59 Zhongguancun Street, Beijing 100872, China. E-mail: xinqigong@ruc.edu.cn; Qiwei Ye, Bio Computing Center, Beijing Academy of Artificial Intelligence, 150 Chengfu Road, Beijing 100084, China. E-mail: qwye@baai.ac.cn

Abstract

Protein pre-training has emerged as a transformative approach for solving diverse biological tasks. While many contemporary methods focus on sequence-based language models, recent findings highlight that protein sequences alone are insufficient to capture the extensive information inherent in protein structures. Recognizing the crucial role of protein structure in defining function and interactions, we introduce 𝒮able, a versatile pre-training model designed to comprehensively understand protein structures. 𝒮able incorporates a novel structural encoding mechanism that enhances inter-atomic information exchange and spatial awareness, combined with robust pre-training strategies and lightweight decoders optimized for specific downstream tasks. This approach enables 𝒮able to consistently outperform existing methods in tasks such as generation, classification, and regression, demonstrating its superior capability in protein structure representation. The code and models can be accessed via GitHub repository at https://github.com/baaihealth/Sable.

Keywords: protein representation learning; protein structural pre-training; antibody design; protein function prediction

Introduction

Pre-training techniques have revolutionized the field of protein research by enabling models to learn meaningful representations from vast amounts of unlabeled data. Inspired by advancements in natural language processing (NLP) [1–3], sequence-based protein pre-training models have achieved significant success. These models leverage large-scale protein sequence data to uncover embedded information such as amino acid interactions [4–6], structural motifs [6–9], and functional domain features [10, 11]. Despite their success in interpreting and predicting protein properties, sequence-based models are inherently limited by the constraints of sequence information. These limitations manifest in unexplained phenomena and uncertainties in predictions. For instance, certain protein families, such as G-protein coupled receptors (GPCRs), exhibit different structures when interacting with various ligands, even with identical amino acid sequences [12]. This highlights the need to incorporate protein structure into modeling frameworks. Protein structures can be represented in multiple ways, each capturing different aspects of their three-dimensional conformation. Among these representations, contact map [13] and distance map [14] provide a detailed view of inter-residue relationships, while surface-based [15, 16] representations and three-dimensional voxelization [17, 18] offer insights into spatial and geometric features. Graph-based representations, where proteins are modeled as graphs with nodes representing atoms or residues and edges representing bonds or spatial proximity, have gained traction due to their intuitive alignment with geometric deep learning methods [19–22]. Graph neural networks (GNNs) have shown immense potential in learning from protein structures and have been widely applied to various geometric deep learning tasks involving three-dimensional structures [23–26].

Despite significant advances in protein pre-training models, several key challenges persist. Different categories of protein-related tasks necessitate models to extract distinct protein representations. For instance, protein–protein interaction prediction tasks focus on identifying interfaces and key binding sites [13], while protein function prediction tasks require capturing information about functional domains and active sites [19]. In contrast, protein design emphasizes constructing novel proteins with specific structural and functional properties, which involves the rational arrangement of amino acid sequences and precise regulation of spatial conformations [27–30]. Existing models often design pre-training frameworks tailored to specific tasks [26, 31, 32]. However, the pursuit of a unified framework that simplifies the protein analysis process, allowing knowledge to generalize and transfer more effectively across various protein-related tasks, to consistently reveal the structural characteristics of proteins, and to enhance the modeling capabilities of the models, thereby achieving consistently outstanding performance across a series of tasks, remains a formidable challenge. However, each structural representation comes with its own set of limitations. Contact

Received: October 29, 2024. **Revised:** January 23, 2025. **Accepted:** February 23, 2025

and distance maps may lose critical spatial information when reduced to two dimensions, potentially oversimplifying the complex three-dimensional landscape. Surface-based representations, while capturing the protein's exterior, often fail to model internal structures accurately. Voxelized representations need to balance between resolution and computational efficiency, as high-resolution voxels offer more detail but are computationally expensive. Graph-based methods, while powerful, often suffer from bottlenecks in message propagation paradigms, where the mechanism of aggregating information from distant nodes can become inefficient, leading to suboptimal performance in tasks requiring accurate long-range information integration [33, 34]. Furthermore, designing self-supervised tasks for protein structure learning remains an area warranting improvement. Current methods tend to focus on tasks such as predicting masked distances, masked dihedral angles [26], and denoising distance matrices [14]. While these methods have advanced the ability to learn protein representations, they often fail to capture the complex long-range interactions and multi-scale information inherent in three-dimensional protein structures [35]. For instance, tasks centered around distance prediction may only address local atomic pair distances, potentially neglecting the overall conformation and spatial arrangement of the protein. A more sophisticated approach to self-supervised task design should aim to incorporate the full range of physical and chemical properties of protein structures. This involves not only understanding local interactions but also integrating global conformational features and hierarchical levels of protein information from the local to the global scale. Effective self-supervised tasks should ensure that models can capture and utilize this multi-scale information, facilitating a comprehensive understanding of protein structure.

To address these challenges, we propose $\mathcal{S}$able, a versatile Structural-understandable protein pre-training model designed to provide a profound and detailed understanding of protein structures through advanced structural encoding and self-supervised learning techniques, achieving outstanding performance across a wide range of downstream tasks. $\mathcal{S}$able features a robust structural encoder capable of direct information exchange between backbone atoms through distance encoding and integrates relative position encoding (RPE) to enhance spatial awareness between residues while maintaining rotational and translational invariance. By combining atomic-level and residue-level structural information, $\mathcal{S}$able improves computational efficiency while preserving the ability to generate all-atom structures. $\mathcal{S}$able employs self-supervised tasks such as masked token prediction and atom distance restoration to further enhance protein representation learning. Through collaboration, these strategies enable the model to effectively learn various protein information, subsequently fine-tuning lightweight task layers to excel in diverse tasks such as antibody design, protein design, protein function prediction, binding affinity prediction, and model quality assessment.

Methods

Our method consists of two key stages: pre-training and fine-tuning, as summarized in Fig. 1. The pre-training stage learns protein representations through self-supervised tasks, while the fine-tuning stage adapts the pre-trained model to various downstream tasks with task-specific layers. This section provides a detailed explanation of each component in our framework.

Protein representation

Given a protein $\mathcal{P}$ as input, we first extract its sequence information and represent it as a sequence of discrete token. Additionally, we incorporate five special tokens: [CLS], which denotes the start of the sequence and serves for classification; [SEP], indicating the end of the sequence; [PAD], used for padding; [UNK], representing unknown elements; and [MASK], for masking specific tokens. This structured tokenization facilitates effective training of the model. We then employ a linear layer to learn the initial single representation $\mathrm{s}_\mathrm{i}^{(0)}$ of the protein sequence. This single representation encapsulates essential features derived from the protein's sequence, thereby laying the groundwork for further integration with structural embeddings. To fully exploit the protein's structural information while preserving the inherent properties of three-dimensional rotation and translation invariance, we employ a set of three modules to extract the protein's structural details and convert them into an initial pair representation $\mathrm{z}_\mathrm{ij}^{(0)}$, capturing residue interactions.

Distance encoding We begin by exploring pairwise interactions between residues based on distance information. From the backbone atom coordinates $\{\mathrm{x}_{\mathrm{i},k}\}$ of the protein, we obtain the distance matrix $\{d_{(\mathrm{i},k)(j,\mathrm{t})}\}$ between backbone atoms, where k, t ranges from 1 to 4 representing backbone atoms N, Cα, C, and O in the ith, jth residue respectively, $d_{(\mathrm{i},k)(j,\mathrm{t})}$ indicating the distance between the kth atom of ith and the tth atom of the jth residue, $i, j \in [1, N]$, N stands for the number of residues. Subsequently, we discretize the continuous distances using the Equal Width Binning strategy to obtain an initial representation based on distance information. As atomic distances increase significantly, the intermolecular forces between them diminish. We apply a maximum distance threshold of 128 for clipping, where any distance between residues exceeding this threshold is not distinguished by this feature. Based on the given supremum of distances and the predefined number of bins, determine the data range represented by each bin. Then, place the distance values into different bins, discretizing the distance values. For instance, if the distance value is 15 and each bin's width is 8, it falls within the second bin. Subsequently, utilize a linear layer to embed the indices of the bins, obtaining the embedding representation of distances at a specific precision. To facilitate hierarchical learning of distance representations at different precision levels, we discretize distances into varying numbers of distance bins and then aggregate to obtain initial distance representations. The specific calculation is as follows:

$$\mathrm{z}_{(\mathrm{i},k)(j,\mathrm{t})}^{\mathrm{distance}} = \sum_{n_{\mathrm{bins}} \in \psi} \mathrm{bining}\left(d_{(\mathrm{i},k)(j,\mathrm{t})}, n_{\mathrm{bins}}\right) \tag{1}$$

The set $\psi = \{16, 64, ..., 16384\}$ represents the varying number of bins from coarse to fine granularity. Linear layers are employed to embed the distance intervals at each level, followed by an aggregation step to obtain the initial distance representation. This hierarchical learning approach allows for the extraction of more nuanced and fine-grained distance representations, enhancing the model's ability to capture subtle structural features and relationships within residues. The algorithm for the binning process is summarized in Appendix Algorithm 1.

We design a purposeful projection that accurately aligns atomic-level representations to the residue level.

$$\mathrm{z}_{\mathrm{ij}}^{\mathrm{distance}} = \mathrm{Linear}\left(\sum_{k=1}^{4}\sum_{\mathrm{t}=1}^{4} W \mathrm{z}_{(\mathrm{i},k)(j,\mathrm{t})}^{\mathrm{distance}}\right) \tag{2}$$

where $W \in \mathbb{R}^{c_z}$ is a trainable parameter, c_z stand for the dimension of the pair representation. The projection aggregates relevant information from multiple atoms, enhancing

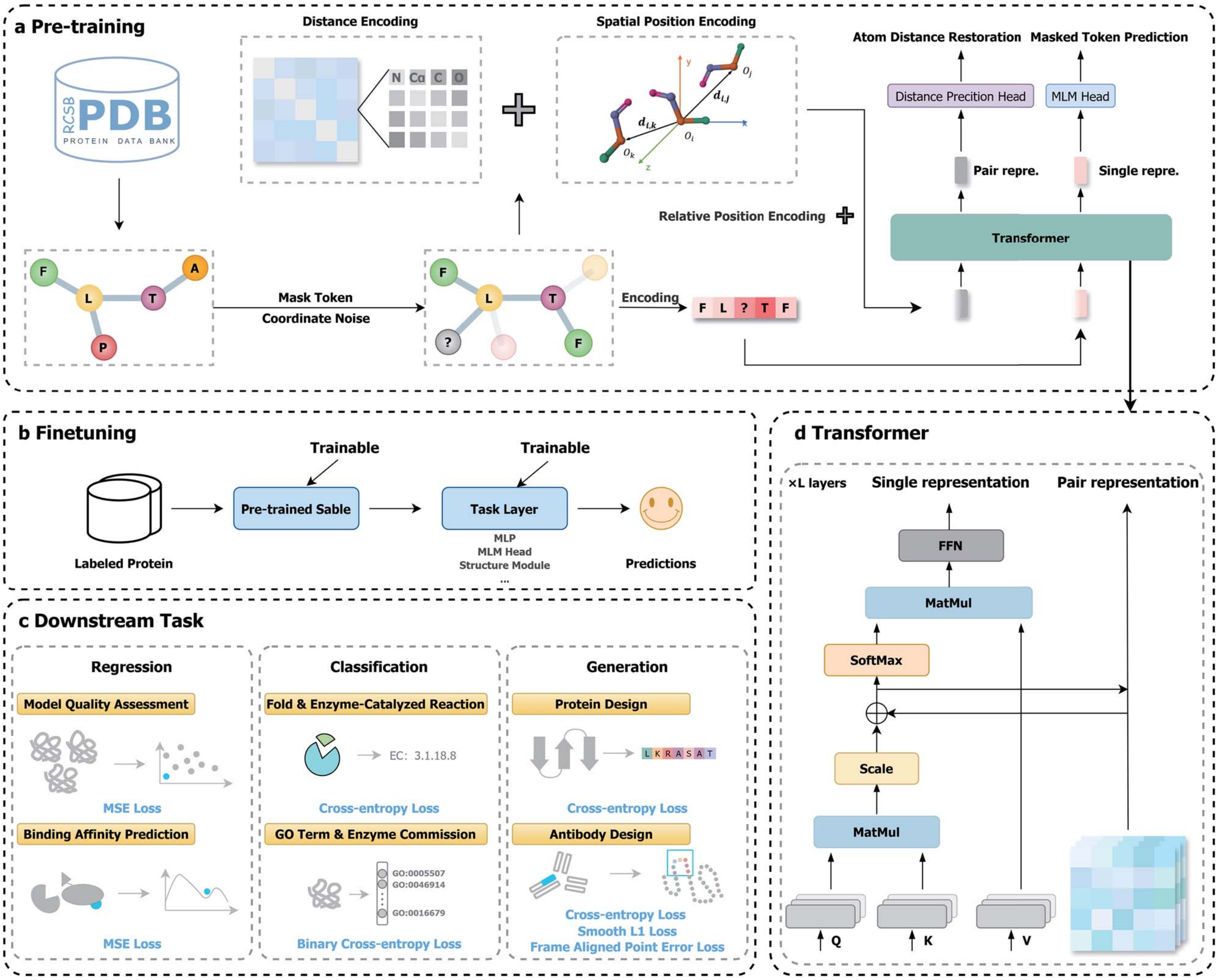

Figure 1. **$\mathcal{S}$able overview. (a)** Pre-training stage: $\mathcal{S}$able utilizes distance encoding, SPE, and RPE to capture the essential features of protein structures. By incorporating self-supervised tasks like masked token prediction and atom distance restoration, $\mathcal{S}$able significantly improves its capability to learn detailed and robust protein representations. **(b)** Finetuning stage: the pre-trained $\mathcal{S}$able model, combined with task-specific layers, enables accurate predictions on various downstream tasks using the pre-training fine-tuning paradigm. **(c)** Visualization of different downstream tasks, including their respective categories and loss functions (shown in blue font). **(d)** Detailed architecture of the Transformer in pre-training.

the representation's ability to capture important structural features.

Spatial position encoding (SPE) The distance-based encoding method compresses three-dimensional structural information into two-dimensional distance information. While this effectively reduces dimensionality and captures key information, it also results in the loss of geometric and directional information between atoms, thereby limiting the ability to precisely represent and analyze spatial arrangements. To overcome these limitations, we introduce SPE to capture the spatial geometric relationships between residues while maintaining rotational and translational invariance. Given any residues r_i and r_j, with the Cα atom of residue i serving as the coordinate origin, a local Cartesian coordinate system, denoted as $\mathcal{O}_i$, is established based on $x_{i,1}$, $x_{i,2}$, and $x_{i,3}$ by Schmidt orthogonalization. Subsequently, we project the Cα atom of residue j onto the local frame $\mathcal{O}_i$ to obtain $\vec{d}_{ij}$, which represents the coordinates of $x_{j,2}$ in $\mathcal{O}_i$. Finally, we partition the continuous coordinates into equally spaced bins, convert them into embeddings, and employ them as SPE.

$$z_{ij}^{\mathrm{spe}} = \mathrm{SPE}\left(x_{i,1}, x_{i,2}, x_{i,3}, x_{j,2}\right) \tag{3}$$

Appendix Algorithm 4 and Fig. B2 elucidate the specific operations of SPE.

Relative position encoding (RPE) To enrich the network with information about the positional context of residues within the sequence, we introduce relative positional encoding into the initial pair representations, referred as z_{ij}^{rpe}). Specifically, we employ a one-hot encoding scheme to represent the relative distance between position r_i^{index} and position r_j^{index} in the sequence as a vector. This encoding strategy is constrained to distances less than a predefined threshold, ensuring the effective capture of significant relative positional relationships.

$$z_{ij}^{\mathrm{rpe}} = \mathrm{RPE}\left(r_i^{\mathrm{index}}, r_j^{\mathrm{index}}\right) \tag{4}$$

Appendix Algorithm 3 delineates the detailed operations involved in RPE.

At this stage, we obtain our initial pair representation:

$$z_{ij}^{(0)} = z_{ij}^{\mathrm{distance}} + z_{ij}^{\mathrm{spe}} + z_{ij}^{\mathrm{rpe}} \tag{5}$$

Since the distance and relative position information are invariant to rotation and translation, the structural encoder ensures this invariance, allowing for more robust representations.

Backbone network

Driven by the transformer's intrinsic self-attention mechanism, which enables efficient computation across the entire protein sequence and captures long-range associations between distant residues, we adopt the transformer architecture as the backbone of our network. This allows us to effectively capture global features and interactions within protein structures. Additionally, we incorporate an interaction mechanism that integrates sequence and structural information, further enhancing the model's ability to leverage insights from both dimensions.

The transformer architecture consists of stacked layers of transformers, where initialized single representations serve as input. Each transformer layer comprises two main components: a self-attention module and a feed-forward network. The update of the single representation in the lth layer is achieved as follows:

$$\begin{aligned}\text{Attention}(Q_i^{l,h}, K_i^{l,h}, V_i^{l,h}) = \\ \sum_j \text{softmax}\left(\frac{Q_i^{l,h}(K_j^{l,h})^T}{\sqrt{d_k}} + z_{ij}^{l-1,h}\right) V_j^{l,h}\end{aligned} \tag{6}$$

where $Q_i^{l,h}$, $K_i^{l,h}$, and $V_i^{l,h}$ correspond to the *Query*, *Key*, and *Value* for the ith residue, in the lth layer and the hth head, $h \in \{1, 2, \dots, H\}$, H is the number of attention heads, d_k represents the dimension of the *Key*, and $z_{ij}^{l-1,h}$ denotes the pair representation of the residue pair ij in the $l - 1$-th layer and the hth head. Furthermore, we utilize the attention weights obtained from the self-attention mechanism to update the pair representations as follows:

$$z_{ij}^{l,h} = z_{ij}^{l-1,h} + \text{concat}_h\left(\frac{Q_i^{l,h}(K_j^{l,h})^T}{\sqrt{d_k}}\right) \tag{7}$$

where concat(·) represents the concatenation operation.

