DeepProtein: Deep Learning Library and Benchmark for Protein Sequence Learning

Abstract

In recent years, deep learning has revolutionized the field of protein science, enabling advancements in predicting protein properties, structural folding and interactions. This paper presents DeepProtein, a comprehensive and user-friendly deep learning library specifically designed for protein-related tasks. DeepProtein integrates a couple of state-of-the-art neural network architectures, which include convolutional neural network (CNN), recurrent neural network (RNN), transformer, graph neural network (GNN), and graph transformer (GT). It provides user-friendly interfaces, facilitating domain researchers in applying deep learning techniques to protein data. Also, we curate a benchmark that evaluates these neural architectures on a variety of protein tasks, including protein function prediction, protein localization prediction, and protein-protein interaction prediction, showcasing its superior performance and scalability. Additionally, we provide detailed documentation and tutorials to promote accessibility and encourage reproducible research. This is a library that is extended from a well-known drug discovery library, DeepPurpose. The library is publicly available at https://anonymous.4open.science/r/DeepProtein-F8FE.

1 Introduction

Understanding the representation of proteomics is vital in developing traditional biological and medical progress [Wu et al., 2022b, Fu et al., 2024], multi-omics genomics [Wu et al., 2022a], and curing human diseases [Chen et al., 2024c,b]. Being the working house of the cell, it provides lots of functions that support human's daily life, such as catalyzing biochemical reactions that happen in the body as a role of enzymes and providing helpful immune responses against detrimental substance that acts as immunoglobulin. Under the necessity of analyzing those useful proteins, several related protein databases are available to researchers [Berman et al., 2000, Bairoch and Apweiler, 2000, Consortium, 2015, Pontén et al., 2008]. Apart from the 2D database, some recent 3D Protein Database used AlphaFold 2.0 [Jumper et al., 2021] is important to better assist in learning those representations in 3d-dimensional space. The success of AlphaFold 2.0 has sparked a significant increase in interest in using machine learning techniques for protein learning tasks, of which the goal is to improve our understanding of proteins' biochemical mechanisms.

Deep learning has shown its power in protein learning tasks, including protein-protein interaction [Gainza et al., 2020], protein folding [Jumper et al., 2021, Lu, 2022, Panou and Reczko, 2020, Chen et al., 2016], protein-ligand interaction [Li et al., 2021b, Xia et al., 2023], and protein function and property prediction [Gligorijević et al., 2021, Sevgen et al., 2023]. Convolutional neural network (CNN) [Shanehsazzadeh et al., 2020] and TAPE Transformer [Rao et al., 2019] on proteins are models with regard to protein sequence-to-sequence (Seq2Seq) learning, including the pretrained transformer such as ProtBert [Brandes et al., 2022]. With the help of 3D information intrinsics, graph neural networks (GNN) are applied, especially some pretrained GNN models [Jing et al., 2020, Zhang et al., 2022], and graph transformers [Yuan et al., 2022, Gu et al., 2023]. While transformer claims to be the

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

state-of-the-art in the previous benchmark [Xu et al., 2022], the analysis of potential powerful deep learning models such as GNN models and graph transformer models are neglected, which prompts us to integrate those methods in our benchmark.

Challenges. Previous benchmarks related to molecular learning have offered valuable insights regarding their respective libraries and implementation interfaces. DeepPurpose[1] [Huang et al., 2020] has provided an interface that implements the task with a majority of drug discovery tasks, which only has protein-protein interaction and protein function prediction implemented. Datasets on proteins are lacking as well. TorchProtein[2] [Xu et al., 2022], also named as PEER, implemented most of the tasks in the protein field. In terms of models, the focus has largely been on *sequential* learning methods: Convolutional Neural Networks (CNNs), Transformers, and ESM architectures. This suggests that there are still many *structural* methods available for consideration. Furthermore, PEER interface is not user-friendly without prior domain knowledge in graphs and biochemistry. This situation presents an opportunity to improve the existing interface with regard to simplicity and comprehensibility.

Solutions. To address these challenges, in this paper, we propose `DeepProtein`, which aims to benchmark mainstream and cutting-edge deep learning models on a wide range of AI-solvable protein sequence learning tasks. We investigate the performance of various deep learning models on a wide range of protein sequence learning tasks. We analyze each method's advantages and disadvantages when performing each task (working as the explainer for each task). We have provided user-friendly and well-wrapped interfaces to facilitate domain experts' research.

