# PROTEINADAPTER: ADAPTING PRE-TRAINED LARGE PROTEIN MODELS FOR EFFICIENT PROTEIN REPRESENTATION LEARNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

The study of proteins is crucial in various scientific disciplines, but understanding their intricate multi-level relationships remains challenging. Inspired by the sequence and structure understanding of Large Protein Models (LPMs), we introduce a new Mamba-based **ProteinAdapter**, to efficiently transfer the broad knowledge encapsulated in multiple LPMs, *e.g.*, ESM-1b and ESM-IF, to task-specific insights. ProteinAdapter could largely save labor-intensive analysis on the 3D position and the amino acid order. Specifically, (1) with a modest number of additional parameters, ProteinAdapter facilitates multi-level protein representation learning by integrating both sequence and geometric structure embeddings from LPMs; (2) based on the learned embedding, we further scale up the proposed ProteinAdapter to various tasks with a unified Multi-Scale Predictor, which optimally harnesses the learned embeddings through task-specific attention. Albeit simple, the proposed method is scalable to multiple downstream tasks without bells and whistles. Extensive experiments on over 20 tasks show that ProteinAdapter outperforms state-of-the-art methods under both single-task and multi-task scenarios.

## 1 INTRODUCTION

Proteins serve as vital constituents and functional units for life, underscoring the significance of protein research for life sciences. As understanding their intricate structure-function relationships remains costly and time-consuming, there is an urgent need to develop a discriminative protein representation for enhancing various computational biological analyses. Recently, Large Protein Models (LPMs) have been verified as superior representation learners from different structure levels, containing 1D Protein Language Models (PLMs) (Rives et al., 2021; Meier et al., 2021; Lin et al., 2022) and 3D Protein Structure Models (PSMs) (Hsu et al., 2022; Zhang et al., 2023b;a). These works motivate us to ride on the coattails of LPMs, seize the windfalls of multi-level proteomic knowledge, and explore multi-level protein embeddings for efficient protein representation learning. However, considering the complex characteristics of proteins, there remain two key challenges: (1) *multi-level complementarity*, and (2) *multi-scale integration*. **On the one hand**, for various protein tasks, different levels of structure, such as amino acid sequences (primary structure) and geometric coordinates (tertiary structure) exhibit complementarity. Namely, for the protein-protein interaction prediction task, 1D sequence information can be used to predict potential protein partners, while 3D coordinate information can further help clarify the specific details of these interactions. **On the other hand**, a protein can encompass multiple smaller substructures that are meaningful at different scales. Namely, for the protein function classification task, the sequence lengths of different functional regions or domains exhibit variability, ranging from a few amino acid residues to hundreds.

In an attempt to address both limitations, we propose a new Mamba-based multi-level adapter, called **ProteinAdapter**, to take advantage of existing pre-trained large models for efficient protein representation learning. Our ProteinAdapter explicitly captures the interrelations and complementarity among multi-level protein representations. Specifically, (1) to facilitate *multi-level interrelations*, the proposed ProteinAdapter directly takes the intermediate feature from pretrained ESM-1b (Rives et al., 2021) and ESM-IF1 (Hsu et al., 2022) as inputs, and outputs a multi-level mixed representation embedding containing knowledge from both 1D and 3D structure levels. ProteinAdapter consists of a pre-alignment module and stacked Mamba Fusion Block (MFBlock), in which the mamba-based

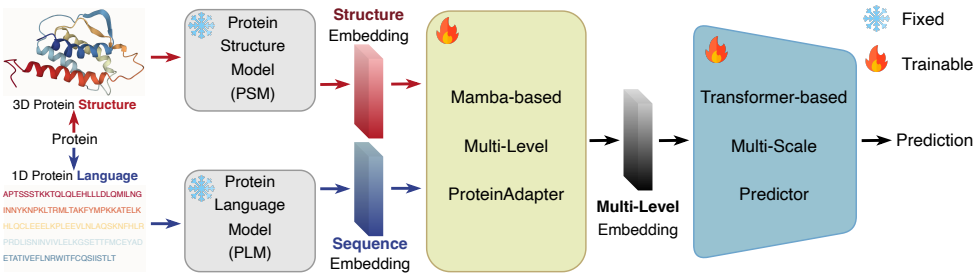


Figure 1: Overview of the proposed framework. The primary 1D sequence and the tertiary 3D structure are first individually fed into two Large Protein Models (LPMs) to obtain their corresponding embeddings. The proposed ProteinAdapter is to fuse the complementary embeddings from both levels to obtain the multi-level embedding. Finally, the merged multi-level embedding is fed into a multi-scale predictor to further take both local and global protein properties into consideration.

pre-align module effectively captures long-range features from single-level protein embeddings while the MFBlock aims to combine complementary information from multi-level protein features. (2) Aiming at *multi-scale integration*, as a minor contribution, we further design a multi-scale predictor for various downstream tasks. Specifically, the predictor adopts a hierarchical pyramid structure that dynamically adjusts the weights between different sizes to ensure a comprehensive understanding of the mixed representation. Furthermore, different from conventional multi-task methods using single-scale generic representations, our multi-scale predictor naturally enables applicability in multi-task scenarios. Our contributions can be summarized as follows:

- A Mamba-based multi-level adapter, dubbed **ProteinAdapter**, is proposed for parameter-efficient fine-tuning on pre-trained large protein models, which effectively merges protein embeddings from different structure levels.
- A multi-scale hierarchical predictor is further designed to utilize the merged multi-level protein representation for various protein tasks, which fully integrate and utilize the protein embeddings across different scales.
- Due to the use of frozen protein models and the lightweight adapter, our method is more compute-efficient than existing state-of-the-arts. Extensive experiments on over 20 tasks show that the proposed method surpasses previous methods by large margins. Furthermore, our ProteinAdapter trained in a multi-task setting still consistently outperforms existing counterparts on most tasks, indicating the ability of protein representation and generalization.

## 2 Related Work

**Protein Representation Learning.** Proteins exhibit multi-level structures. Existing protein representation methods mainly focus on the 1D primary and the 3D tertiary structures understanding. For the primary structure, regarding protein sequences as the language of life, many Protein Language Models (PLMs) (Bepler & Berger, 2019; Strodthoff et al., 2020; Vig et al., 2020; Rives et al., 2021; Amidi et al., 2018; Elnaggar et al., 2023) have been proposed for sequence-based protein representation learning with large-scale protein sequence corpora. For the tertiary structure, several Protein Structure Models (PSMs) (Hsu et al., 2022; Fan et al., 2022; 2023; Hermosilla et al., 2020; Zhang et al., 2023b) propose to extract features directly from the geometric information of amino acids or atoms. Since each structure of the protein has its own merit and driving forces in describing specific characteristics, several works (Wang et al., 2023; Zhang et al., 2023a; Fan et al., 2022; 2023) have been proposed to explicitly model the complementary information between different levels. However, due to the computation limits, these methods usually process adjacent amino acids within limited neighboring graph nodes (Zhang et al., 2023b; Sun et al., 2022) or small convolution kernels (Fan et al., 2022; 2023) with stacked downsampling to expand the receptive field for global perception. The former local propagation pattern with a fixed small region limits the perception field for protein functional regions. The latter downsampling operations with fixed length also break the relation between neighbor functional regions in proteins. Moreover, these methods still require pretraining

from scratch on large protein datasets, followed by fine-tuning for deployment to each subtask. In this work, we directly use the power of previous well-trained Large Protein Models (LPMs) to acquire discriminative protein embeddings. Different from these methods, our ProteinAdapter leverages the trend of large models by directly utilizing two single-level models focused on different structures (*i.e.*, ESM-1b (Rives et al., 2021) for 1D sequence and ESM-IF1 (Hsu et al., 2022) for 3D coordinates), and achieves multi-level protein representations through an efficient Mamba-based adapter.

**State Space Model (SSM).** State Space Models (SSM) (Gu et al., 2021a;b; Smith et al., 2022), originating from classical control theory, have become practical components for constructing deep networks due to their cutting-edge performance in analyzing long sequential data. Structured State Space Sequence Models (S4) (Gu et al., 2021a) introduces the concept of normal plus low-rank, thereby effectively reducing the computational complexity associated with the SSM. Subsequently, S5 (Smith et al., 2022) and H3 (Fu et al., 2022) further introduce a parallel scan on a diagonalized linear SSM, narrowing the performance gap between the SSM and Transformer. Mamba (Gu & Dao, 2023) further proposes a selection mechanism for dynamically extracting features from input data, which outperforms Transformer on various 1D datasets while requiring significantly fewer computational resources. However, the methods discussed above primarily focus on Mamba's application and directionality, leaving the potential of the SSM in protein representation largely unexplored. Recently, ProteinMamba (Xu et al., 2024) introduces a Mamba-based two-stage model for protein representation, yet the performance is severely limited due to only considering the sequence information, and it still requires time-consuming pretraining on large-scale datasets. In contrast, our ProteinAdapter is able to leverage both primary and tertiary protein information and can be efficiently deployed to various downstream tasks with a minimal number of parameters.

**Parameter Efficient Fine-Tuning (PEFT).** To utilize the rich evolutionary and biological patterns from these pretrained LPMs, ESM-GearNet (Zhang et al., 2023a) makes the first attempt by replacing the original graph node with the well-learned 1D sequence embedding produced by ESM-1b (Rives et al., 2021). However, this method mainly focuses on the pretraining of a complex local structure encoder from scratch. To fully unleash and efficiently utilize the power of the off-the-shelf large models, Parameter-Efficient Fine-Tuning (PEFT) methods (Li & Liang, 2021; Lester et al., 2021; Liu et al., 2022; Houlsby et al., 2019; Hu et al., 2021) have prevailed recently in both Natural Language Processing (NLP) and Computer Vision (CV) communities. Existing PEFT methods can be roughly divided into three parts: Prefix-tuning (Li & Liang, 2021; Lester et al., 2021; Liu et al., 2022), Adapter-tuning (Houlsby et al., 2019), and LoRA (Hu et al., 2021). In this paper, we resort to the adapter by adding only a few trainable parameters for different downstream protein tasks, while the parameters of the original LPMs are fixed. To the best of our knowledge, this is the first adapter-based work exploring the parameter-efficient fine-tuning of pre-trained Large Protein Models (LPMs).