Pre-training dataset

Our pre-training dataset is sourced from the Protein Data Bank (PDB), encompassing approximately 200 000 protein structure entries released up to May 1, 2023. During the data preparation process, we implement stringent filtering and cleansing protocols to exclude elements such as RNA, DNA, small molecules, water molecules, and heterogeneous residues from the PDB files. Furthermore, for missing backbone atom coordinates, we substitute them with the centroid coordinates of the respective residues. To mitigate the risk of data leakage associated with downstream task fine-tuning, we conduct rigorous scrutiny to ensure data isolation. Our focus is primarily on model quality assessment and protein design tasks. Although pre-training has different training objectives and loss functions from these downstream tasks, we remove any structural data from the pre-training dataset that overlaps with the testing sets of these tasks. For other downstream tasks, we incorporate additional annotated data during fine-tuning, while protein sequence and structural information serve as known conditions. This approach eliminates the necessity to exclude test data from the pre-training dataset.

Pre-training strategies

We utilize two self-supervised tasks: masked token prediction and atom distance restoration, tailored specifically to extract universal representations from extensive protein structure data.

Masked token prediction We introduce a masked language modeling Head that receives the hidden layer feature of the masked token s_i^L after network updates as input. Start by selecting a specific proportion of residues and replace their types with a mask token, unknown residue types, or leave the original residue types unchanged. To accommodate different scenarios, the specific proportion can be randomly chosen from 15, 50, and 100% with varying weights. Through a fully connected layer, activation function, and normalization, it projects the features to a dimension equal to the size of the vocabulary, where each dimension represents the predicted probability of different residue types. Then, we use the softmax function to normalize the projected features, obtaining the probability distribution for each residue type. Finally, we select the residue type with the highest probability as the prediction result.

$$\begin{aligned} h_i &= \text{LayerNorm}(\text{GELU}(\text{Linear}(s_i^L))) \\ \hat{s}_i &= \text{argmax}(\text{softmax}(\text{Linear}(h_i))) \end{aligned} \tag{8}$$

where s_i^L denotes the output of the final layer of the backbone network.

Atom distance restoration To further improve the robustness of this reconstruction process, where selected residues can efficiently reconstruct the original residue types through the interplay between single and pairwise representations, we also utilize atom distance restoration. For the residue r_i replaced by a masked token, its coordinate $x_{i,k}$ is subjected to a certain amount of noise, given by $\tilde{x}_{i,k} = x_{i,k} + \epsilon_{i,k}$, where $\epsilon_{i,k}$ represents the noise term defined as

$$\epsilon_{i,k} = \alpha \cdot \eta_{i,k} \tag{9}$$

Here, $\eta_{i,k}$ denotes Gaussian noise with a mean of 0 and a variance of 1 ($\eta_{i,k} \sim \mathcal{N}(0, 1)$). The parameter α is sampled from the set $\{0, 0.1, 1\}$ with different probabilities. Consequently, the distance encoding and SPE generated based on the backbone atom coordinates are subjected to a certain degree of perturbation. We introduce a Distance Prediction Head, which is used to predict the distances between backbone atoms from perturbed pair representations z_{ij}^L. This module takes input features z_{ij}^L and first extracts atom-level information from aggregated residue-level representations. Then, through linear transformation, activation function, and normalization operations, the input features are projected into the space of interatomic distances, resulting in the output of a predicted distance matrix.

$$\begin{aligned} z_{(i,k)(j,t)}^L &= z_{ij}^L \otimes W \\ h_{(i,k)(j,t)} &= \text{LayerNorm}(\text{GELU}(\text{Linear}(z_{(i,k)(j,t)}^L))) \\ \hat{d}_{(i,k)(j,t)} &= \text{Project}(h_{(i,k)(j,t)}) \end{aligned} \tag{10}$$

where $\otimes$ denotes element-wise multiplication, W represents the parameter matrix for extracting atom-level information, and z_{ij}^L denotes the pair representation from the final layer of the network. $z_{(i,k)(j,t)}^L$ indicates the atom-level pair representation. $\hat{d}_{(i,k)(j,t)}$ represents the predicted distance between the kth atom of the ith residue and the tth atom of the jth residue.

Fine-tuning strategies

As illustrated in Fig. 1b, our fine-tuning strategy leverages labeled data, which is processed through the pre-trained model and complemented by a lightweight task layer to yield precise predictions. To enhance model effectiveness and ensure it adapts to specific task requirements, we allow both the pre-trained $\mathcal{S}$able and the task layer to remain trainable during the fine-tuning phase.

Antibody design

Due to the high conservation of antibodies, particularly the constant regions and the framework region sequences within the variable domains, and the specificity of binding primarily determined by the complementarity-determining regions (CDRs), our main objective is to design CDRs. This task involves addressing several computational challenges: first, modeling the intrinsic relationship between CDR sequences and their 3D structures; second, modeling the distribution of CDRs conditioned on the remaining antibody sequences; and third, generating CDRs that specifically bind to antigens while ensuring real-time docking capabilities. This task gives rise to two essential goals: the co-design of antibody sequences and structures, and the design of antigen-specific antibodies.

To accomplish the sequence-structure co-design of the antibody's CDRs, we primarily focus on the heavy chain of the antibody. Taking CDR-H3 as an example, residues belonging to CDR-H3 are masked, and the coordinates for these residues are assigned by averaging the Cα coordinates of the two closest residues outside this region. This process yields initial representations with noise. Subsequently, we leverage the pre-trained framework to predict the residue types of CDR-H3 and refine the representations accordingly. The updated single and pair representations are then fed into the structure module [36], facilitating the generation of the complete heavy chain structure. Additionally, we introduce an additional distance head based on the representations to ensure accurate prediction of the Euclidean distance between corrupted atom pairs in CDRs.

When contemplating the generation of antigen-specific antibodies, we incorporate both the sequence and structural information of the antigen, while preserving the framework region information of the antibody heavy chain. It is worth noting that we assume the relative positions of the antigen and antibody heavy chain to be unknown, indicating a lack of inter-chain information. To construct the initial representation of the antibody-antigen complex, we utilize the pre-trained model to obtain single and pair representations of the antigen and antibody heavy chain separately. The single representations are concatenated to obtain a single representation of the complex. For the pair representation of the complex, the positions along the diagonal (representing intra-chain information) are replaced with the pair representations of the antigen and antibody heavy chain. Conversely, the positions along the anti-diagonal (representing inter-chain information) remain vacant. A shallow decoder, similar to the pre-training framework, is trained to take the concatenated complex representation as input and reconstruct the inter-chain information. Subsequently, akin to the process of predicting solely the antibody heavy chain, the complex structure is generated through a structure module and distance head.

To guide the sequence generation and structure generation processes, we employ a loss function comprising three components: cross-entropy loss, smooth L1 loss, and frame alignment point error. These components are weighted equally at a ratio of 1:1:1, ensuring a balanced optimization approach. In fact, our approach can be straightforwardly extended to simultaneously predict all six CDRs regions of antibody-antigen complexes, once the sequence and structural information of the light chain variable region is introduced.

Protein design

Protein sequence design involves identifying amino acid sequences capable of adopting a specific protein backbone conformation. During the sequence generation phase, the types of residues at each position are replaced by a masking token. $\mathcal{S}$able utilizes structure encoding to adjust attention weights and update the single representation. This is achieved through the incorporation of distance encoding and RPE modules. The distance encoding module learns the pairwise relationships between backbone atoms and maps them to the distance relationships between residues, capturing atomic-level structural information while reducing computational complexity. The RPE module establishes a local frame for each residue using the backbone atoms and determines the distance and direction between residues based on this frame. A classification-like strategy is employed to predict residue types, with c denoting the size of the residue type dictionary.

Protein function

In our approach to predicting protein function, we employ a straightforward linear layer as the classifier within our model. We obtain the representation h_i^L by processing the final single representation s_i^L through a fully connected layer and an activation function, followed by normalization. The probability for each individual category is computed using $\mathbf{softmax}(\mathbf{avg}(\{h_i^L\}_{i=1}^n \mathrm{W}_c + b_c))$, where $\{h_i^L\}$ signifies the final single representation of the ith residues, c represents the number of classes, W_c denotes the learnable parameter matrix, and b_c stands for the bias term. In the fold classification task, $c = 1195$ indicates that there are 1195 identified folds. In the Enzyme-Catalyzed Reaction Classification task, $c = 384$ represents there are 384 different Enzyme Commission (EC) numbers. The class with the highest predicted probability or confidence serves as the final prediction result. In the multi-label classification task, it can be treated as multiple binary classification tasks by considering each class separately. Each binary classification task determines whether an instance belongs to one class or not. The ultimate predictions are determined based on classes whose probabilities surpass a predefined threshold.

Binding affinity prediction

The change in binding affinity is calculated by the formula $\Delta\Delta G = \Delta G_{\text{wild\_type}} - \Delta G_{\text{mutant}}$. Given the assumption that the structure of wild-type and mutant structures does not undergo significant changes, we exclusively consider the single representations to compute the change in binding affinity values, as shown below:

$$\Delta\Delta G = \mathbf{avg}(I_{\psi}(\mathbf{Linear}(\mathbf{MLP}(f_{wm}^{i}) - \mathbf{MLP}(f_{mw}^{i})))) \tag{11}$$

where I is the indicator function that equals 1 when $i \in \psi$, the set ψ represents the indices of mutant residues. $f_{wm}^{i} = \mathbf{concat}(s_{w,i}^{L}, s_{m,i}^{L})$, $f_{mw}^{i} = \mathbf{concat}(s_{m,i}^{L}, s_{w,i}^{L})$, where $s_{m,i}^{L}$ and $s_{w,i}^{L}$, respectively, denote the single representations of the ith wild-type and mutant residues in the final layer output.

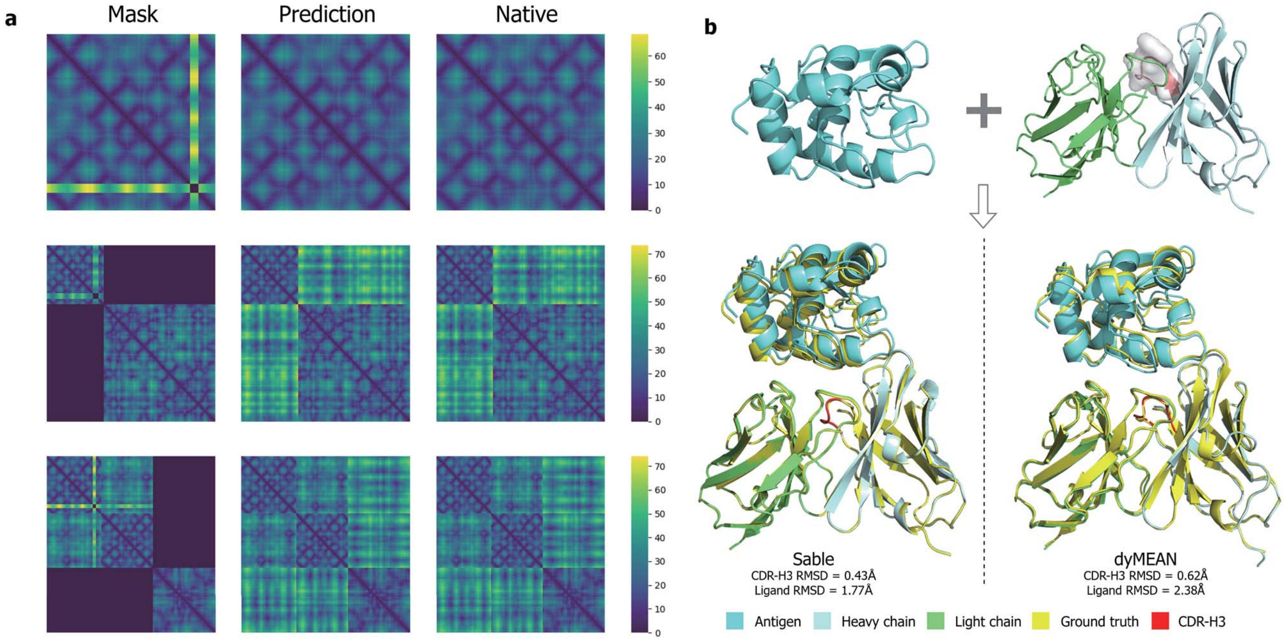

Figure 2. **Exploring $\mathcal{S}$able's performance in generation tasks. (a)** Visualizing the distance matrix provides insights into the process of structure recovery. Taking the CDR-H3 region as an example, only the heavy chain of the antibody (top), the addition of antigen information (middle), and the incorporation of light chain information (bottom) are considered. The relative positioning between the antigen and antibody remains unknown throughout. **(b)** Antigen-specific antibody design schematic (PDB ID: 1IC7).

Model quality assessment

Model quality assessment in the context of protein structure prediction is to evaluate the accuracy and reliability of predicted protein structures. We utilize the pre-trained model as the backbone and incorporate a two-layer MLP as the predictor. Our objective is to predict the quality of both global and local structures. Initially, we conduct column-wise and row-wise aggregation on pair representations, then concatenate these aggregated representations with single representations and use a MLP along with the sigmoid function to map them into the (0,1) range, signifying scores for each residue. For assessing global structural quality, we follow the same procedure, ultimately averaging the scores at the residue level to derive the global score.

Results

Evaluation of $\mathcal{S}$able on downstream tasks

In order to verify the effectiveness of our proposed pre-training model, we conduct experiments on several downstream tasks. Information on the datasets, baselines, and evaluation metrics is provided in Appendix C.

Antibody design

Here we focus on two key tasks: the design of heavy chain CDRs and the design of antigen-specific antibodies. Experimental results demonstrate that $\mathcal{S}$able achieves significantly better amino acid recovery (AAR) and root mean square deviation (RMSD) in most cases, and is comparable to the current best methods in a few instances.

We apply $\mathcal{S}$able for the design of antibody heavy chain CDRs, swiftly locating the position of each CDR by renumbering the PDB according to the IMGT scheme. Taking the design of CDR-H3 as an example, the positions of residues within CDR-H3 are disrupted, resulting in the loss of internal distance and interaction with other residues in the variable region, as shown in Fig. 2a (top). Conditioned on the remaining variable regions, $\mathcal{S}$able successfully restores the missing 3D positions of CDR-H3. Despite initializing the coordinates of missing residues in CDR-H3 using the average coordinates of the preceding and succeeding residues outside CDR-H3, we find that $\mathcal{S}$able achieves equally impressive performance even when initialized with arbitrary coordinates such as the origin. We compare $\mathcal{S}$able with seven state-of-the-art antibody design methods, and the results are summarized in Table 1. Under the first data splitting scheme, our method demonstrates outstanding performance in both sequence design and structure generation for CDR-H2 and CDR-H3. The AAR rate for CDR-H2 improves by 7.96%, while for CDR-H3, the AAR increases by 1.65%. The AAR for CDR-H1 is on par with the leading methods, and our approach also predicts more favorable structures. In 10-fold cross-validation, our method achieves the highest AAR for the most challenging CDR-H3 predictions and provides the best CDR-H2 structure predictions. In contrast to advanced methods such as RefineGNN and MEAN, which only generate the backbone atoms of CDRs, our method can produce more consistent all-atom structures. Moreover, $\mathcal{S}$able is capable of generating the overall structure of the heavy chain, resulting in smoother connectivity between the known framework regions and CDRs (see Fig. C3).

We evaluate $\mathcal{S}$able's performance on a more challenging task: antigen-specific antibody design. Due to the highly specific binding between antibodies and antigens, it is often necessary to design antibodies that specifically bind to a particular antigen. However, the high variability of CDR-H3 and the unknown binding modes between antibodies and antigens, as shown in Fig. 2a (middle), result in a complete lack of inter-chain information, posing significant challenges for generative models. By incorporating additional antigen information, we aim to simultaneously generate antigen-specific antibody CDRs and predict the binding modes. Compared to antibody heavy chain design with given framework regions, antigen information provides additional guidance for the generation of heavy chain CDRs, resulting in a slight improvement of 0.98% in AAR and a significant reduction in RMSD

Table 1. Results of antibody design: sequence-structure co-design. **Top:** standard data partitioning is based on RefineGNN [22]. **Bottom:** 10-fold cross-validation. The best and runner-up results are highlighted in **bold** and underlined, respectively

Method	CDR-H1		CDR-H2		CDR-H3	
	AAR %	RMSD ↓	AAR %	RMSD ↓	AAR %	RMSD ↓
LSTM [37, 38]	28.02	–	24.39	–	18.92	–
AR-GNN [39]	41.88	2.87	41.18	2.34	18.93	3.19
RefineGNN [22]	30.07	0.97	27.70	0.73	27.60	2.12
AbBERT-HMPN [40]	55.56	0.91	51.46	0.67	31.08	2.38
MEAN [41]	**62.78**	0.94	52.04	0.89	39.87	2.20
Sable	62.59	**0.88**	**60.00**	**0.67**	**41.52**	**2.12**

Method	CDR-H1		CDR-H2		CDR-H3	
	AAR %	RMSD ↓	AAR %	RMSD ↓	AAR %	RMSD ↓
LSTM [37, 38]	40.98	–	28.50	–	15.69	–
C-LSTM [37, 38]	40.93	–	29.24	–	15.48	–
RefineGNN [22]	39.40	3.22	37.06	3.64	21.13	6.00
C-RefineGNN [22]	33.19	3.25	33.53	3.69	18.88	6.22
MEAN [41]	58.29	0.98	47.15	0.95	36.38	2.21
AbODE [42]	**70.5**	**0.65**	**55.7**	0.73	39.8	**1.73**
AntiDesigner [43]	64.34	0.82	55.52	0.79	37.37	1.97
Sable	57.22	0.84	55.09	**0.59**	**41.25**	2.00

Table 2. Results of antibody design: antigen specific design. the best and the runner-up results are highlighted in **bolded** and underlined, respectively

Model	AAR %	RMSD ↓
RAbD [44]	28.6	–
LSTM [37, 38]	22.36	–
CondRefineGNN [22]	33.2	–
HSRN [45]	34.1	–
MEAN [41]	36.77	1.81
dyMEAN [46]	**43.65**	–
AbODE [42]	39.95	**1.54**
ADesigner [43]	40.94	1.55
Sable	42.50	1.70

from 2.12 to 1.70, with fewer outliers (see Table 2 and Fig. C4). Unlike traditional approaches that break down antibody design into separate tasks such as sequence optimization, structure prediction, and docking simulations, our method provides a comprehensive, end-to-end solution. By integrating all these steps into a unified process, we streamline the design workflow, eliminating the need for manual transitions between different stages.

