PEER: A Comprehensive and Multi-Task Benchmark for Protein Sequence Understanding

Minghao Xu[1,2 *] **Zuobai Zhang**[1,2 *] **Jiarui Lu**[1,2] **Zhaocheng Zhu**[1,2]
Yangtian Zhang[3] **Chang Ma**[4] **Runcheng Liu**[5] **Jian Tang**[1,6,7 †]
[*]equal contribution [†]corresponding author
[1]Mila - Québec AI Institute [2]Université de Montréal [3]Shanghai Jiao Tong University
[4]Peking University [5]Tsinghua University [6]HEC Montréal [7]CIFAR AI Research Chair
contacts: <minghao.xu, zuobai.zhang>@mila.quebec, jian.tang@hec.ca

Abstract

We are now witnessing significant progress of deep learning methods in a variety of tasks (or datasets) of proteins. However, there is a lack of a standard benchmark to evaluate the performance of different methods, which hinders the progress of deep learning in this field. In this paper, we propose such a benchmark called **PEER**, a comprehensive and multi-task benchmark for **P**rotein s**E**quence und**ER**standing. PEER provides a set of diverse protein understanding tasks including protein function prediction, protein localization prediction, protein structure prediction, protein-protein interaction prediction, and protein-ligand interaction prediction. We evaluate different types of sequence-based methods for each task including traditional feature engineering approaches, different sequence encoding methods as well as large-scale pre-trained protein language models. In addition, we also investigate the performance of these methods under the multi-task learning setting. Experimental results show that large-scale pre-trained protein language models achieve the best performance for most individual tasks, and jointly training multiple tasks further boosts the performance. The datasets and source codes of this benchmark are all available at `https://github.com/DeepGraphLearning/PEER_Benchmark`.

1 Introduction

Proteins are working horses of life and play a critical role in many biological processes. Understanding the functions of proteins is therefore important in a variety of applications [99, 51, 9]. Thanks to the recent progress of protein sequencing [50, 49], a large number of protein sequences are available. For example, in the UniProt database [14], more than 200 million protein sequences are available. This offers a big opportunity for machine learning methods to analyze these sequences and understand the functions of proteins.

Indeed, a variety of deep learning methods have been successfully applied to different protein tasks such as protein function prediction [25, 88, 98], protein structure prediction [35, 5] and protein engineering [51, 9]. These works generally borrow techniques from the natural language processing (NLP) community to learn effective protein representations with different sequence encoders, *e.g.*, CNNs [72], LSTMs [62] and Transformers [62, 20, 64]. However, these approaches are usually evaluated on different tasks/datasets, and there lacks a standard benchmark to systematically evaluate the performance of different techniques, which hinders the progress of deep learning for protein understanding.

In this paper, we are inspired by the success of deep learning in computer vision and natural language processing community, where, to a large extent, the progresses are driven by benchmark datasets such as ImageNet [18] and GLUE [86]. Therefore, we take the initiative of building a comprehensive

36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks.

benchmark for **P**rotein s**E**quence und**ER**standing (**PEER**), to facilitate the progress of deep learning in protein understanding. The PEER benchmark includes fourteen biologically relevant tasks that cover diverse aspects of protein understanding, including protein function prediction, protein structure prediction, protein localization prediction, protein-protein interaction prediction and protein-ligand interaction prediction. In each benchmark task, the training/validation/test splits are carefully designed to evaluate the generalization ability of deep learning techniques in real-world settings. For example, in protein engineering tasks, low-order mutants of wild type sequences are used as training data while high-order mutants are used as testing data to evaluate the out-of-distribution generalization ability of deep learning approaches.

For each individual task, we evaluate the performance of different types of sequence-based approaches, including traditional feature engineering approaches, different protein sequence encoders such as CNNs, LSTMs and Transformers, and large-scale pre-trained protein language models. In addition, we also expect to develop general approaches that can perform well across different protein tasks and benefit from knowledge sharing and transferring. We therefore also evaluate different approaches under the multi-task learning setting. Experimental results show that pre-trained protein language models achieve the best performance on most individual tasks by either directly using the pre-trained protein language models as sequence encoders or fine-tuning them with task-specific supervised data. Jointly training multiple tasks can further enhance the performance, showing the potential of multi-task learning.

We maintain the leaderboards of PEER benchmark at `https://torchprotein.ai/benchmark`, where we will receive new benchmark results from the community in the near future. We hope this benchmark will serve as a good starting point and spark the interest of machine learning community in working on deep learning for protein understanding. In the future, we will further extend the current benchmark by going beyond sequence-based approaches to structure-based approaches.

2 Related Work

Protein representation learning. Protein representation learning has been studied for decades and recently drawn increasing interest from different domains. Early methods on protein representation learning define different rules to extract physiochemical or statistical features from protein sequences [42, 41, 69, 21, 87]. Advancements of deep learning and natural language processing motivate the development of protein sequence models to utilize large-scale protein sequence corpus. Early works along this line adapt the idea of word2vec [53] and doc2vec [44] to protein sequences [84, 39, 92, 52]. To increase model capacity, deeper sequence encoders originally developed by the NLP community are pre-trained on million- or billion-scale protein sequences. Well-known works include UniRep [2] and ProtXLNet [20], which are pre-trained with the next amino acid prediction task, and TAPE Transformer [62], ProtBert [20], ProtAlbert [20], ProtElectra [20], ProtT5 [20] and ESM [64] with masked language modeling task, PMLM with pairwise masked language modeling loss [28] and CPCProt with contrastive predictive coding loss [48]. Recently, there are also some works exploring protein multiple sequence alignments (MSAs) [63, 9, 51], 3D structures [30, 98] and surfaces [23, 80, 16] for learning effective protein representations. The structure-based approaches outperform sequence-based approaches on some tasks, but sequence-based approaches still dominate the performance for most tasks thanks to the large number of protein sequences. Therefore, in this benchmark, we mainly focus on sequence-based approaches.

Protein modeling benchmarks. To fairly compare different protein modeling methods, computational biologists have devoted a lot of effort to build large-scale and comprehensive benchmarks. Among them, the best-known one is the biennial Critical Assessment of protein Structure Prediction (CASP) [43], which focuses on protein structure prediction and becomes a golden-standard assessment in this area. Accompanied with CASP, the Critical Assessment of Functional Annotation (CAFA) challenge [99] is held for the evaluation of protein function prediction. To comprehensively compare different machine learning methods, the TAPE benchmark [62] is built on five tasks spread across different domains of protein biology and evaluate the performance of protein sequence encoders. FLIP [17] proposes three protein landscape benchmarks for fitness prediction evaluation. A recent work [83] focuses on the evaluation of unsupervised protein representations and evaluates 23 typical methods. TDA [34] contains protein-related datasets and tasks for drug discovery. ATOM3D [81] provides benchmark datasets for 3D structure based biomolecule understanding. Another recent work [10] builds a benchmark containing 7 downstream tasks for evaluating self-supervised protein representation learning.

Table 1: Benchmark task descriptions. Each task, along with its category, the source of dataset, the size of each split and evaluation metric are shown below. *Abbr.*, Reg.: regression; Cls.: classification; Acc: accuracy; RMSE: root-mean-square error.

Task	Task Category	Data Source	#Train	#Validation	#Test	Metric
Protein Function Prediction Tasks						
Fluorescence prediction	Protein-wise Reg.	Sarkisyan's dataset [70]	21,446	5,362	27,217	Spearman's ρ
Stability prediction	Protein-wise Reg.	Rocklin's dataset [65]	53,571	2,512	12,851	Spearman's ρ
β-lactamase activity prediction	Protein-wise Reg.	Envision [26]	4,158	520	520	Spearman's ρ
Solubility prediction	Protein-wise Cls.	DeepSol [38]	62,478	6,942	1,999	Acc
Protein Localization Prediction Tasks						
Subcellular localization prediction	Protein-wise Cls.	DeepLoc [3]	8,945	2,248	2,768	Acc
Binary localization prediction	Protein-wise Cls.	DeepLoc [3]	5,161	1,727	1,746	Acc
Protein Structure Prediction Tasks						
Contact prediction	Residue-pair Cls.	ProteinNet [4]	25,299	224	40	L/5 precision
Fold classification	Protein-wise Cls.	DeepSF [31]	12,312	736	718	Acc
Secondary structure prediction	Residue-wise Cls.	NetSurfP-2.0 [40]	8,678	2,170	513	Acc
Protein-Protein Interaction Prediction Tasks						
Yeast PPI prediction	Protein-pair Cls.	Guo's dataset [27]	1,668	131	373	Acc
Human PPI prediction	Protein-pair Cls.	Pan's dataset [57]	6,844	277	227	Acc
PPI affinity prediction	Protein-pair Reg.	SKEMPI [54]	2,127	212	343	RMSE
Protein-Ligand Interaction Prediction Tasks						
Affinity prediction on PDBbind	Protein-ligand Reg.	PDBbind [47]	16,436	937	285	RMSE
Affinity prediction on BindingDB	Protein-ligand Reg.	BindingDB [45]	7,900	878	5,230	RMSE

These previous benchmarks mainly focus on specific types of protein modeling tasks, which are insufficient to evaluate the versatility and general effectiveness of a protein encoder. In this work, we propose the comprehensive PEER benchmark, including tasks for protein function/localization/structure prediction, protein-protein interaction prediction and protein-ligand interaction prediction. Besides benchmarking under conventional single-task learning, we additionally evaluate the impact of multi-task learning on protein sequence modeling, which has shown great potential in the NLP community according to the GLUE [86] and SuperGLUE [85] benchmarks.