Contribution. The major contributions of our paper are summarized as

- **Comprehensive Benchmarking**: We curate a benchmark to evaluate the performance of 8 neural network architectures (including CNNs, RNNs, transformers, and various graph neural networks) on these 7 essential protein learning tasks (including protein function prediction, protein localization prediction, protein-protein interaction prediction, antigen epitope prediction, antibody paratope prediction, CRISPR repair outcome prediction, and antibody developability prediciton), demonstrating superior performance and scalability.
- **User-friendly Library**: We develop `DeepProtein`, a specialized deep learning library that integrates these neural network architectures (CNNs, RNNs, transformers, and graph neural networks) on these 7 protein-related tasks such as protein function prediction, protein localization prediction, protein-protein interaction prediction, antigen epitope prediction, and antibody developability prediciton. For each task, our library enables one command line to run every method.
- **Enhanced Accessibility**: We provide extensive documentation and tutorials to facilitate user engagement and promote reproducible research, building upon the foundation of the well-known drug discovery library, DeepPurpose [Huang et al., 2020].

2 Related Works

Benchmarks and libraries are crucial in AI-based therapeutic science, e.g., multi-omics data [Lu, 2018], protein learning [Xu et al., 2022], small-molecule drug discovery [Gao et al., 2022, Xu et al., 2024], and drug development (clinical trial) [Chen et al., 2024a, Wang et al., 2024]. They provide standardized metrics for evaluating the performance of various algorithms and models. These benchmarks enable researchers to compare different approaches systematically, ensuring reproducibility and reliability of results.

In this section, we briefly discuss the benchmark studies in this area. Proteins are vital in drug discovery because they often serve as the primary targets for therapeutic agents, influencing disease mechanisms and biological pathways. Additionally, proteins play key roles in various cellular processes, making them essential for identifying potential drug candidates and biomarkers in the drug development pipeline. A couple of protein learning benchmarks are developed, including PEER [Xu et al., 2022], DeepPurpose [Huang et al., 2020], FLIP [Dallago et al., 2021], TAPE [Rao et al., 2019]. Table 1 compares `DeepProtein` with existing AI-based protein learning benchmarks. We extend the scope of existing protein learning benchmarks by incorporating more protein learning datasets, more cutting-edge deep learning models, and enhancing user-friendliness.

[1] `https://github.com/kexinhuang12345/DeepPurpose`

[2] `https://github.com/DeepGraphLearning/PEER_Benchmark`

Table 1: Comparison of benchmark studies on protein sequence learning. TDC provides AI-ready datasets but does not contain protein learning benchmarks (denoted ⋄).

Datasets	DeepPurpose	FLIP	TAPE	PEER	TDC (data only)	DeepProtein
References	[Huang et al., 2020]	[Dallago et al., 2021]	[Rao et al., 2019]	[Xu et al., 2022]	[Huang et al., 2021]	ours
Fluorescence	×	✓	✓	✓	×	✓
β-lactamase	×	×	×	✓	×	✓
Solubility	×	×	×	✓	×	✓
Stability	×	✓	✓	✓	×	✓
Subcellular (Binary)	×	×	×	✓	×	✓
PPI Affinity	×	×	×	✓	×	✓
Yeast PPI	×	×	×	✓	×	✓
Human PPI	×	×	×	✓	×	✓
IEDB	×	×	×	×	⋄	✓
PDB-Jespersen	×	×	×	×	⋄	✓
SAbDab-Liberis	×	×	×	×	⋄	✓
TAP	×	×	×	×	⋄	✓
SAbDab-Chen	×	×	×	×	⋄	✓
CRISPR-Leenay	×	×	×	×	⋄	✓
user-friendly	✓	×	×	×	✓	✓

3 DeepProtein Library and Benchmark

3.1 AI-solvable Protein Problems

In this section, we elaborate on a couple of AI-solvable protein problems and the related datasets.