## 3 METHODS

Aiming at efficient utilization of off-the-shelf pre-trained models for protein representation learning, there are two key properties that differentiate protein language from natural language: *multi-level* and *multi-scale*. First, a protein can target various structural levels to carry out its functions. Each level possesses distinct advantages and underlying factors in elucidating particular attributes. Consequently, it is necessary to consider both 1D and 3D structures for comprehensive protein representation. Second, due to their distinct biological functions, the scale of functional regions in different proteins typically varies. Likewise, due to genetic differences, the scale of the same functional regions is often inconsistent across different species (Gabaldón, 2008). Thus the predictor should possess multi-scale perceptual capabilities to fully leverage the protein representation for various downstream tasks. Considering these two properties, as shown in Figure 1, our method consists of three key components: pre-trained protein models, a multi-level ProteinAdapter, and a multi-scale predictor.

### 3.1 ACQUIRING MULTI-LEVEL PROTEIN EMBEDDINGS WITH PRE-TRAINED MODELS

Recently, large Protein Language Models (PLMs) (Vig et al., 2020; Rao et al., 2020; Rives et al., 2021) have demonstrated strong capabilities in understanding protein sequences, which encourages us to leverage pre-trained sequence embeddings with rich information In our approach, we use a powerful PLM, ESM-1b (Rives et al., 2021), as our 1D protein encoder, which takes protein sequences as input and outputs the sequence embedding $E_{seq}$.

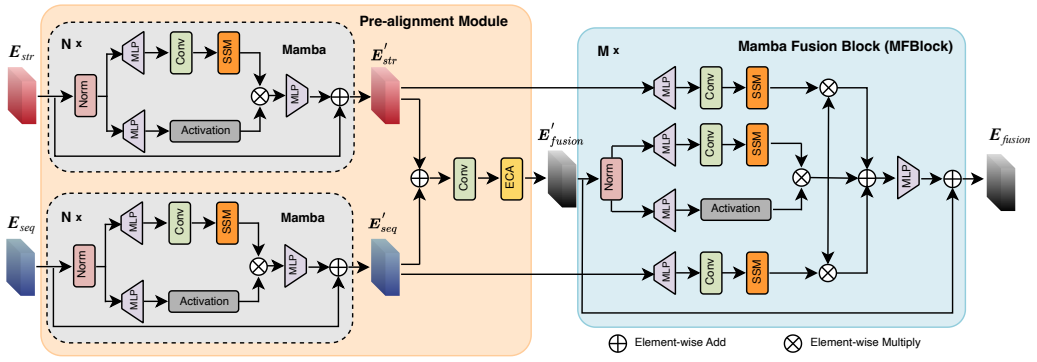


Figure 2: Architecture of the ProteinAdapter, which consists of a pre-alignment module and multiple MFBlocks. The sequence embedding $E_{seq}$ and structure embedding $E_{str}$ from pretrained LPMs first undergo processing through their respective Mamba blocks, after which they are summed to obtain the initial fused embedding. The processed single-level embedding $E^{'}_{str}$, $E^{'}_{str}$, and the preliminary multi-level embedding $E^{'}_{fusion}$ are then jointly fed into a series of MFBlocks for deep integration.

However, considering that PLMs do not directly incorporate protein structures as input, they are limited in capturing intricate structural features. Given the importance of protein structures in determining functions, we adopt another Protein Structure Model (PSM), ESM-IF1(Hsu et al., 2022), as the 3D structure encoder. As a multi-level complement to the sequence embedding, the structure embedding $E_{str}$ obtained with ESM-IF1 effectively encapsulates geometric information on proteins within sequence embedding $E_{seq}$.

### 3.2 Integrating Sequence-Structure Features with ProteinAdapter

Despite the superiority of existing PLMs and PSMs in sequence and structure understanding, they are individually trained on single-level protein data. This implies the need to effectively integrate their embeddings while minimizing the reduction in their respective unimodal representation capabilities. To preserve such unimodal information when injecting the cross-level information from each other, in this subsection, we propose to integrate sequence-structure features with a new ProteinAdapter.

The current Mamba architecture cannot effectively handle multiple-feature integration as it lacks mechanisms similar to cross-attention. As an improvement, we designed a Mamba-based fusion mechanism in our ProteinAdapter for multi-level protein representation. As shown in Figure 2, ProteinAdapter consists of a pre-alignment module followed by several MFBlocks.

**Pre-alignment.** We first deploy several respective Mamba blocks (Gu & Dao, 2023) on each single-level embedding as feature preprocessing. Subsequently, they undergo a fuse operation by simple addition to obtain an initial fusion feature $E_{fusion}$. To reduce channel redundancy and enhance the expressive power of different channels, we further integrate Efficient Channel Attention (ECA) (Wang et al., 2020) to the fused embedding. Overall, the output after pre-alignment can be expressed as:

$$E^{'}_{fusion} = \mathrm{ECA}(\mathrm{Conv}(\mathrm{Mamba}(E_{str}) + \mathrm{Mamba}(E_{seq}))). \tag{1}$$

More details on Mamba are given in the Appendix A.

**Mamba Fusion Block (MFBlock).** The MFBlock consists of three branches based on the Mamba architecture, which leverages level-specific protein features to guide multi-level embedding fusion, aiming to capture local detail characteristics from various protein levels. The input to the MFBlock is the initial fused feature $E^{'}_{fusion}$ derived from the pre-alignment module. Additionally, two more branches are incorporated, each receiving protein features $E^{'}_{seq}$ and $E^{'}_{str}$ from distinct levels. Each of the three branches goes through layer normalization, 1D convolution, SiLU activation, and parameter discretization, followed by passing through the SSM to generate an output. These outputs are modulated by a gating factor and added to the output from the previous block, producing the final multi-level protein embedding $E_{fusion}$ guided by level-specific features. More details on the MFBlock are given in Algorithm 1.

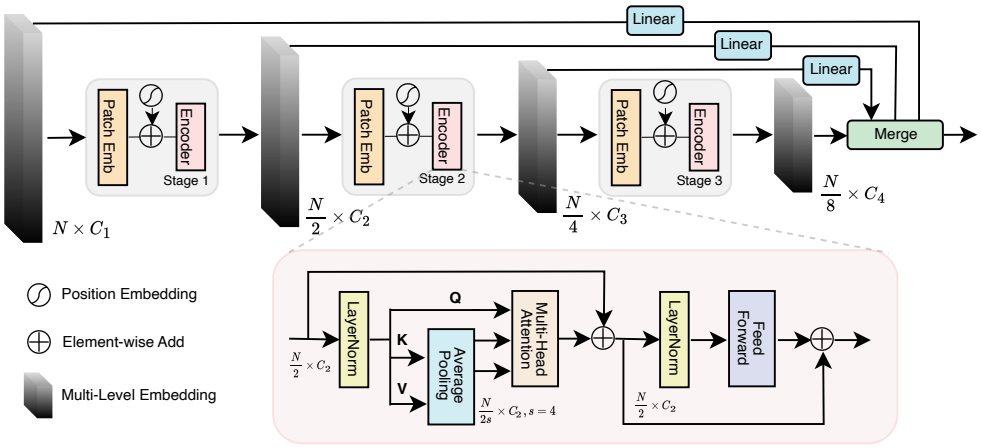


Figure 3: Architecture of the Multi-Scale Predictor. In each stage, the input embedding first undergoes an overlapping patch embedding block for $2\times$ downsamping, and then undergoes the encoder for multi-head self-attention. $N$ and $C$ represent the number of features and channels respectively. We deploy an efficient transformer as the Encoder, where the key and value is downsampled with $s = 4$.

Finally, considering amino acids primarily react with their surrounding neighbors, we further employ 1D convolution with sliding windows after the last MFBlock, to extract local features from the concatenated multi-level embeddings. After such alignment, the PSM is enriched by the valuable property information within PLM, to capture the protein property information with different levels while preserving its original representation powers.

### 3.3 Enhancing pyramidal representations with multi-scale predictor

After thoroughly fusing the sequence and geometric embeddings through ProteinAdapter, the obtained multi-level embedding contains rich evolutionary and biological patterns underlying both levels of protein structures. Now we move a further step to fully utilize the multi-level embedding for downstream tasks. As shown in Figure 3, we design a multi-scale predictor to efficiently extract and utilize protein features in a hierarchical manner.

**Patch Embedding.** During each stage, an overlapping patch embedding block (Wang et al., 2021a; 2022) with downsampling rate 2 is performed to halve the protein resolution and build a pyramid architecture. Specifically, given the embedding $\boldsymbol{E}_{i-1} \in \mathbb{R}^{N_{i-1} \times C_{i-1}}$ from the previous stage $i-1$ as input, we feed it to a 1D convolution to acquire the downsampled embedding $\boldsymbol{E}_i \in \mathbb{R}^{N_i \times C_i}$, where $N_i$ is computed as $N/2^i$. In our implementation, we set channel numbers $C_1$, $C_2$, $C_3$, $C_4$ as 512, 512, 1024, 1024. The 1D convolution is set as the stride 2, kernel size 3, zero padding size of 1, and the kernel number of $C_i$. This downsampling patch embedding block flexibly adjusts the feature scale in each stage, making it possible to construct a feature pyramid for the Transformer. In this way, our method can handle longer input protein embedding with limited resources.

**Transformer Encoder.** Following existing efficient transformer methods (Wang et al., 2022; Han et al., 2023), we adopt zero padding position encoding into the transformer encoder. In our implementation, we further use a linear attention mechanism (Katharopoulos et al., 2020) that uses average pooling to reduce the sequence length $n$ with a pooling size $s = 4$ before the attention operation, as shown in Figure 3. We also employ a focused function (Han et al., 2023)on the attention map, to pay more attention to those effective functional regions. In our implementation, we use dynamic batching (Rives et al., 2021) to handle variable-length sequences for higher computational efficiency.