3 Benchmark Tasks

The PEER benchmark includes 14 benchmark tasks within 5 task groups in total. We represent a protein x as a sequence of amino acids (*a.k.a.*, residues) $x = (x_1, x_2, \cdots, x_L)$ of length L. For protein-ligand interaction prediction, we represent a ligand g as a molecular graph $g = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the atom set and the bond set, respectively. We list the task category, data source, dataset statistics and evaluation metric of each task in Tab. 1.

3.1 Protein Function Prediction

This group of tasks intend to predict functional values of proteins (either discrete or continuous). We select tasks from important protein engineering applications, and the data splits are designed to evaluate the performance of machine learning methods in real-world scenarios.

Fluorescence prediction asks the model to predict the fitness of green fluorescent protein mutants. The target $y \in \mathbb{R}$ is the logarithm of fluorescence intensity annotated by Sarkisyan *et al.* [70]. We adopt the dataset splits from TAPE [62], where the training and validation sets consist of mutants with three or less mutations, and the test set is composed of mutants with four or more mutations. This task measures the transferability of the model from training on lower-order mutants to evaluating on higher-order mutants.

Impact: Green fluorescent protein is an important marker protein, enabling scientists to see the presence of the particular protein in an organic structure by its green fluorescence [82]. This task could reveal the mutational patterns of enhancing/reducing such biological property.

Stability prediction attempts to evaluate the stability of proteins under natural environment. The target $y \in \mathbb{R}$ indicates the experimental measurement of stability. We adopt the dataset from Rocklin *et al.* [65] and follow the dataset splits of TAPE [62], where the proteins from four rounds of experimental design are used for training and validation, and the top candidates with single mutations

are used for test. This task evaluates the generalization ability by training on the data with multiple mutations and applying to discover the top candidates with single mutations.

Impact: The stability of a protein affects whether its function can be carried out in the body [73]. This benchmark task simulates the real-world application scenario of selecting functional mutants that possess decent stability.

β-lactamase activity prediction studies the activity among first-order mutants of the TEM-1 beta-lactamase protein. The target $y \in \mathbb{R}$ is the experimentally tested fitness score which records the scaled mutation effect for each mutant. The sequences of mutants along with their labels are taken from Envision [26], and the dataset split protocol follows Rives *et al.* [64]. The models with high capacity are expected to discriminate proteins with only one-residue difference in the dataset.

Impact: TEM-1 beta-lactamase is the most widespread enzyme that endows gram-negative bacteria with beta-lactam antibiotic resistance [56]. This task studies how to enhance the activity of this important enzyme by single mutations.

Solubility prediction aims to predict whether a protein is soluble or not (*i.e.*, with label $y \in \{0, 1\}$). We adopt the training, validation, and test splits from DeepSol [38], where protein sequences with sequence identity $\geqslant 30\%$ to any sequence in the test set are removed from the training set. This task evaluates a model's ability to generalize across dissimilar protein sequences.

Impact: Protein solubility plays a critical role in pharmaceutical research and industry field, since good solubility is an essential property for a functional protein [38]. This task aims to boost the development of effective in silico sequence-based protein solubility predictor.

3.2 Protein Localization Prediction

The localization of proteins is highly related to their *in vivo* functionality. This task group contains two levels of protein localization prediction.

Subcellular localization prediction expects the model to predict where a natural protein locates in the cell. For example, the proteins naturally existing in the lysosome will be attached with a categorical label *"lysosome"*. There are 10 possible localizations, inducing the label $y \in \{0, 1, \cdots, 9\}$. We adopt the training and test sets introduced in DeepLoc [3]. In DeepLoc, homologous protein sequences are clustered with 30% sequence identity and split into five folds. Four of them are used for training, and the rest one is held out for testing. We randomly split out a validation set from the training set with a 4:1 training/validation ratio. This task evaluates the ability of the model to correctly predict the subcellular localization of homologous proteins.

Impact: Acquiring the subcellular localization of a protein can greatly improve target identification during drug discovery [61]. A high-throughput and accurate prediction tool of subcellular localization can accelerate this whole process. This task facilitates the development of such a tool.

Binary localization prediction is a much simpler version of the task above, where a model is asked to coarsely classify each protein to be either "membrane-bound" or "soluble" (*i.e.*, with label $y \in \{0, 1\}$). The training and test sets are also from DeepLoc [3], where we retain the samples attached with binary localization labels. We randomly hold out a validation set from training with a 4:1 training/validation ratio. This task also evaluates the generalization across homologous proteins.

Impact: The "soluble" proteins are free molecules in the body, while the "membrane-bound" proteins may contain some catalytic activity by binding to the membrane [24]. This task boosts the efficient discrimination of these two types of proteins by machine learning techniques.

3.3 Protein Structure Prediction

The accurate prediction of protein folding structures is critical to understand their various functions. This group involves three sub-problems of general protein structure prediction.

Contact prediction estimates the contact probability of each pair of residues, where each residue pair is associated with a binary label $y \in \{0, 1\}$ indicating whether they contact (*i.e.*, within a distance threshold δ) or not. We adopt the ProteinNet dataset [4] for this task. Following TAPE [62], we use the ProteinNet CASP12 test set for evaluation, which is filtered against the training set at a 30% sequence identity. According to the standard of CASP [55], we report the precision of the L/5 most likely contacts for medium- and long-range contacts on the test set. Such evaluation measures the ability of a contact prediction model on predicting the folded structures of diverse protein sequences.

Impact: The prediction of amino acid contacts from protein sequence is a crucial step towards the prediction of folded protein structures [8]. The evaluation of this task pays particular attention to medium- and long-range contacts for their critical roles in protein folding.

Fold classification classifies the global structural topology of a protein on the fold level, represented as a categorical label $y \in \{0, 1, \cdots, 1194\}$. The label is determined by the backbone coordinates of the corresponding protein structure. We adopt training, validation, and test sets from Hou's dataset [31], originally derived from the SCOP 1.75 database [22]. All proteins of a given fold class are further categorized into related *superfamilies*. Entire superfamilies are held out from training to compose the test set, allowing us to evaluate the ability of the model to detect the proteins with similar structures but dissimilar sequences, *i.e.*, performing remote homology detection [62].

Impact: Protein fold classification is important for both functional analysis and drug design [11]. The SCOPe database [22] only categorizes a small portion of proteins in PDB [7]. This task aims to empower automatic fold classification from protein sequences by machine learning.

Secondary structure prediction predicts the local structures of protein residues in their natural state. A secondary structure label $y \in \{0, 1, 2\}$ (*i.e.*, coil, strand or helix) is assigned to each residue. We adopt the training set from Klausen's dataset [40], which is filtered such that no two proteins have greater than 25% sequence identity. For test, we use the CB513 dataset [15], and it is filtered at 25% sequence identity against training to evaluate the generalization across dissimilar protein sequences.

Impact: The accurate prediction of protein secondary structure is useful in multiple aspects, *e.g.*, protein function understanding [40] and multiple sequence alignment [74]. This benchmark task approaches such a goal by enabling the training and generalization test of machine learning models.

3.4 Protein-Protein Interaction Prediction

Protein-protein interaction (PPI) prediction is crucial for protein complex structure modeling and protein function understanding. This group involves three PPI prediction tasks.

Yeast PPI prediction predicts whether two yeast proteins interact or not (*i.e.*, with a binary label $y \in \{0, 1\}$). We use Guo's yeast PPI dataset [27], where negative pairs are from different subcellular locations. For dataset split, we first remove redundancy within all protein sequences in the dataset with a 90% sequence identity cut-off, and then randomly split these filtered sequences into training/validation/test splits. After that, we remove the redundancy between each split pair with a 40% sequence identity cut-off. The generalization across dissimilar protein sequences are thus evaluated.

Impact: It is of broad scientific interest to construct complete and precise yeast interactome network maps [95, 60, 6]. The benchmark task will aid to this project by predicting binary yeast protein interactions with machine learning models.

Human PPI prediction predicts whether two human proteins interact or not (*i.e.*, with a binary label $y \in \{0, 1\}$). We adopt Pan's human PPI dataset [57] that contains positive protein pairs from Human Protein Reference Database (HPRD) [59] and negative pairs from different subcellular locations. The dataset splitting scheme follows that of yeast PPI prediction, except that a 8:1:1 train/validation/test ratio is adopted here. This task also evaluates generalization across dissimilar protein sequences.

Impact: Unraveling the human protein interactome is vital to understand mechanisms of disease and uncover unknown disease genes, motivating many projects [67, 96, 66]. This benchmark task is expected to contribute by boosting effective machine learning models for human PPI prediction.

PPI affinity prediction estimates the binding affinity $y \in \mathbb{R}$ measured by pK_d between two proteins. We utilize the SKEMPI dataset [54] and split it according to the number of mutations. In specific, the training set consists of wild-type complexes as well as the mutants with at most 2 mutations; the validation set consists of the mutants with 3 or 4 mutations; the test set is composed of the mutants with more than 4 mutations. Therefore, this task evaluates model's generalization ability under a multi-round protein binder design scenario.

Impact: Predicting the relative binding strength among candidate binders is important for protein binder design [46, 71]. This task provides a test field for machine learning models in such a real-world application.

3.5 Protein-Ligand Interaction Prediction

Protein-ligand interaction (PLI) prediction seeks to model the interaction strength between pairs of *protein* and *ligand*. We involve two tasks from different data sources here. Both tasks aim to estimate the binding affinity $y \in \mathbb{R}$ measured by pK_d.

Table 2: Baseline model descriptions. *Abbr.*, Params.: parameters; feats.: features; dim.: dimension; attn.: attention; conv.: convolutional. "-" indicates a nonexistent component for a model.

