Multimodal Distillation of Protein Sequence, Structure, and Function

Anonymous Authors[1]

Abstract

Proteins serve as the foundational building blocks of life, orchestrating vital biological functions. The acquisition of effective protein representations holds paramount importance for various applications. While language models (LMs) and graph neural networks (GNNs) exhibit promising performance in protein modeling, the presence of multiple data modalities, such as sequence, structure, and functional annotations, poses a challenge. Frameworks that seamlessly integrate these diverse sources without relying on extensive pre-training are currently limited. In response, we introduce ProteinSSA, a novel multimodal knowledge distillation framework that integrates **Protein** **S**equence, **S**tructure, and Gene Ontology (GO) **A**nnotation for the purpose of learning unified representations. Our approach employs a teacher-student model architecture connected through knowledge distillation. The student GNN is designed to encode protein sequences and structures, while the teacher model utilizes both GNN and an auxiliary GO encoder to incorporate extra functional knowledge that results in the generation of hybrid multimodal embeddings, which are then passed to the student for learning function-enriched representations through distribution approximation. Experiments on tasks like protein function and enzyme commission (EC) number prediction show that ProteinSSA significantly outperforms state-of-the-art baselines, demonstrating the benefits of our multimodal framework.

1. Introduction

Proteins are essential molecules that serve as the basic structural and functional components of cells and organisms. A natural protein consists of a linear sequence of amino acids that are linked together by peptide bonds, which fold into a three-dimensional (3D) structure. Recent advances like AlphaFold2 (Jumper et al., 2021) have enabled highly accurate protein structure prediction, facilitating the application of artificial intelligence techniques for proteins. There exists a major scientific challenge to figure out the relationships between a protein's sequence, structure, and function, while this knowledge is crucial for elucidating disease mechanisms (Serçinoğlu & Ozbek, 2020). Protein representation learning is an active research area that aims to learn underlying patterns from raw protein data, which are useful in different downstream tasks (Unsal et al., 2022).

Recently, protein LMs have been developed to process protein sequences and have demonstrated an ability to learn the certain 'grammar of life' from large numbers of protein sequences (Lin et al., 2022). Models like ProtTrans (Elnaggar et al., 2021), ESM series (Rives et al., 2019; Rao et al., 2021; 2020; Lin et al., 2022), and xTrimoPGLM (Chen et al., 2024) leverage transformers, and attention mechanisms to learn intrinsic patterns in a self-supervised manner, pre-training on large-scale data. Unlike sequences, protein structures exhibit continuous 3D coordinate data (Fan et al., 2023), requiring different modeling approaches. To represent protein 3D structures, GNN-based models have been designed and adapted (Baldassarre et al., 2021; Hermosilla & Ropinski, 2022). For example, GearNet (Zhang et al., 2023b) encodes the sequential and spatial features of proteins by passing messages between nodes and edges in an alternating pattern on multiple types of protein graphs.

Though protein LMs and GNNs have achieved remarkable performance in various protein-related applications, such as predicting protein mutational stability and EC numbers (Hu et al., 2023b), while most of these methods ignore functional information (Elnaggar et al., 2021; Lin et al., 2022). Incorporating functional annotations is also important for enhancing model capabilities and uncovering the intrinsic relationships between protein sequences and functions (Zhou et al., 2023; Hu et al., 2023b). Recent works explore token-level protein knowledge by pre-training on biomedical texts, containing sequential and functional information (Zhou et al., 2023; Xu et al., 2023). However, protein sequences vastly outnumber available structures and annotations (Ashburner et al., 2000). For example, there are over 500M (million) sequences in UniParc (Consor-

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

tium, 2013) versus about 190K (thousand) structures in the Protein Data Bank (PDB) (Berman et al., 2000b) versus approximately 5M triplets in ProteinKG25 (Zhang et al., 2022), including about 600K proteins and 50K attribute terms. This scale difference makes it difficult to bring the same success of sequence pre-training into structure and function pre-training.

Considering the difference in data categories and sizes of protein sequences, structures, and functions, we propose ProteinSSA, a multimodal framework for comprehensive protein representation learning. This is a teacher-student framework based on domain-adapted knowledge distillation without relying on the large-scale pre-training mechanism. ProteinSSA utilizes the teacher model to learn from the triplets of sequences, structures, and functions, distilling this knowledge to guide the training of the student model. The teacher model requires extra functions as input for protein representation learning, however, not even 1% of sequenced proteins have functional annotations at present (Torres et al., 2021; Ibtehaz et al., 2023), such information is not always available in various downstream tasks for the teacher model. Therefore, we tend to train a sequence-structure student model that can be applied in downstream tasks conveniently, and the teacher model is only used to provide functional knowledge for the student model. To transfer teacher knowledge, we employ domain adaptation techniques to align the latent space distributions between the teacher model and the student model. Specifically, we calculate the Kullback-Leibler (KL) divergence to minimize the distribution discrepancy between the teacher domain and the student domain.

The contributions of this paper are summarized as (1) We propose ProteinSSA to incorporate multiple categories of protein data, including sequence, structure, and functional annotations. The teacher-student framework allows learning unified representations without large-scale pre-training for applicability to various downstream tasks. (2) We are the first to adapt the knowledge distillation method to connect the protein teacher-student network, injecting the functional information into the student representations via distribution approximation and domain adaptation. (3) We validate ProteinSSA by surpassing currently prevalent protein representation methods on extensive tasks, including predicting protein fold, enzyme reactions, GO terms, and EC numbers.

2. Related Work

2.1. Representation Learning for Protein

Self-supervised pre-training methods have been proposed to learn representations directly from amino acid sequences (Rao et al., 2019), with significant efforts to increase model sizes or dataset scales (Rao et al., 2020; Elnaggar et al., 2021; Nijkamp et al., 2022; Ferruz et al., 2022; Rao et al., 2019; Heinzinger et al., 2022). In protein multimodality learning, to leverage tertiary structures, GearNet (Zhang et al., 2023b) represent sequential and geometric features as the graph node and edge features, using the message passing mechanism to encode them (Hermosilla et al., 2021; Jing et al., 2020a). Considering SE(3)-equivariant properties in protein structures, equivariant and invariant features are designed as model inputs (Jing et al., 2020a; Guo et al., 2022). Moreover, CDConv (Fan et al., 2023) proposes a continuous-discrete convolution to model the sequential and geometric features. ProNet (Wang et al., 2023) provides complete geometric representations at multiple structure levels of granularity.

Moreover, factual biological knowledge has been shown to improve pre-trained LMs on protein sequences (Zhang et al., 2022). ProteinBERT (Brandes et al., 2022) are pre-trained on proteins and frequent GO annotations from UniRef90 (Boutet et al., 2016). KeAP (Zhou et al., 2023) and ProtST (Xu et al., 2023) train biomedical LMs using masked language modeling (Devlin et al., 2018) as the pretext task. Notably, MASSA (Hu et al., 2023b) first obtains sequence-structure embeddings from existing pre-trained models (Rao et al., 2020; Jing et al., 2020a), then globally aligns them with GO embeddings using five pre-training objectives. Brief comparisons of these models are shown in Table 1. The learned student model is deemed to contain information from protein sequences, structure, and functions.

Table 1. Comparisons of existing protein representation learning methods. Seq: sequence; Struct: structure; Func: function.

Method	Information	Pre-training
GearNet	Seq, Struct	✓
KeAP	Seq, Func	✓
MASSA	Seq, Struct, Func	✓
ProteinSSA (Student)	Seq, Struct, Func	✗

2.2. Knowledge Distillation

Knowledge distillation involves the transfer of knowledge from a larger teacher model to a smaller student model (Hinton et al., 2015). There has been considerable progress in graph-based knowledge distillation, with many proposed methods (Liu et al., 2023; Tian et al., 2022). RDD (Zhang et al., 2020) mandates the student model to faithfully replicate the complete node embeddings of the teacher, ensuring the transfer of more informative knowledge. Another noteworthy approach, GraphAKD (He et al., 2022a), employs adversarial learning to distill node representations from the teacher to the student. This method effectively distills knowledge from both local and global perspectives, demonstrating superior performance compared to earlier graph distillation methods (He et al., 2022b).

2.3. Domain Adaptation

Domain adaptation generally seeks to learn a model from source-labeled data that can be generalized to a target domain by minimizing differences between domain distributions (Farahani et al., 2021; Wilson & Cook, 2020; Wang & Deng, 2018). Distribution alignment methods minimize marginal and conditional representation distributions between source and target (Nguyen et al., 2022; Long et al., 2015). Adversarial learning approaches have shown impressive performance in reducing divergence between source and target domains (Ganin & Lempitsky, 2015; Long et al., 2018; Pei et al., 2018). Semi-supervised domain adaptation reduces source-target discrepancy given limited labeled target data (Saito et al., 2019; Kim & Kim, 2020; Jiang et al., 2020; Qin et al., 2021). We adapt domain adaptation methods to align the distributions of representations from teacher and student networks trained on different protein tasks.

3. Method

3.1. Preliminaries

In this subsection, we provide the problem definitions and relevant notations. The background knowledge of the local coordinate system is also introduced, which is closely associated with the protein graph edge features.