**Multi-Scale Prediction.** Finally, the features from various scales are linearly mapped to the dimensions corresponding to downstream tasks. Then, through a set of learnable parameters, the weights of different scales are dynamically adjusted, thereby achieving adaptability to different tasks and the possibility of multi-task learning.

# 4 EXPERIMENTS

## 4.1 EVALUATION TASKS AND DATASETS

To validate the efficacy of ProteinAdapter, we conduct tests on over 20 tasks. Since our model explicitly leverages multi-level protein data, we separately assess the model's ability to protein sequences and structures understanding on the PEER (Xu et al., 2022), ATOM3D (Townshend et al., 2020), and several other benchmarks. Since the PEER benchmark only contains 1D sequences, we further validate the performance on four sequence-structure paired datasets. We also conduct experiments with predicted protein structures (see Section 4.2). Subsequently, we evaluate the performance in a multi-task setting on the PEER benchmark, verifying its generalization capability. We then compare the efficiency superiority of ProteinAdapter against the current LPMs. Finally, we conduct a series of ablation studies to confirm the effectiveness of our multi-level adapter and multi-scale predictor. More details on the benchmarks and each task can be found in our Appendix C.

## 4.2 TRAINING SETUP

For the sequence-based PEER benchmark, due to the lack of corresponding 3D coordinate information, we simply replaced the structural embedding with another layer of sequence embedding (*i.e.*, layer 32 and 33 of ESM-1b). Another alternative approach is to use a pre-trained structure prediction model ESMFold (Lin et al., 2022) to generate pseudo structures. Due to computational limitations, we only conducted multi-level performance tests on the Stability Prediction and Fold Classification tasks for the PEER benchmark. The Stability Prediction dataset has the shortest average sequence length (less than 70). For the Fold Classification task, we can collect the corresponding structures from RCSB-PDB (Berman et al., 2000b). Moreover, for the PPI task that takes paired proteins as inputs, we take the multi-level embeddings obtained for each of the two proteins, and then feed them into an MLP predictor defined based on the concatenation of the embeddings of the two proteins. For the PLI task which has protein-ligand inputs, we follow previous practices (Hu et al., 2019; Sun et al., 2019), and involve an additional Graph Isomorphism Network (GIN) (Xu et al., 2018) with 4 layers and 256 hidden dimensions as the ligand graph encoder. As for the ATOM3D benchmark (Townshend et al., 2020), we can directly extract their protein sequences from the mdb or pdb files. Our method is implemented based on PyTorch 1.13.1 with CUDA 11.7. All experiments are conducted on one NVIDIA Tesla A100 (80GB). As for other compared methods, we deploy the same default training settings in the benchmark (Xu et al., 2022; Townshend et al., 2020).

## 4.3 SINGLE-TASK *vs.* MULTI-TASK

**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 delve deeper into multi-task learning to validate the generalization ability of ProteinAdapter. Similar to PEER(Xu et al., 2022), our training objective comprises a *primary task* and a *supportive task*. Following the principle that "protein structures determine their functions" (Hegyi & Gerstein, 1999), we employ three structure prediction tasks, *i.e.*, contact prediction, fold classification and secondary structure prediction, as the auxiliary task. In more detail, when presented with a primary task $t_p$ characterized by loss $\mathcal{L}_{t_p}$ and a supportive task $t_s$ characterized by loss $\mathcal{L}_{t_s}$, our multi-task model follows the framework of hard parameter sharing (Ruder, 2017). Within this framework, we utilize a universal protein sequence encoder for all tasks, and a ligand graph encoder for protein-ligand interaction prediction tasks. Throughout the multi-task learning process, the network parameters are fine-tuned based on the combined loss of the primary and supportive tasks: $\mathcal{L} = \mathcal{L}_{t_p} + \lambda \mathcal{L}_{t_s}$. Here, $\lambda$ represents the balancing factor for the two objectives, defaulting to 1.0 unless otherwise mentioned. The training iterations are consistent with single-task learning on the primary task, and we only evaluate one primary task to maintain consistency with the PEER benchmark.

**Optimization Objective.** Following PEER Benchmark (Xu et al., 2022), the fluorescence, stability, $\beta$-lactamase activity, PPI affinity, PDBbind and BindingDB prediction tasks are trained with Mean Squared Error (MSE); the solubility, subcellular localization, binary localization, fold, secondary

Table 1: Experimental results on PEER benchmark and other datasets under single-task learning setting. ↑ indicates that higher values correspond to better performance. To ensure a fair comparison, we only conducted the evaluation of ProteinAdapter to other residue sequence-only methods. The best performance is marked in **blod** and the second performance is underlined.

| Task | Vanilla Encoder | | Pre-trained LPMs | | | ProteinMamba (Xu et al., 2024) | SaProt (Su et al., 2024) | ProteinAdapter (w/o structure) |
|---|---|---|---|---|---|---|---|---|
| | Transformer | CNN | ProtBert | ESM-1b | ESM-2 | | | |
| Flu ↑ | 0.643 | 0.682 | 0.679 | 0.701 | 0.697 | 0.683 | 0.699 | **0.703** |
| Sta ↑ | 0.649 | 0.639 | 0.771 | 0.779 | 0.735 | 0.753 | 0.755 | **0.785** |
| $\beta$-lac ↑ | 0.261 | 0.781 | 0.731 | 0.855 | **0.904** | 0.788 | 0.896 | 0.897 |
| Sol ↑ | 70.12 | 64.43 | 68.15 | 70.23 | 70.12 | 68.25 | **70.88** | 70.65 |
| GB1 ↑ | 0.271 | 0.502 | 0.634 | 0.685 | 0.701 | 0.706 | **0.711** | 0.709 |
| AAV ↑ | 0.681 | 0.746 | 0.794 | 0.785 | **0.795** | 0.754 | 0.792 | **0.795** |
| Thermo ↑ | 0.545 | 0.494 | 0.660 | **0.687** | 0.677 | 0.674 | 0.679 | 0.683 |

Table 2: Experiments with GT structures on various datasets. ↑ indicates higher values correspond to better performance. The best performance is marked in **blod** with the second performance underlined. (3+1)D represents methods with sequence-structure paired inputs.

| Input | Method | Fold Classification | | | Enzyme | Gene Ontology | | | Enzyme |
|---|---|---|---|---|---|---|---|---|---|
| | | Fold ↑ | Superfamily ↑ | Family ↑ | Reaction ↑ | BP ↑ | MF ↑ | CC ↑ | Commission ↑ |
| 1D | CNN (Shanehsazzadeh et al., 2020) | 11.3 | 13.4 | 53.4 | 51.7 | 0.244 | 0.354 | 0.287 | 0.545 |
| | Transformer (Rao et al., 2019b) | 9.22 | 8.81 | 40.4 | 26.6 | 0.264 | 0.211 | 0.405 | 0.238 |
| | ESM-1b (Rives et al., 2021) | 26.8 | 60.1 | 97.8 | 83.3 | 0.452 | 0.657 | 0.477 | 0.864 |
| 3D | GCN (Kipf & Welling, 2016) | 16.8 | 21.3 | 82.8 | 67.3 | 0.252 | 0.195 | 0.329 | 0.320 |
| | GAT (Veličković et al., 2017) | 12.4 | 16.5 | 72.7 | 55.6 | 0.284 | 0.317 | 0.385 | 0.368 |
| | 3D CNN (Derevyanko et al., 2018) | 31.6 | 45.4 | 92.5 | 72.2 | 0.240 | 0.147 | 0.305 | 0.077 |
| (3+1)D | IEConv (Hermosilla et al., 2020) | 47.6 | 70.2 | 99.2 | 87.2 | 0.421 | 0.624 | 0.431 | — |
| | GearNet-Edge (Zhang et al., 2023b) | 44.0 | 66.7 | 99.1 | 86.6 | 0.403 | 0.580 | 0.450 | 0.810 |
| | CDConv (Fan et al., 2022) | **56.7** | 77.7 | 99.6 | 88.5 | 0.453 | 0.654 | 0.479 | 0.820 |
| | SaProt-PDB (Su et al., 2024) | 52.5 | 77.8 | 99.6 | 87.7 | 0.465 | 0.669 | 0.415 | **0.888** |
| | ESM-GearNet (Zhang et al., 2023a) | 55.1 | **82.0** | **99.9** | 88.4 | 0.491 | 0.677 | **0.501** | 0.883 |
| | **ProteinAdapter (w/ GT structure)** | 56.5 | 81.7 | **99.9** | **89.2** | **0.495** | **0.681** | **0.501** | 0.885 |

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. As for ATOM3D benchmark (Townshend et al., 2020), we use MSE loss for regression tasks and cross-entropy loss for classification tasks.

### 4.4 Comparisons

**Evaluation on Single-Task Learning.** (1) *Sequence-only input*: Table 1 shows the performance of ProteinAdapter on the PEER (Xu et al., 2022) benchmark. Notably, as the PEER benchmark is designed for protein sequence understanding and only provides sequence information, we instead replace the structural embedding in our baseline with sequence features from different layers of ESM-1b, as mentioned in Section 4.2. It can be observed that for sequence-based tasks in the PEER benchmark, our ProteinAdapter achieves favorable performance against other sequence-only encoders. This is mainly because our adapter not only performs multi-level feature interactions but also can optimize each single-level protein representation through Mamba. (2) *Sequence-Structure input*. In addition to sequence-only experiments, we also tested our ProteinAdapter on four downstream tasks using sequence-structure paired data to fully explore its potential in fusing multi-level protein features. Results in Table 2 demonstrate the effectiveness of the proposed ProteinAdapter that utilizes features from pre-trained LPMs and further addresses the multi-level complementarity through the Mamba fusion mechanism. As for the structure-based tasks in the ATOM3D benchmark, Table 3 shows that our ProteinAdapter consistently achieves state-of-the-art performance, since our method explicitly utilizes multi-level protein knowledge.

**Evaluation on Multi-Task Learning.** We then conduct multi-task learning experiments following the settings of the PEER benchmark, the results are shown in Table 4. It can be observed that although our method has already made full use of the sequence information, using other structure-related prediction tasks as an auxiliary can still improve the performance of the central task to some extent. This experiment indicates the complementary role of structural information (even when implicitly embedded in the protein sequence) for sequence tasks, and also demonstrates that our method has decent generalization capability.