Model	Model Type	Input Layer	Hidden Layers	Output Layer	#Params.
Feature Engineer					
DDE [69]	MLP	400-dim. statistical feats.	linear (hidden dim.:512) + ReLU	-	205.3K
Moran [21]	MLP	240-dim. physicochemical feats.	linear (hidden dim.:512) + ReLU	-	123.4K
Protein Sequence Encoder					
LSTM [62]	LSTM	640-dim. token embedding (21 entries)	3 × bidirectional LSTM layers (hidden dim.: 640)	weighted sum over all residues + linear (output dim.: 640) + Tanh	26.7M
Transformer [62]	Transformer	512-dim. embedding (24 entries)	4 × Transformer blocks (hidden dim.: 512; #attn. heads: 8; activation: GELU)	linear (output dim.: 512) + Tanh upon `[CLS]` token	21.3M
CNN [72]	CNN	21-dim. one-hot residue type	2 × 1D conv. layers (hidden dim.: 1024; kernel size: 5; stride: 1; padding: 2)	max pooling over all residues	5.4M
ResNet [62]	CNN	512-dim. token embedding (21 entries) + 512-dim. positional embedding	8 × residual blocks (hidden dim.: 512; kernel size: 3; stride: 1; padding: 1)	attentive weighted sum over all residues	11.0M
Pre-trained Protein Language Model					
ProtBert [20]	Transformer	1024-dim. token embedding (30 entries) + 1024-dim. positional embedding	30 × Transformer blocks (hidden dim.: 1024; #attn. heads: 16; activation: GELU)	linear (output dim.: 1024) + Tanh upon `[CLS]` token	419.9M
ESM-1b [64]	Transformer	1280-dim. token embedding (33 entries)	33 × Transformer blocks (hidden dim.: 1280; #attn. heads: 20; activation: GELU)	mean pooling over all residues	652.4M

PLI prediction on PDBbind adopts the PDBbind-2019 dataset [47]. We choose the test set according to the CASF-2016 benchmark [77] to evaluate model generalization. To avoid redundancy, we first remove training sequences against test ones with a 90% sequence identity cut-off, and then cluster the training sequences and randomly split the clusters into training/validation splits with a 9:1 ratio. Note that, we use only the refined-set in PDBbind for better binding affinity data quality.

Impact: The recognition of the interactions between small molecules and target proteins is a prominent research topic in the field of drug discovery [93, 90]. This benchmark task seeks to assess the ability of machine learning models to accomplish such a goal.

PLI prediction on BindingDB adopts the BindingDB dataset [45]. We follow the dataset splitting scheme in DeepAffinity [36], where 4 protein classes (ER, GPCR, ion channels and receptor tyrosine kinases) are held out from training and validation for generalization test.

Impact: Similar as the task above, this task is attractive to the drug discovery community, and it focuses on the evaluation of ligands' interactions with four specific classes of proteins.

4 Methods

4.1 Baselines

We consider three types of baseline models in our benchmark, *i.e.,* feature engineers, protein sequence encoders and pre-trained protein language models. We summarize each model along with its model type, input layer, hidden layers, output layer and number of parameters in Tab. 2.

Feature engineers. We adopt two typical protein sequence feature descriptors, *i.e.*, Dipeptide Deviation from Expected Mean (DDE) [69] and Moran correlation (Moran) [21]. The DDE feature descriptor (400 dimensions) is based on the dipeptide frequency within the protein sequence, and it is centralized and normalized by the theoretical mean and variance computed by amino acid codons. The Moran feature descriptor (240 dimensions) defines the distribution of amino acid properties along a protein sequence. Following iFeature [12], we retrieve 8 physicochemical properties $\{P^k\}_{k=1}^8$ from AAindex Database [37] to construct the Moran feature descriptor. Upon these two kinds of feature vectors, we use a nonlinear MLP to project them to the latent space for task prediction. More details about these two feature engineers are provided in Appendix A.

Protein sequence encoders. We employ four broadly-studied protein sequence encoders, *i.e.*, LSTM [62], Transformer [62], shallow CNN [72] and ResNet [62]. These encoders aim to capture the short-range interactions (shallow CNN and ResNet) and long-range interactions (LSTM and Transformer) within the protein sequence. The final output layer aggregates the representations of different residues into a protein-level representation for task prediction.

Pre-trained protein language models. We also evaluates two representative protein language models pre-trained on large-scale protein sequence corpuses, *i.e.*, ProtBert [20] and ESM-1b [64]. Both models are huge Transformer encoders exceeding the size of BERT-Large [19]. ProtBert is pre-trained on 2.1 billion protein sequences from the BFD database [76] with the masked language

modeling (MLM) objective, and ESM-1b is pre-trained on 24 million protein sequences from UniRef50 [79] by MLM. We study two evaluation settings in our benchmark by either (1) learning the prediction head with protein language model parameters frozen or (2) fine-tuning the protein language model along with the prediction head.

4.2 Model Pipeline

In general, we utilize three model pipelines to solve different types of tasks in the PEER benchmark.

Protein function, localization and structure prediction tasks learn a function $y = f_\theta(x)$ that maps protein x to the label y, where f_θ is parameterized by a sequence-based encoder and an MLP predictor. The predictor is applied upon residue-level embeddings for secondary structure prediction, upon the embeddings of residue pairs for contact prediction and upon protein-level embeddings for all other tasks in these three groups.

Protein-protein interaction prediction tasks learn a function $y = f_\theta(x, x')$ that maps a pair of proteins (x, x') to the label y, where f_θ is parameterized by a pair of siamese sequence-based encoders and an MLP predictor defined upon the concatenation of the embeddings of two proteins.

Protein-ligand interaction prediction tasks learn a function $y = f_\theta(x, g)$ that maps a protein-ligand pair (x, g) to the label y, where f_θ is parameterized by a protein sequence encoder, a ligand graph encoder and an MLP predictor defined upon the concatenated protein-ligand embedding.

4.3 Single-Task Learning *vs* Multi-Task Learning

Single-Task Learning. A simple and common strategy to solve these protein-related tasks is to train a model exclusively on one task of interest at a time, known as single-task learning. Given a task $t \in \mathcal{T}$ from the pool $\mathcal{T}$ of benchmark tasks, a task-specific loss function $\mathcal{L}_t$ is defined to measure the correctness of model predictions on training samples against ground truth labels. The objective of learning this single task is to optimize model parameters to minimize the loss $\mathcal{L}_t$ on this task.

Multi-Task Learning. We further study multi-task learning as a way to enhance the generalization ability of the model. In our benchmark, the sizes of the training sets on different tasks could be significantly different. To have a fair comparison with single-task learning under comparable training budget, we focus on the multi-task learning setting with *a center task* and *an auxiliary task*. Specifically, given a center task t_c with loss $\mathcal{L}_{t_c}$ and an auxiliary tasks t_a with loss $\mathcal{L}_{t_a}$, we seek to boost the performance on the center task by leveraging the knowledge learned from the auxiliary task. To achieve this goal, we build our model f_θ under the regime of hard parameter sharing [68], in which we employ a single protein sequence encoder shared across all tasks, and a ligand graph encoder is additionally involved in protein-ligand interaction prediction tasks. During training, the model parameters are optimized by the joint loss of center and auxiliary tasks: $\mathcal{L} = \mathcal{L}_{t_c} + \alpha\mathcal{L}_{t_a}$, where α is the tradeoff parameter balancing two objectives. The number of training iterations is identical to that of single-task learning on center task, and we only test on center task to ensure fair comparison.

5 Experiments

5.1 Experimental Setups

Dependencies. The codebase of PEER benchmark is developed based on PyTorch [58] (BSD License) and TorchProtein platform, an extension of TorchDrug [100] (Apache License 2.0) specific to protein applications.

Model setups. For protein function, localization and structure prediction tasks, given the protein embedding, we apply a 2-layer MLP with a ReLU nonlinearity in between to perform prediction. For protein-protein interaction prediction tasks, upon the concatenation of the embeddings of two proteins, a 2-layer MLP activated by ReLU serves as the predictor. For protein-ligand interaction prediction tasks, we involve an additional Graph Isomorphism Network (GIN) [91] with 4 layers and 256 hidden dimensions as the ligand graph encoder, which follows previous practices [33, 78]. Based on the concatenation of protein and ligand embedding, a 2-layer MLP activated by ReLU is applied for prediction. For each atom of an input ligand, we construct its input features by concatenating the one-hot embeddings of atomic symbol, atomic chirality tag, atom degree, formal charge, number of

Table 3: Benchmark results on single-task learning. We report *mean (std)* for each experiment. We use four color scales of green to denote the first, second, third and fourth best performance among all models; SOTA model performance from literature is in gray; "-" indicates a non-applicable setting.