Problem Statement We represent a protein graph as $G = (\mathcal{V}, \mathcal{E}, X, E)$, where $\mathcal{V} = \{v_i\}_{i=1,\dots,n}$ and $\mathcal{E} = \{\varepsilon_{ij}\}_{i,j=1,\dots,n}$ denote the vertex and edge sets with n residues, respectively. We use the coordinate of C_α to represent the position of a residue, and the position matrix is denoted as $\mathcal{P} = \{P_i\}_{i=1,\dots,n}$, where $P_i \in \mathbb{R}^{3\times1}$. The node and edge feature matrices are $X = [\boldsymbol{x}_i]_{i=1,\dots,n}$ and $E = [\boldsymbol{e}_{ij}]_{i,j=1,\dots,n}$, the feature vectors of node and edge are $\boldsymbol{x}_i \in \mathbb{R}^{d_1}$ and $\boldsymbol{e}_{ij} \in \mathbb{R}^{d_2}$, d_1 and d_2 are the initial feature dimensions. The GO annotations are denoted as $A = \{A_i\}_{i=1,\dots,k}$ with k terms in total for proteins, where $A_i \in \{0, 1\}$ is the indicator for annotation i. The goal of protein graph representation learning is to form a set of low-dimensional embeddings z for each protein.

There is a source domain S for the teacher model with the data distribution $p_S(z_S|G_S, A)$ in the latent space, and there is also a target domain T for the student model with the data distribution $p_T(z_T|G_T)$ in the latent space. z_S, z_T are latent embeddings from the teacher and student networks for protein graphs G_S and G_T.

Local Coordinate System In order to avoid the usage of complicated SE(3)-equivariant models, the invariant and locally informative features are developed from the local coordinate system (Ingraham et al., 2019), shown in Figure 4 in appendix, which is defined as:

$$\boldsymbol{O}_i = [\boldsymbol{b_i} \quad \boldsymbol{n_i} \quad \boldsymbol{b_i} \times \boldsymbol{n_i}] \tag{1}$$

where $\boldsymbol{u}_i = \frac{P_i - P_{i-1}}{\|P_i - P_{i-1}\|}, \boldsymbol{b_i} = \frac{\boldsymbol{u}_i - \boldsymbol{u}_{i+1}}{\|\boldsymbol{u}_i - \boldsymbol{u}_{i+1}\|}, \boldsymbol{n}_i = \frac{\boldsymbol{u}_i \times \boldsymbol{u}_{i+1}}{\|\boldsymbol{u}_i \times \boldsymbol{u}_{i+1}\|}$.

$$\boldsymbol{e}_{ij} = \mathrm{Concat}(\|P_i - P_j\|, \boldsymbol{O}_i^T \cdot \frac{P_i - P_j}{\|P_i - P_j\|}, \boldsymbol{O}_i^T \cdot \boldsymbol{O}_j) \tag{2}$$

Note that the edge feature vector $\boldsymbol{e}_{ij}$ is the concatenation of the geometric features for protein 3D structures, including distance, direction, and orientation, where $\|\cdot\|$ denotes the l^2-norm.

3.2. A Preliminary Exploration

For large-scale pre-training, it is unclear whether one or a few self-supervision tasks are sufficient for learning effective representations and which task would be beneficial (Hu et al., 2023b). Thus, the performance of pre-trained models is limited by model size, dataset scale, and choice of pre-training tasks. We conducted a preliminary experiment to illustrate this. CDConv (Fan et al., 2023) designs an effective fundamental operation to encapsulate the protein structure without any pre-training or self-supervised learning, achieving comparable accuracy to pre-training methods. It is currently an effective, publicly available method for modeling protein sequence and structure.

In the field of protein pre-training, we select the current state-of-the-art knowledge-enhanced model, KeAP (Zhou et al., 2023), to generate universal sequence-function embeddings, which are used to enhance the CDConv model. ESM-2 (650M) (Lin et al., 2022) is one of the most prevalent sequence pre-training models and is chosen to output sequence embeddings as a comparison with KeAP. By incorporating the embeddings from KeAP and ESM-2 to enhance the embeddings obtained from CDConv, we can compare the quality and performance of the embeddings from these two pre-trained models. The averaged results are shown in Table 2. More details about these experimental settings are provided in Appendix B.1.

As shown in Table 2, the sequence embeddings from ESM-2 provide better enhancement compared to the sequence-function embeddings from KeAP when used with CDConv. This observation demonstrates the limitations of the current sequence-function pre-trained model. To overcome these limitations while better utilizing functional information, we propose a multimodal knowledge distillation framework.

3.3. Overall Framework

The overall framework of ProteinSSA is illustrated in Figure 1. It consists of two branches that train a teacher model and a student model via iterative knowledge distillation.

Table 2. $\mathrm{F}_{\max}$ of GO term and EC number prediction. The base model, CDConv (Fan et al., 2023), is enhanced by sequence and sequence-function embeddings from ESM-2 (Lin et al., 2022) and KeAP (Zhou et al., 2023), respectively.

Algorithm	GO-BP	GO-MF	GO-CC	EC
CDConv	0.453	0.654	0.479	0.820
Enhanced by the sequence embeddings (CDConv + ESM-2)	0.471	0.665	0.538	0.862
Enhanced by the sequence-function embeddings (CDConv + KeAP)	0.467	0.671	0.529	0.842

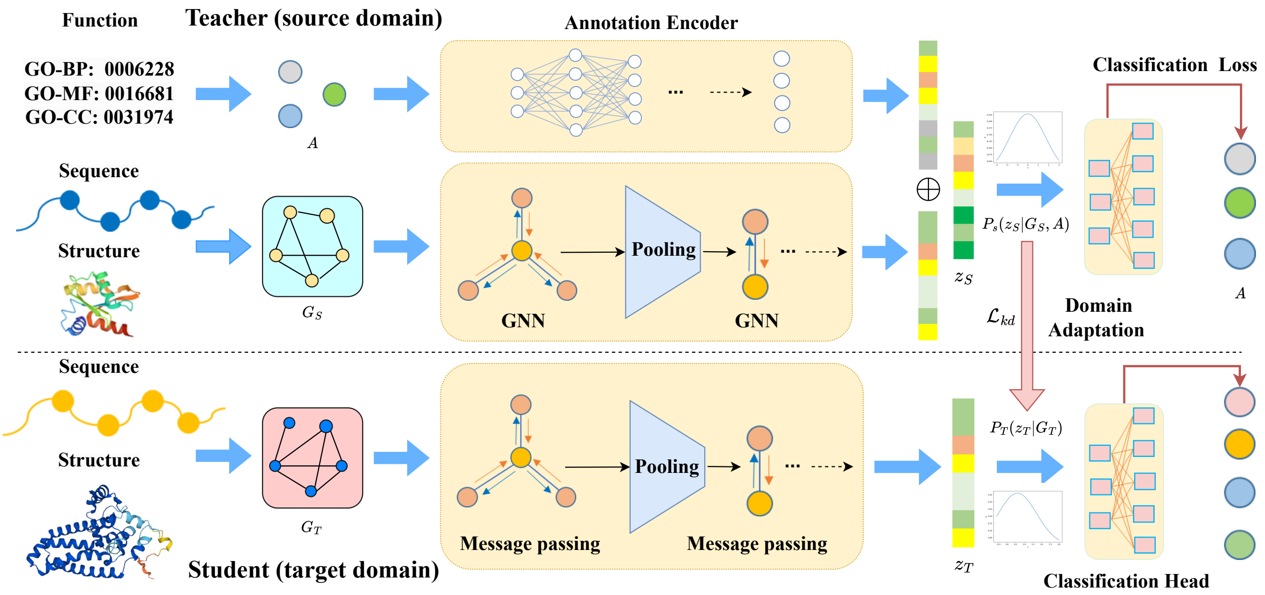


Figure 1. The overall framework of ProteinSSA consists of two branches: a teacher model in the source domain and a student model in the target domain, connected by domain-adapted knowledge distillation.

Compared to the student, the teacher has an additional annotation encoder module comprised of several fully connected layers. This transforms GO annotations into functional embeddings, combined with sequence-structure embeddings from the GNNs to form the final knowledge-enhanced embeddings z_S. Previous works have successfully utilized label-augmented techniques to enhance model training (Bengio et al., 2010; Sun et al., 2017). This technique involves encoding labels and combining them with node attributes through concatenation or summation. By doing so, it improves feature representation and enables the model to effectively utilize valuable information from labels. Importantly, instead of directly minimizing the distances between sample-dependent embeddings, denoted as z_S and z_T, we introduce a sample-independent method. This is accomplished by aligning the latent space of the student with that of the teacher, achieved through the approximation of distributions of embeddings from both networks. This distribution alignment approach avoids dependence on individual sample inputs. It is noteworthy that our primary objective is to derive comprehensive embeddings for the student model, with less emphasis on the training specifics of the teacher model. Consequently, the teacher model can be trained on a larger dataset or multiple datasets, eliminating the requirement for the student to have access to identical information.