Table 3: Atom-level results on ATOM3D benchmark results under single-task learning. "—" indicates a non-applicable setting. "*" indicates that the original implementation in ATOM3D benchmark. The best performance is marked in **blod** and the second performance is underlined.

| Task | CNN* | GNN* | ENN* | GearNet-Edge (Zhang et al., 2023b) | GVP-GNN (Jing et al., 2021) | SiamDiff (Zhang et al., 2024) | ProteinAdapter (w/ GT structure) |
|---|---|---|---|---|---|---|---|
| SMP ↓ | 0.754 | 0.501 | 0.052 | 0.067 | 0.049 | 0.023 | **0.019** |
| PIP ↑ | 0.844 | 0.669 | — | 0.868 | 0.866 | 0.882 | **0.884** |
| RES ↑ | 0.451 | 0.082 | 0.072 | 0.443 | 0.527 | 0.460 | **0.529** |
| MSP ↑ | 0.574 | 0.609 | 0.574 | 0.632 | 0.680 | **0.695** | 0.677 |
| LBA ↓ | 1.416 | 1.601 | 1.568 | 1.330 | 1.594 | **1.057** | 1.063 |
| LEP ↑ | 0.589 | 0.681 | 0.663 | 0.625 | 0.628 | 0.711 | **0.731** |
| PSR ↑ | 0.789 | 0.750 | — | 0.780 | **0.845** | 0.831 | 0.827 |
| RSR ↑ | 0.372 | 0.512 | — | 0.397 | 0.330 | 0.341 | **0.533** |

Table 4: PEER benchmark results under multi-task learning setting. Red means the average results outperform the original single-task learning baseline; gray results are the same as the baseline; blue results underperform the baseline; "-" indicates not applicable for this setting. Abbr., Ori.: original; Avg.: average performance under three auxiliary tasks. ↑ indicates that higher values correspond to better performance. The best performance among three auxiliary tasks is marked in **blod**.

| Task | Transformer | | | | | ESM-1b | | | | | ProteinAdapter (w/o structure) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Ori. | +Cont | +Fold | +SSP | Avg. | ori. | +Cont | +Fold | +SSP | Avg. | Ori. | +Cont | +Fold | +SSP | Avg. |
| **Function Prediction** | | | | | | | | | | | | | | | |
| Flu ↑ | 0.643 | 0.612 | 0.648 | **0.656** | 0.638 | 0.701 | **0.704** | 0.702 | **0.704** | 0.703 | 0.703 | 0.702 | 0.703 | **0.704** | 0.703 |
| Sta ↑ | 0.649 | 0.620 | **0.672** | 0.667 | 0.653 | 0.779 | 0.782 | 0.783 | **0.789** | 0.785 | 0.785 | 0.781 | **0.788** | 0.787 | 0.786 |
| $\beta$-lac ↑ | 0.261 | 0.142 | **0.276** | 0.197 | 0.205 | 0.855 | **0.899** | 0.882 | 0.881 | 0.887 | 0.897 | **0.899** | 0.893 | 0.891 | 0.894 |
| Sol ↑ | 70.12 | **70.03** | 68.85 | 69.81 | 69.56 | 70.23 | **70.46** | 64.80 | 70.03 | 68.43 | 70.65 | **71.08** | 70.61 | 71.07 | 70.92 |
| **Localization Prediction** | | | | | | | | | | | | | | | |
| Sub ↑ | 56.02 | 52.92 | **56.74** | 56.70 | 55.45 | 78.13 | **78.86** | 78.43 | 78.00 | 78.43 | 85.20 | **85.70** | 85.65 | 85.33 | 85.56 |
| Bin ↑ | 75.74 | 74.98 | **76.27** | 75.20 | 75.48 | 92.40 | **92.50** | 91.83 | 92.26 | 92.19 | 93.55 | 93.51 | **93.57** | 93.50 | 93.53 |
| **Structure Prediction** | | | | | | | | | | | | | | | |
| Cont ↑ | 17.50 | — | 2.04 | **12.76** | 7.40 | 45.78 | — | **35.86** | 32.03 | 33.94 | 55.15 | — | **55.21** | 55.17 | 55.19 |
| Fold ↑ | 8.52 | **9.16** | — | 8.14 | 8.65 | 28.17 | **32.10** | — | 28.63 | 30.36 | 32.60 | **33.21** | — | 33.18 | 33.19 |
| SSP ↑ | 59.62 | **63.10** | 50.93 | — | 57.00 | 82.73 | **83.21** | 82.27 | — | 82.74 | 83.11 | **83.15** | 83.10 | — | 83.13 |
| **Protein-Protein Interaction Prediction** | | | | | | | | | | | | | | | |
| Yst ↑ | 54.12 | 52.86 | **54.00** | **54.00** | 53.62 | 57.00 | 58.50 | **64.76** | 62.06 | 61.77 | 68.45 | 68.41 | **68.44** | 68.37 | 68.41 |
| PPI ↑ | 59.58 | 60.76 | **67.33** | 54.80 | 60.96 | 78.17 | 81.66 | 80.28 | **83.00** | 81.64 | 79.12 | 79.05 | **79.11** | 79.01 | 79.06 |
| Aff ↓ | 2.499 | 2.733 | **2.524** | 2.651 | 2.636 | 2.281 | **1.893** | 2.002 | 2.031 | 1.975 | 2.073 | 2.077 | 2.085 | **2.071** | 2.078 |
| **Protein-Ligand Interaction Prediction** | | | | | | | | | | | | | | | |
| PDB ↓ | 1.455 | 1.574 | 1.531 | **1.387** | 1.497 | 1.559 | 1.458 | 1.435 | **1.419** | 1.437 | 1.153 | 1.155 | 1.153 | **1.151** | 1.153 |
| BDB ↓ | 1.566 | 1.490 | **1.464** | 1.519 | 1.491 | 1.556 | 1.490 | 1.511 | **1.482** | 1.494 | 1.344 | 1.339 | **1.331** | 1.340 | 1.337 |

**Pseudo 3D Structures.** Additionally, we further evaluate the performance with multi-level inputs on the Stability Prediction and Fold Classification tasks. We use a pretrained structure prediction network ESMFold (Lin et al., 2022) to obtain pseudo-3D labels as structure inputs (see Section 4.2). The compared methods are conducted with the same predicted structures for a fair comparison. Table 5 shows that our performance is further improved with additional structural information, indicating the effectiveness of our ProteinAdapter in learning multi-level representations.

### 4.5 Discussion

**Ablation Study.** To fully validate the effectiveness of our proposed multi-level adapter and multi-scale predictor, we conducted ablation experiments on two paired protein sequence-structure tasks: Gene Ontology Term Prediction and Enzyme Commission Number Prediction. **1. *multi-level adapter***: We first directly remove the adapter and input the concatenated embeddings of the two LPMs into the predictor ($a$). Additionally, we also conduct tests by replacing the pre-alignment module ($b$) with direct concatenation after respective MLP, and replacing each MFBlock ($c$) with a three-layer MLP. **2. *multi-scale predictor***: Subsequently, we remove the predictor and directly input the fused embedding from the adapter into a three-layer MLP ($d$). Performances are compared with various multi-level methods. $F_{max}$ accuracy is used as the evaluation metric for these two tasks.

Table 5: Experiments with pseudo structures on the PEER benchmark. ↑ indicates higher values correspond to better performance. (3+1)D represents methods with sequence-structure inputs. We also show the performance gap before and after using pseudo-structure labels.

| Input | Method | Stability Prediction ↑ | Fold Classification ↑ |
|---|---|---|---|
| 1D | ResNet (Rao et al., 2019b) | 0.655 | 18.20 |
| | ProteinMamba (Xu et al., 2024) | 0.753 | — |
| | **ProteinAdapter w/o structure** | 0.785 | 32.60 |
| (3+1)D | IEConv (Hermosilla et al., 2020) | 0.785 | 53.10 |
| | GearNet-Edge (Zhang et al., 2023b) | 0.793 | 56.50 |
| | CDConv (Fan et al., 2022) | 0.801 | 63.70 |
| | ESM-GearNet (Zhang et al., 2023a) | 0.818 | 68.20 |
| | **ProteinAdapter w/ pseudo structure** | **0.823** (+0.038) | **68.40** (+35.80) |

Table 6: Ablation and efficiency study on Gene Ontology (GO) term prediction and Enzyme Commission (EC) number prediction tasks. ↑ indicates higher values correspond to better performance. The best performance is marked in **blod** and the second performance is underlined.

| Methods & Variants | GO | | | EC ↑ |
|---|---|---|---|---|
| | BP ↑ | MF ↑ | CC ↑ | |
| GearNet-Edge (Zhang et al., 2023b) (63.5M) | 0.403 | 0.580 | 0.450 | 0.810 |
| CDConv (Fan et al., 2022) (40.7M) | 0.453 | 0.654 | 0.479 | 0.820 |
| SaProt-PDB (Su et al., 2024) (35M) | 0.465 | 0.669 | 0.415 | **0.888** |
| ESM-GearNet (Zhang et al., 2023a) (60M) | 0.491 | 0.677 | **0.501** | 0.883 |
| *a*. w/o adapter, directly removed (15.5M) | 0.457 | 0.658 | 0.477 | 0.868 |
| *b*. w/o pre-alignment module, MLP instead (21.1M) | 0.463 | 0.665 | 0.482 | 0.872 |
| *c*. w/o MFBlock, MLP instead (25.5M) | 0.468 | 0.667 | 0.486 | 0.874 |
| *d*. w/o predictor, MLP instead (13.8M) | 0.475 | 0.674 | 0.492 | 0.879 |
| **ProteinAdapter w/ GT structure (22.5M)** | **0.495** | **0.681** | **0.501** | 0.885 |

As shown in Table 6, it can be seen that both the multi-level adapter and multi-scale predictor are indispensable in our method. Without our mamba-based feature processing and integration, there is a clear performance drop on variant *a* due to the absence of our ProteinAdapter. Replacing the pre-alignment module with MLPs provides some mitigation (*b*), but its representation quality is not as good as the features after initial integration (*c*). Note that unlike single-scale MLP (*d*) or previous methods *e.g.*, CDConv (Fan et al., 2022) that use pooling layers to reduce the number of nodes and learn representations at different scales, our multi-scale predictor does not disrupt the continuity of amino acids. In contrast, our method represents proteins at various scales and then combines these representations to ensure a comprehensive understanding.