Task	Feature Engineer		Protein Sequence Encoder				Pre-trained Protein Language Model				Literature SOTA
	DDE	Moran	LSTM	Transformer	CNN	ResNet	ProtBert	ProtBert*	ESM-1b	ESM-1b*	
Function Prediction											
Flu	0.638$_{(0.003)}$	0.400$_{(0.001)}$	0.494$_{(0.071)}$	0.643$_{(0.005)}$	0.682$_{(0.002)}$	0.636$_{(0.021)}$	0.679$_{(0.001)}$	0.339$_{(0.003)}$	0.679$_{(0.002)}$	0.430$_{(0.002)}$	0.69 (Shallow CNN [72])
Sta	0.652$_{(0.033)}$	0.322$_{(0.011)}$	0.533$_{(0.101)}$	0.649$_{(0.056)}$	0.637$_{(0.010)}$	0.126$_{(0.094)}$	0.771$_{(0.020)}$	0.697$_{(0.013)}$	0.694$_{(0.073)}$	0.750$_{(0.010)}$	0.79 (Evoformer [32])
β-lac	0.623$_{(0.019)}$	0.375$_{(0.008)}$	0.139$_{(0.051)}$	0.261$_{(0.015)}$	0.781$_{(0.011)}$	0.152$_{(0.029)}$	0.731$_{(0.226)}$	0.616$_{(0.002)}$	0.839$_{(0.053)}$	0.528$_{(0.009)}$	0.89 (ESM-1b [72])
Sol	59.77$_{(1.21)}$	57.73$_{(1.33)}$	70.18$_{(0.63)}$	70.12$_{(0.31)}$	64.43$_{(0.25)}$	67.33$_{(1.46)}$	68.15$_{(0.92)}$	59.17$_{(0.21)}$	70.23$_{(0.75)}$	67.02$_{(0.40)}$	77.0 (DeepSol [38])
Localization Prediction											
Sub	49.17$_{(0.40)}$	31.13$_{(0.47)}$	62.98$_{(0.37)}$	56.02$_{(0.82)}$	58.73$_{(1.05)}$	52.30$_{(3.51)}$	76.53$_{(0.93)}$	59.44$_{(0.16)}$	78.13$_{(0.49)}$	79.82$_{(0.18)}$	86.0 (LA-ProtT5 [75])
Bin	77.43$_{(0.42)}$	55.63$_{(0.85)}$	88.11$_{(0.14)}$	75.74$_{(0.74)}$	82.67$_{(0.32)}$	78.99$_{(4.41)}$	91.32$_{(0.89)}$	81.54$_{(0.09)}$	92.40$_{(0.35)}$	91.61$_{(0.10)}$	92.34 (DeepLoc [3])
Structure Prediction											
Cont	-	-	26.34$_{(0.65)}$	17.50$_{(0.77)}$	10.00$_{(0.20)}$	20.43$_{(0.74)}$	39.66$_{(1.21)}$	24.35$_{(0.44)}$	45.78$_{(2.73)}$	40.37$_{(0.22)}$	82.1 (MSA Transformer [63])
Fold	9.57$_{(0.46)}$	7.10$_{(0.56)}$	8.24$_{(1.61)}$	8.52$_{(0.63)}$	10.93$_{(0.35)}$	8.89$_{(1.45)}$	16.94$_{(0.42)}$	10.74$_{(0.93)}$	28.17$_{(2.05)}$	29.95$_{(0.21)}$	56.5 (GearNet-Edge [98])
SSP	-	-	68.99$_{(0.76)}$	59.62$_{(0.94)}$	66.07$_{(0.06)}$	69.56$_{(0.20)}$	82.18$_{(0.05)}$	62.51$_{(0.06)}$	82.73$_{(0.21)}$	83.14$_{(0.10)}$	86.41 (DML_SS$^{\text{embed}}$ [94])
Protein-Protein Interaction Prediction											
Yst	55.83$_{(3.13)}$	53.00$_{(0.50)}$	53.62$_{(2.72)}$	54.12$_{(1.27)}$	55.07$_{(0.02)}$	48.91$_{(1.78)}$	63.72$_{(2.80)}$	53.87$_{(0.38)}$	57.00$_{(6.38)}$	66.07$_{(0.58)}$	-
Hum	62.77$_{(2.30)}$	54.67$_{(4.43)}$	63.75$_{(5.12)}$	59.58$_{(2.09)}$	62.60$_{(1.67)}$	68.61$_{(3.78)}$	77.32$_{(1.10)}$	83.61$_{(1.34)}$	78.17$_{(2.91)}$	88.06$_{(0.24)}$	-
Aff	2.908$_{(0.043)}$	2.984$_{(0.026)}$	2.853$_{(0.124)}$	2.499$_{(0.156)}$	2.796$_{(0.071)}$	3.005$_{(0.244)}$	2.195$_{(0.073)}$	2.996$_{(0.462)}$	2.281$_{(0.250)}$	3.031$_{(0.014)}$	-
Protein-Ligand Interaction Prediction											
PDB	-	-	1.457$_{(0.131)}$	1.455$_{(0.070)}$	1.376$_{(0.008)}$	1.441$_{(0.064)}$	1.562$_{(0.072)}$	1.457$_{(0.024)}$	1.559$_{(0.164)}$	1.368$_{(0.076)}$	1.181 (SS-GNN [97])
BDB	-	-	1.572$_{(0.022)}$	1.566$_{(0.052)}$	1.497$_{(0.022)}$	1.565$_{(0.033)}$	1.549$_{(0.019)}$	1.649$_{(0.022)}$	1.556$_{(0.047)}$	1.571$_{(0.032)}$	1.34 (DeepAffinity [36])

* Used as a feature extractor with pre-trained weights frozen.

hydrogens on the atom, number of radical electrons on the atom, the atom's hybridization, binary aromatic flag and binary within-a-ring flag. These one-hot embeddings form 66-dimensional atom features as the input node features of GIN.

Training setups. We train all models with three seeds (0, 1 and 2) on each task and report the mean and standard deviation of three runs' results. For each run, we train with an Adam optimizer for 50 epochs on contact and human PPI prediction and for 100 epochs on other tasks. We perform 10 times of validation uniformly along the training process, and the test performance of the best validation epoch model is reported. For each model, we search for its learning rate among $[1 \times 10^{-5}, 2 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}, 2 \times 10^{-4}]$ and its batch size among $[1, 2, 4, 8, 16, 32, 64, 128, 256]$ based on the validation performance on the β-lactamase activity prediction task, and the searched parameters are applied to all tasks for that model. For fine-tuning ProtBert and ESM-1b, we set their learning rate as one-tenth of that of the MLP predictor. Unless specified, the tradeoff parameter α for multi-task learning is set as 1.0. The fluorescence, stability, β-lactamase activity, PPI affinity, PDBbind and BindingDB prediction tasks are trained with mean squared error; the solubility, subcellular localization, binary localization, fold, secondary structure, yeast PPI and human PPI prediction tasks are trained with cross entropy loss; the contact prediction task is trained with binary cross entropy loss. For all models except for ESM-1b, we use the full protein sequence as the input on all tasks. Because of the intrinsic limit on input sequence length, ESM-1b truncates those sequences with more than 1022 residues by keeping the first 1022 residues. All experiments are conducted on 4 Tesla V100 GPUs (32GB). In Appendix B, we further compare different models on two class-imbalanced tasks, *i.e.*, subcellular localization prediction and fold classification, using the balanced metric weighted F1 [1]. We provide ablation studies for important hyperparameters in Appendix C.

5.2 Benchmark Results on Single-Task Learning

In Tab. 3, we report the benchmark results for single-task learning, in which we evaluate ten models in three model groups. Based on these results, we have following findings:

- **Statistical features are informative for protein sequence understanding.** On all available tasks, the 2-gram based DDE featurization significantly outperforms Moran, the physicochemical featurization scheme. Such results demonstrate that the statistical characteristics of protein sequence segments can effectively reveal some biological properties of proteins.
- **Among models trained from scratch, shallow CNN is the best model.** Among four protein sequence encoders trained from scratch, the shallow CNN performs best on 6 out of 14 tasks and achieves competitive performance on others. These results align with the finding in Shanehsaz-

Table 4: Benchmark results on multi-task learning. We report *mean (std)* for each experiment. Red results outperform the *original* single-task learning baseline; gray results are same as the baseline; blue results underperform the baseline; **bold results** are best for the specific model; "-" indicates not applicable for this setting.