Protein Graph Message Passing A protein sequence consists of n residues, which are deemed as graph nodes. We concatenate the one-hot encoding of residue types with the physicochemical properties of each residue, namely, a steric parameter, hydrophobicity, volume, polarizability, isoelectric point, helix probability, and sheet probability (Xu et al., 2022; Hanson et al., 2019), which are used as the graph node features $\boldsymbol{x}_i$. These node features capture meaningful biochemical characteristics, enabling the model to learn which residues tend to be buried, exposed, tightly packed, *etc.* We define the sequential distance, $l_{ij} = \|i - j\|$, and spatial distance $d_{ij} = \|P_i - P_j\|$, where P_i is the 3D coordinate of the C_α atom of the i-th residue. An edge ε_{ij} exists if:

$$l_{ij} < l_s \quad \text{and} \quad d_{ij} < r_s \tag{3}$$

where l_s, r_s are predefined radius thresholds, $\boldsymbol{e}_{ij}$ consists of geometric features of the protein structure, defined in Eq. 2. We convolve node and edge features from sequence and structure simultaneously and formulate the message passing mechanism as:

$$\begin{aligned} \boldsymbol{h}_i^{(0)} &= \mathrm{BN}\left(\mathrm{FC}\left(\boldsymbol{x}_i\right)\right), \\ \boldsymbol{u}_i^{(l)} &= \sigma(\mathrm{BN}(\sum_{v_j \in \mathcal{N}(v_i)} W \boldsymbol{e}_{ij} \boldsymbol{h}_j^{(l-1)}), \\ \boldsymbol{h}_i^{(l)} &= \boldsymbol{h}_i^{(l)} + \mathrm{Dropout}(\mathrm{FC}(\boldsymbol{u}_i^{(l)})) \end{aligned} \tag{4}$$

This mechanism (as shown in Eq. 4) can fuse and update the node and edge features, which includes aggregation and update functions, where $\mathrm{FC}(\cdot)$, $\mathrm{BN}(\cdot)$, $\mathrm{Dropout}(\cdot)$ represent fully connected, batch normalization, and dropout layers, $\sigma(\cdot)$ is the activation function LeakyReLU and W is the learnable convolutional kernel. $\mathcal{N}(v_i)$ refers to the neighbors of node v_i, and $\boldsymbol{h}_i^{(l)}$ is the representation of node v_i in the l-th message passing layer. The node and edge features are processed together in Eq. 4. After message passing operations, a sequence pooling layer is applied to reduce the sequence length, providing a simple but effective way to aggregate key patterns. After average pooling, the residue number is halved; we expand the radius r_s to $2r_s$ to update the edge conditions and perform the message passing and pooling operations again. These operations can make the GNNs cover more distant nodes gradually. The teacher and student models share the same GNN architecture to process protein sequences and structures. Finally, a global pooling layer is applied to obtain the graph-level protein embeddings, denoted as h_S and z_T for the teacher and student. Detailed model descriptions are presented in Appendix B.2.

Protein Domain Adaption As shown in Figure 1, the teacher model consists of GNNs and an auxiliary annotation encoder, which is a multi-layer perceptron (MLP) that provides function-friendly protein representations. The annotations associated with G_S serve as the input for the annotation encoder, resulting in the extraction of feature vector h_A. Therefore, we can combine h_A and the graph-level protein embeddings h_S learned from G_S together:

$$\begin{aligned} h_A &= \mathrm{MLP}(A) \\ z_S &= h_A + \alpha h_S \end{aligned} \tag{5}$$

where α is a hyper-parameter, controlling the balance between the contribution of the annotation embeddings h_A and the protein embeddings h_S in the combined representations.

As depicted in Figure 1, the generated protein embeddings z_S contain sequence, structure, and function information, guiding the training of the student model. Since knowledge-enhanced embeddings z_S are intended to be aligned with z_T, we obtain z_S from the entire protein and GO term datasets to avoid dependence on individuals. Then, we calculate the distributions of z_S and z_T to better capture the inherent uncertainty in the teacher's and student's latent spaces, in which the real distributions are sample-independent. The minibatch is adopted to approximate the quantities $p_S(z_S)$ and $p_T(z_T)$:

$$\begin{aligned} p_S(z_S) &= \mathbb{E}_{p_S(G_S,A)}[p(z_S|G_S,A)] \\ &\approx \frac{1}{B_S}\sum_{i=1}^{B_S} p_S(z_S|G_S^{(i)},A^{(i)}) \\ p_T(z_T) &= \mathbb{E}_{p_T(G_T)}[p_T(z_T|G_T)] \\ &\approx \frac{1}{B_S}\sum_{i=1}^{B_S} p_T(z_T|G_T^{(i)}) \end{aligned} \tag{6}$$

where B_S is the batch size. A Gaussian distribution Θ is assumed for protein embeddings, which exhibit smoothness and symmetry properties that can reasonably mimic the expected continuity and unimodality of the embeddings aggregated over many residues. We employ the reparameterization trick (Kingma & Welling, 2013) to sample the protein embeddings.

$$p_S(z_S) = \Theta(\mu_S, \sigma_S^2); \quad p_T(z_T) = \Theta(\mu_T, \sigma_T^2) \tag{7}$$

where μ_S, σ_S^2 and μ_T, σ_T^2 are the mean and variance values of the embeddings for the teacher and student models, providing a summary of the distribution using first- and second-order statistics.

Proposition 2 in Appendix E shows that the conditional misalignment in the representation space is bounded by the conditional misalignment in the input space. We have:

$$\mathcal{L}^*_{\text{student}} \leq \mathcal{L}_{\text{teacher}} + \frac{M}{\sqrt{2}} C \tag{8}$$

$$C = \sqrt{\mathrm{KL}\left[p_S(z) \parallel p_T(z)\right] + \mathbb{E}_{p_S(G)}\left[\mathrm{KL}\left[p_S(y|G) \parallel p_T(y|G)\right]\right]} \tag{9}$$

where $\mathcal{L}^*_{\text{student}}$ is the ideal target domain loss, and $\mathcal{L}_{\text{teacher}}$ is the teacher's supervised loss, M is a bound, see Appendix E. $\mathbb{E}_{p_S(G)}\left[\mathrm{KL}\left[p_S(y|G) \parallel p_T(y|G)\right]\right]$ is often small and fixed (not dependent on the representation z, and y is the function label). To reduce the generalization bound, we can focus on optimizing the marginal misalignment with a hyper-parameter β:

$$\mathcal{L}_{\text{teacher}} + \beta(\mathrm{KL}\left[p_S(z) \parallel p_T(z)\right]) \tag{10}$$

Eq. 10 can be used in an unsupervised way for the student to predict functions, which is near the ideal target domain loss. For the proposed framework ProteinSSA (Figure 1), we use the $\mathcal{L}_{\text{teacher}}$ to first train the teacher model, we adopt a hybrid loss $\mathcal{L}$ to train the student model using the labeled data in the target domain, where the $\mathcal{L}_{kd} = \mathrm{KL}\left[p_S(z)|p_T(z)\right]$ is to optimize the marginal misalignment between teacher and student models. Therefore, the final loss $\mathcal{L}$ with a hyper-parameter β for the student model is formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{student}} + \beta \mathcal{L}_{kd} \tag{11}$$

Table 3. Accuracy (%) of fold classification and enzyme reaction classification. The best results are shown in bold.

Input	Method	Fold Classification			Enzyme
		Fold	SuperFamily	Family	Reaction
Sequence	CNN (Shanehsazzadeh et al., 2020)	11.3	13.4	53.4	51.7
	ResNet (Rao et al., 2019)	10.1	7.21	23.5	24.1
	LSTM (Rao et al., 2019)	6.41	4.33	18.1	11.0
	Transformer (Rao et al., 2019)	9.22	8.81	40.4	26.6
Structure	GCN (Kipf & Welling, 2016)	16.8	21.3	82.8	67.3
	GAT (Velickovic et al., 2017)	12.4	16.5	72.7	55.6
	3DCNN_MQA (Derevyanko et al., 2018)	31.6	45.4	92.5	72.2
Sequence-Structure	GraphQA (Baldassarre et al., 2020)	23.7	32.5	84.4	60.8
	GVP (Jing et al., 2020b)	16.0	22.5	83.8	65.5
	ProNet-Amino Acid (Wang et al., 2023)	51.5	69.9	99.0	86.0
	ProNet-Backbone (Wang et al., 2023)	52.7	70.3	99.3	86.4
	ProNet-All-Atom (Wang et al., 2023)	52.1	69.0	99.0	85.6
	CRL (Hermosilla & Ropinski, 2022)	47.6	70.2	99.2	87.2
	GearNet (Zhang et al., 2023b)	28.4	42.6	95.3	79.4
	GearNet-IEConv (Zhang et al., 2023b)	42.3	64.1	99.1	83.7
	GearNet-Edge (Zhang et al., 2023b)	44.0	66.7	99.1	86.6
	GearNet-Edge-IEConv (Zhang et al., 2023b)	48.3	70.3	99.5	85.3
	CDConv (Fan et al., 2023)	56.7	77.7	99.6	88.5
	ProteinSSA (Student)	**60.5**	**79.4**	**99.8**	**89.4**

The objective function of the teacher model $\mathcal{L}_{\text{teacher}}$ is the cross entropy loss for protein graph classification. It is important to note that the training of the teacher model can be considered distinct from traditional pre-training, as it does not involve unsupervised or self-supervised learning on a large dataset. The hybrid loss for the student model has a cross entropy loss $\mathcal{L}_{\text{student}}$ for classification and a regularization loss $\mathcal{L}_{kd}$ for domain-adapted knowledge distillation.