**Efficiency.** By integrating two pre-trained LPMs with our lightweight ProteinAdapter, our method can achieve higher performance with fewer parameters compared to training a single-level large model from scratch. As shown in Table 6, with the default setting of three MFBlocks in the adapter and three downsampling times in the predictor, our model has 20× fewer parameters than the previous LPMs (22.5M of ProteinAdapter vs 650M of ESM-1b).

## 5 Conclusion

The proposed ProteinAdapter offers a scalable solution for parameter-efficient fine-tuning on pre-trained Large Protein Models (LPMs). The multi-level adapter seamlessly merges protein embeddings, eliminating the need for resource-intensive pretraining. The method demonstrates superior computing efficiency through frozen models and a lightweight Mamba-based adapter. Extensive experiments across over 20 tasks showcase its substantial performance gains over existing methods. In a multi-task setting, ProteinAdapter consistently outperforms its counterparts, establishing its versatility and generalization capabilities. We will consider combining more powerful protein representation structure prediction models in the future.

## REFERENCES


Afshine Amidi, Shervine Amidi, Dimitrios Vlachakis, Vasileios Megalooikonomou, Nikos Paragios, and Evangelia I Zacharaki. Enzynet: enzyme classification using 3d convolutional neural networks on spatial representation. *PeerJ*, 6:e4750, 2018. 2

Nina M Antikainen and Stephen F Martin. Altering protein specificity: techniques and applications. *Bioorganic & medicinal chemistry*, 13(8):2701–2716, 2005. 22

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. 20

Tristan Bepler and Bonnie Berger. Learning protein sequence embeddings using information from structure. *arXiv preprint arXiv:1902.08661*, 2019. 2

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, 2000a. 19

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, 2000b. 6

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):8039, 2021. 19

Daozheng Chen, Xiaoyu Tian, Bo Zhou, Jun Gao, et al. Profold: Protein fold classification with additional structural features and a novel ensemble classifier. *BioMed research international*, 2016, 2016. 19

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*, pp. 2021–11, 2021. 18, 19

Natalie L Dawson, Tony E Lewis, Sayoni Das, Jonathan G Lees, David Lee, Paul Ashford, Christine A Orengo, and Ian Sillitoe. Cath: an expanded resource to predict protein function through structure and sequence. *Nucleic acids research*, 45(D1):D289–D295, 2017. 22

Georgy Derevyanko, Sergei Grudinin, Yoshua Bengio, and Guillaume Lamoureux. Deep convolutional networks for quality assessment of protein folds. *Bioinformatics*, 34(23):4046–4053, 2018. 7

Ahmed Elnaggar, Hazem Essam, Wafaa Salah-Eldin, Walid Moustafa, Mohamed Elkerdawy, Charlotte Rochereau, and Burkhard Rost. Ankh: Optimized protein language model unlocks general-purpose modelling. *bioRxiv*, pp. 2023–01, 2023. 2

Hehe Fan, Zhangyang Wang, Yi Yang, and Mohan Kankanhalli. Continuous-discrete convolution for geometry-sequence modeling in proteins. In *The Eleventh International Conference on Learning Representations*, 2022. 2, 7, 9

Hehe Fan, Linchao Zhu, Yi Yang, and Mohan Kankanhalli. Pointlistnet: Deep learning on 3d point lists. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17692–17701, 2023. 2

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. 19

Daniel Y Fu, Tri Dao, Khaled K Saab, Armin W Thomas, Atri Rudra, and Christopher Ré. Hungry hungry hippos: Towards language modeling with state space models. *arXiv preprint arXiv:2212.14052*, 2022. 3

Toni Gabaldón. Comparative genomics-based prediction of protein function. *Genomics Protocols*, pp. 387–401, 2008. 3

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. 18

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):3168, 2021. 23

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 3, 4, 16

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021a. 3

Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585, 2021b. 3

Dongchen Han, Xuran Pan, Yizeng Han, Shiji Song, and Gao Huang. Flatten transformer: Vision transformer using focused linear attention. *arXiv preprint arXiv:2308.00442*, 2023. 5

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. 6

Pedro Hermosilla, Marco Schäfer, Matěj Lang, Gloria Fackelmann, Pere Pau Vázquez, Barbora Kozlíková, Michael Krone, Tobias Ritschel, and Timo Ropinski. Intrinsic-extrinsic convolution and pooling for learning on 3d protein structures. *arXiv preprint arXiv:2007.06252*, 2020. 2, 7, 9, 23

Jie Hou, Badri Adhikari, and Jianlin Cheng. Deepsf: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, 34(8):1295–1303, 2018. 23

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pp. 2790–2799, 2019. 3

Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. In *International Conference on Machine Learning*, pp. 8946–8970, 2022. 1, 2, 3, 4

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 3

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. 6

Bowen Jing, Stephan Eismann, Pratham N Soni, and Ron O Dror. Equivariant graph neural networks for 3d macromolecular structure. *arXiv preprint arXiv:2106.03843*, 2021. 8

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020. 5

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. 19

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 7

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. 20

Andriy Kryshtafovych, Torsten Schwede, Maya Topf, Krzysztof Fidelis, and John Moult. Critical assessment of methods of protein structure prediction (casp)—round xiii. *Proteins: Structure, Function, and Bioinformatics*, 87(12):1011–1020, 2019. 19

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 3

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 3

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, 2022:500902, 2022. 1, 6, 8

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. 20

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 61–68, 2022. 3

Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alex Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in Neural Information Processing Systems*, 34:29287–29303, 2021. 1

Alexey G Murzin, Steven E Brenner, Tim Hubbard, and Cyrus Chothia. Scop: a structural classification of proteins database for the investigation of sequences and structures. *Journal of molecular biology*, 247(4):536–540, 1995. 23

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. 18

Karl Pearson. X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175, 1900. 23

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. 20

Asha Rajagopal and Sanford M Simon. Subcellular localization and activity of multidrug resistance proteins. *Molecular biology of the cell*, 14(8):3389–3399, 2003. 17

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32, 2019a. 20

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32, 2019b. 7, 9

Roshan Rao, Joshua Meier, Tom Sercu, Sergey Ovchinnikov, and Alexander Rives. Transformer protein language models are unsupervised structure learners. *Biorxiv*, pp. 2020–12, 2020. 3

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):e2016239118, 2021. 1, 2, 3, 5, 7

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. 20

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. 20

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. 6

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. 18

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. 20

Amir Shanehsazzadeh, David Belanger, and David Dohan. Is transfer learning necessary for protein landscape prediction? *arXiv preprint arXiv:2011.03443*, 2020. 7

VA Simossis and J Heringa. Integrating protein secondary structure prediction and multiple sequence alignment. *Current Protein and Peptide Science*, 5(4):249–266, 2004. 20

Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*, 2022. 3

C. Spearman. "general intelligence," objectively determined and measured. *The American Journal of Psychology*, 15(2):201–292, 1904. 24

Nils Strodthoff, Patrick Wagner, Markus Wenzel, and Wojciech Samek. Udsmprot: universal deep sequence models for protein classification. *Bioinformatics*, 36(8):2401–2409, 2020. 2

Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. Saprot: Protein language modeling with structure-aware vocabulary. In *The Twelfth International Conference on Learning Representations*, 2024. 7, 9

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. 21

Chao Sun, Zhedong Zheng, Xiaohan Wang, Mingliang Xu, and Yi Yang. Self-supervised point cloud representation learning via separating mixed shapes. *IEEE Transactions on Multimedia*, 2022. 2

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. 6

Wen Torng and Russ B Altman. 3d deep convolutional neural networks for amino acid environment similarity analysis. *BMC bioinformatics*, 18:1–23, 2017. 22

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. 6, 7, 17

Roger Y Tsien. The green fluorescent protein. *Annual review of biochemistry*, 67(1):509–544, 1998. 18

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 7

Jesse Vig, Ali Madani, Lav R Varshney, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. Bertology meets biology: interpreting attention in protein language models. *arXiv preprint arXiv:2006.15222*, 2020. 2, 3

Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11534–11542, 2020. 4

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 568–578, 2021a. 5

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022. 5

Zeyuan Wang, Qiang Zhang, HU Shuang-Wei, Haoran Yu, Xurui Jin, Zhichen Gong, and Huajun Chen. Multi-level protein structure pre-training via prompt learning. In *The Eleventh International Conference on Learning Representations*, 2023. 2

Zichen Wang, Steven A Combs, Ryan Brand, Miguel Romero Calvo, Panpan Xu, George Price, Nataliya Golovach, Emmanuel 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*, pp. 2021–09, 2021b. 23

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. 21

Bohao Xu, Yingzhou Lu, Yoshitaka Inoue, Namkyeong Lee, Tianfan Fu, and Jintai Chen. Protein-mamba: Biological mamba models for protein function prediction. *arXiv preprint arXiv:2409.14617*, 2024. 3, 7, 9

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018. 6

Minghao Xu, Zuobai Zhang, Jiarui Lu, Zhaocheng Zhu, Yangtian Zhang, Ma Chang, Runcheng Liu, and Jian Tang. Peer: a comprehensive and multi-task benchmark for protein sequence understanding. *Advances in Neural Information Processing Systems*, 35:35156–35173, 2022. 6, 7, 17

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. 21

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. 20

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. 20

Zuobai Zhang, Minghao Xu, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Enhancing protein language models with structure-based encoder and pre-training. In *International Conference on Learning Representations Machine Learning for Drug Discovery Workshop*, 2023a. 1, 2, 3, 7, 9

Zuobai Zhang, Minghao Xu, Arian Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. In *International Conference on Learning Representations*, 2023b. 1, 2, 7, 8, 9

Zuobai Zhang, Minghao Xu, Aurelie C Lozano, Vijil Chenthamarakshan, Payel Das, and Jian Tang. Pre-training protein encoder via siamese sequence-structure diffusion trajectory prediction. *Advances in Neural Information Processing Systems*, 36, 2024. 8

## A Details on Mamba

**State Space Models (SSM)**. State space models are typically regarded as linear time-invariant systems that map input signals from $x(t) \in \mathbb{R}^N$ to output responses $y(t) \in \mathbb{R}^N$ through an intermediate hidden state $h(t) \in \mathbb{R}^N$. These systems are mathematically described by linear ordinary differential equations (ODEs):

$$h'(t) = Ah(t) + Bx(t), \tag{2}$$

$$y(t) = Ch(t), \tag{3}$$

where $N$ is the state dimension, $A \in \mathbb{R}^{N\times N}$ denotes the evolution parameter, $B \in \mathbb{R}^{N\times 1}$ and $C \in \mathbb{R}^{N\times 1}$ are projection parameters. This equation suggests that the SSM exhibits global awareness, as the current output is influenced by all preceding input data.