Due to the high variability of the six CDRs in antibodies, it is crucial to generate both the sequence and structure of the entire CDR ensemble, rather than individually designing each CDR. To accomplish this, we incorporate information from the light chain and mask residue types in all CDRs. Sable utilizes contextual information to comprehensively consider the characteristics of CDRs, thereby enhancing the accuracy and efficiency of the design process. This integrated optimization strategy provides us with greater design flexibility and leads to significant advancements in designing all six CDRs. By successfully simultaneously designing the sequences and structures of all six CDRs, we achieve higher AAR in five out of the six CDRs, as shown in Table C3. This highlights the effectiveness of Sable in achieving successful design outcomes while considering the unique functionalities of each CDR.

Protein design

We achieve the second-highest AAR on the TS50 benchmark, surpassing all methods except SPDesign [50] by a notable 6.19%. Our performance on the CATH v4.2 test set is comparable, yielding results similar to other protein representation methods, although slightly inferior to those specifically tailored for protein design tasks. To ensure a fair comparison with other approaches, we report the performance of Sable on TS50 using the standard training set, with an AAR of 54.68%, outperforming other protein representation learning methods fine-tuned for protein design tasks. For methods that do not explicitly use the standard training set, we test the models fine-tuned on the CATH standard training set on TS50, achieving an AAR of 64.91%. The results of protein design are summarized in Table 3. Comparing the results of the two different training sets on the TS50 clearly reveals the significant impact of data leakage on this task. To prevent data leakage during pre-training, we remove any overlapping data between the pre-training dataset and the protein test set. Although Sable does not directly learn to map structural information to sequences during pre-training, communication between single and pair representations still captures this association during fine-tuning. However, underperformance on the CATH test set prompt us to propose a new pre-training mechanism to enhance the ability to represent sequences through structural information. One potential approach is provided in the Appendix D. By concentrating the pre-training self-supervised task on masked token prediction, we observe a 3.2% improvement in Sable's AAR on the CATH test set.

Protein function

In the task of protein function prediction, Sable achieves state-of-the-art results on four out of eight datasets and performs on par with other established baselines on the remaining four datasets. Function prediction tasks can be viewed as classification tasks, comprising two main types: single-label classification and multi-label classification. Single-label classification involves categorizing proteins into different fold types (fold classification) or identifying specific enzyme types (enzyme-catalyzed

Table 3. Results of different Protein Design methods. [¶] denotes results taken from [50], [†] denotes results taken from [30], [♮] denotes results taken from [25], and [‡] represents the results as reported in their respective papers. Method lacks specification regarding the construction of a canonical training set for TS50. **Bold** and underline indicate the top two results, respectively

Method	CATH		TS50	
	PPL ↓	AAR %	PPL ↓	AAR %
Structured Transformer [27][†]	6.63	35.82	5.60	42.20
GraphTrans [47][¶]	6.63	35.82	5.40	43.89
ESM-IF [28][†]	6.44	38.3	–	–
ProteinMPNN [29][♮]	4.61	45.96	3.93	54.43
SPIN-CGNN [48][‡]	4.05	54.81	–	–
PiFold [30][†]	4.55	51.66	3.86	58.72
LM-DESIGN(PiFold) [49][‡]	4.52	55.65	3.50	57.89
SPDesign [50][‡]	**2.43**	**67.05**	**2.72**	**68.64**
GVP-GNN [23][♮]	5.29	40.2	–	44.9
GBPNet [51][‡]	5.03	42.70	–	–
DW-GCN [25][♮]	3.94	47.5	–	53.8
DW-GIN [25][♮]	3.85	47.8	–	52.7
DW-GAT [25][♮]	4.13	46.7	–	54.5
$\mathcal{S}$able	–	–	5.91	64.91
$\mathcal{S}$able(canonical)	8.16	46.23	6.40	54.68

reaction classification). In these tasks, each protein is assigned a single label, representing either its fold type or enzyme type. $\mathcal{S}$able demonstrates remarkable accuracy in fold classification, accurately assigning protein structures to their respective fold categories. This showcases $\mathcal{S}$able's ability to effectively capture crucial structural patterns and features essential for fold discrimination. Across three different levels of testing sets, $\mathcal{S}$able achieves state-of-the-art classification accuracy in the superfamily category and performs competitively close to the baseline tasks in the fold and family subsets, highlighting its competitive edge. In enzyme catalytic reaction classification, $\mathcal{S}$able exhibits commendable precision in predicting EC numbers. It surpasses the current baseline model in terms of mean accuracy, underscoring the potential of our model in enzyme-catalyzed reaction classification.

Multi-label classification involves finer-grained functional predictions for proteins, including predicting Gene Ontology (GO) terms and EC numbers. GO term prediction refers to predicting multiple functional annotations that a protein may have, represented in the form of GO terms. For example, a protein may be predicted to be involved in molecular function(MF), biological process(BP), cellular component(CC). EC number prediction refers to predicting one or more enzyme activities that a protein may possess, represented by the third and fourth levels of the EC tree, which describe the types of reactions catalyzed by enzymes. As shown in Fig. 3, $\mathcal{S}$able surpasses other established models in terms of MF and EC number prediction, while matching the top performance in BP and CC prediction. It demonstrates consistent performance across datasets, showcasing its broad generalization capabilities. However, due to imbalanced sample distributions, all methods exhibit relatively lower performance in GO prediction, leaving ample room for improvement. The results are summarized in Fig. 3 and Table C4.

Binding affinity

$\mathcal{S}$able can predict the impact of any type of mutation without restrictions, achieving state-of-the-art performance on S8338 and M1707. It significantly reduces root mean square error (RMSE) on all test datasets compared to other baseline methods, further proving its superiority. These results are summarized in Table 4. Binding affinity prediction aims to forecast the changes in binding affinity resulting from amino acid mutations. Here we compare $\mathcal{S}$able with five state-of-the-art methods across five different benchmark datasets. Among these datasets, three of them contain only single-point mutation data, while the remaining two include multi-point mutations, with S8338 and M1707 also involving some reverse mutations. As $\mathcal{S}$able excels in characterizing complex structures and accurately capturing interchain interactions, it demonstrates exceptional capability in precisely detailing the various interchain interactions within these complexes. Although $\mathcal{S}$able performs a regression task, if binding affinity prediction is framed as a binary classification task, predicting whether a mutation makes the binding more stable or less stable, $\mathcal{S}$able still achieves an accuracy of 88.3% on M1707.

Model quality assessment

$\mathcal{S}$able achieves state-of-the-art performance across various evaluation metrics on the CASP14 test set, and attains state-of-the-art results in five out of eight evaluation metrics on the CASP15 test set.

The critical assessment of protein structure prediction (CASP) is the most authoritative and prestigious event in the field of protein structure prediction, held every 2 years. Therefore, for our evaluation, we select the target and decoy construction test datasets from the two most recent rounds of CASP. It is worth noting that since the majority of methods are published before CASP15, the results of all baselines are reproduced using publicly available model parameters. Further details can be found in the Appendix C.3.5. $\mathcal{S}$able demonstrates its optimization capabilities in predicting both local and global structural quality, outperforming DeepAccNet [61], DeepUMQA [62], and Ornate [60], which solely focus on local structural quality assessment. Additionally, $\mathcal{S}$able exhibits stability in performance across decoys of varying quality.

In CASP15, the performance of participating teams significantly improves due to the remarkable advancements made by AlphaFold2 [36] in the field of structure prediction. Specifically, 81.3% of the decoys in the CASP15 test set have LDDT scores in

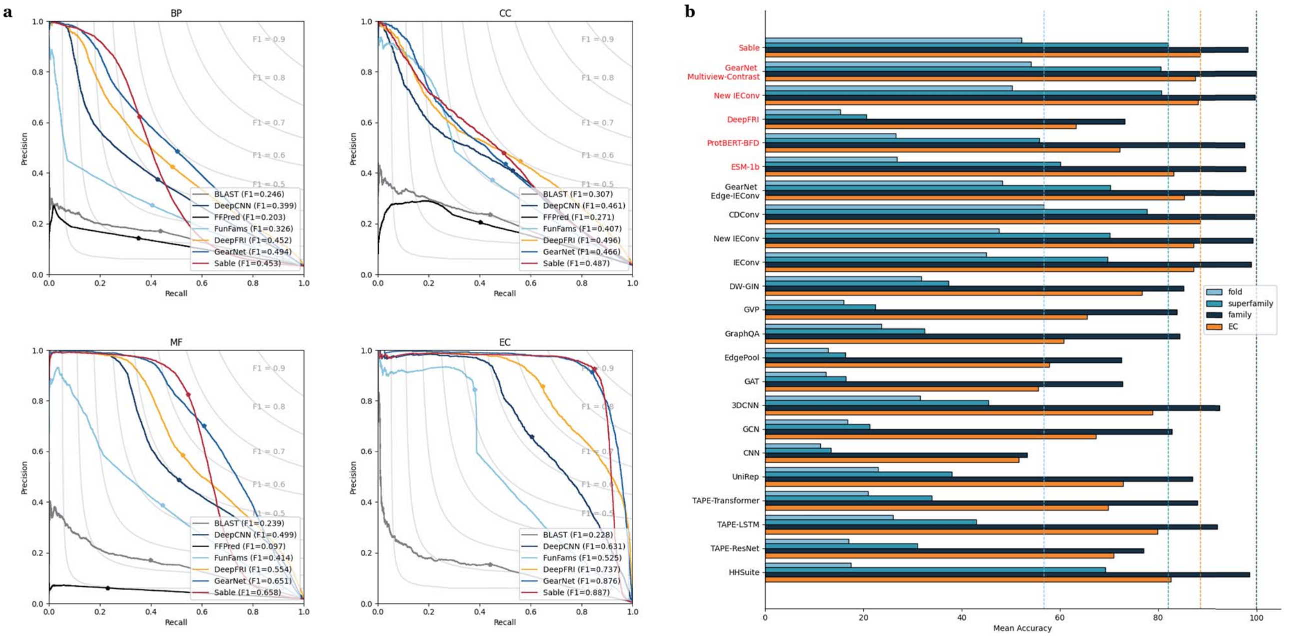

Figure 3. **Overall comparison between 𝓢able and other methods for protein function prediction. (a)** The precision-recall curves illustrating the performance of different methods on GO-BP (top left), GO-CC (top right), GO-MF (bottom left), and EC (bottom right) are displayed. 𝓢able exceeds other methods in MF and EC number prediction, while matching the top performance in BP and CC prediction. **(b)** The bar plots illustrate the results for the fold classification and Enzyme-Catalyzed Reaction classification tasks, with different colored dashed lines representing the levels of state-of-the-art performance across various test sets.

Table 4. Results of various binding affinity prediction methods on the mutation dataset. [†] and [♭] denote results taken from [55] and [56], respectively. The top two results are highlighted in **bold** and underlined, respectively

Method	S1131		S4169		S8338		M1101		M1707	
	r	RMSE ↓	*r*	RMSE ↓	*r*	RMSE ↓	*r*	RMSE ↓	*r*	RMSE ↓
FoldX [52]†	0.46	2.18	0.27	2.73	0.44	2.73	0.34	2.39	0.49	3.02
MutaBind2 [53]†	–	–	–	–	–	–	–	–	0.72	2.25
TopGBT [54]†	0.32	2.31	0.41	1.60	0.61	1.61	–	–	–	–
TopNetTree [54]†	0.29	2.4	0.39	1.65	0.59	1.65	–	–	–	–
GeoPPI-ECDO [55]†	0.58	2.01	0.52	1.48	0.68	1.49	0.53	1.81	0.74	2.21
GeoPPI [55]†	**0.85**	–	**0.78**	–	0.85	–	**0.78**	–	0.89	–
ddg predictor [56]♭	0.65	–	–	–	–	–	–	–	0.59	–
𝓢able	0.72	**1.66**	0.76	**1.17**	**0.92**	**0.80**	0.65	**1.69**	**0.93**	**1.38**

the range of 0.7 to 1, a sharp contrast to only 17.1% in CASP14 (see Fig. C5). Furthermore, most of the decoys are refined using AlphaFold2, leading to a large number of structurally similar and high-quality decoys, which presents new challenges for structure quality assessment methods. With the exception of 𝓢able and AF2Rank, all other methods experience a significant drop in performance on the CASP15 test set, resulting in varying degrees of increase in the RMSE. AF2Rank exhibits a completely different trend in performance between CASP14 and CASP15 when compared to other baselines. In the CASP14 test set, AF2Rank shows relatively poor prediction performance, while in CASP15, it achieves the highest correlation coefficient between predicted and true values. However, since AF2Rank is based on AlphaFold2, it performs poorly when assessing decoys generated by other structure prediction methods, which limits its applicability. In contrast, 𝓢able consistently demonstrates lower RMSE across both test datasets (see Fig. 4a), highlighting its robust prediction capability and adaptability to various task scenarios. The results are summarized in Fig. 4 and detailed in Tables 5 and 6.

Furthermore, we explore the advantages of using pretrained models. Given the ease of access to training data for model quality assessment and the ability to generate a large number of decoys, enlarging the training data size without pretrained models appears to be a viable option. We generate a training dataset containing 1 million decoys and conduct experiments under two settings: with and without pre-training. The results are shown in Tables C7 and C8. Augmenting training data with decoys of varying degrees of distortion does not significantly enhance the model's capacity to discern structural quality. Conversely, the pretrain-finetune paradigm suggests that even with a small amount of fine-tuning data, superior results can be achieved.

Ablation study

We conduct comprehensive ablation experiments aimed at assessing the effectiveness and contributions of individual components within 𝓢able. By systematically disabling or weakening different components of the model, we are able to determine

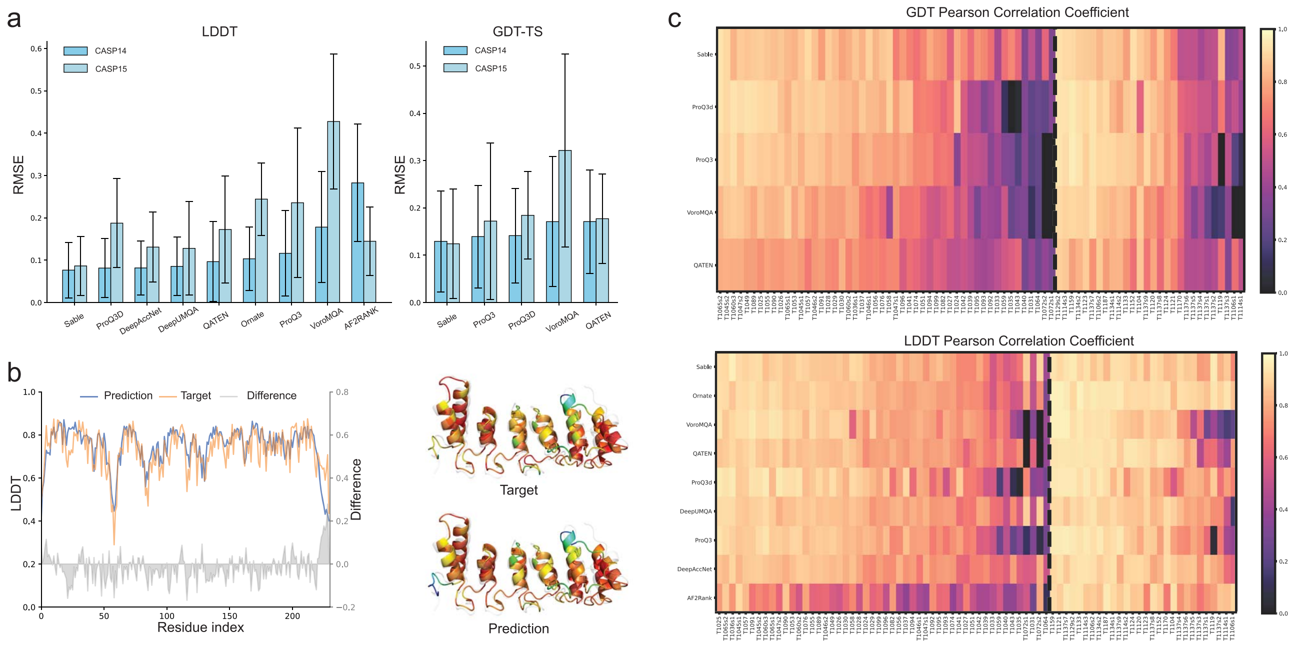

Figure 4. **Comparison evaluations of 𝒮able and baseline methods on model quality assessment tasks. (a)** The bar plot shows the RMSE for two different test sets. **(b)** Visualizing an example (decoy ID: T1134s1TS427_1, PDB ID: 7ubz_A). The protein cartoon (right) is color-coded with a rainbow scheme, indicating the ground truth LDDT scores (top) and the predicted scores (bottom). The specific values are provided by the line plot (left). **(c)** Heatmap displaying the Pearson correlation coefficients of GDT-TS (top) and LDDT (bottom) metrics for each target in the CASP14 (left of the dashed line) and CASP15 (right of the dashed line) datasets.