Task	CNN				Transformer				ESM-1b			
	Ori.	+Cont	+Fold	+SSP	Ori.	+Cont	+Fold	+SSP	Ori.	+Cont	+Fold	+SSP
Function Prediction												
Flu	0.682$_{(0.002)}$	0.680$_{(0.001)}$	0.682$_{(0.001)}$	**0.683**$_{(0.001)}$	0.643$_{(0.005)}$	0.642$_{(0.017)}$	0.648$_{(0.004)}$	**0.656**$_{(0.002)}$	0.678$_{(0.001)}$	**0.681**$_{(0.001)}$	0.679$_{(0.001)}$	**0.681**$_{(0.002)}$
Sta	0.637$_{(0.010)}$	0.661$_{(0.006)}$	0.472$_{(0.170)}$	**0.695**$_{(0.016)}$	0.649$_{(0.056)}$	0.620$_{(0.004)}$	**0.672**$_{(0.010)}$	0.667$_{(0.063)}$	0.694$_{(0.073)}$	0.733$_{(0.007)}$	0.728$_{(0.002)}$	**0.759**$_{(0.002)}$
β-lac	0.781$_{(0.011)}$	**0.835**$_{(0.009)}$	0.736$_{(0.012)}$	0.811$_{(0.014)}$	0.261$_{(0.015)}$	0.142$_{(0.063)}$	**0.276**$_{(0.029)}$	0.197$_{(0.017)}$	0.839$_{(0.053)}$	**0.899**$_{(0.001)}$	0.882$_{(0.007)}$	0.881$_{(0.001)}$
Sol	64.43$_{(0.25)}$	**70.63**$_{(0.34)}$	69.23$_{(0.10)}$	69.85$_{(0.62)}$	**70.11**$_{(0.30)}$	70.03$_{(0.42)}$	68.85$_{(0.43)}$	69.81$_{(0.46)}$	70.23$_{(0.75)}$	**70.46**$_{(0.16)}$	64.80$_{(0.49)}$	70.03$_{(0.15)}$
Localization Prediction												
Sub	58.73$_{(1.05)}$	**59.07**$_{(0.45)}$	56.54$_{(0.65)}$	56.64$_{(0.33)}$	56.01$_{(0.81)}$	52.92$_{(0.64)}$	**56.74**$_{(0.29)}$	56.70$_{(0.16)}$	78.13$_{(0.49)}$	**78.86**$_{(0.75)}$	78.43$_{(0.28)}$	78.00$_{(0.34)}$
Bin	**82.67**$_{(0.32)}$	82.67$_{(0.72)}$	81.14$_{(0.40)}$	81.83$_{(0.86)}$	75.74$_{(0.74)}$	74.98$_{(0.77)}$	**76.27**$_{(0.57)}$	75.20$_{(1.23)}$	92.40$_{(0.34)}$	**92.50**$_{(0.26)}$	91.83$_{(0.20)}$	92.26$_{(0.20)}$
Structure Prediction												
Cont	**10.00**$_{(0.20)}$	-	5.87$_{(0.21)}$	5.73$_{(0.66)}$	**17.50**$_{(0.77)}$	-	2.04$_{(0.31)}$	12.76$_{(1.62)}$	**45.78**$_{(2.72)}$	-	35.86$_{(1.27)}$	32.03$_{(12.2)}$
Fold	10.93$_{(0.35)}$	11.07$_{(0.38)}$	-	**11.67**$_{(0.56)}$	8.62$_{(0.62)}$	**9.16**$_{(0.91)}$	-	8.14$_{(0.76)}$	28.10$_{(2.05)}$	**32.10**$_{(0.72)}$	-	28.63$_{(1.55)}$
SSP	66.07$_{(0.06)}$	**66.13**$_{(0.06)}$	65.93$_{(0.04)}$	-	59.62$_{(0.94)}$	**63.10**$_{(0.43)}$	50.93$_{(0.20)}$	-	82.73$_{(0.20)}$	**83.21**$_{(0.32)}$	82.27$_{(0.23)}$	-
Protein-Protein Interaction Prediction												
Yst	**55.07**$_{(1.68)}$	54.50$_{(1.61)}$	53.28$_{(1.91)}$	54.12$_{(2.87)}$	**54.12**$_{(1.26)}$	52.86$_{(1.15)}$	54.00$_{(2.58)}$	54.00$_{(1.17)}$	57.00$_{(6.37)}$	58.50$_{(2.15)}$	**64.76**$_{(1.42)}$	62.06$_{(5.98)}$
Hum	62.60$_{(1.67)}$	65.10$_{(2.26)}$	**69.03**$_{(2.68)}$	66.39$_{(0.86)}$	59.58$_{(2.08)}$	60.76$_{(6.87)}$	**67.33**$_{(2.68)}$	54.80$_{(2.06)}$	78.16$_{(2.90)}$	81.66$_{(2.88)}$	80.28$_{(1.27)}$	**83.00**$_{(0.88)}$
Aff	2.796$_{(0.071)}$	**1.732**$_{(0.044)}$	2.392$_{(0.041)}$	2.270$_{(0.041)}$	**2.499**$_{(0.156)}$	2.733$_{(0.126)}$	2.524$_{(0.146)}$	2.651$_{(0.034)}$	2.280$_{(0.249)}$	**1.893**$_{(0.064)}$	2.002$_{(0.065)}$	2.031$_{(0.031)}$
Protein-Ligand Interaction Prediction												
PDB	1.376$_{(0.008)}$	1.328$_{(0.033)}$	1.316$_{(0.064)}$	**1.295**$_{(0.030)}$	1.455$_{(0.069)}$	1.574$_{(0.215)}$	1.531$_{(0.181)}$	**1.387**$_{(0.019)}$	1.559$_{(0.164)}$	1.458$_{(0.003)}$	1.435$_{(0.015)}$	**1.419**$_{(0.026)}$
BDB	1.497$_{(0.022)}$	1.501$_{(0.035)}$	**1.462**$_{(0.044)}$	1.481$_{(0.036)}$	1.566$_{(0.051)}$	1.490$_{(0.058)}$	**1.464**$_{(0.007)}$	1.519$_{(0.050)}$	1.556$_{(0.047)}$	1.490$_{(0.033)}$	1.511$_{(0.017)}$	**1.482**$_{(0.014)}$

zadeh *et al.* [72] that, when trained from scratch, a shallow CNN model is no worse or even better than deeper models like LSTM, Transformer and ResNet.

- **ESM-1b is a superior model for various benchmarks.** ESM-1b obtains the best performance on 10 out of 14 benchmark tasks, and it performs consistently well either as a feature extractor or fine-tuned along with the predictor. These results verify ESM-1b as a superior protein language model that captures rich evolutionary and biological patterns underlying protein sequences.

5.3 Benchmark Results on Multi-Task Learning

In Tab. 4, we report the benchmark results for multi-task learning. Following the principle that "protein structures determine their functions" [29], we employ three structure prediction tasks, *i.e.*, contact prediction, fold classification and secondary structure prediction, as the auxiliary task, and we evaluate the effect of training each of these tasks along with the center task. We perform multi-task learning on three models with different capacities, *i.e.*, shallow CNN, Transformer and ESM-1b. According to experimental results, we have following findings:

- **MTL well benefits shallow CNN.** By using contact prediction or secondary structure prediction as the auxiliary task, the performance of CNN is improved on 9 out of 13 applicable tasks over the single-task learning baseline. Therefore, these two tasks can be deemed as the suitable auxiliary task for CNN. By comparison, the MTL assisted by fold classification is less benefical to CNN, which leads to 7 degenerated results.
- **MTL least benefits the Transformer trained from scratch.** Compared to CNN and ESM-1b, the Transformer model achieves least performance gain from MTL. The most effective auxiliary task for Transformer is fold classification, which only improves 7 out of 13 applicable tasks. Contact prediction and secondary structure prediction leads to 9 and 8 degenerated results, respectively.
- **MTL best benefits ESM-1b.** The ESM-1b model is broadly benefited by applying all the three auxiliary tasks. In particular, contact prediction succeeds in improving over the single-task learning baseline on all the 13 applicable tasks. Fold classification and secondary structure prediction are also effective, which both achieve 9 increased results.
- **Contact prediction is easily dominated by auxiliary task.** Under all three considered models, the performance of contact prediction degenerates significantly when trained along with the auxiliary task. This phenomenon indicates that the training objective of contact prediction is more vulnerable than those of other benchmark tasks.

In summary, MTL has great potential on boosting the performance of CNN and ESM-1b, while negative effects show on the Transformer trained from scratch. We thus suggest CNN and ESM-1b as baseline models for future MTL research. In addition, suitable auxiliary tasks could vary across models, requiring an extra auxiliary task selection scheme. We suggest it as a future direction.

6 Conclusions and Future Work

In this work, we build a comprehensive benchmark for general protein sequence understanding, named as PEER. Upon this benchmark, we compare the performance of both single- and multi-task learning among three types of models. The benchmark results for single-task learning demonstrate the shallow CNN as a decent baseline model which owns competitive performance and low computational cost, and ESM-1b is verified as a superior protein language model. The benchmark results for multi-task learning demonstrate the effectiveness of MTL on boosting the performance of CNN and ESM-1b, and they also illustrate the importance of selecting suitable auxiliary tasks.

The current PEER benchmark could be strengthened by adding more tasks and models. Therefore, in the future, we will continue extending our benchmark with more tasks and datasets (*e.g.*, functional terms prediction on Gene Ontology [13] and reaction classification based on Enzyme Commission number [89]), and we will evaluate more baseline models (*e.g.*, ProtT5 [20] and UniRep [2]) on our benchmark. Also, we will continue promoting the efforts on MTL for protein sequence understanding, and we would like to collaborate with the community to push the edge of this research topic.

7 Broader Societal Impacts

This work focuses on building a comprehensive and multi-task benchmark for protein sequence understanding. In this benchmark, five types of protein understanding tasks are leveraged to evaluate the general effectiveness of protein sequence encoding methods. By evaluating on the proposed benchmark, we can comprehensively assess whether a protein sequence encoding approach could be promising in various real-world applications. Therefore, this benchmark lays a solid foundation for the application of machine learning techniques on pharmaceutical research.

However, it cannot be denied that some harmful activities could be boosted by the powerful models validated by our benchmark, *e.g.*, designing harmful drugs. Therefore, our future works will seek to mitigate these issues by formulating guidelines for the responsible usage of our benchmark.

Acknowledgments

This project is supported by AIHN IBM-MILA partnership program, the Natural Sciences and Engineering Research Council (NSERC) Discovery Grant, the Canada CIFAR AI Chair Program, collaboration grants between Microsoft Research and Mila, Samsung Electronics Co., Ltd., Amazon Faculty Research Award, Tencent AI Lab Rhino-Bird Gift Fund, a NRC Collaborative R&D Project (AI4D-CORE-06) as well as the IVADO Fundamental Research Project grant PRF-2019-3583139727.

The authors would like to thank Meng Qu, Shengchao Liu, Chence Shi, Minkai Xu, Huiyu Cai and Hannes Stärk for their helpful discussions and comments.

References

[1] sklearn.metrics.f1 score. `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html`.

[2] Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, 16(12):1315–1322, 2019.

[3] José Juan Almagro Armenteros, Casper Kaae Sønderby, Søren Kaae Sønderby, Henrik Nielsen, and Ole Winther. Deeploc: prediction of protein subcellular localization using deep learning. *Bioinformatics*, 33(21):3387–3395, 2017.

[4] Mohammed AlQuraishi. Proteinnet: a standardized data set for machine learning of protein structure. *BMC bioinformatics*, 20(1):1–10, 2019.

[5] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

[6] Anastasia Baryshnikova, Michael Costanzo, Yungil Kim, Huiming Ding, Judice Koh, Kiana Toufighi, Ji-Young Youn, Jiongwen Ou, Bryan-Joseph San Luis, Sunayan Bandyopadhyay, et al. Quantitative analysis of fitness and genetic interactions in yeast on a genome scale. *Nature methods*, 7(12):1017–1024, 2010.