- **Protein Function Prediction**. Protein function prediction involves determining the biological roles and activities of proteins based on their sequences or structures. This process is crucial for understanding cellular mechanisms and interactions, as a protein's function is often linked to its sequence composition and the context of its cellular environment. Machine learning algorithms are employed to analyze known protein databases, identifying patterns and features that correlate with specific functions. Accurate predictions can facilitate drug discovery, help elucidate disease mechanisms, and support advancements in synthetic biology by providing insights into how proteins can be engineered for desired activities [Zhang et al., 2021]. We consider the following datasets.
  - **Fluorescence** [Sarkisyan et al., 2016]. Protein fluorescence refers to the phenomenon where certain proteins can emit light of a specific wavelength when excited by light of a shorter wavelength. It is a widely used technique to study protein structure, dynamics, interactions, and function. The dataset consists of 54,025 protein sequences with real-valued groundtruth. The label is the logarithm of fluorescence intensity.
  - **Stability** [Rocklin et al., 2017]. Protein stability is the capacity of a protein to preserve its three-dimensional structure and functional characteristics across different environmental conditions. This stability is essential for the proper functioning and longevity of proteins within biological systems. A protein's stability is influenced by its ability to withstand denaturation, aggregation, and degradation. The dataset comprises 68,934 protein sequences with real-valued groundtruth.
  - **β-lactamase** [Gray et al., 2018]. This task aims to predict the increased activity of β-lactamase, the most common enzyme that provides gram-negative bacteria with resistance to beta-lactam antibiotics through single mutations. The dataset consists of 5,198 protein sequences with real-valued groundtruth. The groundtruth refers to the experimentally determined fitness score, which measures the scaled mutation effect for each mutant.
  - **Solubility** [Khurana et al., 2018]. Protein solubility is the capacity of a protein to dissolve or remain dispersed in a solution. This property is crucial for determining how the protein behaves and functions in various biological and industrial contexts. Several factors influence a protein's solubility, including its amino acid composition, ionic strength, pH, temperature, and the presence of other molecules in the solution. The dataset consists of 71,419 protein sequences with binary labels.
- **Protein Localization Prediction**. Accurate localization predictions can enhance drug development by informing target identification and improving therapeutic efficacy, particularly in treating diseases linked to protein mislocalization. Additionally, insights gained from localization predictions facilitate the mapping of biological pathways, aiding in the identification of new therapeutic targets and potential disease mechanisms.

- **Subcellular** [Almagro Armenteros et al., 2017]. The task predicts the location of a natural protein within the cell. The dataset consists of 13,961 data samples with categorical labels (10 classes, $\{0, 1, 2, \cdots, 9\}$).
- **Binary** [Almagro Armenteros et al., 2017]. It is a simpler version of the previous task (10-category classification), where the model is trained to roughly forecast each protein as either "membrane-bound" or "soluble" (i.e., binary classification). The dataset comprises 8,634 data samples with binary labels.

- **Protein-Protein Interaction (PPI).** Proteins are the essential functional units in human biology, but they seldom operate in isolation; rather, they typically interact with one another to perform various functions. Understanding protein-protein interactions (PPIs) is crucial for identifying potential therapeutic targets for disease treatment. Traditionally, determining PPI activity requires costly and time-consuming wet-lab experiments. PPI prediction seeks to forecast the activity of these interactions based on the amino acid sequences of paired proteins.
  - **PPI Affinity** [Moal and Fernández-Recio, 2012]. It consists of 2,682 protein-protein pairs with real-valued groundtruth.
  - **Yeast** [Guo et al., 2008]. The dataset comprises 2,172 protein-protein pairs with binary labels.
  - **Human PPI** [Pan et al., 2010]. The dataset comprises 7,348 protein-protein pairs with binary labels.

- **Epitope Prediction**. An epitope, also known as an antigenic determinant, is the region of a pathogen that can be recognized by antibodies and cause an adaptive immune response. The epitope prediction task is to distinguish the active and non-active sites from the antigen protein sequences. Identifying the potential epitope is of primary importance in many clinical and biotechnologies, such as vaccine design and antibody development, and for our general understanding of the immune system [Wu et al., 2024]. In epitope prediction, the machine learning model makes a binary prediction for each amino acid residue. This is also known as ***residue-level classification***.
  - **Immune Epitope Database (IEDB)** [Vita et al., 2019]. It consists of 3,159 antigens with binary labels on each amino acid. The label indicates whether the amino acid belongs to the epitope, i.e., active position in binding. It can be downloaded from TDC (`https://tdcommons.ai/single_pred_tasks/epitope/`).
  - **PDB-Jespersen** [Jespersen et al., 2017]. It consists of 447 antigens with binary labels on each amino acid. It is curated by [Jespersen et al., 2017] and is extracted from PDB (Protein Data Bank). It can be downloaded from TDC (`https://tdcommons.ai/single_pred_tasks/epitope/`).

- **Paratope Prediction**. Antibodies, or immunoglobulins, are large, Y-shaped proteins that can recognize and neutralize specific molecules on pathogens, known as antigens. They are crucial components of the immune system and serve as valuable tools in research and diagnostics. The paratope, also referred to as the antigen-binding site, is the region that specifically binds to the epitope. While we have a general understanding of the hypervariable regions responsible for this binding, accurately identifying the specific amino acids involved remains a challenge. This task focuses on predicting which amino acids occupy the active positions of the antibody that interact with the antigen. In paratope prediction, the machine learning model makes a binary prediction for each amino acid residue. This is also known as ***residue-level classification***.
  - **SAbDab-Liberis** [Liberis et al., 2018] is curated from SAbDab [Dunbar et al., 2014]. It consists of 1,023 antibody chain sequences; each antibody contains both heavy and light chain sequences. It can be downloaded from TDC (`https://tdcommons.ai/single_pred_tasks/paratope/#sabdab-liberis-et-al`).