Table 5. Comparison of different model quality assessment methods on the CASP14 test sets. All results are obtained based on publicly released source code

Method	GDT-TS				LDDT			
	RMSE ↓	r	ρ	τ	RMSE ↓	r	ρ	τ
ProQ3 [57]	0.18	0.67	0.69	0.50	0.15	0.67	0.69	0.51
ProQ3D [58]	0.17	0.65	0.70	0.50	0.11	0.76	0.77	0.58
VoroMQA [59]	0.22	0.45	0.48	0.34	0.22	0.49	0.51	0.37
Ornate [60]	–	–	–	–	0.13	0.72	0.72	0.53
DeepAccNet [61]	–	–	–	–	0.10	0.77	0.78	0.58
DeepUMQA [62]	–	–	–	–	0.11	0.78	0.76	0.57
AF2Rank [63]	–	–	–	–	0.31	0.55	0.55	0.38
QATEN [21]	0.20	0.61	0.61	0.44	0.14	0.59	0.62	0.47
𝒮able	**0.17**	**0.76**	**0.76**	**0.57**	**0.10**	**0.81**	**0.80**	**0.61**

Table 6. Comparison of different model quality assessment methods on the CASP15 test sets. All results are obtained based on publicly released source code

Method	GDT-TS				LDDT			
	RMSE ↓	r	ρ	τ	RMSE ↓	r	ρ	τ
ProQ3 [57]	0.24	0.67	0.64	0.49	0.29	0.43	0.44	0.30
ProQ3D [58]	0.21	0.77	0.72	0.54	0.22	0.64	0.57	0.40
VoroMQA [59]	0.38	0.57	0.50	0.34	0.46	0.30	0.32	0.22
Ornate [60]	–	–	–	–	0.26	0.75	0.64	0.47
DeepAccNet [61]	–	–	–	–	0.16	0.71	0.71	0.53
DeepUMQA [62]	–	–	–	–	0.17	0.64	0.66	0.48
AF2Rank [63]	–	–	–	–	0.16	**0.84**	**0.80**	**0.61**
QATEN [21]	0.20	0.68	0.67	0.48	0.21	0.54	0.58	0.42
𝒮able	**0.17**	**0.78**	**0.75**	**0.56**	**0.11**	0.79	0.69	0.51

their respective contributions to model performance. Our main evaluation is conducted on downstream tasks, specifically protein function prediction and protein design. We implement strict variable controls to ensure that each experiment uses the same hyperparameters for both pre-training and fine-tuning as 𝒮able.

Firstly, we conduct detailed ablation studies on the key encoder, assessing the necessity of incorporating structural information and its effectiveness in representing protein structures. While we acknowledge SPE, distance encoding, and RPE as holistic approaches to representing protein structure, each component is indispen𝒮able. However, we disable each component individually during pre-training to verify their respective contributions. Disabling SPE leads to a slight improvement in Enzyme-Catalyzed Reaction Classification but results in a significant decline in both fold type prediction and protein design performance. Disabling Distance or RPE shows noticeable decreases in Enzyme-Catalyzed Reaction Classification, with the absence of Distance information severely affecting 𝒮able's protein design capability, resulting in a performance loss of nearly 10%. Furthermore, to verify the necessity of integrating structural information, we simultaneously disable SPE and distance encoding, using RPE as the attention bias to communicate with a single representation. This deprives the model of its structural-awareness module, and thus we choose not to evaluate this configuration in the protein design task. The experimental results in Table C10 demonstrate that although protein function prediction exhibits relatively lower dependence on protein structural information, removing the robust structural representation encoder still significantly affects the results of protein function prediction. This effect becomes even more pronounced in tasks such as protein design that heavily rely on structural representation.

We conduct a thorough analysis to determine whether backbone atomic coordinates can effectively represent protein structure. In 𝒮able-residue, the 3D positions of residues are initially represented using the coordinates of the Cα atoms. Specifically, we use the Cα atom's position as a proxy for the entire residue's spatial coordinates. However, constructing the local frame of each residue solely based on a single atom is not feasible, and thus, the Side-Chain Encoding (SPE) is disabled. Substituting residue positional information with Cα atom coordinates disregards a significant amount of interatomic distance information and makes the orientation between residues difficult to define, which results in a noticeable performance decrease in protein design tasks. Nevertheless, as shown in Table 7, this approach significantly improves computational efficiency by reducing the number of model parameters, memory usage, and both training and inference times. The training and inference times measure the computational overhead for processing a single sample during the training and inference phases, where the inference time is the average time measured on the CATH test set of the downstream protein design task. To further evaluate the impact of richer structural representations, we extend 𝒮able by introducing an additional virtual atom whose position is determined as the centroid of all side-chain atoms. This virtual atom, combined with backbone atoms, forms an all-atom model, providing 𝒮able with more comprehensive atomic information. However, as shown in Table 7, despite the inclusion of richer side-chain information, the all-atom model achieves performance comparable to 𝒮able in most cases and even exhibits a significant drop in accuracy on the fold and superfamily test sets of the fold classification task. Furthermore, the all-atom model considerably increases the number of model parameters and computational costs, leading to longer training and inference times. A comparison of the experimental results reveals that while the Cα-based representation sacrifices some geometric information between residues, it significantly reduces computational complexity and improves efficiency. In contrast, the all-atom model attempts to incorporate more information but offers minimal improvements, while adding complexity and computational cost. The limited benefits of this improvement may be related to the uneven distribution of side-chain atoms and the relatively low degrees of freedom in modeling, making the decision to use all-atom information worth further investigation.

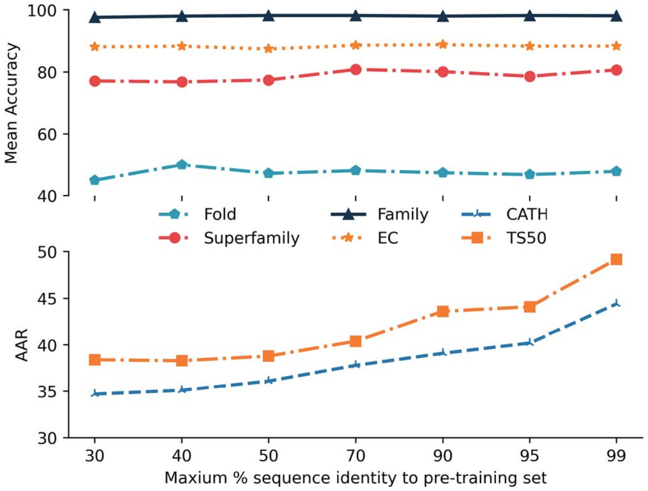

Figure 5. Line plot illustrating the performance of different pre-training data scales on both protein function prediction and protein design tasks. A moderately sized pre-training dataset is sufficient for achieving competitiveness in protein function tasks. However, in protein design tasks, utilizing a more diverse dataset leads to significant performance gains.

Furthermore, we extensively investigate how the size of the pre-training dataset affects the performance of the model. We construct seven different pre-training datasets through sequence clustering with varying sequence identity thresholds (see Appendix D.1 for details). Subsequently, we perform pre-training on these datasets, following the same setting as with 𝒮able. The results reveal that enlarging the pre-training dataset and enhancing its diversity significantly improve the model's performance. In protein function prediction, as depicted in Fig. 5, the performance tends to stabilize across different dataset sizes, indicating that a moderately sized pre-training dataset is sufficient for the model to achieve competitiveness in this task. However, in protein design tasks, a more diverse dataset yields notable performance gains of nearly 10%, as shown in Table 8. Notably, when employing a pre-training dataset generated with a clustering threshold of 99%, the model's performance approaches levels achieved by training on the entire protein structure dataset. Furthermore, utilizing this pre-training dataset also halves the training time, making it a viable alternative for 𝒮able.

We also explore the potential impact of various pre-training strategies, including whether to perform pre-training, whether to freeze pre-training parameters during fine-tuning, and more. For detailed information, please refer to Appendix D.

Discussion

We propose 𝒮able, a structure-centric self-supervised pre-training framework. 𝒮able significantly enhances protein representation learning, enabling various prediction tasks such as generation, classification, and regression. By leveraging Structure Position Encoding and distance information, 𝒮able comprehensively captures structural characteristics of proteins, enhancing its understanding and representation capabilities while maintaining three-dimensional rotational and translational invariance. Through the introduction of a hybrid pre-training strategy, 𝒮able incorporates more accurate atom-level information into pair representations.

Table 7. Comparison of computational efficiency at different levels of structural representation

Methods	Params	GPU memory	Training time	Inference time
$\mathcal{S}$able - residue	50.43M	9032 MiB	0.26 seconds	0.050 seconds
$\mathcal{S}$able - backbone atom	50.50M	24346 MiB	0.56 seconds	0.060 seconds
$\mathcal{S}$able - all atom	50.50M	30442 MiB	0.79 seconds	0.068 seconds

Table 8. Impact of pre-training data size on downstream tasks. By clustering the pre-training data using different sequence identity thresholds as cut-off values, we evaluated its performance in classification and protein design tasks. Average accuracy was used as the evaluation metric for classification tasks, while AAR rate served as the evaluation metric for protein design tasks

Cut-off	Cluster size	Data size	Fold			EC	CATH	TS50
			Fold	Sup	Family			
30%	34,554	29,415	45.0	77.1	97.6	88.1	34.7	38.4
40%	41,492	35,520	**50.0**	76.8	98.0	88.3	35.1	38.3
50%	48,021	41,070	47.2	77.4	98.2	87.4	36.1	38.8
70%	58,877	49,758	48.1	**80.8**	98.2	88.6	37.8	40.4
90%	73,276	59,429	47.4	80.1	98.0	**88.8**	39.1	43.6
95%	85,637	66,347	46.8	78.6	**98.2**	88.3	40.2	44.1
99%	137,521	93,086	47.9	80.6	98.1	88.3	**44.4**	**49.2**

Furthermore, $\mathcal{S}$able, combined with different lightweight fine-tuning strategies, effectively integrates knowledge obtained from large protein structure data. Experimental results validate the outstanding performance of $\mathcal{S}$able across various tasks, particularly in the comprehension of protein complexes.

However, we acknowledge certain limitations of $\mathcal{S}$able. Firstly, there is room for improvement in mapping protein structure information to protein sequences, hence the accuracy of protein sequence design needs refinement. Secondly, the pre-training self-supervised task aimed at recovering distance information between backbone atoms still imposes certain constraints on pair representation learning. We advocate for enhancing $\mathcal{S}$able's capability to directly generate all-atom structures. Lastly, exploring larger-scale pre-training may further enhance protein representation learning effectiveness. In the future, our aim is to extend $\mathcal{S}$able to a wider range of practical problems and application domains.

Key Points

- We introduce $\mathcal{S}$able, a pioneering structural pre-training model that leverages advanced encoding techniques to enhance protein representation learning.
- $\mathcal{S}$able excels in a diverse range of downstream tasks, demonstrating state-of-the-art performance across antibody design, protein design, protein function prediction, binding affinity prediction, and model quality assessment.
- By integrating atom-level and residue-level structural information, $\mathcal{S}$able achieves computational efficiency while maintaining the ability to generate accurate all-atom structures.
- The model's demonstrated applicability in this study indicates its potential to extend beyond the tasks addressed here, making it a versatile framework for a broad spectrum of biological computation challenges.

A Appendix 1 Related work

Protein structure matters

Protein structure is essential to tackle most downstream tasks, as highlighted by the complexity of the protein folding problem. Even when two proteins share similar amino acid sequences, they can fold into entirely distinct three-dimensional structures. This discrepancy becomes particularly apparent during post-translational modifications following protein translation, such as glycosylation, phosphorylation, methylation, acetylation, which profoundly alter the protein's structure and function. Anomalies in these modifications can even lead to serious diseases like leukemia, pancreatic dysfunction, and Alzheimer's disease [64]. In the context of Alzheimer's disease, for instance, a portion of beta-amyloid protein may form toxic plaques due to misfolding, exerting detrimental effects on neural cells [24, 65]. Furthermore, GPCRs in proteins undergo conformational changes in their extracellular regions upon binding with excitatory signaling molecules like odors, hormones, neurotransmitters, and chemokines [66, 67]. Figure A1 presents a specific example illustrating the conformational changes that occur in the Gα subunit (comprising two subdomains, the Ras domain and the AHD domain) during receptor-mediated G protein nucleotide exchange. This further emphasizes the critical role of protein structure in regulating biological functions.

Protein representation learning

Protein representation learning presents a fundamental challenge in bioinformatics, as it seeks to develop effective methodologies for accurately describing the structure and function of proteins. There are two primary approaches to tackling this challenge: one centers on the analysis of amino acid sequences, while the other focuses on the study of three-dimensional protein structures.

Protein sequence representation learning In recent years, advancements in protein sequence representation learning introduce various techniques that apply NLP methodologies to biological sequences through transfer learning [68, 69]. The

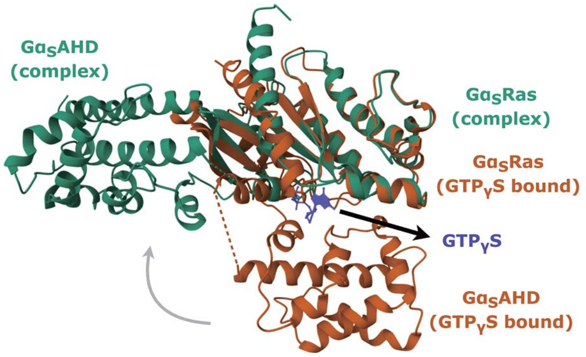

Figure A1. **Interaction-mediated conformational changes.** The figure depicts structural changes between receptor-bound and nucleotide-free $G\alpha_S$ (turquoise, PDBID 3SN6) and $G\alpha_S$(burnt sienna, PDBID 1AZT) bound to $GTP_\gamma S$ (indigo). Research reveals that the receptor for Gs induces a movement of the α-helical domain ($G\alpha_S$AHD) of $G\alpha_s$, causing it to shift outward relative to its position in the GTPγS-bound state, thereby triggering conformational changes (the receptor of G_S is not shown in the figure). This example is derived from [12].

TAPE benchmark framework is developed for semi-supervised learning of proteins [4], ESM-1b explores the application of self-supervised language models in the protein domain by training on unlabeled amino acid sequences in a manner akin to character-level language models [5]. ProtTrans extends this approach by scaling up the dataset and comparing the impact of autoregressive and autoencoding pre-training on subsequent supervised training success [8]. ProGen optimizes the model by predicting the probability of the next amino acid type, without relying on explicit structural information or co-evolutionary assumptions [70]. xTrimoPGLM adopts the general language model (GLM) framework, leveraging its bidirectional attention and autoregressive blank filling objectives to effectively handle both protein understanding and generation tasks [6].

Protein structure representation learning Given the critical role of protein structure in determining function, structure-based representation methods emerge as a superior solution. In the past, these methods often rely on manually designed feature extraction techniques, such as using Voronoi tessellation to describe protein contact areas [71] or employing 3D Zernike descriptors to characterize protein surface properties [72–74]. Although these methods are effective to some extent, they struggle to capture complex protein structural information. With the advancement of deep learning, a new generation of methods continuously emerges. In the early stages, 3D convolutional neural networks (3D CNNs) are employed to voxelate protein structures [17, 18]. Subsequently, GNNs gain prominence by abstracting protein structures into graphs. Some methods even integrate multiple general GNN frameworks to introduce geometric information [23] or maintain SO(3)-equivariance properties [25], aiming for a more precise representation of protein structures. Furthermore, the representation of local protein structures also garners significant attention. For instance, some methods concentrate on extracting information from the protein surface, as seen in MaSIF [15] and dMaSIF [16]. This is crucial for identifying potential protein–protein interaction interfaces. Uni-Mol, on the other hand, focuses on learning universal representations, with particular emphasis on pseudo protein pockets that could form interfaces [14]. Some approaches attempt to enhance their performance by fine-tuning parameters of sequence-based pre-trained models and introducing structure-aware modules [75]. Contrastive learning has emerged as a new trend in the field of pre-training, aiming to learn structural representations by maximizing the distance metric between different protein structures as a training objective [26, 76]. These methods are pretrained on large-scale datasets, enhancing the quality and generalizability of the representations.

B Appendix 2: Model details

Protein representation

This section provides additional details of the $\mathcal{S}$able, including four specific algorithms. These algorithms are critical for understanding the various processing and encoding steps used in the model.

Algorithm 1 Distance binning

```
Definition binning($d_{ij}$, $n_{\text{bins}}$, supremum=128)
    width = supremum/$n_{\text{bins}}$
    indices = $\lfloor d_{ij}/\text{width} \rfloor$
    $\tilde{d}_{ij}$ = Linear(indices)
    return $\tilde{d}_{ij}$
end
```

Algorithm 1 is a distance binning algorithm used to map continuous distance values into discrete distance intervals. The core idea of the algorithm is to divide the distance range into multiple equally sized intervals and map the distance values into the corresponding intervals. Through this process of discretization, it is possible to reduce the dimensionality of the data, extract features, or transform the distance measure into a more manageable form.

Figure B1 and Algorithm 4 presented describe the SPE method, which encodes the spatial positions of residues into a higher-dimensional space by utilizing local coordinate frames. In the SPE process, an orthogonal basis is constructed using the coordinates of the N, Cα, and C atoms within residue i via the Gram-Schmidt procedure. This basis forms a local frame $\mathcal{O}_i$, which is then employed to project the displacement vectors between residues. The displacement information, combined with its norm, is discretized into bins, one-hot encoded, and linearly transformed to produce the final encoded positions. This encoding captures the geometric relationships between residues, facilitating accurate representation of spatial configurations in the model.