[7] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[8] Wendy M Billings, Connor J Morris, and Dennis Della Corte. The whole is greater than its parts: ensembling improves protein contact prediction. *Scientific Reports*, 11(1):1–7, 2021.

[9] Surojit Biswas, Grigory Khimulya, Ethan C Alley, Kevin M Esvelt, and George M Church. Low-n protein engineering with data-efficient deep learning. *Nature Methods*, 18(4):389–396, 2021.

[10] Henriette Capel, Robin Weiler, Maurits JJ Dijkstra, Reinier Vleugels, Peter Bloem, and K Anton Feenstra. Proteinglue: A multi-task benchmark suite for self-supervised protein modeling. *bioRxiv*, 2021.

[11] Daozheng Chen, Xiaoyu Tian, Bo Zhou, and Jun Gao. Profold: Protein fold classification with additional structural features and a novel ensemble classifier. *BioMed research international*, 2016, 2016.

[12] Zhen Chen, Pei Zhao, Fuyi Li, André Leier, Tatiana T Marquez-Lago, Yanan Wang, Geoffrey I Webb, A Ian Smith, Roger J Daly, Kuo-Chen Chou, et al. ifeature: a python package and web server for features extraction and selection from protein and peptide sequences. *Bioinformatics*, 34(14):2499–2502, 2018.

[13] Gene Ontology Consortium. The gene ontology (go) database and informatics resource. *Nucleic acids research*, 32(suppl_1):D258–D261, 2004.

[14] UniProt Consortium. Uniprot: A worldwide hub of protein knowledge. *Nucleic Acids Research*, 47:D506–D515, 2019.

[15] James A Cuff and Geoffrey J Barton. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics*, 34(4):508–519, 1999.

[16] Bowen Dai and Chris Bailey-Kellogg. Protein interaction interface region prediction by geometric deep learning. *Bioinformatics*, 2021.

[17] Christian Dallago, Jody Mou, Kadina E Johnston, Bruce J Wittmann, Nicholas Bhattacharya, Samuel Goldman, Ali Madani, and Kevin K Yang. Flip: Benchmark tasks in fitness landscape inference for proteins. *bioRxiv*, 2021.

[18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[20] Ahmed Elnaggar *et al.* 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*, pages 1–1, 2021.

[21] Zhi-Ping Feng and Chun-Ting Zhang. Prediction of membrane protein types based on the hydrophobic index of amino acids. *Journal of Protein Chemistry*, 19(4):269–275, 2000.

[22] Naomi K Fox, Steven E Brenner, and John-Marc Chandonia. Scope: Structural classification of proteins—extended, integrating scop and astral data and classification of new structures. *Nucleic Acids Research*, 42(D1):D304–D309, 2014.

[23] Pablo Gainza, Freyr Sverrisson, Frederico Monti, Emanuele Rodola, D Boscaini, MM Bronstein, and BE Correia. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2):184–192, 2020.

[24] Marina Gimpelev, Lucy R Forrest, Diana Murray, and Barry Honig. Helical packing patterns in membrane and soluble proteins. *Biophysical journal*, 87(6):4075–4086, 2004.

[25] Vladimir Gligorijević, P Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C Taylor, Ian M Fisk, Hera Vlamakis, et al. Structure-based protein function prediction using graph convolutional networks. *Nature Communications*, 12(1):1–14, 2021.

[26] Vanessa E Gray, Ronald J Hause, Jens Luebeck, Jay Shendure, and Douglas M Fowler. Quantitative missense variant effect prediction using large-scale mutagenesis data. *Cell Systems*, 6(1):116–124, 2018.

[27] Yanzhi Guo, Lezheng Yu, Zhining Wen, and Menglong Li. Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences. *Nucleic Acids Research*, 36(9):3025–3030, 2008.

[28] Liang He, Shizhuo Zhang, Lijun Wu, Huanhuan Xia, Fusong Ju, He Zhang, Siyuan Liu, Yingce Xia, Jianwei Zhu, Pan Deng, et al. Pre-training co-evolutionary protein representation via a pairwise masked language model. *arXiv preprint arXiv:2110.15527*, 2021.

[29] Hedi Hegyi and Mark Gerstein. The relationship between protein structure and function: a comprehensive survey with application to the yeast genome. *Journal of molecular biology*, 288(1):147–164, 1999.

[30] Pedro Hermosilla and Timo Ropinski. Contrastive representation learning for 3d protein structures. In *Submitted to The Tenth International Conference on Learning Representations*, 2022.

[31] Jie Hou, Badri Adhikari, and Jianlin Cheng. Deepsf: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, 34(8):1295–1303, 2018.

[32] Mingyang Hu, Fajie Yuan, Kevin K Yang, Fusong Ju, Jin Su, Hui Wang, Fei Yang, and Qiuyang Ding. Exploring evolution-based &-free protein language models as protein function predictors. *arXiv preprint arXiv:2206.06583*, 2022.

[33] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.

[34] Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *arXiv preprint arXiv:2102.09548*, 2021.

[35] John Jumper *et al.* Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[36] Mostafa Karimi, Di Wu, Zhangyang Wang, and Yang Shen. Deepaffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks. *Bioinformatics*, 35(18):3329–3338, 2019.

[37] Shuichi Kawashima and Minoru Kanehisa. Aaindex: amino acid index database. *Nucleic Acids Research*, 28(1):374–374, 2000.

[38] Sameer Khurana, Reda Rawi, Khalid Kunji, Gwo-Yu Chuang, Halima Bensmail, and Raghvendra Mall. Deepsol: a deep learning framework for sequence-based protein solubility prediction. *Bioinformatics*, 34(15):2605–2613, 2018.

[39] Dhananjay Kimothi, Akshay Soni, Pravesh Biyani, and James M Hogan. Distributed representations for biological sequence analysis. *arXiv preprint arXiv:1608.05949*, 2016.

[40] Michael Schantz Klausen, Martin Closter Jespersen, Henrik Nielsen, Kamilla Kjaergaard Jensen, Vanessa Isabell Jurtz, Casper Kaae Soenderby, Morten Otto Alexander Sommer, Ole Winther, Morten Nielsen, Bent Petersen, et al. Netsurfp-2.0: Improved prediction of protein structural features by integrated deep learning. *Proteins: Structure, Function, and Bioinformatics*, 87(6):520–527, 2019.

[41] Petr Klein, John A Jacquez, and Charles Delisi. Prediction of protein function by discriminant analysis. *Mathematical biosciences*, 81(2):177–189, 1986.

[42] Petr Klein, Minoru Kanehisa, and Charles DeLisi. The detection and classification of membrane-spanning proteins. *Biochimica et Biophysica Acta (BBA)-Biomembranes*, 815(3):468–476, 1985.

[43] Andriy Kryshtafovych, Torsten Schwede, Maya Topf, Krzysztof Fidelis, and John Moult. Critical assessment of methods of protein structure prediction (casp)—round xiv. *Proteins: Structure, Function, and Bioinformatics*, 89(12):1607–1617, 2021.

[44] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR, 2014.

[45] Tiqing Liu, Yuhmei Lin, Xin Wen, Robert N Jorissen, and Michael K Gilson. Bindingdb: a web-accessible database of experimentally determined protein–ligand binding affinities. *Nucleic Acids Research*, 35(suppl_1):D198–D201, 2007.

[46] Xianggen Liu, Yunan Luo, Pengyong Li, Sen Song, and Jian Peng. Deep geometric representations for modeling effects of mutations on protein-protein binding affinity. *PLoS computational biology*, 17(8):e1009284, 2021.

[47] Zhihai Liu *et al.* Forging the basis for developing protein-ligand interaction scoring functions. *Accounts of Chemical Research*, 50(2):302–309, 2017.

[48] Amy X Lu, Haoran Zhang, Marzyeh Ghassemi, and Alan Moses. Self-supervised contrastive learning of protein representations by mutual information maximization. *BioRxiv*, 2020.

[49] Bin Ma. Novor: real-time peptide de novo sequencing software. *Journal of the American Society for Mass Spectrometry*, 26(11):1885–1894, 2015.

[50] Bin Ma and Richard Johnson. De novo sequencing and homology searching. *Molecular & cellular proteomics*, 11(2), 2012.

[51] Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alexander Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *bioRxiv*, 2021.

[52] María Katherine Mejía-Guerra and Edward S Buckler. A k-mer grammar analysis to uncover maize regulatory architecture. *BMC plant biology*, 19(1):1–17, 2019.

[53] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[54] Iain H Moal *et al.* Skempi: a structural kinetic and energetic database of mutant protein interactions and its use in empirical models. *Bioinformatics*, 28(20):2600–2607, 2012.

[55] John Moult, Krzysztof Fidelis, Andriy Kryshtafovych, Torsten Schwede, and Anna Tramontano. Critical assessment of methods of protein structure prediction (casp)—round xii. *Proteins: Structure, Function, and Bioinformatics*, 86:7–15, 2018.

[56] TIMOTHY Palzkill and DAVID Botstein. Identification of amino acid substitutions that alter the substrate specificity of tem-1 beta-lactamase. *Journal of bacteriology*, 174(16):5237–5243, 1992.

[57] Xiao-Yong Pan, Ya-Nan Zhang, and Hong-Bin Shen. Large-scale prediction of human protein-protein interactions from amino acid sequence based on latent topic features. *Journal of Proteome Research*, 9(10):4992–5001, 2010.

[58] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NeurIPS Workshop*, 2017.

[59] Suraj Peri, J Daniel Navarro, Ramars Amanchy, Troels Z Kristiansen, Chandra Kiran Jonnalagadda, Vineeth Surendranath, Vidya Niranjan, Babylakshmi Muthusamy, TKB Gandhi, Mads Gronborg, et al. Development of human protein reference database as an initial platform for approaching systems biology in humans. *Genome Research*, 13(10):2363–2371, 2003.