4. Experiments

4.1. Training Details

The proposed multimodal knowledge distillation framework, ProteinSSA, undergoes a two-step training process. A dataset comprising approximately 30,000 proteins, each associated with 2,752 Gene Ontology (GO) annotations from the GO dataset, is utilized without further categorization into biological process (BP), molecular function (MF), and cellular component (CC) classes (Gligorijević et al., 2021). These classes serve as input to the annotation encoder of the teacher model, yielding an overall $\mathrm{F}_{\max}$ of 0.489 for the teacher model. Subsequently, the student model is trained. The optimization is performed using the Adam optimizer through the PyTorch library, and the performance metrics are computed as mean values over three initializations. Further details regarding experimental settings are available in Appendix B.3.

4.2. Tasks and Baselines

Following the tasks in IEconv (Hermosilla et al., 2021) and CDConv (Fan et al., 2023), we evaluate ProteinSSA on four protein tasks: protein fold classification, enzyme reaction classification, GO term prediction, and EC number prediction. Detailed task descriptions are presented in Appendix C. Dataset statistics are shown in Table 6.

Baselines The proposed method is compared with existing protein representation learning methods, which are classified into three categories based on their inputs, which could be a sequence, 3D structure, or both sequence and structure. 1) Sequence-based encoders, including CNN (Shanehsazzadeh et al., 2020), ResNet (Rao et al., 2019), LSTM (Rao et al., 2019) and Transformer (Rao et al., 2019). 2) Structure-based methods (GCN (Kipf & Welling, 2016), GAT (Velickovic et al., 2017), 3DCNN_MQA (Derevyanko et al., 2018) 3) Sequence-structure based models, *e.g.*, GVP (Jing et al., 2020b), CRL (Hermosilla & Ropinski, 2022), ProNet (Wang et al., 2023), GearNet (Zhang et al., 2023b), CDConv (Fan et al., 2023), *etc*. GearNet-IEConv and GearNetEdge-IEConv (Zhang et al., 2023b) add the IEConv (Hermosilla et al., 2021) layer on GearNet. We also compare ProteinSSA with prevalent pre-training models, the baselines and results are shown in Appendix G.

Table 4. $\mathrm{F}_{\max}$ of GO term prediction and EC number prediction. The best results are shown in bold.

Category	Method	GO-BP	GO-MF	GO-CC	EC
Sequence	CNN (Shanehsazzadeh et al., 2020)	0.244	0.354	0.287	0.545
	ResNet (Rao et al., 2019)	0.280	0.405	0.304	0.605
	LSTM (Rao et al., 2019)	0.225	0.321	0.283	0.425
	Transformer (Rao et al., 2019)	0.264	0.211	0.405	0.238
Structure	GCN (Kipf & Welling, 2016)	0.252	0.195	0.329	0.320
	GAT (Velickovic et al., 2017)	0.284	0.317	0.385	0.368
	3DCNN_MQA (Derevyanko et al., 2018)	0.240	0.147	0.305	0.077
Sequence-Structure	GraphQA (Baldassarre et al., 2020)	0.308	0.329	0.413	0.509
	GVP (Jing et al., 2020b)	0.326	0.426	0.420	0.489
	CRL (Hermosilla & Ropinski, 2022)	0.421	0.624	0.431	-
	GearNet (Zhang et al., 2023b)	0.356	0.503	0.414	0.730
	GearNet-IEConv (Zhang et al., 2023b)	0.381	0.563	0.422	0.800
	GearNet-Edge (Zhang et al., 2023b)	0.403	0.580	0.450	0.810
	GearNet-Edge-IEConv (Zhang et al., 2023b)	0.400	0.581	0.430	0.810
	CDConv (Fan et al., 2023)	0.453	0.654	0.479	0.820
	ProteinSSA (Student)	**0.464**	**0.667**	**0.492**	**0.857**

4.3. Results of Fold and Enzyme Reaction Classification

Table 3 shows performance comparisons on protein fold and enzyme reaction prediction across different methods, reported as average values. From the table 3, we can see that the proposed ProteinSSA achieves the best performance among all methods on the four test sets for both fold and reaction prediction tasks. Sequence-structure based methods generally outperform sequence- or structure-only methods, indicating the benefits of co-modeling sequence and structure. Notably, on the Fold test set, ProteinSSA improves accuracy by over 6.7% compared to prior techniques, demonstrating its effectiveness at learning sequence, structure, and function mappings. Additionally, both CDConv and ProteinSSA use sequence-structure convolution architectures, but ProteinSSA outperforms the CDConv model. This suggests the teacher-student training paradigm in ProteinSSA helps the student learn superior protein embeddings.

4.4. Results of GO Term and EC Number Prediction

Following the protocol in GearNet (Zhang et al., 2023b), the test sets for GO term and EC number prediction only contain PDB chains with less than 95% sequence identity to the training set, ensuring rigorous evaluation. The student model conducts the experiments, and the teacher model's annotations are not classified into these classes, avoiding data leakage. Table 4 shows comparative results between different protein modeling methods on these tasks, with performance measured by $\mathrm{F}_{\max}$, which balances precision and recall, working well even if positive and negative classes are imbalanced. The mean values of three independent runs are reported. ProteinSSA achieves the highest $\mathrm{F}_{\max}$ across all test sets for both GO and EC prediction, outperforming other approaches. This demonstrates ProteinSSA's strong capabilities for predicting protein functions and activities. Compared to preliminary results in Table 2, ProteinSSA even exceeds CDConv (Fan et al., 2023) augmented with sequence-function embeddings got from the large-scale pretrained model, KeAP (Zhou et al., 2023) on EC number prediction, while being comparable on GO term prediction. Overall, the consistent improvements verify the benefits of injecting functional information into sequence-structure models, as done in ProteinSSA's teacher-student framework. The results cement ProteinSSA's effectiveness using knowledge distillation techniques.

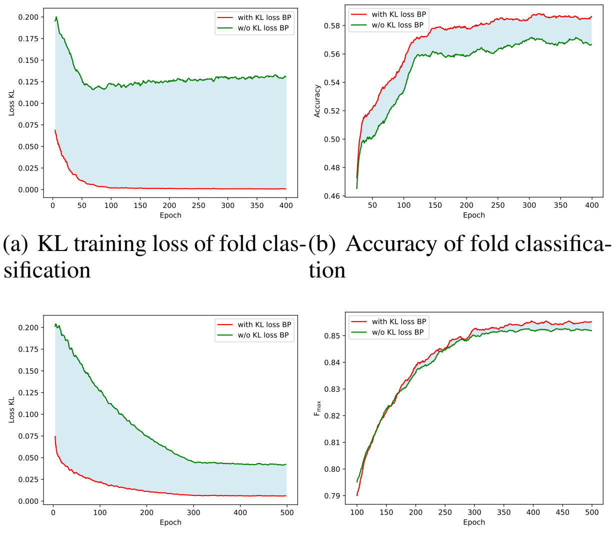
4.5. Ablation Study

Table 5 presents ablation studies of the proposed ProteinSSA model on the four downstream tasks. We examine the impact of removing the teacher model, which means removing the $\mathcal{L}_{kd}$. We also remove the annotation encoder in the teacher, which means that we incorporate function information into the loss function for the teacher models. As shown in Table 5, removing the teacher model altogether (w/o Teacher) leads to substantial performance drops across all tasks compared to the full ProteinSSA. This shows the teacher's knowledge distillation provides useful signals for the student model. Besides, removing the annotation encoder in the teacher (w/o AE-T) also degrades performance, though less severely. Despite being a label-augmented strategy, the annotation encoder exhibits minimal influence, indicating low sensitivity and limited impact on test perfor-

Table 5. Ablation experiments of our proposed method. w/o AE-T denotes without the annotation encoder in the teacher model. w/o teacher means without the teacher model and directly using the student model, which also means without $\mathcal{L}_{kd}$.

Method	Fold Classification			Enzyme	GO			EC
	Fold	Superfamily	Family	Reaction	BP	MF	CC	
ProteinSSA	60.5	79.4	99.8	89.4	0.464	0.667	0.492	0.857
w/o AE-T	60.4	79.1	99.7	88.9	0.454	0.664	0.490	0.854
w/o Teacher	57.8	78.7	99.6	88.6	0.458	0.660	0.484	0.851

mance. Our student model is specifically designed to process protein sequences and structures as inputs, enabling it to function independently without the need for guidance from the teacher model.



(a) KL training loss of fold classification

(b) Accuracy of fold classification

(c) KL training loss of EC number prediction

(d) $\mathrm{F}_{\max}$ of EC number prediction

Figure 2. The KL training loss curves (a), (c) and test performance (b), (d) on the tasks of fold classification and EC number prediction. The red curve denotes that $\mathcal{L}_{\mathrm{kd}}$ conducts its function, while the green curve denotes we calculated the value of $\mathcal{L}_{\mathrm{kd}}$, but it is not involved in the process of the gradient backpropagation (BP).