**Discretization**. To integrate time-continuous SSM into deep learning frameworks, discretization is essential. A method often used is zero-order hold (ZOH), allowing us to discretize the system described above. The discretized equations can be written as:

$$h_t = \bar{A}h_{t-1} + \bar{B}x_t, \tag{4}$$

$$y_t = Ch_t, \tag{5}$$

where $\bar{A} = \exp(\Delta A)$ and $\bar{B} = (\Delta A)^{-1}(\exp(\Delta A) - I) \cdot \Delta B$ are the discretized system matrices, and $\Delta$ represents the discretization step size.

**Selective Scan**. In Mamba, the Selective Scan Mechanism is introduced to overcome the challenge of parallelizing SSM operations due to input-dependent parameters, which prevent reformulating them in a convolutional form. This mechanism combines *kernel fusion*, *parallel scan*, and *recomputation* to improve efficiency. Kernel fusion minimizes memory access overhead, parallel scan speeds up sequence processing, and recomputation reduces memory usage by recalculating values as needed. Together, these techniques enable Mamba to achieve faster computation with lower memory requirements.

## B Details on the Mamba Fusion Block (MFBlock)

**Algorithm 1** Mamba Fusion Block (MFBlock).
$B$: batch size, $N$: length of the protein sequence, $C$: number of channels
*Disc* and *SSM* represent Eq. 4 and Eq. 5 implemented by selective scan in Mamba (Gu & Dao, 2023).

**Input**: single-level protein embedding $E'_{seq}$: ($B$, $N$, $C$), $E'_{str}$: ($B$, $N$, $C$),
initial fused embedding $E'_{fusion}$: ($B$, $N$, $C$)
**Output**: fused embedding $E^l_{fusion}$: ($B$, $N$, $C$) after the $l$-th MFBlock, and $l \in \{0, 1, 2, \ldots, M\text{-}1\}$.

**for** i in $seq$, $str$, $fusion$ **do**
  $E''_i : (B, N, C) \leftarrow \text{LayerNorm}(E'_i)$
  $x_i : (B, N, C') \leftarrow \text{MLP}_{x_i}(E''_i)$.
  $x'_i : (B, N, C') \leftarrow \text{SiLU}(Conv_i(x_i))$.
  $\bar{\mathbf{A}} : (B, N, C', D), \bar{\mathbf{B}} : (B, N, C', D), \mathbf{C} : (B, N, C', D) \leftarrow Disc(\mathbf{x}'_i)$
  $\mathbf{y}_i : (B, N, C') \leftarrow SSM(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \mathbf{C})(\mathbf{x}'_i)$
**end for**
$\mathbf{z} : (B, N, C') \leftarrow \text{MLP}_z(E''_{fusion})$
$\mathbf{y}'_{seq} : (B, N, C') \leftarrow \mathbf{y}_{seq} \odot \text{SiLU}(\mathbf{z})$
$\mathbf{y}'_{str} : (B, N, C') \leftarrow \mathbf{y}_{str} \odot \text{SiLU}(\mathbf{z})$
$E^l_{fusion} : (B, N, C) \leftarrow \text{MLP}_F(\mathbf{y}'_{seq} + \mathbf{y}'_{str}) + E^{l-1}_{fusion}$
**Return** $E^l_{fusion}$.

# C Details on Evaluation Tasks

PEER (Xu et al., 2022) is a comprehensive and multi-task benchmark for protein sequence understanding. ATOM3D (Townshend et al., 2020) is a structure-based benchmark designed primarily for 3D structural prediction and related tasks in molecular biology. The benchmark encompasses a wide array of 3D structures of proteins, small molecules, and protein-ligand complexes. We select 22 kinds of the most common tasks from these two benchmarks, focusing on different tasks including localization prediction, function prediction, structure prediction, Protein-Protein Interaction (PPI) prediction, and Protein-Ligand Interaction (PLI) prediction.

Table 7: Task descriptions. Each task, along with its acronym, category, the size of each dataset split (train/valid/test), and evaluation metric are shown below. *Abbr.*, Reg.: regression; Cls.: classification; $R_S$: Spearman correlation; $R_P$: Pearson correlation; Acc: accuracy; RMSE: root-mean-square error; AUROC: area under the receiver operating characteristic curve; MAE: mean absolute error.

| Name (Acronym) | Task Category | Dataset Split | Metric |
|---|---|---|---|
| **PEER Benchmark** | | | |
| Fluorescence prediction (Flu) | Protein-wise Reg. | 21,446 / 5,362 / 27,217 | $R_S$ |
| Stability prediction (Sta) | Protein-wise Reg. | 53,571 / 2,512 / 12,851 | $R_S$ |
| $\beta$-lactamase activity prediction ($\beta$-lac) | Protein-wise Reg. | 4,158 / 520 / 520 | $R_S$ |
| GB1 fitness (GB1) | Protein-wise Reg. | 381 / 43 / 8,309 | $R_S$ |
| AAV fitness (AAV) | Protein-wise Reg. | 28,626 / 3,181 / 50,776 | $R_S$ |
| Thermostability (Thermo) | Protein-wise Reg. | 5,149 / 643 / 1,366 | $R_S$ |
| Contact prediction (Cont) | Residue-pair Cls. | 25,299 / 224 / 40 | L/5 precision |
| Subcellular localization prediction (Sub) | Protein-wise Cls. | 8,945 / 2,248 / 2,768 | Acc |
| Binary localization prediction (Bin) | Protein-wise Cls. | 5,161 / 1,727 / 1,746 | Acc |
| Solubility prediction (Sol) | Protein-wise Cls. | 62,478 / 6,942 / 1,999 | Acc |
| Fold classification (Fold) | Protein-wise Cls. | 12,312 / 736 / 718 | Acc |
| Secondary structure prediction (SSP) | Residue-wise Cls. | 8,678 / 2,170 / 513 | Acc |
| Yeast PPI prediction (Yst) | Protein-pair Cls. | 1,668 / 131 / 373 | Acc |
| Human PPI prediction (PPI) | Protein-pair Cls. | 6,844 / 277 / 227 | Acc |
| PPI affinity prediction (Aff) | Protein-pair Reg. | 2,127 / 212 / 343 | RMSE |
| PLI prediction on PDBbind (PDB) | Protein-ligand Reg. | 16,436 / 937 / 285 | RMSE |
| PLI prediction on BindingDB (BDB) | Protein-ligand Reg. | 7,900 / 878 / 5,230 | RMSE |
| **ATOM3D Benchmark** | | | |
| Small Molecule Properties (SMP) | Protein-wise Reg. | 103,547 / 12,943 / 12,943 | MAE |
| Protein Interface Prediction (PIP) | Protein-pair Cls. | 1,240 / 155 / 155 | AUROC |
| Residue Identity (RES) | Protein-wise Cls. | 21,147 / 964 / 3,319 | Acc |
| Mutation Stability Prediction (MSP) | Protein-wise Cls. | 1,660 / 210 / 210 | AUROC |
| Ligand Binding Affinity (LBA) | Protein-ligand Reg. | 3,678 / 460 / 460 | RMSE |
| Ligand Efficacy Prediction (LEP) | Protein-ligand Cls. | 4,220 / 501 / 870 | AUROC |
| Protein Structure Ranking (PSR) | Protein-wise Reg. | 508 / 85 / 56 | $R_S$ |
| RNA Structure Ranking (RSR) | Protein-wise Reg. | 13,182 / 4,056 / 4,056 | $R_S$ |

## C.1 PEER Benchmark

**Subcellular localization prediction (Sub).**

*Impact:* Determining the subcellular positioning of a protein can significantly enhance target pinpointing in drug development (Rajagopal & Simon, 2003). A tool that predicts subcellular localization swiftly and precisely can expedite this procedure. This endeavor aids in the creation of such an instrument.

*Target:* The task requires the model to determine the cellular location of a native protein. For instance, proteins inherently present in the lysosome will be designated with a category tag *"lysosome"*. Ten potential localizations exist, leading to the label $y \in \{0, 1, ..., 9\}$.

*Split:* We randomly split out a validation set from the training set with a 4:1 training/validation ratio.

**Binary Protein Localization (Bin).**

*Impact:* Identifying whether a protein is "soluble" or "membrane-bound" plays a pivotal role in comprehending its function. "Soluble" proteins operate as free molecules in organisms, while "membrane-bound" proteins might exhibit catalytic functions upon membrane attachment (Gimpelev et al., 2004). Efficiently distinguishing these two protein categories via computational methods can streamline biological research.

*Target:* The primary objective of the model in this task is a coarse classification of proteins into one of two categories: "membrane-bound" or "soluble". Consequently, the label for these proteins is $y \in \{0, 1\}$.

*Split:* For validation, we allocate a subset from the training data, maintaining a 4:1 ratio from training to validation. This task also assesses the model's capability to generalize over related protein sequences.

**Fluorescence Prediction (FLu).**

*Impact:* The green fluorescent protein acts as a crucial marker, allowing researchers to identify the existence of specific proteins in organic entities through its green glow (Tsien, 1998). This task aims to uncover the mutation trends that amplify or diminish such a biological characteristic.