- **Antibody Developability Prediction**. Immunogenicity, instability, self-association, high viscosity, polyspecificity, and poor expression can hinder an antibody from being developed as a therapeutic agent, making early identification of these issues crucial. The goal of antibody developability prediction is to predict an antibody's developability from its amino acid sequences. A fast and reliable developability predictor can streamline antibody development by minimizing the need for wet lab experiments, alerting chemists to potential efficacy and safety concerns, and guiding necessary modifications. While previous methods have used 3D structures to create accurate developability indices, acquiring 3D information is costly. Therefore, a machine learning approach that calculates developability based solely on sequence data is highly advantageous.

  - **TAP** [Raybould et al., 2019]. It contains 242 antibodies with real-valued groundtruth. Given the sequences of the antibody's heavy and light chains, we need to predict its developability (continuous value). The input consists of a list containing two sequences: the first representing the heavy chain and the second representing the light chain. It can be downloaded from TDC (`https://tdcommons.ai/single_pred_tasks/develop/`).
  - **SAbDab-Chen** [Chen et al., 2020]. It consists of 2,409 antibodies with real-valued groundtruth. It is extracted from SAbDab (the structural antibody database)[3], which is a database containing all the antibody structures available in the PDB (Protein Data Bank), annotated and presented in a consistent fashion [Dunbar et al., 2014]. Given the antibody's heavy chain and light chain sequence, predict its developability (binary label). It can be downloaded from TDC (`https://tdcommons.ai/single_pred_tasks/develop/`).
- **CRISPR Repair Outcome Prediction**. CRISPR-Cas9 is a gene editing technology that allows for the precise deletion or modification of specific DNA regions within an organism. It operates by utilizing a custom-designed guide RNA that binds to a target site upstream, which results in a double-stranded DNA break facilitated by the Cas9 enzyme. The cell responds by activating DNA repair mechanisms, such as non-homologous end joining, leading to a range of gene insertion or deletion mutations (indels) of varying lengths and frequencies. This task aims to predict the outcomes of these repair processes based on the DNA sequence. Gene editing marks a significant advancement in the treatment of challenging diseases that conventional therapies struggle to address, as demonstrated by the FDA's recent approval of gene-edited T-cells for the treatment of acute lymphoblastic leukemia. Since many human genetic variants linked to diseases arise from insertions and deletions, accurately predicting gene editing outcomes is essential for ensuring treatment effectiveness and reducing the risk of unintended pathogenic mutations.
  - **CRISPR-Leenay** [Leenay et al., 2019]. The dataset comprises 1,521 DNA sequences (including guide RNA and PAM) with five measured repair outcomes, assessed across various donor populations of primary T cells. It can be downloaded from TDC (`https://tdcommons.ai/single_pred_tasks/CRISPROutcome/`).

In this library, we follow the train-validation-test split in PEER benchmark [Xu et al., 2022] and TDC [Huang et al., 2022]. Each individual split is reported from Table 2 to 6.

3.2 Cutting-edge Deep Learning Methods

At the core of deep learning lies the artificial neural network, a machine learning technique inspired by the architecture and functionality of the human brain. What distinguishes deep learning from other machine learning approaches is its exceptional capacity to recognize and analyze complex, nonlinear patterns in data, leading to enhanced performance and accuracy. Concretely, we incorporate several cutting-edge neural network architectures into two groups: 1) sequential-based learning and 2) structural-based learning. Detailed model architectures are described as follows:

Sequential based learning It generally takes a sequence as an input, uses one-hot encoding to pre-encode the input characters. Such learning methods include convolutional neural network, recurrent neural network and transformers.