Figure 2 illustrates the comparisons of the knowledge distillation loss $\mathcal{L}_{kd}$ with and without involvement in backpropagation during training. When $\mathcal{L}_{kd}$ is excluded from the gradient backpropagation process, it exhibits a decrease alongside the classification loss $\mathcal{L}$student. However, its value remains substantially higher compared to when $\mathcal{L}_{kd}$ is included in the training process. Similar observations are presented for the accuracy and $\mathrm{F}_{\max}$ on the fold classification and EC number prediction. The notable disparity observed between the distillation loss and the test performance with and without involvement in backpropagation suggests that the KL loss does indeed play a significant role in guiding the student model's learning process. Its presence influences the model's performance and convergence.

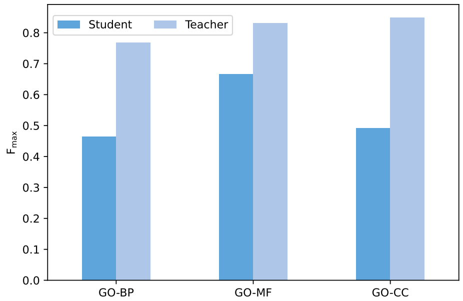


Figure 3. Performance comparisons of the teacher and student of ProteinSSA on GO term prediction.

We compare the performance of the teacher and the student on the tasks of GO term prediction. From the provided Figure 3, it is evident that incorporating functional information as the input of the annotation encoder significantly enhances performance, particularly for MF and CC term prediction. These two classes have fewer categories and are more accessible, resulting in higher scores.

5. Conclusion

In this paper, we propose ProteinSSA, a multimodal protein representation learning framework integrating the information from protein sequences, structures, and annotations. Unlike large-scale pre-training methods, ProteinSSA efficiently integrates functional knowledge. Importantly, we estimate the latent embedding distributions for the teacher-student model and learn annotation-enriched student representations by distribution approximation. Compared to mainstream protein representation learning techniques, ProteinSSA achieves superior performance in predicting protein families, reactions, GO terms, and EC numbers. These consistent improvements across benchmarks highlight the advantages of this approach for informative protein representation learning. However, the student is restricted by the teacher's ability. Therefore, this framework could be improved by training the teacher on larger annotation datasets with more information, like protein domains, motifs, regions, and shapes, *etc.*, or by designing a much more complicated annotation encoder, which is a promising direction.

6. Social Impact

Our proposed framework, ProteinSSA, can enable advanced protein analyses and provide effective and comprehensive representations that incorporate the information from protein sequences, structures, and functions. However, there may exist broader impacts and harmful activities. In detail, the representations are potentially misused, *e.g.*, designing harmful molecules or proteins based on these representations. Wet lab experiments may be needed for the newly found mechanisms or functions of proteins based on the learned representations.

References

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. Gene ontology: tool for the unification of biology. *Nature Genetics*, pp. 25–29, May 2000. doi: 10.1038/75556. URL http://dx.doi.org/10.1038/75556.

Baldassarre, F., Hurtado, D. M., Elofsson, A., and Azizpour, H. Graphqa: protein model quality assessment using graph convolutional networks. *Bioinformatics*, 2020.

Baldassarre, F., Menéndez Hurtado, D., Elofsson, A., and Azizpour, H. Graphqa: protein model quality assessment using graph convolutional networks. *Bioinformatics*, pp. 360–366, Apr 2021. doi: 10.1093/bioinformatics/btaa714. URL http://dx.doi.org/10.1093/bioinformatics/btaa714.

Bateman, A. Uniprot: A worldwide hub of protein knowledge. *Nucleic Acids Research*, 2019.

Bengio, S., Weston, J., and Grangier, D. Label embedding trees for large multi-class tasks. *Advances in neural information processing systems*, 23, 2010.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. The protein data bank. *Nucleic acids research*, 28(1): 235–242, 2000a.

Berman, H. M., Westbrook, J. D., Feng, Z., Gilliland, G. L., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. The protein data bank. *Nucleic Acids Research*, 2000b.

Boutet, E., Lieberherr, D., Tognolli, M., Schneider, M., Bansal, P., Bridge, A. J., Poux, S., Bougueleret, L., and Xenarios, I. Uniprotkb/swiss-prot, the manually annotated section of the uniprot knowledgebase: how to use the entry view. *Plant bioinformatics: methods and protocols*, pp. 23–54, 2016.

Brandes, N., Ofer, D., Peleg, Y., Rappoport, N., and Linial, M. Proteinbert: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110, 2022.

Chen, B., Cheng, X., Li, P., Geng, Y.-a., Gong, J., Li, S., Bei, Z., Tan, X., Wang, B., Zeng, X., et al. xtrimopglm: unified 100b-scale pre-trained transformer for deciphering the language of protein. *arXiv preprint arXiv:2401.06199*, 2024.

Consortium, U. Update on activities at the universal protein resource (uniprot) in 2013. *Nucleic Acids Research*, 2013.

Derevyanko, G., Grudinin, S., Bengio, Y., and Lamoureux, G. Deep convolutional networks for quality assessment of protein folds. *Bioinformatics*, 34(23):4046–4053, 2018.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Yu, W., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., Bhowmik, D., and Rost, B. Prottrans: Towards cracking the language of lifes code through self-supervised deep learning and high performance computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Fan, H., Wang, Z., Yang, Y., and Kankanhalli, M. Continuous-discrete convolution for geometry-sequence modeling in proteins. In *The Eleventh International Conference on Learning Representations*, 2023.

Farahani, A., Voghoei, S., Rasheed, K., and Arabnia, H. R. A brief review of domain adaptation. *Advances in data science and information engineering: proceedings from ICDATA 2020 and IKE 2020*, pp. 877–894, 2021.

Ferruz, N., Schmidt, S., and Höcker, B. Protgpt2 is a deep unsupervised language model for protein design. *Nature Communications*, Jul 2022. doi: 10.1038/s41467-022-32007-7. URL http://dx.doi.org/10.1038/s41467-022-32007-7.

Ganin, Y. and Lempitsky, V. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pp. 1180–1189. PMLR, 2015.

Gligorijević, V., Renfrew, P. D., Kosciolek, T., Leman, J. K., Berenberg, D., Vatanen, T., Chandler, C., Taylor, B. C., Fisk, I. M., Vlamakis, H., et al. Structure-based protein function prediction using graph convolutional networks. *Nature communications*, 12(1):3168, 2021.

Guo, Y., Wu, J., Ma, H., and Huang, J. Self-supervised pre-training for protein embeddings using tertiary structures. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6801–6809, 2022.

Hanson, J., Paliwal, K., Litfin, T., Yang, Y., and Zhou, Y. Improving prediction of protein secondary structure, backbone angles, solvent accessibility and contact numbers by using predicted contact maps and an ensemble of recurrent and residual convolutional neural networks. *Bioinformatics*, 35(14):2403–2410, 2019.

He, H., Wang, J., Zhang, Z., and Wu, F. Compressing deep graph neural networks via adversarial knowledge distillation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 534–544, 2022a.

He, R., Sun, S., Yang, J., Bai, S., and Qi, X. Knowledge distillation as efficient pre-training: Faster convergence, higher data-efficiency, and better transferability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9161–9171, 2022b.

Heinzinger, M., Littmann, M., Sillitoe, I., Bordin, N., Orengo, C., and Rost, B. Contrastive learning on protein embeddings enlightens midnight zone. *NAR genomics and bioinformatics*, 4(2):lqac043, 2022.

Hermosilla, P. and Ropinski, T. Contrastive representation learning for 3d protein structures. *arXiv preprint arXiv:2205.15675*, 2022.

Hermosilla, P., Schäfer, M., Lang, M., Fackelmann, G., Vázquez, P. P., Kozlíková, B., Krone, M., Ritschel, T., and Ropinski, T. Intrinsic-extrinsic convolution and pooling for learning on 3d protein structures. *International Conference on Learning Representations*, 2021.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network, 2015.

Hou, J., Adhikari, B., and Cheng, J. Deepsf: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, 34(8):1295–1303, 2018.

Hu, B., Tan, C., Xia, J., Zheng, J., Huang, Y., Wu, L., Liu, Y., Xu, Y., and Li, S. Z. Learning complete protein representation by deep coupling of sequence and structure. *bioRxiv*, pp. 2023–07, 2023a.

Hu, F., Hu, Y., Zhang, W., Huang, H., Pan, Y., and Yin, P. A multimodal protein representation framework for quantifying transferability across biochemical downstream tasks. *Advanced Science*, pp. 2301223, 2023b.

Ibtehaz, N., Kagaya, Y., and Kihara, D. Domain-pfp allows protein function prediction using function-aware domain embedding representations. *Communications Biology*, 6 (1):1103, 2023.

Ingraham, J., Garg, V., Barzilay, R., and Jaakkola, T. Generative models for graph-based protein design. *Advances in neural information processing systems*, 32, 2019.

Jiang, P., Wu, A., Han, Y., Shao, Y., Qi, M., and Li, B. Bidirectional adversarial training for semi-supervised domain adaptation. In *IJCAI*, pp. 934–940, 2020.

Jing, B., Eismann, S., Suriana, P., Townshend, R. J., and Dror, R. Learning from protein structure with geometric vector perceptrons. *arXiv preprint arXiv:2009.01411*, 2020a.