*Target:* This challenge tasks the model to forecast the fitness of green fluorescent protein variants. The target label $y \in R$ corresponds to the logarithmic value of the fluorescence intensity as annotated by Sarkisyan et al. (Sarkisyan et al., 2016).

*Split:* We retain the division strategy from TAPE, emphasizing training on simpler mutants (with up to three mutations) and evaluating the model's performance on more complex mutants (with four or more mutations).

**Stability Prediction (Sta).**

*Impact:* The stability of a protein is pivotal for its functional efficacy in the body (Sarkisyan et al., 2016). This benchmarking task mirrors the practical application setting where functional mutants with satisfactory stability are chosen.

*Target:* The challenge here is to assess the stability of proteins in their natural environments. The target label $y \in R$ reflects the experimental stability measurement.

*Split:* We align with TAPE's splitting method, emphasizing training on proteins with multiple mutations and testing the model's capabilities on top-tier candidates having only a single mutation.

**$\beta$-lactamase activity prediction ($\beta$-lac).**

*Impact:* The TEM-1 beta-lactamase is the predominant enzyme granting gram-negative bacteria resistance to beta-lactam antibiotics (Palzkill & Botstein, 1992). This task delves into the improvement of this critical enzyme's activity through singular mutations.

*Target:* The aim is to analyze the activity among primary mutants of the TEM-1 beta-lactamase protein. The target label $y \in R$ corresponds to the empirically determined fitness score, which captures the proportional effect of mutations for each variant.

*Split:* High-capacity models are anticipated to discern proteins that differ by only a single amino acid residue in the dataset.

**GB1 fitness (GB1).**

*Impact:* GB1 fitness assesses the fitness values of potential mutants of the GB1 protein, which plays a crucial role in protein engineering. Specifically, the goal of this research is to understand how different mutations affect the protein's functionality.

*Target:* The study focuses on protein G, which is an immunoglobulin-binding protein. The GB1 domain within protein G is essential for its binding function. The research aims to enhance the fitness of this domain by exploring the interactions between mutations, ultimately improving the engineered protein's performance.

*Split:* The data source is from the FLIP benchmark (Dallago et al., 2021). The ground truth is a continuous fitness value. The training, validation, and test set contain 381, 43, and 8309 data points, respectively.

**AAV fitness (AAV).**

*Impact:* AAV fitness is critical in the field of gene therapy, as it evaluates the fitness scores of mutants in the VP-1 AAV capsid proteins. By optimizing these capsid proteins, the manipulation of Adeno-associated virus (AAV) holds significant promise for improving gene delivery efficiency. This can greatly enhance the ability of the virus to deliver therapeutic DNA to target cells, making gene therapy treatments more effective.

*Target:* The goal of evaluating AAV fitness is to understand how different mutations in these proteins affect the virus's capacity to effectively deliver its genetic payload into specific target cells.

*Split:* The data source is from the FLIP benchmark (Dallago et al., 2021). The ground truth (fitness score) is a continuous value. The training, validation, and test sets contain 28,626/3,181/50,776 data points, respectively.

**Thermostability (Thermo).**

*Impact:* Thermostability is vital for enhancing the durability of proteins in drug development and industrial processes, allowing them to maintain functionality under high-temperature conditions.

*Target:* The focus is on improving protein structure to withstand heat, making proteins more suitable for therapeutic use and industrial applications like enzyme catalysis.

*Split:* The data source is from the FLIP benchmark (Dallago et al., 2021). The ground truth (temperature) is a continuous value. The training/validation/test sets consist of 5149, 643,and 1366 data samples, respectively.

**Solubility prediction (Sol).**

*Impact:* In the realms of pharmaceutical research and industry, protein solubility stands as a paramount attribute, as optimal solubility is indispensable for a protein's functionality (Khurana et al., 2018). This endeavor seeks to enhance the design of efficient computational tools that predict protein solubility based on sequences.

*Target:* The challenge revolves around forecasting a protein's solubility. To specify, it determines if a protein is soluble, resulting in a label $y \in \{0, 1\}$.

*Split:* The division of data evaluates the proficiency of models in extrapolating across diverse protein sequences.

**Contact prediction (Cont).**

*Impact:* Estimating amino acid contacts derived from protein sequences is pivotal in predicting folded protein structures (Billings et al., 2021). This benchmark emphasizes medium- and long-range contacts, which play an instrumental role in the protein folding process.

*Target:* This assignment seeks to determine the contact likelihood between residue pairs. Each pair of residues is labeled with a binary value,$y \in \{0, 1\}$, signifying whether they establish contact within a predefined distance threshold $\delta$ or remain distant.

*Split:* In line with the CASP criteria (Kryshtafovych et al., 2019), our assessment hones in on the precision of the top L/5 contact predictions for medium and long-range contacts within the test dataset, thereby evaluating the prowess of contact prediction models in discerning the folded conformations of a diverse array of protein sequences.

**Fold classification (Fold).**

*Impact:* Discerning the overarching structural topology of a protein at the fold level is invaluable for functional elucidation and drug design initiatives (Chen et al., 2016). Given that the SCOPe database (Fox et al., 2014) only categorizes a fractional segment of proteins in PDB (Berman et al., 2000a), there's a pronounced need to harness machine learning for automated fold classification directly from protein sequences.

*Target:* The objective is to classify the protein based on its global structural contour at the fold tier. This is denoted by a categorical label, $y \in \{0, 1, ..., 1194\}$, defined by the backbone coordinates of its structure.

*Split:* Superfamilies in entirety are excluded from the training phase and make up the test set. Such an arrangement offers a unique opportunity to assess the model's competency in recognizing proteins with structurally akin attributes but sequence differences, which is a hallmark of remote homology detection (Rao et al., 2019a).

**Secondary structure prediction (SSP).**

*Impact:* Accurately discerning the local structures of protein residues in their native conformation has multifaceted benefits, including insights into protein functionality (Klausen et al., 2019) and refining multiple sequence alignments (Simossis & Heringa, 2004). This benchmark exercise seeks to foster the development and testing of machine learning models tailored for such predictions.

*Target:* The mission is to prognosticate the local configurations of protein residues as they exist naturally. Each residue is earmarked with a secondary structure label $y \in \{0, 1, 2\}$, corresponding to coil, strand, or helix.

*Split:* While the primary source of data is Klausen's dataset for training, evaluation pivots on the CB513 dataset, ensuring a rigorous assessment of model generalization across variegated protein sequences.

**Yeast PPI prediction (Yst).**

*Impact:* Constructing comprehensive and accurate yeast interactome network maps is of paramount scientific significance (Yu et al., 2008; Pu et al., 2009; Baryshnikova et al., 2010). By forecasting binary yeast protein interactions using machine learning models, this benchmark task makes strides towards realizing this ambitious objective.

*Target:* The challenge mandates the model to ascertain whether a pair of yeast proteins engages in interaction. Pairs of proteins are designated with a binary label, symbolized as $y \in \{0, 1\}$, indicating the presence or absence of an interaction.

*Split:* We commence by pruning redundancies from all protein sequences in the dataset, setting a $90\%$ sequence identity threshold. Following this, these refined sequences are indiscriminately apportioned into training, validation, and test segments. Subsequently, redundancy elimination is carried out between each pair of these segments, with a stricter $40\%$ sequence identity cut-off. This ensures rigorous appraisal of the model's capacity for generalization across disparate protein sequences.

**Human PPI prediction (PPI).**

*Impact:* Deciphering the intricate web of the human protein interactome plays a crucial role in shedding light on disease mechanisms and unearthing novel disease-associated genes (Rual et al., 2005; Yu et al., 2011; Rolland et al., 2014). With this benchmark task, there's a hopeful anticipation of enhancing potent machine learning models adept at predicting human protein-protein interactions.

*Target:* The objective at hand is for the model to discern if a pair of human proteins is interactive. Each pairing is accompanied by a binary label, represented as $y \in \{0, 1\}$, indicating their interactive status.

*Split:* Our data partitioning strategy mirrors that of the yeast PPI prediction. However, we opt for an 8:1:1 division ratio for train, validation, and test segments, respectively. Just as before, this task evaluates the model's proficiency in generalizing across diverse protein sequences.

**PPI affinity prediction (Aff).**

*Impact:* The capability to forecast the relative binding vigor among potential binding candidates holds paramount importance in the realm of protein binder design (Liu et al., 2021; Shan et al., 2022). This task seeks to provide a pragmatic arena for machine learning models to demonstrate their efficacy in such a tangible application.

*Target:* The primary objective for the model is to compute the binding affinity, denoted as $y \in R$, gauged through pKd, between two protein entities.

*Split:* Delving deeper into the dataset segmentation, our training set amalgamates wild-type complexes alongside mutants possessing a maximum of two mutations. The validation set envelops mutants with a mutation count of three or four. Lastly, the test set encompasses mutants that exhibit more

than four mutations. With this delineation, the task is positioned to assess the model's generalization prowess in a phased protein binder design context.

**PLI prediction on PDBbind (PDB).**

*Impact:* The elucidation of interactions between minor molecular entities and their corresponding target proteins emerges as a salient focus in drug discovery research (Yamanishi et al., 2010; Wen et al., 2017). This benchmark task is meticulously crafted to gauge the prowess of machine learning models in realizing this intricate objective.

*Target:* The onus is on the model to predict the interactions between small molecules and target proteins.

*Split:* To initiate, we diligently eradicate training sequences that parallel test sequences, deploying a 90% sequence identity threshold. Subsequently, the remaining training sequences undergo clustering. These clusters are then randomly apportioned into training and validation sets, abiding by a 9:1 distribution ratio. For assessment of model generalizability, the CASF-2016 benchmark (Su et al., 2018) is the chosen paradigm.

**PLI prediction on BindingDB (BDB).**

*Impact:* Recognizing the interactions between ligands and specific protein classes remains a pivotal endeavor in the realm of drug discovery. This benchmark task resonates with the aspirations of the drug discovery fraternity, emphasizing the evaluation of ligand interactions across four distinct protein classes.

*Target:* The core objective is for the model to ascertain ligand interactions, particularly focusing on four protein classes: ER, GPCR, ion channels, and receptor tyrosine kinases.