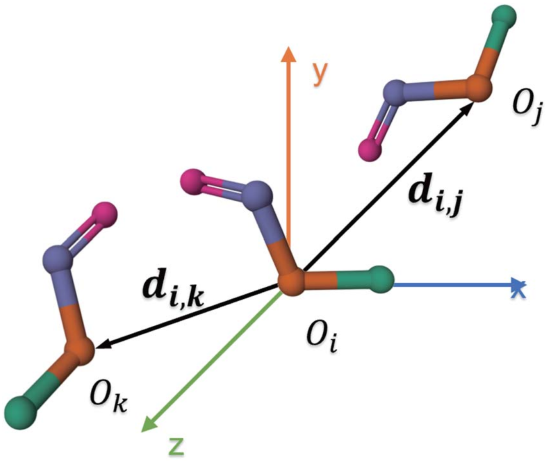

Figure B1. The sketch map of SPE, with N, Cα, C, and O colored in teal, burnt orange, periwinkle, and magenta, respectively.

Algorithm 2 converts a scalar x into a one-hot encoded vector based on predefined bins. The steps are as follows:

Algorithm 2 One-hot encoding

```
Definition   one_hot($x$, $v_{\text{bins}}$)
    $p = 0$
    $b = \arg\min(|x - v_{\text{bins}}|)$
    $p_b = 1$
    return $p$
end
```

Algorithm 3 encodes the relative positions of residues using predefined bins and a linear transformation. The steps are as follows:

Algorithm 3 Relative position encoding(RPE)

```
Definition   RPE($r_i^{\text{index}}, r_j^{\text{index}}, v_{\text{bins}} = [-32, -31, \ldots, 32]$)
    $r_{ij} = r_i^{\text{index}} - r_j^{\text{index}}$
    $\tilde{r}_{ij} = \text{Linear}(\text{one\_hot}(r_{ij}, v_{\text{bins}}))$
    return $\tilde{r}_{ij}$
end
```

C Appendix 3: Experiments detail & reproduce

Dataset statistics

Table C1 showcases the dataset statistics for both pre-training and downstream tasks, with data splitting principles primarily drawn from well-established benchmarks in the field. For tasks such as antibody design and binding affinity prediction, cross-validation is employed, and the table provides the approximate data size for each fold. Further details can be found in the "Datasets Details." subsection of the respective downstream tasks.

Pre-training implementation details

For the two self-supervised tasks corresponding to pre-training, namely "masked token prediction" and "atom distance restoration", we employ two distinct loss functions, specifically cross-entropy loss and Smooth L1 loss. To facilitate effective model training, we combine these two loss functions with equal weights of 1:1, constituting the overall loss function during the pre-training phase. All models are trained on 8 NVIDIA A100 40GB GPUs. Additionally, further hyperparameter configurations related to pre-training can be found in Table C2.

Fine-tuning implementation details

Antibody design

Datasets details

We collect data from the Structural Antibody Database (SAbDab) [77], which contains structural information for antibody-antigen complexes. In the antibody sequence-structure co-design task, we adopt two approaches to construct the datasets. The first approach follows the partitioning procedure described in RefineGNN [22]. Taking the construction process of CDR-H3 as an example, we first perform clustering of all CDR-H3 sequences using the MMseqs2 tool, with a sequence identity threshold set at 40%. From the clustering results, we select one representative sequence from each cluster. These representative sequences are then randomly divided into training, validation, and test sets in an 8:1:1 ratio. The same clustering and partitioning process is applied to construct datasets for CDR-H1 and CDR-H2.

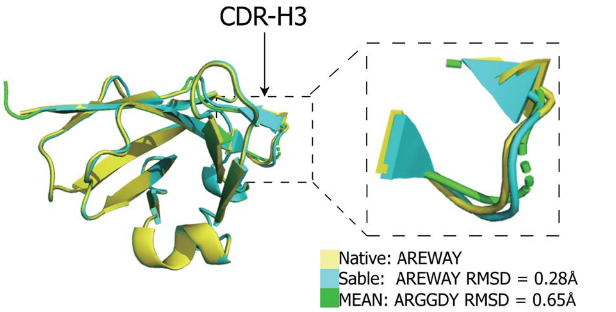

Figure C1. Schematic illustration of the co-designed sequence structure of CDR-H3 (PDB ID: 5KQV)

The second approach employs k-fold cross-validation, following the procedure in MEAN [41]. After removing antibody-antigen complexes lacking light chains or antigens, we cluster the remaining 3124 complexes with a sequence identity threshold of 40%. After clustering, we retain 765 clusters for CDR-H1, 1094 clusters for CDR-H2, and 1658 clusters for CDR-H3. Each of the clusters is then randomly divided into 10 equal parts. The ith fold is used as the test set, the (i-1)-th fold is used as the validation set, and the remaining folds are used as the training set. In the case where the first fold is used as the test set, the 10th fold is designated as the validation set.

In the antigen-specific antibody design task, we follow the partitioning procedure described in HSRN [45]. First, we filter antibody structures without antigen information from the SAbDab database and remove duplicate data. To evaluate the model's ability to generalize to new antigen binding sites, we exclude sequences from the training set that have more than 70% sequence identity with those in the test set, ensuring that the model does not encounter similar antibody sequences during training. The test set for this task is derived from the benchmark dataset for antigen binding site design constructed in RAbD [44]. This benchmark dataset consists of 60 antibody-antigen complexes with diverse antigen types.

Baselines

In the experiments involving co-design of antibody sequence and structure, we initiate our investigation by considering a sequence-based LSTM model [37, 38]. This approach primarily focuses on modeling sequence information. Subsequently, we introduce RefineGNN [22], which incorporates three-dimensional structural information and employs an iterative optimization strategy for autoregressive co-design of antibody sequence and structure. AbBERT-HMPN [40] capitalizes on an antibody pre-trained language model, enabling one-shot generation of antibody sequences. Additionally, we employ a multi-round 3D equivariant model MEAN [41]. AbODE [42] is a generative model that extends graph PDEs to jointly model antibody-antigen interactions. By formulating a coupled neural ODE system, it enables the end-to-end, one-shot generation of antibody sequences and structures while accounting for contextual and external interactions. AntiDesigner [43] proposes a novel geometric modeling approach using a protein complex invariant embedding network, which

Algorithm 4 Spatial position encoding (SPE)

```
Definition  $\mathrm{SPE}(x_{i,1}, x_{i,2}, x_{i,3}, x_{j,2} \in \mathbb{R}^3, v_{\mathrm{bins}} = [0, ..., 127])$
    $\vec{v}_{i,1} = x_{i,1} - x_{i,2}; \vec{v}_{i,2} = x_{i,3} - x_{i,2}$
    $\vec{e}_{i,1}, \vec{e}_{i,2} = \text{Gram-Schmidt}(\vec{v}_{i,1}, \vec{v}_{i,2})$        ▷ Compute an orthogonal basis
    $\vec{e}_{i,3} = \vec{e}_{i,1} \times \vec{e}_{i,2}$
    $\mathcal{O}_i = \mathrm{concat}(\vec{e}_{i,1}, \vec{e}_{i,2}, \vec{e}_{i,3})$        ▷ Local frame constructed by the $i$-th residue
    $\vec{d}_{ij} = x_{i,2} - x_{j,2}$
    $p_{ij} = \mathrm{concat}(\|\vec{d}_{ij}\|, \vec{d}_{ij} \circ \mathcal{O}_i)$
                                             ▷ $\circ$ represent the projection of $\vec{d}_{ij}$ in the local frame $\mathcal{O}_i$
    $\tilde{p}_{ij} = \mathrm{Linear}(\mathrm{one\_hot}(\mathrm{binning}(p_{ij}), |v_{\mathrm{bins}}|), v_{\mathrm{bins}})$
                                             ▷ $|v_{\mathrm{bins}}|$ refers to the number of elements contained in $v_{\mathrm{bins}}$.
    return $\tilde{p}_{ij}$
end
```

Table C1. Dataset statistics for pre-training and downstream tasks

DATASETS	# TRAIN	# VALID	# TEST	TASK
Pre-training	197,170	1,178	–	Pre-training
Antibody Design - *CDR-H1*	4,050	356	326	Generation
Antibody Design - *CDR-H2*	3,874	483	376	Generation
Antibody Design - *CDR-H3*	3,896	403	437	Generation
Antibody Design - *CDR-H1-kfold*	612	76	77	Generation
Antibody Design - *CDR-H2-kfold*	875	109	110	Generation
Antibody Design - *CDR-H3-kfold*	1,327	165	166	Generation
Antibody Design - *RAbD*	2,237	155	56	Generation
Protein Design - *CATH v4.2*	18,024	608	1,120	Generation
Protein Design - *TS50(canonical)*	17,669	577	50	Generation
Fold Classification - *Fold*	12,312	736	718	Classification
Fold Classification - *Superfamily*	12,312	736	1,254	Classification
Fold Classification - *Famliy*	12,312	736	1,272	Classification
Enzyme-Catalyzed Reaction Classification	29,215	2,562	5,651	Classification
Gene Ontology Term Prediction	29,896	3,321	3,415	Classification
Enzyme Commission Numbers Prediction	15,549	1,728	1,919	Classification
Binding Affinity - *S1131*	904	113	114	Regression
Binding Affinity - *S4169*	3,336	416	417	Regression
Binding Affinity - *S8338*	6,671	833	834	Regression
Binding Affinity - *M1101*	824	102	102	Regression
Binding Affinity - *M1707*	1,366	170	171	Regression
Model Quality Assessment - *CASP14*	33,510	3,705	24,313	Regression
Model Quality Assessment - *CASP15*	33,510	3,705	13,260	Regression

captures intra- and inter-component interactions at the atomic level.

Expanding our scope to encompass the design of antigen-specific binding antibodies, we introduce an additional set of methodologies. Among these, we incorporate the physics-based traditional approach RAbD [44]. Furthermore, we integrate the hierarchical model HSRN [45], tailor-made for antibody-antigen interface design. To enhance the design of antibody heavy chains, MEAN [41] and the end-to-end 3D equivariant model dyMEAN [46] not only consider antigen but also incorporate antibody light chain information into the known conditions.

Evaluation metrics

We employ AAR to evaluate the quality of generated CDR sequences, RMSD to assess the quality of generated CDR structures, and ligand-RMSD to gauge the quality of the antibody-antigen complex structure. AAR measures the similarity between the generated CDR sequences and the target sequences, quantifying the proportion of target amino acids successfully recovered in the generated CDRs, thus capturing sequence-level quality. RMSD focuses on the spatial configuration of the CDR structure, calculating the average deviation of atomic coordinates between the generated and target CDR structures. Ligand-RMSD quantifies structural differences by aligning the receptors and computing the deviation in atomic coordinates between the two ligands. For the calculations in Fig. 2, we consider the antibody as the receptor.

Protein design

Datasets details

We collect data from the protein structure classification database CATH. In the CATH v4.2 40% non-redundant dataset, 18 024 chains are collected as the training set, 608 chains as the validation set, and 1120 chains as the test set according to the way [27] divides the datasets. In addition, we also demonstrate the model's performance on TS50 [78], a universal benchmark dataset for protein design tasks. Due to the lack of a canonical training set specifically for the TS50 test dataset, we follow the approach of [23, 25, 79] and remove 435 protein structure data similar to TS50 from the training dataset of CATH v4.2 as a new training set.

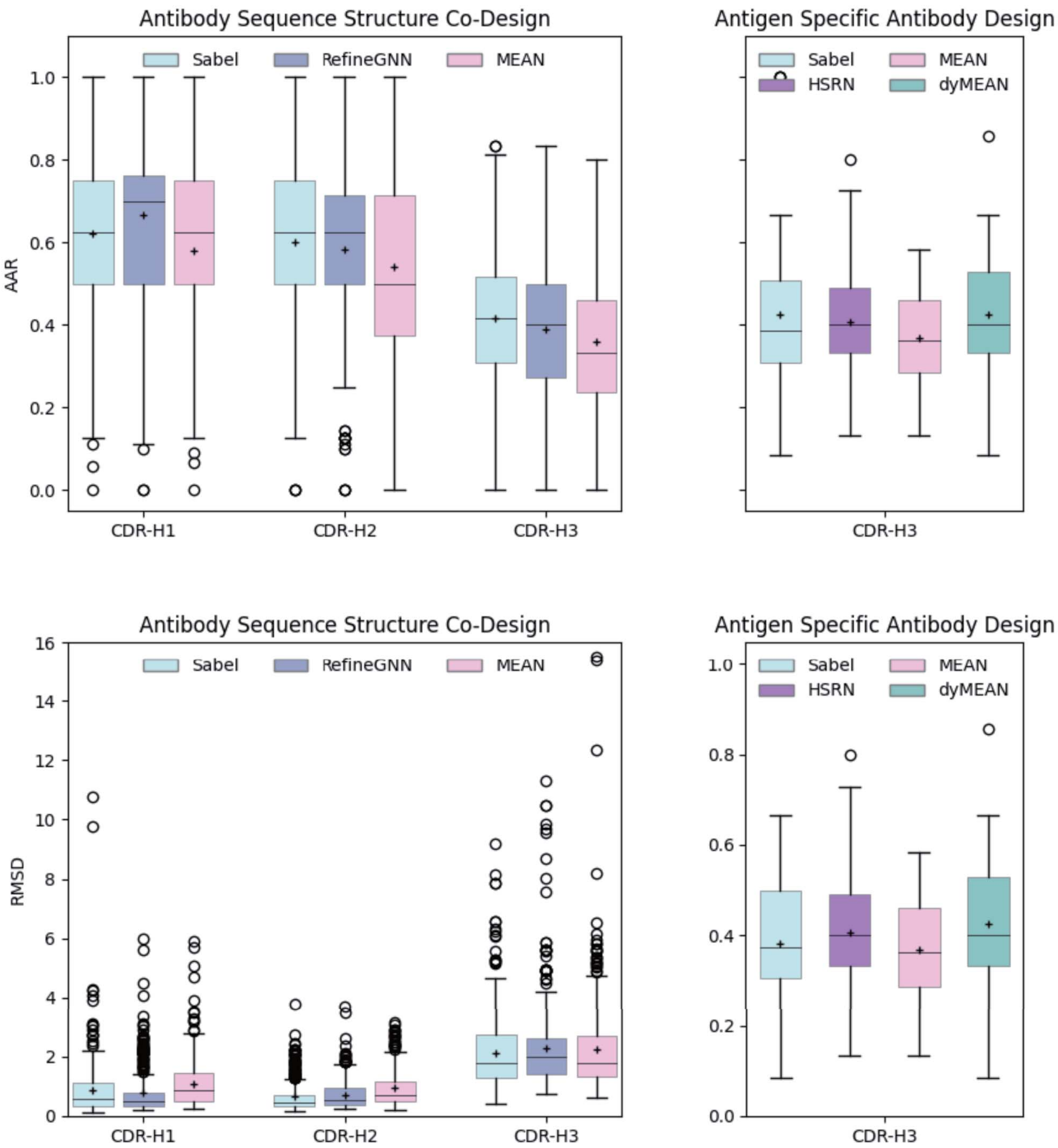

Figure C2. Boxplots of AAR (top) and Cα-RMSD (bottom) for antibody sequence-structure co-design (left) and antigen-specific antibody design (right).

Baselines

We conduct a comparative analysis of our pre-trained model with various baseline approaches, encompassing specialized generative models tailored for protein design and methods focusing on protein representation learning. Structured Transformer [27], GraphTrans [47], ESM-IF [28], ProteinMPNN [29], SPIN-CGNN [48], PiFold [30] and LM-DESIGN [49] SPDesign [50] are state-of-the-art methods for protein design, while GVP-GNN [23], GBPNet [51], DW-GCN, DW-GIN, and DW-GAT [25] aim to construct general protein representation methods, achieving advanced performance in protein design tasks as well. With method ESM-IF utilizing CATH v4.3 for training, the remaining methods are trained using CATH v4.2. All protein representation methods, indicated by gray shading in Table 3, employ a canonical training set for the TS50, while the training sets used by the other methods are not explicitly specified.

Evaluation metrics

To evaluate protein sequence generation tasks, we utilize perplexity and AAR as evaluation metrics. Perplexity quantifies the model's uncertainty during sequence generation, with lower values indicating better alignment between the model's predictions and the native sequence. AAR quantifies the proportion of matching amino acids between the generated and target sequences. Higher AAR values indicate greater similarity and better performance of the model.

Table C2. Hyperparameters setup during pre-training

Hyperparameters	Base size
Layers	15
Hidden size	512
FFN hidden size	2048
Attention heads	4
Attention head size	128
Training epochs	500
Batch size	32
Adam ϵ	1e-12
Adam β	(0.9, 0.82)
Peak learning rate	1e-4
Learning rate schedule	Polynomial
Warmup steps	5000
Gradient clip norm	1.0
Dropout	0.1
Weight decay	1e-4
Activation function	GELU
Sequence crop size	256
Spatial crop ratio	0.5
Mask ratio	(0.15, 0.5, 1.0)
Mask ratio probability	(0.6, 0.2, 0.2)
Noise type	$\mathcal{N}(0, 0.1), \mathcal{N}(0, 1)$
Noise probability	(0.2, 0.8)
Vocabulary size (residue types)	24

Protein function

Datasets details

To assess the model's proficiency in protein function prediction, we utilize four tasks, comprising two single-classification tasks: fold classification and enzyme reaction classification, along with two multi-label classification tasks: GO term prediction and EC number prediction. The fold classification task reveals the relationship between protein structure based on three-dimensional similarity and evolution. We collect protein structure data from the SCOP v1.75 database [80] after clustering with 95% sequence identity. Following the data processing method outlined in [81], redundancy is eliminated among the training, validation, and test sets. Evaluation is conduct on three distinct test sets at different levels: fold (proteins not belonging to the same superfamily as those in the training set), superfamily (proteins not belonging to the same family as those in the training set), and family (proteins that appear in the same family as those in the training set). Enzymes play a crucial role as catalysts in proteins. The EC defines a numbering and naming scheme for different categories of enzymes, consisting of four digits. Protein data annotated with EC numbers were collected from the SIFTS database [82], and the dataset was partitioned into training, validation, and test sets, ensuring a sequence identity of less than 50% between different data splits. The GO describes knowledge of the biological domain with respect to three aspects: MF, BP, CC. Data are sourced from the PDB, and a 95% sequence identity clustering is applied to all annotated PDB chains. A representative PDB chain is selected from each cluster, and their associated GO terms are retrieved from the SIFTS and UniProt knowledgebase databases. EC number prediction entails forecasting the EC numbers of proteins based on their characteristics and functions. Specifically, the enzyme classifications are obtained from levels 3 and 4 of the EC tree. The GO and EC datasets for multi-label classification tasks are divided into training, validation, and testing sets following the partitioning methodology outlined by [19].