Jing, B., Eismann, S., Suriana, P., Townshend, R. J. L., and Dror, R. O. Learning from protein structure with geometric vector perceptrons. *Learning*, 2020b.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Kim, T. and Kim, C. Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pp. 591–607. Springer, 2020.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, 2022.

Liu, J., Zheng, T., Zhang, G., and Hao, Q. Graph-based knowledge distillation: A survey and experimental evaluation, 2023.

Long, M., Cao, Y., Wang, J., and Jordan, M. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pp. 97–105. PMLR, 2015.

Long, M., Cao, Z., Wang, J., and Jordan, M. I. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018.

Microsoft. Neural Network Intelligence, 1 2021. URL https://github.com/microsoft/nni.

Nguyen, A. T., Tran, T., Gal, Y., Torr, P. H. S., and Baydin, A. G. Kl guided domain adaptation, 2022.

Nijkamp, E., Ruffolo, J., Weinstein, E. N., Naik, N., and Madani, A. Progen2: Exploring the boundaries of protein language models, 2022.

Omelchenko, M. V., Galperin, M. Y., Wolf, Y. I., and Koonin, E. V. Non-homologous isofunctional enzymes: a systematic analysis of alternative solutions in enzyme evolution. *Biology direct*, 5:1–20, 2010.

Pei, Z., Cao, Z., Long, M., and Wang, J. Multi-adversarial domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Qin, C., Wang, L., Ma, Q., Yin, Y., Wang, H., and Fu, Y. Contradictory structure learning for semi-supervised domain adaptation. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pp. 576–584. SIAM, 2021.

Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, X., Canny, J., Abbeel, P., and Song, Y. S. Evaluating protein transfer learning with tape. *bioRxiv*, 2019.

Rao, R., Liu, J., Verkuil, R., Meier, J., Canny, J. F., Abbeel, P., Sercu, T., and Rives, A. Msa transformer. *bioRxiv*, 2021. doi: 10.1101/2021.02.12.430858. URL https://www.biorxiv.org/content/10.1101/2021.02.12.430858v1.

Rao, R. M., Meier, J., Sercu, T., Ovchinnikov, S., and Rives, A. Transformer protein language models are unsupervised structure learners. *bioRxiv*, 2020. doi: 10.1101/2020.12.15.422761. URL https://www.biorxiv.org/content/10.1101/2020.12.15.422761v1.

Rives, A., Goyal, S., Meier, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., and Fergus, R. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences of the United States of America*, 2019.

Saito, K., Kim, D., Sclaroff, S., Darrell, T., and Saenko, K. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8050–8058, 2019.

Serçinoğlu, O. and Ozbek, P. Sequence-structure-function relationships in class i mhc: A local frustration perspective. *PloS one*, 15(5):e0232849, 2020.

Shanehsazzadeh, A., Belanger, D., and Dohan, D. Is transfer learning necessary for protein landscape prediction? *arXiv preprint arXiv:2011.03443*, 2020.

Su, J., Han, C., Zhou, Y., Shan, J., Zhou, X., and Yuan, F. Saprot: Protein language modeling with structure-aware vocabulary. In *International Conference on Learning Representations*, 2024.

Sun, X., Wei, B., Ren, X., and Ma, S. Label embedding network: Learning label representation for soft training of deep networks. *arXiv preprint arXiv:1710.10393*, 2017.

Tian, Y., Zhang, C., Guo, Z., Zhang, X., and Chawla, N. V. Nosmog: Learning noise-robust and structure-aware mlps on graphs. *arXiv preprint arXiv:2208.10010*, 2022.

Torres, M., Yang, H., Romero, A. E., and Paccanaro, A. Protein function prediction for newly sequenced organisms. *Nature Machine Intelligence*, 3(12):1050–1060, 2021.

Unsal, S., Atas, H., Albayrak, M., Turhan, K., Acar, A. C., and Doğan, T. Learning functional properties of proteins with language models. *Nature Machine Intelligence*, 4 (3):227–245, 2022.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., et al. Graph attention networks. *stat*, 1050 (20):10–48550, 2017.

Wang, L., Liu, H., Liu, Y., Kurtin, J., and Ji, S. Learning hierarchical protein representations via complete 3d graph networks. In *The Eleventh International Conference on Learning Representations*, 2023.

Wang, M. and Deng, W. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

Wang, Z., Combs, S. A., Brand, R., Calvo, M. R., Xu, P., Price, G., Golovach, N., Salawu, E. O., Wise, C. J., Ponnapalli, S. P., et al. Lm-gvp: A generalizable deep learning framework for protein property prediction from sequence and structure. *bioRxiv*, pp. 2021–09, 2021.

Webb, E. C. et al. *Enzyme nomenclature 1992. Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzymes.* Academic Press, 1992.

Wilson, G. and Cook, D. J. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46, 2020.

Xu, G., Wang, Q., and Ma, J. Opus-rota4: a gradient-based protein side-chain modeling framework assisted by deep learning-based predictors. *Briefings in Bioinformatics*, 23(1):bbab529, 2022.

Xu, M., Yuan, X., Miret, S., and Tang, J. Protst: Multimodality learning of protein sequences and biomedical texts, 2023.

Zhang, N., Bi, Z., Liang, X., Cheng, S., Hong, H., Deng, S., Lian, J., Zhang, Q., and Chen, H. Ontoprotein: Protein pretraining with gene ontology embedding. *arXiv preprint arXiv:2201.11147*, 2022.

Zhang, W., Miao, X., Shao, Y., Jiang, J., Chen, L., Ruas, O., and Cui, B. Reliable data distillation on graph convolutional network. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pp. 1399–1414, 2020.

Zhang, Z., Wang, C., Xu, M., Chenthamarakshan, V., Lozano, A., Das, P., and Tang, J. A systematic study of joint representation learning on protein sequences and structures. *Preprint at http://arxiv. org/abs/2303.06275*, 2023a.

Zhang, Z., Xu, M., Jamasb, A., Chenthamarakshan, V., Lozano, A., Das, P., and Tang, J. Protein representation learning by geometric structure pretraining. In *International Conference on Learning Representations*, 2023b.

Zhou, H.-Y., Fu, Y., Zhang, Z., Bian, C., and Yu, Y. Protein representation learning via knowledge enhanced primary structure modeling, 2023.

A. Local Coordinate System

We have introduced the local coordinate system (Ingraham et al., 2019) $\boldsymbol{Q}_i$ in the Section 3.1, which defines the geometric properties of the point v_i. It is shown in Figure 4. From this figure, we can easily find that b_i is the negative bisector of the angle between the rays $(P_{i-1} - P_i)$ and $(P_{i+1} - P_i)$.

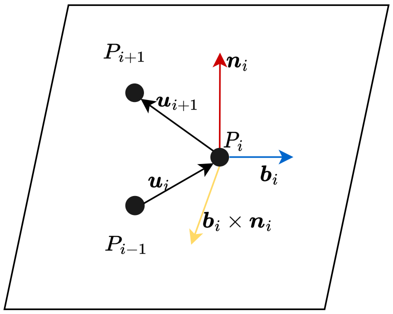


Figure 4. The local coordinate system $\boldsymbol{Q}_i$ related to protein graph node v_i, P_i is the coordinate of residue i.

B. Experiment Setup

B.1. Detailed Experimental Settings in Section 3.2

As stated in Section 3.2, embeddings generated from the pre-trained models, ESM-2[1] (Lin et al., 2022) and KeAP[2] (Zhou et al., 2023), are used to enhance the sequence-structure model CDConv[3] (Fan et al., 2023). As shown in Figure 5. A two-layer MLP is used to encode the generated embeddings, which are then added to the CDConv embeddings. The MLP has feature dimensions of 1024 and 2048, with other hyper-parameters remaining the same as the base models. This allows the integration of knowledge from large-scale pre-trained LMs into the sequence-structure framework for improving protein characterization.

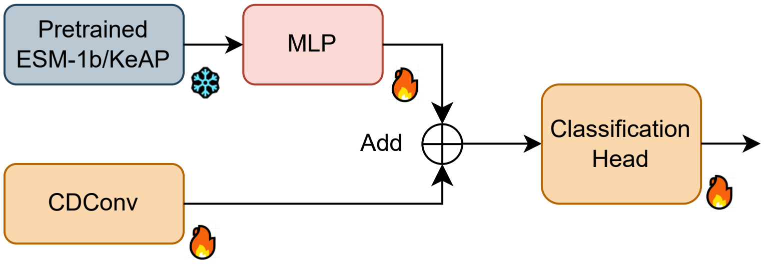


Figure 5. An illustration of the enhanced CDConv model.

B.2. Model Details

The radius r_s threshold increases from 4 to 16, and l_s is 11. We set two message passing layers with one average sequence pooling per GNN. After the pooling layer, the number of residues is halved, and we update the edge conditions before performing another round of message passing and pooling operations, as illustrated in the model Figure 1. The final GNNs include eight message-massing and four pooling layers, which are sufficient for achieving satisfactory results. The number of initial feature channels is 256, increased to 2048. The annotation encoder has 2 FC layers changing feature channels from 2752 to 2048. The classification head is a liner layer for predicting classes. For the teacher model, we use z_S to get the predicted annotations by the classification head and calculate the loss by $\mathcal{L}_{\text{teacher}}$. The final loss $\mathcal{L}$ is used for the training of the student model.