[60] Shuye Pu, Jessica Wong, Brian Turner, Emerson Cho, and Shoshana J Wodak. Up-to-date catalogues of yeast protein complexes. *Nucleic acids research*, 37(3):825–831, 2009.

[61] Asha Rajagopal and Sanford M Simon. Subcellular localization and activity of multidrug resistance proteins. *Molecular biology of the cell*, 14(8):3389–3399, 2003.

[62] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S Song. Evaluating protein transfer learning with tape. *Advances in Neural Information Processing Systems*, 32:9689, 2019.

[63] Roshan Rao, Jason Liu, Robert Verkuil, Joshua Meier, John F Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. *bioRxiv*, 2021.

[64] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15), 2021.

[65] Gabriel J Rocklin *et al.* Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science*, 357(6347):168–175, 2017.

[66] Thomas Rolland, Murat Taşan, Benoit Charloteaux, Samuel J Pevzner, Quan Zhong, Nidhi Sahni, Song Yi, Irma Lemmens, Celia Fontanillo, Roberto Mosca, et al. A proteome-scale map of the human interactome network. *Cell*, 159(5):1212–1226, 2014.

[67] Jean-François Rual, Kavitha Venkatesan, Tong Hao, Tomoko Hirozane-Kishikawa, Amélie Dricot, Ning Li, Gabriel F Berriz, Francis D Gibbons, Matija Dreze, Nono Ayivi-Guedehoussou, et al. Towards a proteome-scale map of the human protein–protein interaction network. *Nature*, 437(7062):1173–1178, 2005.

[68] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[69] Vijayakumar Saravanan and Namasivayam Gautham. Harnessing computational biology for exact linear b-cell epitope prediction: a novel amino acid composition-based feature descriptor. *Omics: A Journal of Integrative Biology*, 19(10):648–658, 2015.

[70] Karen S Sarkisyan, Dmitry A Bolotin, Margarita V Meer, Dinara R Usmanova, Alexander S Mishin, George V Sharonov, Dmitry N Ivankov, Nina G Bozhanova, Mikhail S Baranov, Onuralp Soylemez, et al. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603):397–401, 2016.

[71] Sisi Shan, Shitong Luo, Ziqing Yang, Junxian Hong, Yufeng Su, Fan Ding, Lili Fu, Chenyu Li, Peng Chen, Jianzhu Ma, et al. Deep learning guided optimization of human antibody against sars-cov-2 variants with broad neutralization. *Proceedings of the National Academy of Sciences*, 119(11):e2122954119, 2022.

[72] Amir Shanehsazzadeh *et al.* Is transfer learning necessary for protein landscape prediction? *arXiv preprint arXiv:2011.03443*, 2020.

[73] Brian K Shoichet, Walter A Baase, Ryota Kuroki, and Brian W Matthews. A relationship between protein stability and protein function. *Proceedings of the National Academy of Sciences*, 92(2):452–456, 1995.

[74] VA Simossis and J Heringa. Integrating protein secondary structure prediction and multiple sequence alignment. *Current Protein and Peptide Science*, 5(4):249–266, 2004.

[75] Hannes Stärk, Christian Dallago, Michael Heinzinger, and Burkhard Rost. Light attention predicts protein location from the language of life. *Bioinformatics Advances*, 1(1):vbab035, 2021.

[76] Martin Steinegger and Johannes Söding. Clustering huge protein sequence sets in linear time. *Nature Communications*, 9(1):1–8, 2018.

[77] Minyi Su, Qifan Yang, Yu Du, Guoqin Feng, Zhihai Liu, Yan Li, and Renxiao Wang. Comparative assessment of scoring functions: the casf-2016 update. *Journal of Chemical Information and Modeling*, 59(2):895–913, 2018.

[78] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. *arXiv preprint arXiv:1908.01000*, 2019.

[79] Baris E Suzek, Hongzhan Huang, Peter McGarvey, Raja Mazumder, and Cathy H Wu. Uniref: comprehensive and non-redundant uniprot reference clusters. *Bioinformatics*, 23(10):1282–1288, 2007.

[80] Freyr Sverrisson, Jean Feydy, Bruno E Correia, and Michael M Bronstein. Fast end-to-end learning on protein surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15272–15281, 2021.

[81] Raphael JL Townshend, Martin Vögele, Patricia Suriana, Alexander Derry, Alexander Powers, Yianni Laloudakis, Sidhika Balachandar, Bowen Jing, Brandon Anderson, Stephan Eismann, et al. Atom3d: Tasks on molecules in three dimensions. *arXiv preprint arXiv:2012.04035*, 2020.

[82] Roger Y Tsien. The green fluorescent protein. *Annual review of biochemistry*, 67(1):509–544, 1998.

[83] Serbulent Unsal, Heval Atas, Muammer Albayrak, Kemal Turhan, Aybar C Acar, and Tunca Doğan. Learning functional properties of proteins with language models. *Nature Machine Intelligence*, 4(3):227–245, 2022.

[84] Fangping Wan and Jianyang Michael Zeng. Deep learning with feature embedding for compound-protein interaction prediction. *bioRxiv*, page 086033, 2016.

[85] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in Neural Information Processing Systems*, 32, 2019.

[86] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[87] Jiawei Wang, Bingjiao Yang, Jerico Revote, Andre Leier, Tatiana T Marquez-Lago, Geoffrey Webb, Jiangning Song, Kuo-Chen Chou, and Trevor Lithgow. Possum: a bioinformatics toolkit for generating numerical sequence feature descriptors based on pssm profiles. *Bioinformatics*, 33(17):2756–2758, 2017.

[88] Zichen Wang, Steven A Combs, Ryan Brand, Miguel Romero Calvo, Panpan Xu, George Price, Nataliya Golovach, Emannuel O Salawu, Colby J Wise, Sri Priya Ponnapalli, et al. Lm-gvp: A generalizable deep learning framework for protein property prediction from sequence and structure. *bioRxiv*, 2021.

[89] Oren F Webb, Tommy J Phelps, Paul R Bienkowski, Philip M Digrazia, David C White, and Gary S Sayler. Enzyme nomenclature. 1992.

[90] Ming Wen, Zhimin Zhang, Shaoyu Niu, Haozhi Sha, Ruihan Yang, Yonghuan Yun, and Hongmei Lu. Deep-learning-based drug-target interaction prediction. *Journal of proteome research*, 16(4):1401–1409, 2017.

[91] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

[92] Ying Xu, Jiangning Song, Campbell Wilson, and James C Whisstock. Phoscontext2vec: a distributed representation of residue-level sequence contexts and its application to general and kinase-specific phosphorylation site prediction. *Scientific reports*, 8(1):1–14, 2018.

[93] Yoshihiro Yamanishi, Masaaki Kotera, Minoru Kanehisa, and Susumu Goto. Drug-target interaction prediction from chemical, genomic and pharmacological data in an integrated framework. *Bioinformatics*, 26(12):i246–i254, 2010.

[94] Wei Yang, Yang Liu, and Chunjing Xiao. Deep metric learning for accurate protein secondary structure prediction. *Knowledge-Based Systems*, 242:108356, 2022.

[95] Haiyuan Yu, Pascal Braun, Muhammed A Yıldırım, Irma Lemmens, Kavitha Venkatesan, Julie Sahalie, Tomoko Hirozane-Kishikawa, Fana Gebreab, Na Li, Nicolas Simonis, et al. High-quality binary protein interaction map of the yeast interactome network. *Science*, 322(5898):104–110, 2008.

[96] Haiyuan Yu, Leah Tardivo, Stanley Tam, Evan Weiner, Fana Gebreab, Changyu Fan, Nenad Svrzikapa, Tomoko Hirozane-Kishikawa, Edward Rietman, Xinping Yang, et al. Next-generation sequencing to generate interactome datasets. *Nature methods*, 8(6):478–480, 2011.

[97] Shuke Zhang, Yanzhao Jin, Tianmeng Liu, Qi Wang, Zhaohui Zhang, Shuliang Zhao, and Bo Shan. Ss-gnn: A simple-structured graph neural network for affinity prediction. *arXiv preprint arXiv:2206.07015*, 2022.

[98] Zuobai Zhang, Minghao Xu, Arian Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. *arXiv preprint arXiv:2203.06125*, 2022.

[99] Naihui Zhou, Yuxiang Jiang, Timothy R Bergquist, Alexandra J Lee, Balint Z Kacsoh, Alex W Crocker, Kimberley A Lewis, George Georghiou, Huy N Nguyen, Md Nafiz Hamid, et al. The cafa challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome biology*, 20(1):1–23, 2019.

[100] Zhaocheng Zhu, Chence Shi, Zuobai Zhang, Shengchao Liu, Minghao Xu, Xinyu Yuan, Yangtian Zhang, Junkun Chen, Huiyu Cai, Jiarui Lu, et al. Torchdrug: A powerful and flexible machine learning platform for drug discovery. *arXiv preprint arXiv:2202.08320*, 2022.

A Baseline Model Details

Dipeptide Deviation from Expected Mean (DDE) [69]. The DDE protein sequence feature vector is defined by the statistical features of dipeptides, *i.e.*, two consecutive amino acids in the protein sequence. The 400-dimensional feature vector corresponds to the feature of 400 different types of dipeptides. For example, the feature of dipeptide "*st*" is defined by its dipeptide composition (D_c), theoretical mean (T_m) and theoretical variance (T_v) as below:

$$D_c(s,t) = \frac{N_{st}}{N-1}, \quad T_m(s,t) = \frac{C_s C_t}{C_N^2}, \quad T_v(s,t) = \frac{T_m(s,t)\big(1 - T_m(s,t)\big)}{N-1}, \tag{1}$$

$$\mathrm{DDE}(s,t) = \frac{D_c(s,t) - T_m(s,t)}{\sqrt{T_v(s,t)}}, \tag{2}$$

where N_{st} is the number of dipeptide "*st*" occurring in the protein sequence, N denotes the protein sequence length, C_s and C_t are the number of codons for amino acid s and t, and $C_N = 61$ is the total number of possible codons, excluding three stop codons.