Baselines

In comparison with the protein function prediction task, we examine a range of baseline methods with the aim of comprehensively assessing the performance of our model and providing reference for further investigation. Firstly, we employ traditional methods such as TMalign [83], BLAST [84], HHSuite [85], and PSI-BLAST [86] as baselines, which have widespread applications in protein structure and sequence similarity analysis. Secondly, our focus turns to sequence-based methods, which primarily utilize the amino acid sequence information of proteins for classification: DeepSF [81], TAPE [4], UniRep [69], and CNN [87]. Additionally, we also delve into structure-based methods, which center on the three-dimensional structural information of proteins, encompassing factors such as inter-amino acid distances and secondary structures: GCN [88], 3DCNN [18], GAT [89], EdgePool [90], GraphQA [20], GVP [23], DW-GIN [25], IEConv [31], and CDConv [32]. Moreover, some methods employ extensive unlabeled data in their model training through pre-training strategies, aiming to enhance the model's feature representation capabilities. For instance, ESM-1b [5] utilizes the UniRef50 dataset, ProtBERT-BFD [8] integrates the BFD database, DeepFRI [19] leverages information from the protein sequence database Pfam for pre-training, New IEConv [76] utilizes the PDB database, and GearNet [26] combines data from AlphaFoldDB. The predicted results of BLAST [84], FunFams [91], DeepCNN [92], and FFPred [93] are obtained from the Source Data provided by DeepFRI.

Evaluation metrics

We evaluate the performance of models in single-label classification tasks (such as Fold classification and Enzyme-Catalyzed Reaction Classification tasks) using mean accuracy, which denotes the proportion of correctly classified predictions among all predictions.

For multi-label classification, we treat it as multiple binary classification tasks and employ the F_{max} as the final evaluation metric for predicting GO terms and EC number tasks. F_{max} refers to the maximum value that the F-measure can achieve across different thresholds. The F-measure is the weighted harmonic mean of precision and recall. Before we introduce the method for calculating the precision and recall, we first define a characteristic function, which maps elements of a given set to 1, and all other elements to 0, i.e.,

$$\chi_A(x) = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{if } x \notin A. \end{cases}$$

For a designated target protein i and a decision threshold t ranging from 0 to 1, the precision and recall are determined as

$$Precision_i(t) = \frac{\sum_f \chi_{P_i(t) \cap T_i}(f)}{\sum_f \chi_{P_i(t)}(f)}$$

and

$$Recall_i(t) = \frac{\sum_f \chi_{P_i(t) \cap T_i}(f)}{\sum_f \chi_{T_i}(f)},$$

where f is a functional term in the ontology, T_i is a set of experimentally function terms for protein i, and $P_i(t)$ is a set of predicted terms for protein i with score not less than t.

Table C3. One-shot generates results for the antibody design of six CDRs simultaneously

Model	CDR-L1	CDR-L2	CDR-L3	CDR-H1	CDR-H2	CDR-H3
dyMEAN [46]	73.55	83.10	52.12	75.72	**68.48**	37.51
$\mathcal{S}$able	**78.19**	**84.86**	**72.21**	**77.33**	68.34	**39.58**

The precision and recall at threshold t are computed as the averages across all proteins:

$$Precision(t) = \frac{1}{m(t)} \cdot \sum_{i=1}^{m(t)} Precision_i(t)$$

and

$$Recall(t) = \frac{1}{n} \cdot \sum_{i=1}^{n} Recall_i(t)$$

Here, $m(t)$ denotes the number of proteins with at least one prediction made above threshold t, and n represents the total number of proteins.

F_{max} can be calculated by iterating over all thresholds ranging from 0 to 1:

$$F_{max} = \max_t \left\{ \frac{2 \cdot Precision(t) \cdot Recall(t)}{Precision(t) + Recall(t)} \right\}$$

Binding affinity

Datasets details

Binding affinity prediction seeks to deduce alterations in protein–protein interactions by predicting the repercussions of amino acid mutations. Proteins are referred to as wild-type when no mutations occur, while those with mutations in specific residues are termed mutant-type. Due to the absence of the native three-dimensional structure of mutants, we assume that the mutational effects do not alter the backbone structure of the protein. We perform a 10-fold cross-validation for binding affinity prediction, utilizing five commonly used datasets: M1101 [94], S1131 [95], M1707 [53], S4169 [96], and S8338. These datasets are derived from the SKEMPI [97], SKEMPI 2.0 [98], and Antibody-Bind (AB-Bind) [94] databases, which compile extensive experimental data widely employed in constructing benchmark datasets for predicting protein–protein interactions. For training, validation, and testing purposes, we randomly divide the datasets into 10 equally sized subsets in an 8:1:1 ratio. Below are detailed descriptions of the five datasets we utilize.

The initial letter in each dataset denotes the type of mutation data it encompasses. “S” signifies that the dataset exclusively includes single-point mutation data, while “M” indicates the presence of multi-point mutation data. The digit following the letter represents the count of mutation data in the dataset. Due to inconsistencies between the residue types of wild-type mutation sites and their corresponding positions in the PDB, the final dataset often contains fewer data points than indicated by the numerical value. The PDB IDs associated with these problematic data are summarized in Table C5.

S1131 is a non-redundant dataset of interface mutations collected from the SKEMPI database. The variants originate from 112 protein complexes, encompassing experimentally determined changes in free energy (ΔΔG) caused by mutations in interface residues.

S4169 is a curated dataset comprising 4169 mutation variants from 319 distinct protein complexes, meticulously collected by filtering single-point mutations with available experimental crystal structures of the wild-type complexes from the SKEMPI 2.0 database.

S8338 contains all the forward ($\Delta\Delta G_{wt\to mut}$) and reverse ($\Delta\Delta G_{mut\to wt}$) mutations from S4169.

M1101 also known as the AB-Bind database, comprises 645 single-point mutations and 456 multi-point mutations. These variants are derived from 32 antibody-antigen complexes, each consisting of 7 to 246 variants, with each variant involving a maximum of 16 mutation sites. Among them, 27 complexes possess experimentally determined structures, while the structures of the remaining 5 complexes are based on homology models constructed from templates with 76–90% sequence similarity.

M1707 is a meticulously curated dataset of multi-point mutations sourced from the SKEMPI 2.0 database. It encompasses variants originating from 120 distinct wild-type protein complexes. The mutations span a variable number of mutation sites, ranging from 2 to 10, and comprise a total of 1337 forward mutations and 370 reverse mutations.

Baselines

We compare method with six recent or established state-of-the-art baselines. FoldX employs an empirical force field to predict the impact of mutations on the binding energy of protein complexes [52]. MutaBind2 utilizes a scoring function composed of seven terms to predict changes in binding affinity [53]. TopGBT and TopNetTree combine topology-based approaches with machine learning techniques [54]. GeoPPI employs a geometric representation that learns encoded topological features of protein structures to predict protein–protein interaction effects [55]. The ddg predictor utilizes an attention-based geometric neural network. By learning the geometric information of mutation pairs within protein structures and using an attention mechanism, it captures crucial interaction features to predict the effects of mutations [56].

Evaluation metrics

We utilize Pearson correlation coefficient (Rp) and RMSE as evaluation metrics to quantify the disparity between predicted binding affinity values and ground-truth. The Pearson correlation coefficient assesses the degree of linear relationship between prediction and ground-truth, with a value closer to 1 indicating a stronger linear relationship. On the other hand, RMSE measures the average magnitude of deviations between predicted and ground-truth, with a smaller value indicating higher prediction accuracy.

Model quality assessment

Datasets details

Model quality assessment aims to evaluate the quality of protein model structures (called decoys), when the experimental

Table C4. Comparison of classification task results with state-of-the-art methods includes F_{max} for EC number prediction and GO term prediction, as well as mean accuracy (%) for Fold and Enzyme-Catalyzed Reaction classification. [‡] denotes results taken from [81], [‡] denotes results taken from [31], [∗] denotes results taken from [25], [♭] denotes results taken from [76], [§] denotes results taken from [32], [♮] denotes results taken from [26], and [¶] represents the reproduced results. Method highlights the integration of structural information. **Bold** and underline indicate the top two results obtained under settings w/o pre-training and w/ pre-training, respectively

	Method	Enzyme	Gene ontology			Fold			Enzyme
		Commission	BP	MF	CC	Fold	Sup	Family	Reaction
w/o pre-training	TMalign [83]♭	–	–	–	–	34.0	65.7	97.5	–
	BLAST [84]¶	0.228	0.246	0.239	0.307	–	–	–	–
	HHSuite [85]♭	–	–	–	–	17.5	69.2	98.6	82.6
	PSI-BLAST [86]‡	–	–	–	–	5.60	42.20	96.80	–
	FunFams [91]¶	0.525	0.326	0.414	0.497	–	–	–	–
	DeepCNN [92]¶	0.631	0.399	0.499	0.461	–	–	–	–
	FFPred [93]¶	–	0.203	0.097	0.271	–	–	–	–
	DeepSF [81]‡	–	–	–	–	40.95	50.71	76.18	–
	TAPE-ResNet [4]♮†	0.605	0.280	0.405	0.304	17.0	31.0	77.0	70.9
	TAPE-LSTM [4]♮†	0.425	0.225	0.321	0.283	26.0	43.0	92.0	79.9
	TAPE-Transformer [4]♮†	0.238	0.264	0.211	0.405	21.0	34.0	88.0	69.8
	UniRep [69]†	–	–	–	–	23.0	38.0	87.0	72.9
	CNN [87]♮	0.545	0.244	0.354	0.287	11.3	13.4	53.4	51.7
	GCN [88]†	0.320	0.252	0.195	0.329	16.8	21.3	82.8	67.3
	3DCNN [18]†	–	–	–	–	31.6	45.4	92.5	78.8
	GAT [89]♮	0.368	0.284	0.317	0.385	12.4	16.5	72.7	55.6
	EdgePool [90]†	–	–	–	–	12.9	16.3	72.5	57.9
	GraphQA [20]†	0.509	0.308	0.329	0.413	23.7	32.5	84.4	60.8
	GVP [23]♮	0.489	0.326	0.428	0.420	16.0	22.5	83.8	65.5
	DW-GIN [25]*	–	–	–	–	31.8	37.3	85.2	76.7
	IEConv [31]†	–	–	–	–	45.0	69.7	98.9	87.2
	New IEConv [76]♭	0.735	0.374	0.544	0.444	47.6	70.2	99.2	87.2
	CDConv [32]§	**0.820**	**0.453**	**0.654**	**0.479**	**56.7**	**77.7**	**99.6**	**88.5**
	GearNet-Edge-IEConv [26]♮	0.810	0.400	0.581	0.430	48.3	70.3	99.5	85.3
w/ pre-training	ESM-1b [5]♮	0.864	0.452	0.657	0.477	26.8	60.1	97.8	83.1
	ProtBERT-BFD [8]♮	0.838	0.279	0.456	0.408	26.6	55.8	97.6	72.2
	DeepFRI [19]♮	0.631	0.399	0.465	0.460	15.3	20.6	73.2	63.3
	LM-GVP [24]♮	0.664	0.417	0.545	**0.527**	–	–	–	–
	New IEConv [76]♭	–	–	–	–	50.3	80.6	99.7	88.1
	GearNet-Multiview-Contrast [26]♮	0.874	**0.490**	0.654	0.488	**54.1**	80.5	**99.9**	87.5
	𝓢able	**0.887**	0.454	**0.658**	0.487	52.2	**82.0**	98.2	**88.5**

Table C5. Summary of problematic data pdb ids for binding affinity prediction task

Datasets	PDB IDs	Total mutations
M1101	1DVF, 1JRH, 1N8Z 2NYY, 2NZ9, 3NPS	73

resolved structures (called native) are unknown. There are multiple metrics available to assess the similarity between predicted and native structures, such as global metrics like GDT-TS and TM-score, as well as local indicators like LDDT and CAD score that provide per-residue scores. For this task, we choose to fit widely-used evaluation metrics, GDT-TS and LDDT, to provide a comprehensive assessment of decoy quality. To construct the decoy training set, we initially selected protein crystal structures stored on the PISCES server [99] up to May 2018 and applied a series of filtering criteria proposed by [61] and [62]: (i) Maximum sequence redundancy limited to 40%. (ii) Minimum resolution set at 2.5. (iii) Number of residues in each protein chain ranging from 50 to 300. (iv) Proteins either exist as monomers or interact with other chains with minimal energy (less than 1 kcal/mol Rosetta energy). After filtering and screening, 7443 protein structures remain. To generate diverse and reasonable decoys for these proteins, we employ RosettaCM [100] for comparative modeling and local structure perturbation, and trRosetta [101] for deep learning guided folding. For each unique protein chain, we select five decoys with varying qualities. The final training set comprises 37 215 decoys, with 33 510 used for training and 3705 for validation.

Our test set consists of decoys submitted by participating teams in the recent two rounds of the Critical Assessment of Protein Structure Prediction (CASP) competition. The set includes targets with experimentally resolved structures, paired with their corresponding decoys. We primarily focus on evaluating monomer structures, although our approach can be easily extended to assess the quality of multimer structures as well. To ensure diversity and reliability, we perform redundancy reduction on these decoys. This involved limiting the sequence similarity

Table C6. Target protein IDs and the corresponding number of decoys in the model quality assessment test set

Datasets	Target ID	Decoys	Target ID	Decoys	Target ID	Decoys	Target ID	Decoys
CASP14	T1024	512	T1025	208	T1026	517	T1027	527
	T1028	201	T1029	529	T1030	501	T1031	519
	T1033	521	T1035	518	T1036s1	204	T1037	500
	T1039	515	T1040	512	T1041	515	T1042	501
	T1043	514	T1045s1	209	T1045s2	610	T1046s1	534
	T1046s2	530	T1047s1	547	T1047s2	549	T1049	512
	T1051	516	T1053	501	T1055	511	T1056	526
	T1057	522	T1058	509	T1059	529	T1060s2	512
	T1060s3	525	T1064	508	T1065s1	561	T1065s2	553
	T1072s1	544	T1072s2	211	T1074	506	T1076	482
	T1082	501	T1089	503	T1090	517	T1091	476
	T1092	496	T1093	481	T1094	485	T1095	493
	T1096	475	T1099	565				
CASP15	T1104	415	T1106s1	504	T1106s2	509	T1114s1	525
	T1114s2	517	T1114s3	515	T1119	514	T1120	447
	T1121	509	T1123	521	T1124	541	T1129s2	494
	T1133	438	T1134s1	512	T1134s2	517	T1137s1	415
	T1137s2	410	T1137s3	405	T1137s4	419	T1137s5	418
	T1137s6	419	T1137s7	424	T1137s8	426	T1137s9	427
	T1152	545	T1159	433	T1170	501	T1187	540

between target proteins to below 70% and strictly controlling the sequence similarity with the training set to be less than 30%. It's important to note that we exclude two specific targets, H1044 and T1169, from the test set. The partitioning of CASP14 target H1044 into multiple domains (e.g. T1031, T1033) and the excessive length of CASP15 target T1169 pose challenges for baseline methods during the inference process. Therefore, to maintain consistency, we remove these targets and their corresponding decoys from the evaluation.

The details of the target proteins and the corresponding number of decoys in the test sets are presented in Table C6. The CASP14 test set includes 50 target protein IDs and a total of 13 260 decoys, while the CASP15 test set contains 28 target protein IDs and 24 313 decoys. These decoy numbers are identical to those reported by the official CASP participating groups. For reproducibility and further analysis, the original decoys for each target protein are publicly accessible through the following links: https://predictioncenter.org/download_area/CASP14/predictions and https://predictioncenter.org/download_area/CASP15/predictions. These repositories include decoys categorized into subfolders such as "regular," "oligo," and others, corresponding to different prediction types submitted during the CASP.

To calculate GDT-TS and LDDT scores for the decoys, we clip the decoys based on the experimentally resolved native structures, discarding decoys with sequence lengths inconsistent with the native structures. We utilize publicly available tools, which can be downloaded and installed from https://zhanggroup.org/TM-score/ and `conda install -c bioconda lddt`, to calculate the GDT-TS and LDDT scores.

Moreover, we delve deeper into investigating the influence of data composition and scale on model performance through the construction of diverse training datasets. While augmenting the training dataset with a large number of decoys is straightforward and viable, we find that leveraging the inherent diversity of native structures proves more effective during training. As delineated in Tables C7 and C8, we assemble two datasets of equivalent size: one composed of decoys corresponding to 7992 native structures, and the other comprising decoys linked to 270 native structures (with a higher decoy-to-native structure ratio). Despite the comparable scale of these training datasets, models trained with the diversity of native structures exhibit superior generalization capabilities. This underscores the critical importance of accurately representing native structures throughout the learning process. This further underscores the robust representation capabilities of our model, which necessitates only simple fine-tuning on a small-scale dataset to attain optimal performance.