We know spatially adjacent residues can still exist even when the sequence distance is large (Hu et al., 2023a). We perform sequence average pooling and change edge conditions after once pooling. These operations enable the protein graph to cover more distant nodes.

B.3. Training Details

Dataset statistics (Zhang et al., 2023b) of the four downstream tasks are summarized in Table 6. The proposed framework conducted experiments on NVIDIA-SMI A100 GPUs and NVIDIA Tesla V100 GPUs, implemented with PyTorch 1.13+cu117 and PyTorch Geometric 2.3.1 with CUDA 11.2.

[1] https://github.com/facebookresearch/esm#available-models
[2] https://github.com/RL4M/KeAP
[3] https://github.com/hehefan/Continuous-Discrete-Convolution

Table 6. Dataset statistics. $\#$X means the number of X.

Dataset	$\#$Train	$\#$Validation	$\#$Test
Enzyme Commission	$15,550$	$1,729$	$1,919$
Gene Ontology	$29,898$	$3,322$	$3,415$
Fold Classification - Fold	$12,312$	736	718
Fold Classification - Superfamily	$12,312$	736	$1,254$
Fold Classification - Family	$12,312$	736	$1,272$
Reaction Classification	$29,215$	$2,562$	$5,651$

In biology, a linear combination of original data with Gaussian noise (Guo et al., 2022) is a simple but effective way to augment the protein data:

$$(P_i, \boldsymbol{x}_i) \leftarrow (P_i, \boldsymbol{x}_i) + \Theta, \Theta \sim (\mu_k, \sigma_k^2) \tag{12}$$

where μ_k and σ_k are selected as the random noise's mean (expectation) and standard deviation.

Hyper-parameters related to the networks are set the same across different datasets: Adam optimizer with learning rate $l_r = 1e - 3$, weight decay $decay = 5e - 4$, epochs $T = 300$, Gaussian noise $\mu_k = 0, \sigma_k = 0.1$, it indicates trivial perturbation is introduced to the protein native structures.

The other dataset-specific hyper-parameters are determined by an AutoML toolkit NNI (Microsoft, 2021) with the search spaces. The loss weight hyper-parameter is related to the value of the task-specific loss $\beta = \{1, 0.1, 0.01, 0.001, 0\}$, and $\alpha = \{10, 1, 0.1, 0.01, 0.001, 0\}$. As for the batch size and training epochs, *etc*., which influence the convergence speed of deep learning models, details about implementation on the NVIDIA-SMI A100 GPUs are shown in Table 7.

Table 7. More details of training setup

Hyper-parameter	Fold	Enzyme Reaction	GO	EC
Batch size	16	8	24	64
Epoch	400	400	500	500

C. Task Introduction

Fold Classification In order to understand how protein structure and evolution interact, it is crucial to be able to predict fold classes (Hou et al., 2018). This dataset contains 16,712 total proteins across 1,195 fold classes. Three test sets are provided. Fold: proteins from the same superfamily are excluded during training; SuperFamily: proteins from the same family are not used for training; and Family: the training set includes proteins from the same family.

Enzyme Reaction Classification Enzyme reaction classification can be viewed as a protein function prediction task based on the enzyme-catalyzed reactions defined by the four levels of enzyme commission numbers (Webb et al., 1992; Omelchenko et al., 2010). We use the dataset (Hermosilla et al., 2021; Berman et al., 2000a) containing 29,215 training proteins, 2,562 validation proteins, and 5,651 test proteins, spanning 384 four-level EC classes.

GO Term Prediction A Gene Ontology (GO) annotation denotes the function of a specific gene and gene product, associating it with a corresponding GO term. These annotations collectively provide a summary of current biological insights, detailing how genes operate at the molecular level, their cellular locations, and the biological processes they contribute to. Importantly, the GO terminology is designed to be universally applicable, encompassing prokaryotic and eukaryotic organisms, as well as both single-celled and multicellular life forms. GO annotations offer a more comprehensive and versatile approach to characterizing protein functions, encompassing a broader range of biological activities and processes[4]. The aim of GO term prediction is to predict whether a given protein should be annotated with a particular GO term. As we have stated before, proteins are categorized into three hierarchical ontologies: MF, BP and CC. Specifically, MF

[4] https://geneontology.org/docs/ontology-documentation/

denotes molecular activities of a protein, BP refers to larger biological processes it is involved in, and CC describes subcellular locations and extracellular components (Bateman, 2019). Accurately assigning GO terms is crucial for understanding protein function and assessing computational methods.

EC Number Prediction This task aims to predict the 538 Enzyme Commission (EC) numbers at the third and fourth level hierarchies for different proteins (Gligorijević et al., 2021), which provide precise information about a protein's enzymatic function, based on the protein's features. The large number of classes at the third and fourth EC levels makes this a challenging multi-class prediction problem in bioinformatics.

D. Evaluation Metric

$\mathrm{F}_{\max}$ provides an overall metric that combines both accuracy and coverage of the predictions. It is calculated by first determining the precision and recall for each protein, then averaging these results over all proteins (Zhang et al., 2023b; Gligorijević et al., 2021). p_i^j is the prediction probability for the j-th class of the i-th protein, given the decision threshold $t \in [0, 1]$, the precision and call are given as:

$$\text{precision}_i(t) = \frac{\sum_j \mathbb{I}[(\left(p_i^j \geq t\right) \cap b_i^j)]}{\sum_j \mathbb{I}[\left(p_i^j \geq t\right)]}, \quad \text{recall}_i(t) = \frac{\sum_j \mathbb{I}[\left(\left(p_i^j \geq t\right) \cap b_i^j\right)]}{\sum_j b_i^j}$$

where $b_i^j \in \{0, 1\}$ is the corresponding binary class label, and $\mathbb{I} \in \{0, 1\}$ is an indicator function. If there are N proteins in total, these protein-level precision and recall values are averaged over all proteins to obtain the overall precision and recall for the dataset, then the average precision and recall are defined as:

$$\text{precision}(t) = \frac{\sum_i^N \text{precision}_i(t)}{\sum_i^N \left(\left(\sum_j \left(p_i^j \geq t\right)\right) \geq 1\right)}, \quad \text{recall}(t) = \frac{\sum_i^N \text{recall}_i(t)}{N}$$

Finally, $\mathrm{F}_{\max}$ is defined as the maximum value of the F-score over all thresholds.

$$\mathrm{F}_{\max} = \max_t \left\{ \frac{2 \cdot \text{precision}(t) \cdot \text{recall}(t)}{\text{precision}(t) + \text{recall}(t)} \right\} \tag{13}$$

E. KL Guided Domain Adaption

Assuming source and target domains have the same support set and share the representation mapping $p(z|G)$, this means these two domains have the same datasets of protein graphs and functions. Given the representation z, we learn a classifier to predict the label y through the predictive distribution $\hat{p}(y|z)$ that is an approximation of the ground truth. During training, the representation network $p(z|G)$ and the classifier $\hat{p}(y|z)$ are trained jointly on the source domain and we hope that they can generalize to the target domain, meaning that both $p(z|G)$ and $\hat{p}(y|z)$ are kept unchanged between training and testing.

We define the predictive distribution of y given G as

$$\hat{p}(y|G) = \mathbb{E}_{p(z|G)}[\hat{p}(y|z)] \tag{14}$$

We have a single z from the source model $p(z|G)$ for each protein. The training objective of the source domain is

$$\mathcal{L}_{\text{teacher}} = \mathbb{E}_{G,y \sim p_S(G,y), z \sim p(z|G)}[-\log \hat{p}(y|z)] = \mathbb{E}_{p_S(z,y)}[-\log \hat{p}(y|z)] \tag{15}$$

We consider the two assumptions of the representation z on the source domain:

Assumption 1. $I_S(z, y) = I_S(G, y)$, where $I_S(\cdot, \cdot)$ is the mutual information term, calculated on the source domain. In particular:

$$I_S(z, y) = \mathbb{E}_{p_S(z,y)} \left[\log \frac{p_S(z, y)}{p_S(z) p_S(y)} \right]; \quad I_S(G, y) = \mathbb{E}_{p_S(G,y)} \left[\log \frac{p_S(G, y)}{p_S(G) p_S(y)} \right] \tag{16}$$

The mutual information quantifies the amount of information shared between the variables z and y (or G and y) in the source domain. It measures the dependence or correlation between these variables in the context of the source domain data. This is often referred to as the 'sufficiency assumption' since it indicates that the representation z has the same information about the label y as the original input protein graph G, and is sufficient for this prediction task in the source domain. Note that the data processing inequality indicates that $I_S(z, y) \leq I_S(G, y)$, so here we assume that z contains maximum information about y.

Assumption 2. $p_S(y|G) = \mathbb{E}_{p(z|G)}\left[p_S(y|z)\right]$

When this assumption holds, the predictive distribution $\hat{p}(y|G)$ will approximate $p_S(y|G)$, as long as $\hat{p}(y|z)$ approximates $p_S(y|z)$.