Moran correlation [21]. The Moran feature descriptor defines the distribution of amino acid properties along a protein sequence. Following iFeature [12], we retrieve 8 physicochemical properties $\{P^k\}_{k=1}^8$ from AAindex Database [37] to construct the Moran feature vector, and each property is centralized and normalized before calculation. The Moran feature vector is with $8M$ dimensions (M is the parameter of maximum lag, setting as 30 following iFeature). The feature for the k-th property with lag m is defined as below ($1 \leqslant k \leqslant 8$, $1 \leqslant m \leqslant M$):

$$\mathrm{Moran}(k,m) = \frac{\frac{1}{N-m}\sum_{i=1}^{N-m}(P_i^k - \bar{P^k})(P_{i+m}^k - \bar{P^k})}{\frac{1}{N}\sum_{i=1}^{N}(P_i^k - \bar{P^k})^2}, \tag{3}$$

where N denotes the protein sequence length, and $\bar{P^k} = \frac{1}{N}\sum_{i=1}^{N} P_i^k$ is the average of property k over the whole sequence.

B More Benchmark Results

B.1 Balanced Metrics on Classification Tasks

Table 5: Balanced metric (weighted F1) compared with accuracy on multi-class classification tasks. We report *mean (std)* for each experiment. We adopt four color scales of green to denote the first, second, third and fourth best performance among all models.

Task	Feature Engineer		Protein Sequence Encoder				Pre-trained Protein Language Model			
	DDE	Moran	LSTM	Transformer	CNN	ResNet	ProtBert	ProtBert*	ESM-1b	ESM-1b*
Fold (weighted F1)	$0.093_{(0.003)}$	$0.030_{(0.003)}$	$0.070_{(0.014)}$	$0.079_{(0.005)}$	$0.120_{(0.011)}$	$0.091_{(0.017)}$	$0.185_{(0.004)}$	$0.069_{(0.005)}$	$0.298_{(0.011)}$	$0.278_{(0.008)}$
Fold (accuracy)	$0.096_{(0.005)}$	$0.071_{(0.006)}$	$0.082_{(0.016)}$	$0.085_{(0.006)}$	$0.109_{(0.004)}$	$0.089_{(0.015)}$	$0.169_{(0.004)}$	$0.107_{(0.009)}$	$0.282_{(0.021)}$	$0.300_{(0.002)}$
Sub (weighted F1)	$0.485_{(0.002)}$	$0.225_{(0.010)}$	$0.624_{(0.009)}$	$0.538_{(0.009)}$	$0.557_{(0.003)}$	$0.515_{(0.005)}$	$0.765_{(0.009)}$	$0.569_{(0.006)}$	$0.778_{(0.005)}$	$0.792_{(0.002)}$
Sub (accuracy)	$0.492_{(0.004)}$	$0.311_{(0.005)}$	$0.630_{(0.004)}$	$0.560_{(0.008)}$	$0.587_{(0.011)}$	$0.523_{(0.035)}$	$0.765_{(0.009)}$	$0.594_{(0.002)}$	$0.781_{(0.005)}$	$0.798_{(0.002)}$

* Used as a feature extractor with pre-trained weights frozen.

It should be noted that there are evident class imbalances in two multi-class classification tasks. In particular, on fold classification, the three smallest classes of training, validation and test splits all contain only one sample, while the largest ones contain tens or hundreds of samples; on subcellular location prediction, the ratio between the number of samples in the largest and smallest classes is greater than 10. Hence, these two tasks are highly imbalanced. We do not observe class imbalances in other classification tasks.

To reflect the ability of models on these imbalanced settings, we report the results of all baselines with a widely used metric to consider data imbalance, weighted F1 [1]. Weighted F1 is defined by first calculating F1 scores on each class and then taking the weighted average according to the number of instances in each class. The results are shown in Tab. 5. The ranking of baselines under weighted F1 is almost unchanged compared to that under accuracy, where shallow CNN is still the best model among models trained from scratch, and ESM-1b remains the SOTA model on these

two tasks. Therefore, the conclusions in Section 5.2 of the main paper still hold. Considering the consistency of experimental conclusions and the comparability with previous benchmark results in the literature where accuracy is commonly reported, we still employ accuracy as the metric for these two tasks in the main paper and provide weight F1 performance in the supplement.

C Ablation Studies

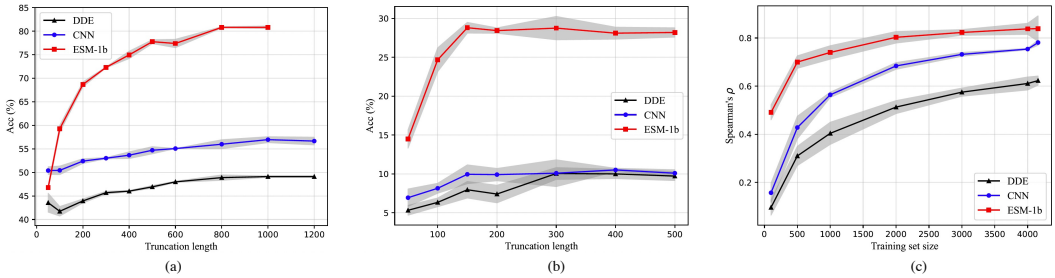


Figure 1: (a) Effect of truncation length on subcellular localization prediction. (b) Effect of truncation length on fold classification. (c) Effect of training set size on β-lactamase activity prediction. All results are averaged over three runs (seeds: 0, 1, 2); the standard deviation is shown by error bar.

C.1 Effect of Truncation Length

In Fig. 1 (a) and (b), we plot the performance of DDE, CNN and ESM-1b on subcellular localization prediction and fold classification under different sequence truncation lengths. The truncation is performed from the start of each protein sequence. It is observed that longer truncation lengths will lead to better performance for all models on both tasks, which matches with the intuition that a longer truncated sequence can contain more information about a protein and thus learn more effective prediction model.

C.2 Effect of Training Set Size

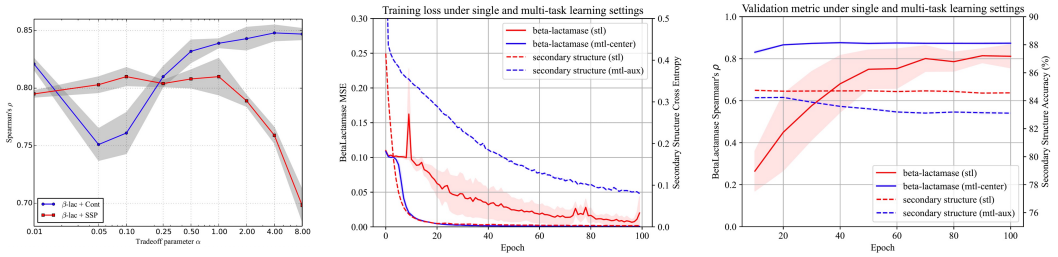
In Fig. 1 (c), we plot the performance of DDE, CNN and ESM-1b on β-lactamase activity prediction under the training set sizes from 100 to 4,158 (the full training set). Training samples are randomly sampled from the full set in required cases. As expected, the performance of all three models monotonously increases as the increase of training set size. These results illustrate the benefit of collecting more labeled data when predicting the fitness landscape of proteins.

C.3 Effect of Tradeoff Parameter

In Fig. 2 (a), we study how the tradeoff parameter α affects the MTL of CNN. When contact prediction serves as the auxiliary task, the center task, β-lactamase activity prediction, is well enhanced by using a larger tradeoff parameter (*i.e.*, $0.5 \leqslant \alpha \leqslant 8.0$). By comparison, when using secondary structure prediction as the auxiliary task, the center task suffers from severe performance decay under large tradeoff parameters, and the peak performance is achieved when α is between 0.1 and 1.0. Both cases suggest $\alpha = 1.0$ as a configuration that achieves stable performance gain, which can be used as a good candidate configuration for MTL.

C.4 Training Center and Auxiliary Tasks under Single- and Multi-Task Learning

In our benchmark experiments, the tradeoff parameter α is generally set to 1. Therefore, it is interesting to see how the optimization of center and auxiliary tasks under the multi-task setting differ from that under the single-task setting. In Fig. 2 (b) and (c), we draw the training loss curves and validation metric curves of β-lacatamase and secondary structure prediction under single- and multi-task learning settings. For multi-task learning, we choose the β-lacatamase as the center task. It can be observed that the model converges faster and generalizes better on the center task under the multi-task setting, which shows the optimization of the center task benefits from the auxiliary task. However, the training of the auxiliary task is worse under multi-task learning. This can be understood



(a) (b) (c)

Figure 2: (a) Effect of tradeoff parameter α on the MTL of CNN. (b,c) Training loss curves (b) and validation metric curves (c) on β-lacatamase and secondary structure prediction under single and multi-task settings. All results are averaged over three runs (seeds: 0, 1, 2); the standard deviation is shown by error bar.

since the number of training iterations is determined by the center task, which leads to the insufficient sampling of the auxiliary dataset. Interestingly, we find that the variance of β-lacatamase is largely decreased (almost stable under different seeds) when including secondary structure prediction as the auxiliary task. One potential reason is that the low-variance auxiliary task makes the training of the center task more stable.