Baselines

We compare method with *seven* recent or established state-of-the-art baselines.

ProQ3 and ProQ3D. We reproduce ProQ3 and ProQ3D using the provided Docker and running scripts (https://bitbucket.org/ElofssonLab/proq3/src/master/). For ProQ3, we utilize `–deep no` and `–quality sscore`, while for ProQ3D, we employ `–deep yes` and `–quality lddt`.

VoroMQA. VoroMQA is a method for assessing the quality of protein structures using Voronoi tessellation and contact area. We download the Linux version of voronota-voromqa from https://gitlab.inria.fr/grudinin/vorocnn to predict scores for each residue in decoys, with all parameters set to default.

Ornate. Ornate employs a residue-wise scoring function based on 3D density maps and a deep 3D CNN to predict both local and global model quality. We utilize version 0.1 of Ornate, available at https://team.inria.fr/nano-d/software/Ornate/, for prediction. Due to its quality measure being CAD score and the output containing only scores for each residue, we solely evaluate its performance in local model quality assessment.

DeepAccNet. DeepAccNet utilizes features such as distance maps and residue properties, which undergo 3D convolution to predict the LDDT score for each residue. Additionally, it iteratively refines the decoy's structure based on error estimation. We utilize the noPyRosetta approach of DeepAccNet for prediction, with both the model and test scripts obtained from the GitHub repository https://github.com/hiranumn/DeepAccNet?tab=readme-ov-file.

Table C7. Comparative analysis of model quality assessment across various training sets, evaluated on the CASP14 testing dataset

DATASETS	GDT-TS				LDDT			
	RMSE ↓	*r*	*ρ*	*τ*	RMSE ↓	*r*	*ρ*	*τ*
Diversity test	0.17	0.70	0.70	0.52	0.12	0.75	0.77	0.58
DeepAccNet w/o pre-training	0.16	0.74	0.75	0.56	0.10	0.77	0.79	0.60
DeepAccNet	0.17	0.72	0.72	0.53	0.11	0.75	0.76	0.57
𝒮able	**0.14**	**0.79**	**0.78**	**0.59**	**0.10**	**0.83**	**0.82**	**0.63**

Table C8. Comparative analysis of model quality assessment across various training sets, evaluated on the CASP15 testing dataset

DATASETS	GDT-TS				LDDT			
	RMSE ↓	*r*	*ρ*	*τ*	RMSE ↓	*r*	*ρ*	*τ*
Diversity test	0.25	0.74	0.74	0.55	0.25	0.60	0.58	0.41
DeepAccNet w/o pre-training	0.20	0.74	0.67	0.48	0.20	0.56	0.56	0.40
DeepAccNet	0.17	0.78	0.75	0.55	0.13	0.72	0.68	0.49
𝒮able	**0.15**	**0.84**	**0.83**	**0.63**	**0.13**	**0.79**	**0.72**	**0.53**

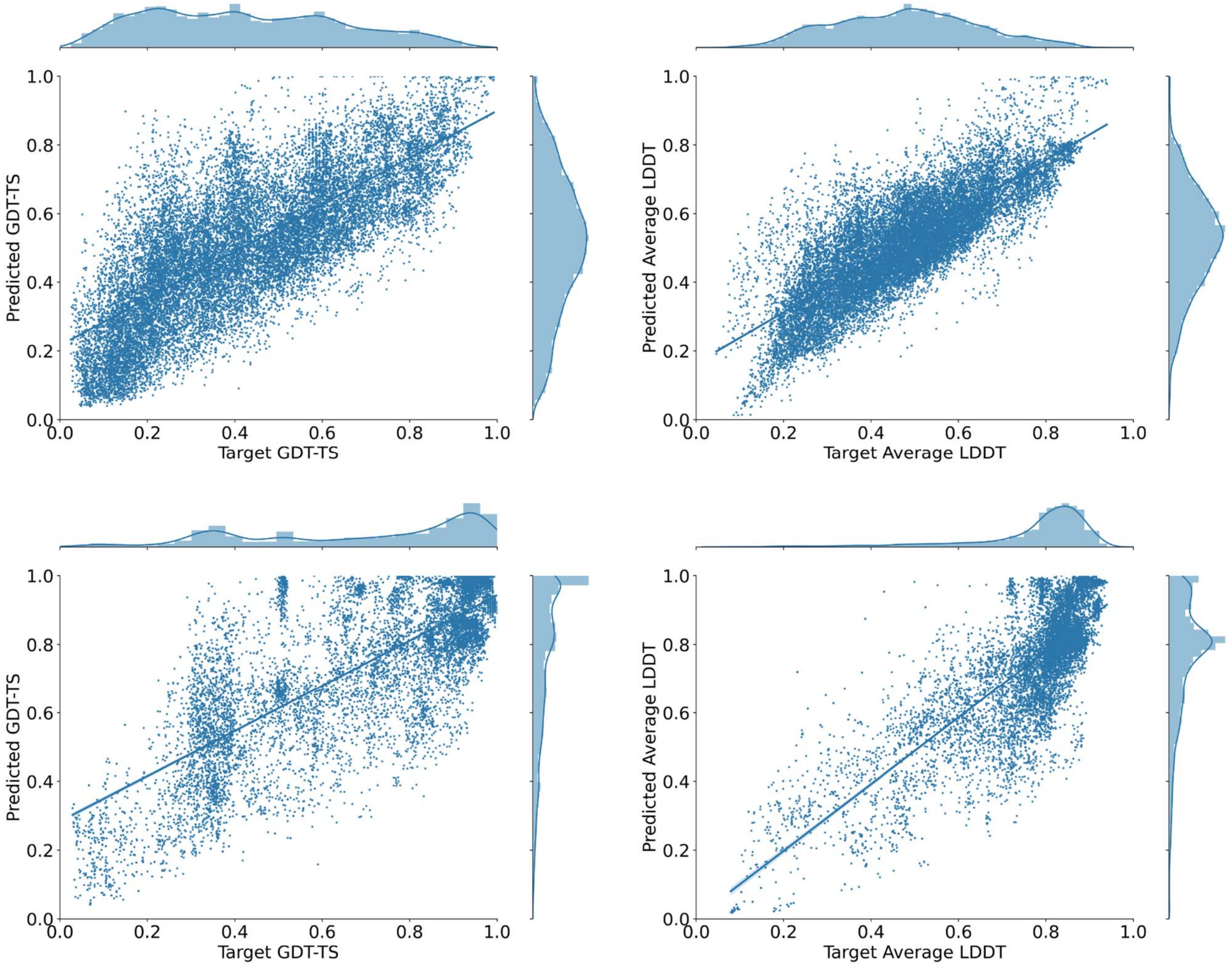

Figure C3. Marginal distribution histogram of CASP14 GDT-TS (upper left), LDDT (upper right), CASP15 GDT-TS (lower left), and LDDT (lower right). In CASP15, there are more high-quality decoys than in CASP14.

Table C9. Hyperparameters setup during fine-tuning

Task	Epoch	Batch size	Learning rate	Dropout	Warm-up	Loss
Model quality assessment	10	64	5e-4	0.2	0.06	MSE
Binding affinity	100	16	3e-4	0.2	0.06	MSE
Fold classification	100	64	5e-4	0.2	0.06	Cross entropy
Enzyme-catalyzed reaction classification	100	64	5e-4	0.2	0.06	Cross entropy
Gene Ontology term prediction - BP	100	64	5e-4	0.2	0.07	Binary cross entropy
Gene Ontology term prediction - MF	100	64	5e-4	0	0.07	Binary cross entropy
Gene Ontology term prediction - CC	100	64	5e-4	0	0.06	Binary cross entropy
Enzyme commission number prediction	100	64	5e-4	0.2	0.05	Binary cross entropy
Protein design	20	64	1e-4	0.2	0.06	Cross entropy
Antibody design	100	8	3e-4	0.2	0.06	Cross entropy & Smooth L1 loss & FAPE

DeepUMQA. DeepUMQA leverages ultrafast shape recognition for efficient feature extraction, followed by feeding these features into a residual neural network to predict LDDT scores. Both the model and code are sourced from the GitHub repository https://github.com/kehan777/DeepUMQA.

AF2Rank. AF2Rank utilizes AlphaFold's learned energy function to rank the quality of protein structure predictions. By leveraging the confidence metrics from AlphaFold's output, AF2Rank provides an assessment of structural accuracy without relying on coevolutionary data. The model and code can be accessed from the GitHub repository https://github.com/jproney/AF2Rank.

QATEN. QATEN integrates a self-attention mechanism, modeling the decoy structure as a graph, enabling simultaneous prediction of LDDT and GDT-TS scores. Test scripts are obtained from the GitHub repository https://github.com/CQ-zhang-2016/QATEN, with all hyperparameters set to default values.

Evaluation metrics

When the native structure is known, several evaluation methods measure the similarity between the decoy and the native structure, providing an assessment of the decoy's quality. We predict the GDT-TS score to evaluate the overall quality of the decoy and the LDDT score to assess the quality of individual residues without relying on the native structure. Additionally, we employ RMSE and three statistical correlation coefficients, Pearson's correlation coefficient r, Spearman's rank correlation coefficient ρ, and Kendall's rank correlation coefficient τ to evaluate the accuracy of the predicted scores.

Downstream task implementation details

During the fine-tuning process, we concurrently optimize the parameters of the pretrained model while adjusting them to fit specific tasks by minimizing the loss function. We employ the commonly used optimizer Adam, combined with a learning rate scheduler for dynamically adjusting the learning rate. All training is conducted on 8 NVIDIA V100 32GB GPUs. Additionally, we summarize the differences in settings among various downstream tasks, as presented in Table C9.

D Appendix 4: Ablation study

We conduct comprehensive ablation experiments to assess the efficacy of each component within the pre-training model. In addition to the delineated ablation studies concerning the protein structure encoder, the characterization of backbone atoms for protein representation, and the influence of pre-training dataset size on model performance, we further delve into the potential impacts of diverse pre-training strategies. Additionally, we provide additional details regarding the construction of pre-training datasets of varying scales.

Data size sensitivity

We process protein sequences from the original pre-training dataset and construct a clustering database containing 714 225 entries (each entry corresponds to a single chain). Utilizing MMseqs2 [102], we perform clustering on this database using different sequence identity thresholds (30% / 40% / 50% / 70% / 90% / 95% / 99%). After clustering, individual chains are assigned to different categories, and the number of categories under each threshold is detailed in the "cluster size" column of Table 8. We generate pre-training datasets of varying sizes based on the proteins associated with each representative chain in the respective category, as indicated in the "data size" column of Table 8. It is worth noting that we exclude categories that containing proteins from both the model quality assessment and the protein design task test set. This precautionary measure is taken to prevent any possibility of data leakage.

Pre-training or not

We conduct an assessment of the effects of pre-training models on large-scale datasets. Acquiring labeled data can be a costly endeavor in many tasks, and often there isn't a sufficient amount of data available to support effective model training. This limitation can hinder the model's ability to generalize effectively. However, by pre-training on extensive datasets, the model can learn more accurate representations, leading to improvements in its performance. As shown in Table C10, the performance drop of the without pre-training models compared to $\mathcal{S}$able is substantial across various tasks, surpassing the performance loss incurred by disabling any individual component. This highlights the significant impact of pre-training data and strategies on model performance. Pre-trained models exhibit enhanced adaptability to diverse tasks, thereby enhancing their generalization capabilities and practicality.

Freeze pre-training parameters or not

During fine-tuning across all downstream tasks, we optimize the parameters of the pre-training model simultaneously. Freezing

Table C10. The results of the ablation study. The first segment pertains to encoder ablation, while the second segment corresponds to pre-training ablation. ✓signifies that the respective component is enabled, × indicates its deactivation, ☑ represents the freezing of pre-trained parameters during fine-tuning. Metrics for the classification task are represented by mean accuracy, whereas for protein design, validation is solely conduct on the CATH v4.2 test set with metrics measured as AAR

	Modifications						**Results**				
	Encoder			**Data level**	**Noise type**	**Pre-training**	**Fold**			**Enzyme**	**CATH**
	SPE	**Distance**	**RPE**				**Fold**	**Sup**	**Family**	**Reaction**	
w/o SPE	×	✓	✓	Backbone atoms	Mix	✓	49.6	80.4	98.0	89.1	45.3
w/o Distance	✓	×	✓	Backbone atoms	Mix	✓	51.9	80.5	98.0	79.4	36.3
w/o RPE	✓	✓	×	Backbone atoms	Mix	✓	44.0	77.9	97.9	79.0	46.1
w/o Structure	×	×	✓	Backbone atoms	Mix	✓	13.9	23.7	83.0	70.7	–
Residue level	×	✓	✓	Residue	Mix	✓	49.4	79.1	98.3	87.6	39.9
All-atom level	✓	✓	✓	All atoms	Mix	✓	44.3	78.8	**98.3**	**89.4**	45.4
w/o pre-training	✓	✓	✓	Backbone atoms	Mix	×	17.7	24.6	84.1	59.2	32.1
w/o fine-tuning parameters	✓	✓	✓	Backbone atoms	Mix	☑	33.7	58.0	96.9	83.3	45.2
Residue type prediction only	✓	✓	✓	Backbone atoms	Mix	✓	49.6	81.0	98.3	88.7	**49.4**
Cα distance restoration only	✓	✓	✓	Backbone atoms	Mix	✓	48.9	79.3	98.1	84.7	46.8
Single noise strategy	✓	✓	✓	Backbone atoms	Single	✓	39.1	73.6	98.1	87.5	38.7
$\mathcal{S}$able	✓	✓	✓	Backbone atoms	Mix	✓	**52.2**	**82.0**	98.2	88.5	46.2

the parameters of the pre-trained model may lead to a certain degree of performance decline. Although we obtain a universal protein representation through pre-training, when fine-tuning for specific downstream tasks, we prefer to transition from the universal representation to task-specific representations.

Pre-training strategy

Based on the $\mathcal{S}$able framework, we investigate the following pre-training strategies:

Residue type prediction only This strategy focuses solely on "masked token prediction" and does not involve the execution of the self-supervised task of "atom distance restoration".

Cα distance restoration only During "atom distance restoration", the emphasis is no longer on restoring the distances between backbone atoms, but rather on reconstructing the distances between Cα atoms.

Single noising strategy We randomly select 15% of the residues to be replaced with mask tokens, and introduce noise following a uniform distribution $\mathcal{U}$(-1, 1) to the coordinates of the residues marked as mask tokens.

The pre-training strategy of "Residue Type Prediction Only" surpasses the existing level of $\mathcal{S}$able in fold classification prediction on the family test set, enzyme-catalyzed reaction classification, and protein design. This suggests that optimizing self-supervised tasks and combining different self-supervised task strategies can offer further improvement opportunities for $\mathcal{S}$able. On the other hand, the pre-training strategies of "Cα Distance Restoration Only" and "Single Noising" exhibit a decrease in performance across all tasks. This highlights the importance of appropriately increasing the difficulty of self-supervised tasks within a reasonable range, as it helps enhance the model's ability to infer correct structural representations from contextual information, thereby demonstrating superior performance.

References

1. Devlin J, Chang M-W, Lee K. *et al.* BERT: pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* 2019;4171–86.
2. Raffel C, Shazeer N, Roberts A. *et al.* Exploring the limits of transfer learning with a unified text-to-text transformer. *J Mach Learn Res* 2020;**21**:1–67.
3. OpenAI. GPT-4 technical report. *arXiv preprint arXiv*:2303.08774, 2023.
4. Rao R, Bhattacharya N, Thomas N. *et al.* Evaluating protein transfer learning with tape. In Wallach H, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox E, Garnett R (Eds.), *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates, Inc.; 2019.
5. Rives A, Meier J, Sercu T. *et al.* Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc Natl Acad Sci* 2021;**118**:e2016239118. https://doi.org/10.1073/pnas.2016239118
6. Chen B, Cheng X, Pan L. *et al.* xTrimoPGLM: unified 100b-scale pre-trained transformer for deciphering the language of protein. *arXiv preprint arXiv*:2401.06199, 2024.
7. Rao RM, Liu J, Verkuil R. *et al.* MSA Transformer. In: *International Conference on Machine Learning*, pp. 8844–56. USA: PMLR, 2021.
8. Elnaggar A, Heinzinger M, Dallago C. *et al.* ProtTrans: toward understanding the language of life through self-supervised learning. *IEEE Trans Pattern Anal Mach Intell* 2021;**44**:7112–27.
9. Lin Z, Akin H, Rao R. *et al.* Evolutionary-scale prediction of atomiclevel protein structure with a language model. *Science* 2023;**379**:1123–30.
10. Elnaggar A, Essam H, Salah-Eldin W. *et al.* Ankh: optimized protein language model unlocks general-purpose modelling. *arXiv preprint arXiv*:2301.06568, 2023.
11. Ruffolo JA, Madani A. Designing proteins with language models. *Nat Biotechnol* 2024;**42**:200–2. https://doi.org/10.1038/s41587-024-02123-4
12. Hilger D, Masureel M, Kobilka BK. Structure and dynamics of GPCR signaling complexes. *Nat Struct Mol Biol* 2018;**25**:4–12. https://doi.org/10.1038/s41594-017-0011-7
13. Wang S, Sun S, Li Z. *et al.* Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLoS*