The above two assumptions ensure that the teacher network has good performance in the source domain. Now, we continue to consider the test loss and how we can reduce it. The loss of the target domain is:

$$\begin{aligned}\mathcal{L}^*_{\text{student}} &= \mathbb{E}_{p_T(G,y)}[-\log \hat{p}(y|G)] = \mathbb{E}_{p_T(G,y)}\left[-\log \mathbb{E}_{p(z|G)}[\hat{p}(y|z)]\right] \\ &\leq \mathbb{E}_{p_T(G,y)}\left[\mathbb{E}_{p(z|G)}[-\log \hat{p}(y|z)]\right] \\ &= \mathbb{E}_{p_T(z,y)}[-\log \hat{p}(y|z)]\end{aligned} \tag{17}$$

Since we do not know the target domain and the target data distribution, there is no way to guarantee the invariance (both marginally and conditionally) of the representation z. Therefore, We introduce the following proposition that ensures a generalization bound of the target domain loss based on the source domain loss and the KL divergence:

Proposition 1. If the loss $-\log \hat{p}(y|z)$ is bounded by M, we have:

$$\begin{aligned}\mathcal{L}^*_{\text{student}} &\leq \mathcal{L}_{\text{teacher}} + \frac{M}{\sqrt{2}}\sqrt{\text{KL}\left[p_S(y, z) \parallel p_T(y, z)\right]} \\ &= \mathcal{L}_{\text{teacher}} + \frac{M}{\sqrt{2}}\sqrt{\text{KL}\left[p_S(z) \parallel p_T(z)\right] + \mathbb{E}_{p_S(z)}\left[\text{KL}\left[p_S(y|z) \parallel p_T(y|z)\right]\right]}\end{aligned} \tag{18}$$

Proposition 2. If Assumption 1 and 2 hold, and if $\frac{p_S(G,y)}{p_T(G,y)} < \infty$ (i.e., there exists N', which can be arbitrarily large, such that $\frac{p_S(G,y)}{p_T(G,y)} < N'$), we have

$$\mathbb{E}_{p_S(G)}\left[\text{KL}\left[p_S(y|z) \parallel p_T(y|z)\right]\right] \leq \mathbb{E}_{p_S(G)}\left[\text{KL}\left[p_S(y|G) \parallel p_T(y|G)\right]\right] \tag{19}$$

This shows that the conditional misalignment in the representation space is bounded by the conditional misalignment in the input space. It then follows that:

$$\mathcal{L}^*_{\text{student}} \leq \mathcal{L}_{\text{teacher}} + \frac{M}{\sqrt{2}}\sqrt{\text{KL}\left[p_S(z) \parallel p_T(z)\right] + \mathbb{E}_{p_S(G)}\left[\text{KL}\left[p_S(y|G) \parallel p_T(y|G)\right]\right]} \tag{20}$$

We know y can represent the underlying functional label for the student model. Although the student model may not have these functional labels, but we can assume that they exist for theoretical reasons. The derived misalignment Eq. 20 and the derived loss Eq. 8 are based on the assumption that the source and target domains have the same support set. Thus, the loss of Eq. 8 can be used in an unsupervised way for the student to predict functions. However, the student model is applied to different downstream tasks, like classification, which has classification classes. Thus, we add the supervised student loss $\mathcal{L}_{\text{student}}$ and the knowledge distillation loss the $\mathcal{L}_{kd}$ as the final hybrid loss for the student to improve its performance on classification tasks.

F. Complexity Analysis

Our main focus is on generating comprehensive embeddings for the student model, with less emphasis placed on the training specifics of the teacher model. Regarding the student model, the computational complexity of one message passing layer in this framework is $\mathcal{O}(nd_n)$, where d_n represents the average node degree, typically much smaller than n. The time complexity is directly related to the computational complexity of the message passing layer; as graph convolution is performed on nodes and edges simultaneously, the time complexity remains $\mathcal{O}(nd_n)$, linear with the number of residues n. Denoting the size of the batch as B_s, the overall computational complexity is merely $\mathcal{O}(B_s nd_n)$.

Table 8. Comparison results with large-scale pre-training models on GO term prediction and EC number prediction, $\mathrm{F}_{\max}$ is presented. The best results are shown in bold. Param., means the number of trainable parameters (B: billion; M: million; K: thousand); Seq: sequence; Struct: structure; Func: function. Results† are from (Fan et al., 2023). Results* are from (Xu et al., 2023). Structures used by SaProt (Su et al., 2024) are AlphaFold2 structures (Jumper et al., 2021).

Method	Input	Param.	Pre-training	GO			EC	Year
			Dataset (Size)	BP	MF	CC		
ESM-1b†	Seq	650M	UniRef50 (24M)	0.470	0.657	0.488	0.864	2019
ProtBERT-BFD†	Seq	420M	BFD (2.1B)	0.279	0.456	0.408	0.838	2021
ESM-2*	Seq	650M	UniRef50 (24M)	0.472	0.662	0.472	0.874	2022
DeepFRI†	Seq, Struct	6.2M	Pfam (10M)	0.399	0.465	0.460	0.631	2021
LM-GVP†	Seq, Struct	420M	UniRef100 (216M)	0.417	0.545	0.527	0.664	2021
CRL†	Seq, Struct	36.6M	PDB (476K)	0.468	0.661	0.516	-	2022
GearNet (Multiview Contrast)†	Seq, Struct	42M	AlphaFoldDB (805K)	0.490	0.654	0.488	0.874	2023
GearNet (Residue Type)†	Seq, Struct	42M	AlphaFoldDB (805K)	0.430	0.604	0.465	0.843	2023
GearNet (Distance)†	Seq, Struct	42M	AlphaFoldDB (805K)	0.448	0.616	0.464	0.839	2023
GearNet (Angle)†	Seq, Struct	42M	AlphaFoldDB (805K)	0.458	0.625	0.473	0.853	2023
GearNet (Dihedral)†	Seq, Struct	42M	AlphaFoldDB (805K)	0.458	0.626	0.465	0.859	2023
ESM-GearNet	Seq, Struct	692M	AlphaFoldDB (805K)	0.488	0.681	0.464	0.890	2023
SaProt	Seq, Struct	650M	AlphaFoldDB (40M)	0.356	0.678	0.414	0.884	2024
ProtST-ESM-1b*	Seq, Func	759M	ProtDescribe (553K)	0.480	0.661	0.488	0.878	2023
ProtST-ESM-2*	Seq, Func	759M	ProtDescribe (553K)	0.482	0.668	0.487	0.878	2023
ProteinSSA (Student)	Seq, Struct	100M	-	0.464	0.667	0.492	0.857	2024

G. Comparison with Pre-training Methods

In the teacher-student framework, the teacher model is usually a well-learned model that serves as a source of knowledge for the student model. The student model aims to mimic the behavior or predictions of the teacher model. ProteinSSA uses annotations for the teacher model, its objective is to learn embeddings in the latent space that contain functional information and provide intermediate supervision during knowledge distillation for the student model. Therefore, the complete training of the teacher model is not our primary concern. Our main focus is to obtain comprehensive embeddings for the student model, which is trained using distillation loss and task loss without the annotations input. As we have mentioned earlier, the training of the teacher model can still be seen as training instead of pre-training because it does not involve unsupervised or self-supervised learning on a large dataset.

As discussed in Section 3.2, we highlight the limitations of pre-training and the absence of a well-learned protein functional encoder to encode functional information. While the teacher network requires extra functions as input, such information is not always available. To address these challenges and make better use of functional information without extensive pre-training, we propose ProteinSSA.

It is not fair for ProteinSSA to compare with the large-scale pre-trained models, but to show its effectiveness, we compare the proposed ProteinSSA (student) to pre-training or self-supervised learning methods: DeepFRI (Gligorijević et al., 2021), ESM-1b (Rives et al., 2019), ProtBERT-BFD (Elnaggar et al., 2021), LM-GVP (Wang et al., 2021), CRL (Hermosilla & Ropinski, 2022), GearNet (GearNet-Edge-IEConv) (Zhang et al., 2023b), ESM-2 (Lin et al., 2022), ESM-GearNet (Zhang et al., 2023a), ProtST (Xu et al., 2023) and SaProt (Su et al., 2024) on these four tasks, including protein fold classification, enzyme reaction classification, GO term prediction, and EC number prediction.

The results are shown in Table 8. Notably, the results demonstrate the competitive performance of our proposed framework, ProteinSSA, even in the absence of extensive pre-training or self-supervised learning. With fewer trainable parameters, ProteinSSA achieves accuracy levels comparable to state-of-the-art pre-training methods. Particularly noteworthy is its superiority over sequence-function based pre-training models in the task of function prediction. This underscores the significance of the hybrid framework, ProteinSSA, in adeptly capturing and modeling the intricate interplay between protein sequences, structures, and functions. Integrating ProteinSSA with other pre-training strategies may be advantageous for enhancing the model's performance. However, managing computational resources required for training and inference, and ensuring compatibility between different model architectures and training objectives. It is not our first objective to combine such pre-trained models, considering the difference in data categories and sizes of protein sequences, structures, and functions, we developed a basic network, ProteinSSA, to incorporate sequential, structural and functional information without large-scale pre-training effectively and efficiently.