*Split:* The dataset segregation strategy, mirroring that of DeepAffinity, ensures that the aforementioned four protein classes are excluded from the training and validation phases, earmarking them exclusively for the generalization test.

### C.2 ATOM3D BENCHMARK

**Ligand Efficacy Prediction (LEP).**

*Impact:* Proteins often activate or deactivate by altering their form. Determining the shape a medication will encourage is pivotal in drug creation.

*Target:* This is approached as a binary classification challenge, where the goal is to ascertain if a molecule, when bound to these structures, will stimulate the protein's function.

*Split:* We categorize the complex pairs based on their protein targets.

**Small Molecule Properties (SMP).**

*Impact:* Estimating the physicochemical attributes of tiny molecules is a standard procedure in pharmaceutical chemistry and materials creation. While quantum-chemical assessments can reveal specific physicochemical characteristics, they are resource-intensive.

*Target:* Our goal is to forecast the properties of the molecules based on their ground-state configurations.

*Split:* We divide the molecules arbitrarily.

**Protein Structure Ranking (PSR).**

*Impact:* Proteins serve as fundamental agents within cells, and discerning their structure is typically vital for comprehending and tailoring their role.

*Target:* We approach this as a regression challenge, aiming to predict the global distance test for each structural blueprint relative to its experimentally defined structure.

*Split:* We segregate structures based on the year of competition.

**RNA Structure Ranking (RSR).** *Impact:* RNA, much like proteins, has pivotal functional responsibilities such as gene regulation and can take on distinct 3D configurations. However, the available data is limited, with only a handful of identified structures.

*Target:* Our aim is to estimate the root-mean-squared deviation (RMSD) for each structural model in comparison to its lab-verified structure.

*Split:* Structures are divided based on the respective year of the competition.

**Protein Interface Prediction (PIP).**

*Impact:* In many situations, proteins interact with one another. For instance, antibodies detect diseases by attaching to antigens. One fundamental challenge in comprehending these interactions is pinpointing the specific amino acids in two proteins that will engage upon binding.

*Target:* Our goal is to determine if two amino acids will come into contact when their parent proteins bind together.

*Split:* Protein complexes are divided ensuring that no protein from the training set shares above 30% sequence similarity with any protein in the DIPS validation set or the DB5 set.

**Ligand Binding Affinity (LBA).**

*Impact:* Proteins often modulate their functions by altering their structures. Determining the favored shape of a drug is crucial in the realm of drug design.

*Target:* We approach this as a binary classification challenge, aiming to discern if a molecule, when attached to these structures, will stimulate the protein's function.

*Split:* We categorize the complex pairs based on their specific protein targets.

**Residue Identity (RES).**

*Impact:* Comprehending the structural contribution of specific amino acids is pivotal for the creation of new proteins. This understanding can be achieved by forecasting the likelihood of various amino acids at a particular protein location, considering the adjacent structural backdrop (Torng & Altman, 2017).

*Target:* We approach this as a classification challenge, aiming to determine the central amino acid's identity by analyzing the surrounding atoms.

*Split:* We segregate environments based on the protein's topological category, as outlined in CATH 4.2 (Dawson et al., 2017), ensuring that environments from proteins of a similar class belong to the same divided dataset.

**Mutation Stability Prediction (MSP).**

*Impact:* Pinpointing mutations that reinforce protein interactions is crucial for crafting new proteins. Given that experimental methods to investigate these mutations are resource-intensive (Antikainen & Martin, 2005), there's a compelling need for streamlined computational approaches.

*Target:* We treat this as a binary classification challenge, aiming to determine if the complex's stability is enhanced due to the mutation.

*Split:* We categorize protein complexes ensuring that no protein in the evaluation set shares more than 30% sequence similarity with any protein in the instructional dataset.

### C.3 Multi-level tasks

**Protein Fold Classification.**

*Impact:* Protein fold classification is important in the study of the relationship between protein structure and protein evolution. The fold classes indicate protein secondary structure compositions, orientations and connection orders.

*Target:* The target is to predict the fold class or category of a protein based on its sequence or structure. This involves assigning the protein to one of several predefined fold types that represent its overall three-dimensional arrangement.

*Split:* We follow IEConv (Hermosilla et al., 2020) to conduct protein fold classification on the training/validation/test splits of the SCOPe 1.75 dataset (Hou et al., 2018), which in total contains 16,712 proteins with 1,195 fold classes. The 3D coordinates of the proteins were collected from the SCOPe 1.75 database (Murzin et al., 1995). The data set provides three different evaluation scenarios. 1) Fold, in which proteins from the same superfamily are not used during training. 2) Superfamily, in which proteins from the same family are not provided during training. 3) Family, in which proteins of the same family are available during training. Mean accuracy is used as the evaluation metric.

**Enzyme Reaction Classification.**

*Impact:* Enzyme Reaction Classification is essential for understanding how enzymes catalyze specific biochemical reactions. Accurate classification helps in predicting enzyme function, aiding drug discovery, metabolic engineering, and biotechnology, where enzymes are used to speed up chemical processes.

*Target:* The goal is to categorize enzymes based on the types of chemical reactions they catalyze, often following the Enzyme Commission (EC) number system, which defines reactions into distinct classes such as oxidoreductases, transferases, and hydrolases.

*Split:* We use the dataset from IEConv (Hermosilla et al., 2020), which includes 384 four-level EC classes and 29,215/2,562/5,651 proteins for training/validation/test, respectively.

**Gene Ontology Term Prediction (GO).**

*Impact:* Accurately predicting a protein's functions using Gene Ontology (GO) terms is pivotal for enhancing our understanding of biological systems. By categorizing proteins based on their specific functions, we can gain deeper insights into cellular processes and mechanisms.

*Target:* The challenge lies in predicting multiple GO terms associated with a protein, effectively making it a multi-label classification task. Specifically, we delve into three ontologies: biological process (BP) with 1,943 categories, molecular function (MF) boasting 489 categories, and cellular component (CC) encompassing 320 categories.

*Split:* The dataset's division earmarks 29,898 proteins for training, 3,322 for validation, and 3,415 for testing. The F max accuracy metric is harnessed to evaluate the predictions.

**Enzyme Commission Number Prediction (EC).**

*Impact:* Predicting the Enzyme Commission (EC) numbers efficiently and accurately plays a pivotal role in understanding enzyme functions and categorizations. This task deviates from merely classifying enzyme reactions and strives to pinpoint the specific three-level and four-level 538 EC numbers, adding granularity to the enzymatic categorization.

*Target:* The overarching goal of this task is to predict the detailed EC numbers associated with each enzyme, which is a multi-label classification task. Such predictions give valuable insights into the specific reactions and pathways these enzymes partake in.

*Split:* The dataset splits are aligned with those detailed in (Gligorijević et al., 2021). As an additional note, in tasks like GO term and EC number predictions, measures are taken to ensure that the test set is comprised only of PDB chains with a sequence identity that doesn't exceed $95\%$ compared to the training set, a standard adhered to in several studies such as (Wang et al., 2021b).

## D Details on Metrics

**Pearson correlation.** Pearson correlation (Pearson, 1900) is a statistic that measures the linear relationship between two continuous variables.

The mathematical formulation of Pearson correlation is given by:

$$R_P = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2}} \tag{6}$$

where $R_P$ is the Pearson correlation coefficient, $X_i$ and $Y_i$ are individual data points from the $X$ and $Y$ variables,$\bar{X}$ and $\bar{Y}$ are the averages of the $X$ and $Y$ variables.

**Spearman correlation.** Spearman correlation (Spearman, 1904) is a statistical measure of the strength and direction of association between two variables. It is a non-parametric method used to assess the monotonic relationship between variables, meaning it doesn't assume a linear relationship between the variables as the Pearson correlation does.

The mathematical formulation of Spearman correlation is given by:

$$R_S = 1 - \frac{6 \sum d^2}{n(n^2 - 1)} \tag{7}$$

where $R_S$ is the Spearman correlation coefficient, $d$ is the difference between the ranks of each pair of corresponding values, $n$ represents the number of data points.

**Root Mean Square Error.** Root Mean Square Error (RMSE) is a common metric used to evaluate the accuracy of a predictive model. It measures the average magnitude of the errors in a set of data.

RMSE is mathematically represented as:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{8}$$

where $y_i$ is the actual or observed value for data point $i$, $\hat{y}_i$ is is the predicted value for data point $i$

**AUROC.** Area Under the Receiver Operating Characteristic curve (AUROC) is a metric used to evaluate the performance of binary classification models. The ROC curve is a graphical representation of a model's ability to discriminate between the positive and negative classes across different threshold values. A higher AUROC suggests better model performance in distinguishing between the two classes.

**Protein-centric maximum F-score.** Protein-centric maximum F-score ($\mathrm{F}_{max}$) is the prediction probability for the $j$-th class of the $i$-th protein, $b_i^j \in \{0, 1\}$ is the corresponding binary class label and $J$ is the number of classes. F-Score is based on the precision and recall of the predictions for each protein.

$$\text{precision}_i(\lambda) = \frac{\sum_j^J ((p_i^J) \cap b_i^j)}{\sum_j^J (p_i^J \geq \lambda)}, \quad \text{recall}_i(\lambda) = \frac{\sum_j^J (p_i^J \geq \lambda)}{\sum_j^J b_i^j} \tag{9}$$

$\mathrm{F}_{max}$ mathematically represented as:

$$\mathrm{F}_{max} = \max_{x \in [0,1]} \left\{ \frac{2 \times \text{precision}(\lambda) \times \text{recall}(\lambda)}{\text{precision}(\lambda) + \text{recall}(\lambda)} \right\} \tag{10}$$

where $\text{precision}(\lambda)$ and $\text{recall}(\lambda)$ represent average precision and recall over all proteins. They are defined as follows:

$$\text{precision}(\lambda) = \frac{\sum_i^N \text{precision}_i(\lambda)}{\sum_i^N ((\sum_j^J (p_i^J \geq \lambda)) \geq 1)}, \quad \text{recall}(\lambda) = \frac{\sum_i^N \text{recall}_i(\lambda)}{N} \tag{11}$$