Comput Biol 2017;**13**:e1005324. https://doi.org/10.1371/journal.pcbi.1005324

14. Zhou G, Gao Z, Ding Q. *et al.* Uni-Mol: a universal 3D molecular representation learning framework. In: *The Eleventh International Conference on Learning Representations*, 2023.
15. Gainza P, Sverrisson F, Monti F. *et al.* Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nat Methods* 2020;**17**:184–92. https://doi.org/10.1038/s41592-019-0666-6
16. Sverrisson F, Feydy J, Correia BE. *et al.* Fast end-to-end learning on protein surfaces. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15272–81. Piscataway, NJ: IEEE, 2021.
17. Amidi A, Amidi S, Vlachakis D. *et al.* EnzyNet: enzyme classification using 3D convolutional neural networks on spatial representation. *PeerJ* 2018;**6**:e4750. https://doi.org/10.7717/peerj.4750
18. Derevyanko G, Grudinin S, Bengio Y. *et al.* Deep convolutional networks for quality assessment of protein folds. *Bioinformatics* 2018;**34**:4046–53. https://doi.org/10.1093/bioinformatics/bty494
19. Vladimir Gligorijević P, Renfrew D, Kosciolek T. *et al.* Structure-based protein function prediction using graph convolutional networks. *Nat Commun* 2021;**12**:3168. https://doi.org/10.1038/s41467-021-23303-9
20. Baldassarre F, Hurtado DM, Elofsson A. *et al.* GraphQA: protein model quality assessment using graph convolutional networks. *Bioinformatics* 2021;**37**:360–6. https://doi.org/10.1093/bioinformatics/btaa714
21. Zhang P, Xia C, Shen H-B. High-accuracy protein model quality assessment using attention graph neural networks. *Brief Bioinform* 2023;**24**:bbac614. https://doi.org/10.1093/bib/bbac614
22. Jin W, Wohlwend J, Barzilay R. *et al.* Iterative refinement graph neural network for antibody sequence-structure co-design. In: *International Conference on Learning Representations*, 2022.
23. Jing B, Eismann S, Suriana P. *et al.* Learning from protein structure with geometric vector perceptrons. In: *International Conference on Learning Representations*, 2021.
24. Wang Z, Combs SA, Brand R. *et al.* LM-GVP: an extensible sequence and structure informed deep learning framework for protein property prediction. *Sci Rep* 2022;**12**:6832.
25. Li J, Luo S, Deng C. *et al.* Orientation-aware graph neural networks for protein structure representation learning. *arXiv preprint arXiv:2201.13299*, 2025.
26. Zhang Z, Xu M, Jamasb AR. *et al.* Protein representation learning by geometric structure pretraining. In: *The Eleventh International Conference on Learning Representations*, 2023.
27. Ingraham J, Garg VK, Barzilay R. *et al.* Generative models for graph-based protein design. In Wallach H, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox E, Garnett R (Eds.), *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates, Inc.; 2019.
28. Hsu C, Verkuil R, Liu J. *et al.* Learning inverse folding from millions of predicted structures. In: *International Conference on Machine Learning*, pp. 8946–70. USA: PMLR, 2022.
29. Dauparas J, Anishchenko IV, Bennett NR. *et al.* Robust deep learning based protein sequence design using ProteinMPNN. *Science (New York, NY)* 2022;**378**:49–56. https://doi.org/10.1126/science.add2187
30. Gao Z, Cheng T, Li SZ. PiFold: toward effective and efficient protein inverse folding. In: *The Eleventh International Conference on Learning Representations*, 2023.
31. Hermosilla P, Schfer M, Lang M. *et al.* Intrinsic-extrinsic convolution and pooling for learning on 3D protein structures. In: *International Conference on Learning Representations*, 2021.
32. Fan H, Wang Z, Yang Y. *et al.* Continuous-discrete convolution for geometry-sequence modeling in proteins. In: *The Eleventh International Conference on Learning Representations*, 2022.
33. Alon U, Yahav E. On the bottleneck of graph neural networks and its practical implications. In: *International Conference on Learning Representations*, 2021.
34. Topping J, Di Giovanni F, Chamberlain BP. *et al.* Understanding over-squashing and bottlenecks on graphs via curvature. In: *International Conference on Learning Representations*, 2022.
35. Nguyen VTD, Hy TS. Multimodal pretraining for unsupervised protein representation learning. *Biol Methods Protoc* 2024;**9**:bpae043. https://doi.org/10.1093/biomethods/bpae043
36. Jumper JM, Evans R, Pritzel A. *et al.* Highly accurate protein structure prediction with alphafold. *Nature* 2021;**596**:583–9. https://doi.org/10.1038/s41586-021-03819-2
37. Saka K, Kakuzaki T, Metsugi S. *et al.* Antibody design using LSTM based deep generative model from phage display library for affinity maturation. *Sci Rep* 2021;**11**:5852. https://doi.org/10.1038/s41598-021-85274-7
38. Akbar R, Robert PA, Weber CR. *et al.* In silico proof of principle of machine learning-based antibody design at unconstrained scale. *MAbs* 2022;**14**:2031482. https://doi.org/10.1080/19420862.2022.2031482
39. Jin W, Barzilay R, Jaakkola T. Multi-objective molecule generation using interpretable substructures. In: *International Conference on Machine Learning*, pp. 4849–59. USA: PMLR, 2020.
40. Gao K-X, Lijun W, Zhu J. *et al.* Incorporating pre-training paradigm for antibody sequence-structure co-design. *arXiv preprint arXiv:2211.08406*, 2022.
41. Kong X, Huang W, Yang L. Conditional antibody design as 3D equivariant graph translation. In: *The Eleventh International Conference on Learning Representations*, 2023.
42. Verma Y, Heinonen M, Garg V. Abode: ab initio antibody design using conjoined odes. In: *International Conference on Machine Learning*, pp. 35037–50. USA: PMLR, 2023.
43. Cheng T, Gao Z, Wu L. *et al.* Cross-gate MLP with protein complex invariant embedding is a one-shot antibody designer. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. **38**, pp. 15222–30. Washington, DC, USA: AAAI Press, 2024.
44. Adolf-Bryfogle J, Kalyuzhniy O, Kubitz M. *et al.* RosettaAntibodyDesign (RAbD): a general framework for computational antibody design. *PLoS Comput Biol* 2018;**14**:e1006112. https://doi.org/10.1371/journal.pcbi.1006112
45. Jin W, Barzilay R, Jaakkola T. Antibody-antigen docking and design via hierarchical structure refinement. In: *International Conference on Machine Learning*, pp. 10217–27. USA: PMLR, 2022.
46. Kong X, Huang W, Liu Y. End-to-end full-atom antibody design. In: *International Conference on Machine Learning*, pp. 17409–29. USA: PMLR, 2023.
47. Wu Z, Jain P, Wright M. *et al.* Representing long-range context for graph neural networks with global attention. *Adv Neural Inf Process Syst* 2021;**34**:13266–79.
48. Zhang X, Yin H, Ling F. *et al.* SPIN-CGNN: improved fixed backbone protein design with contact map-based graph construction and contact graph neural network. *PLoS Comput Biol* 2023;**19**:e1011330. https://doi.org/10.1371/journal.pcbi.1011330
49. Zheng Z, Deng Y, Xue D. *et al.* Structure-informed language models are protein designers. In: *International Conference on Machine Learning*, pp. 42317–38. USA: PMLR, 2023.

50. Wang H, Liu D, Zhao K. *et al.* SPDesign: protein sequence designer based on structural sequence profile using ultrafast shape recognition. *Brief Bioinform* 2024;**25**:bbae146.
51. Aykent S, Xia T. GBPNet: universal geometric representation learning on protein structures. In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4–14. New York, NY, USA: Association for Computing Machinery, 2022.
52. Schymkowitz J, Ferkinghoff-Borg J, Stricher F. *et al.* The FoldX web server: an online force field. *Nucleic Acids Res* 2005;**33**:W382–8. https://doi.org/10.1093/nar/gki387
53. Zhang N, Chen Y, Lu H. *et al.* MutaBind2: predicting the impacts of single and multiple mutations on protein-protein interactions. *iScience* 2020;**23**:100939. https://doi.org/10.1016/j.isci.2020.100939
54. Wang M, Cang Z, Wei G. A topology-based network tree for the prediction of protein–protein binding affinity changes following mutation. *Nat Mach Intell* 2020;**2**:116–23. https://doi.org/10.1038/s42256-020-0149-6
55. Liu X, Luo Y, Song S. *et al.* Deep geometric representations for modeling effects of mutations on protein-protein binding affinity. *PLoS Comput Biol* 2020;**17**:1–28. https://doi.org/10.1371/journal.pcbi.1009284
56. Shan S, Luo S, Yang Z. *et al.* Deep learning guided optimization of human antibody against SARS-CoV-2 variants with broad neutralization. *Proc Natl Acad Sci* 2022;**119**:e2122954119. https://doi.org/10.1073/pnas.2122954119
57. Uziela K, Shu N, Wallner B. *et al.* ProQ3: improved model quality assessments using Rosetta energy terms. *Sci Rep* 2016;**6**:33509. https://doi.org/10.1038/srep33509
58. Uziela K, Hurtado DM, Shu N. *et al.* ProQ3D: improved model quality assessments using deep learning. *Bioinformatics* 2017; **33**:1578–80. https://doi.org/10.1093/bioinformatics/btw819
59. Olechnovič K, Venclovas Č. VoroMQA web server for assessing three-dimensional structures of proteins and protein complexes. *Nucleic Acids Res* 2019;**47**:W437–42. https://doi.org/10.1093/nar/gkz367
60. Pagès G, Charmettant B, Grudinin S. Protein model quality assessment using 3D oriented convolutional neural networks. *Bioinformatics* 2019;**35**:3313–9. https://doi.org/10.1093/bioinformatics/btz122
61. Hiranuma N, Park H, Baek M. *et al.* Improved protein structure refinement guided by deep learning based accuracy estimation. *Nat Commun* 2021;**12**:1340. https://doi.org/10.1038/s41467-021-21511-x
62. Guo S-S, Liu J, Zhou X-G. *et al.* DeepUMQA: ultrafast shape recognition-based protein model quality assessment using deep learning. *Bioinformatics* 2022;**38**:1895–903. https://doi.org/10.1093/bioinformatics/btac056
63. Roney JP, Ovchinnikov S. State-of-the-art estimation of protein model accuracy using alphafold. *Phys Rev Lett* 2022;**129**:238101. https://doi.org/10.1103/PhysRevLett.129.238101
64. Mehboob MZ, Lang M. Structure, function, and pathology of protein O-glucosyltransferases. *Cell Death Dis* 2021;**12**:71. https://doi.org/10.1038/s41419-020-03314-y
65. Hamley IW. The amyloid beta peptide: a chemist's perspective. Role in Alzheimer's and fibrillization. *Chem Rev* 2012;**112**: 5147–92. https://doi.org/10.1021/cr3000994
66. Che T, English J, Krumm BE. *et al.* Nanobody-enabled monitoring of kappa opioid receptor states. *Nat Commun* 2020;**11**:1145. https://doi.org/10.1038/s41467-020-14889-7
67. He Q-T, Xiao P, Huang S-M. *et al.* Structural studies of phosphorylation-dependent interactions between the V2R receptor and arrestin-2. *Nat Commun* 2021;**12**:2396. https://doi.org/10.1038/s41467-021-22731-x
68. Bepler T, Berger B. Learning protein sequence embeddings using information from structure. In: *International Conference on Learning Representations*, 2019.
69. Alley EC, Khimulya G, Biswas S. *et al.* Unified rational protein engineering with sequence-based deep representation learning. *Nat Methods* 2019;**16**:1315–22. https://doi.org/10.1038/s41592-019-0598-1
70. Madani A, Krause B, Greene ER. *et al.* Large language models generate functional protein sequences across diverse families. *Nat Biotechnol* 2023;**41**:1099–106. https://doi.org/10.1038/s41587-022-01618-2
71. Olechnovič K, Venclovas Č. VoroMQA: assessment of protein structure quality using interatomic contact areas. *Proteins* 2017;**85**:1131–45. https://doi.org/10.1002/prot.25278
72. Sael L, Li B, La D. *et al.* Fast protein tertiary structure retrieval based on global surface shape similarity. *Proteins* 2008;**72**: 1259–73. https://doi.org/10.1002/prot.22030
73. Venkatraman V, Yang YD, Sael L. *et al.* Protein-protein docking using region-based 3D Zernike descriptors. *BMC Bioinformatics* 2009;**10**:1–21. https://doi.org/10.1186/1471-2105-10-407
74. Daberdaku S, Ferrari C. Exploring the potential of 3D Zernike descriptors and SVM for protein–protein interface prediction. *BMC Bioinformatics* 2018;**19**:1–23. https://doi.org/10.1186/s12859-018-2043-3
75. Zheng J, Wang G, Huang Y. *et al.* Lightweight contrastive protein structure-sequence transformation. *arXiv preprint arXiv:2303.11783*, 2023.
76. Hermosilla P, Ropinski T. Contrastive representation learning for 3D protein structures. *ArXiv, abs/2205.15675*, 2022.
77. Dunbar J, Krawczyk K, Leem J. *et al.* SAbDab: the structural antibody database. *Nucleic Acids Res* 2014;**42**:D1140–6. https://doi.org/10.1093/nar/gkt1043
78. Li Z, Yang Y, Faraggi E. *et al.* Direct prediction of profiles of sequences compatible with a protein structure by neural networks with fragment-based local and energy-based nonlocal profiles. *Proteins* 2014;**82**:2565–73. https://doi.org/10.1002/prot.24620
79. Qi Y, Zhang JZH. DenseCPD: improving the accuracy of neural-network-based computational protein sequence design with DenseNet. *J Chem Inf Model* 2020;**60**:1245–52. https://doi.org/10.1021/acs.jcim.0c00043
80. Murzin AG, Brenner SE, Hubbard T. *et al.* SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol* 1995;**247**:536–40. https://doi.org/10.1016/S0022-2836(05)80134-2
81. Jie H, Badri A, Jianlin C. DeepSF: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics* 2017;**34**:1295–303. https://doi.org/10.1093/bioinformatics/btx780
82. Jose M, Dana A, Gutmanas N. *et al.* SIFTS: updated structure integration with function, taxonomy and sequences resource allows 40-fold increase in coverage of structure-based annotations for proteins. *Nucleic Acids Res* 2018;**47**:D482–9. https://doi.org/10.1093/nar/gky1114
83. Zhang Y, Skolnick J. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res* 2005;**33**: 2302–9. https://doi.org/10.1093/nar/gki524
84. Radivojac P, Clark WT, Oron TR. *et al.* A large-scale evaluation of computational protein function prediction. *Nat Methods* 2013;**10**:221–7. https://doi.org/10.1038/nmeth.2340

85. Steinegger M, Meier M, Mirdita M. *et al.* HH-suite3 for fast remote homology detection and deep protein annotation. *BMC Bioinformatics* 2019;**20**:1–15. https://doi.org/10.1186/s12859-019-3019-7
86. Madeira F, Pearce M, Tivey ARN. *et al.* Search and sequence analysis tools services from EMBL-EBI in 2022. *Nucleic Acids Res* 2022;**50**:W276–9. https://doi.org/10.1093/nar/gkac240
87. Shanehsazzadeh A, Belanger D, Dohan D. *Is transfer learning necessary for protein landscape prediction? arXiv preprint arXiv:2011.03443*, 2020.
88. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. In: *International Conference on Learning Representations*, 2017.
89. Veličković P, Cucurull G, Casanova A. *et al.* Graph attention networks. In: *International Conference on Learning Representations*, 2018.
90. Diehl F. Edge contraction pooling for graph neural networks. *arXiv preprint arXiv: abs/1905.10990*, 2019.
91. Das S, Lee D, Sillitoe I. *et al.* Functional classification of CATH superfamilies: a domain-based approach for protein function annotation. *Bioinformatics* 2015;**31**:3460–7. https://doi.org/10.1093/bioinformatics/btv398
92. Kulmanov M, Khan MA, Hoehndorf R. *et al.* DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics* 2018;**34**:660–8. https://doi.org/10.1093/bioinformatics/btx624
93. Cozzetto D, Minneci F, Currant H. *et al.* FFPred 3: feature-based function prediction for all gene ontology domains. *Sci Rep* 2016;**6**:31865.
94. Sirin S, Apgar JR, Bennett EM. *et al.* Ab-bind: antibody binding mutational database for computational affinity predictions. *Protein Sci* 2016;**25**:393–409. https://doi.org/10.1002/pro.2829
95. Xiong P, Zhang C, Zheng W. *et al.* BindProfX: assessing mutation-induced binding affinity change by protein interface profiles with pseudo-counts. *J Mol Biol* 2017;**429**:426–34. Computation Resources for Molecular Biology.
96. Rodrigues CHM, Yoochan M, Pires DEV. *et al.* mCSM-PPI2: predicting the effects of mutations on protein-protein interactions. *Nuclc Acids Res* 2019;**47**:w338–44.
97. Moal IH, Fernández-Recio J. SKEMPI: a structural kinetic and energetic database of mutant protein interactions and its use in empirical models. *Bioinformatics* 2012;**28**:2600–7. https://doi.org/10.1093/bioinformatics/bts489
98. Jankauskaite J, Jiménez-García B, Dapkūnas *J. et al.* SKEMPI 2.0: an updated benchmark of changes in protein–protein binding energy, kinetics and thermodynamics upon mutation. *Bioinformatics* 2018;**35**:462–9.
99. Wang G, Dunbrack Jr RL. PISCES: a protein sequence culling server. *Bioinformatics* 2003;**19**:1589–91. https://doi.org/10.1093/bioinformatics/btg224
100. Song Y, DiMaio F, Wang RY. *et al.* High-resolution comparative modeling with RosettaCM. *Structure* 2013;**21**:1735–42. https://doi.org/10.1016/j.str.2013.08.005
101. Yang J, Anishchenko I, Park H. *et al.* Improved protein structure prediction using predicted interresidue orientations. *Proc Natl Acad Sci* 2020;**117**:1496–503.
102. Steinegger M, Söding J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat Biotechnol* 2017;**35**:1026–8.

Briefings in Bioinformatics, 2025, **26(2)**, bbaf120
https://doi.org/10.1093/bib/bbaf120
Problem Solving Protocol