- **Convolutional Neural Network (CNN) (One-dimensional)** captures the local patterns in the data features, commonly used to analyze images and text. **(One-dimensional) Convolutional neural network (CNN)** takes amino acid sequences as the input. CNN has four layers; the number of filters for the four layers is 32, 64, and 96 respectively. The kernel sizes are 4, 8, and 12, respectively. The convolutional layer is followed by a one-layer MLP (multi-layer perceptron) to produce the prediction, which is a scalar.
- **Recurrent Neural Network (RNN)** models sequence data and captures the long-term dependencies in the sequence data. RNN has two well-known variants: long short-term memory networks (LSTMs) [Hochreiter and Schmidhuber, 1996] and gated recurrent units (GRU) [Cho et al., 2014]. The difference between GRU and LSTM is that GRU simplifies LSTM by removing the cell state and reducing the number of gates. We use a two-layer bi-directional GRU following three-layer CNN as the neural network architecture. The dimension of the hidden state in GRU is set to 64. ReLU function is applied after each GRU or CNN layer.

[3]It is publicly available `http://opig.stats.ox.ac.uk/webapps/newSAbDab/SAbDab/`.

- **Transformer** [Vaswani et al., 2017] architecture leverages the power of self-attention mechanisms and parallel computation to enhance the neural network's capability and efficiency in handling sequence data. We use the transformer encoder to represent the amino acid sequence. Two layers of transformer architectures are stacked. The dimension of embedding in the transformer is set to 64. The number of attention heads is set to 4. ReLU function is applied after each self-attention layer. LayerNorm is applied after MLP layers.

  Structural-based learning It generally transforms the input sequence into a valid SMILES string, then transforms the chemical substance into a graph. Then, graph filters are learned toward the input graph signal. Such learning methods are widely called Graph Neural Networks. Recently, graph transformers have shown their power in protein function prediction, and we included them as a part of structural-based learning.

- **Graph Neural Network (GNN)** is a neural network architecture designed to process graph-structured data that takes input from nodes and edges, facilitating the flow of information between connected components to capture their interactions. It learns vector representations for both individual graph nodes and the overall graph structure. We consider the following GNN variants:
  - **Graph Convolutional Network (GCN)** [Kipf and Welling, 2016]. GCN is a GNN variant that iteratively updates the node representation by aggregating the information from its neighbors. GCN has three layers, and the node embedding dimension is set to 64. After GCN, all the node embeddings are aggregated with a readout function (Weighted Sum and Max) to get graph-level embedding, followed by a one-layer MLP to get the final prediction. BatchNorm is applied after MLP layers.
  - **Graph Attention Network (GAT)** [Velickovic et al., 2018]. GAT employs an attention mechanism to introduce anisotropy into the neighborhood aggregation function. This network features a multi-headed architecture that enhances its learning capacity. The node embedding dimension is 64. Readout function is the same as the one deployed in GCN model.
  - **Message Passing Neural Network (MPNN)** [Gilmer et al., 2017]. MPNN is a GNN variant that considers passing messages (and modeling interactions) between both edges and nodes based on their neighbors. Edge features are included necessarily compared with GCN and GAT. Readout function is Sum And Max. Node and edge embedding dimension is 64.
  - **Neural Fingerprint (NeuralFP)** [Duvenaud et al., 2015]. NeuralFP uses Graph convolutional network (GCN) [Kipf and Welling, 2016] to learn a neural network-based molecular embedding (also known as molecular *neural fingerprint*, or NeuralFP) from a large amount of molecule data without labels. The neural fingerprint is essentially a real-valued vector, also known as embedding. Then, the neural fingerprint is fixed and fed into a three-layer MLP to make the prediction. Node embedding dimension is 64. BatchNorm is applied after MLP layers.
  - **Attentive Fingerprint (AttentiveFP)** [Xiong et al., 2019]. AttentiveFP is a variant of graph neural networks that is enhanced by the attention mechanism when evaluating node and edge embedding. The model consists of three AttentiveFP layers with individual readout function: AttentiveFP readout. Node and edge embedding dimension is 64.

- **Graph Transformer** [Yun et al., 2019] is a type of neural network architecture designed to process graph-structured data by leveraging self-attention mechanisms. They extend the principles of traditional transformers, enabling them to capture the relationships and interactions between nodes in a graph effectively.
  - **Path-Augmented Graph Transformer** (**PAGTN**) [Chen et al., 2019]. It used augmented path features to capture long-range ($>$1 hop) graph properties. The model consists of 5 PAGTN layers with LeakyReLU activation. Node embedding dimension is 64.
  - **Graphormer** [Ying et al., 2021]. It utilized transformer on graphs with spatial, centrality, and edge encoding. For simplicity and scalability on large graphs, we only deployed one Graphormer layer with ReLU activation. Node embedding dimension is 64. LayerNorm is applied after MLP layers.

Training setup. For all the models, the maximal training epoch number is set to 100. We employed the Adam optimizer [Kingma and Ba, 2014] for training, with a default learning rate of 0.0001 for sequence-based learning and 0.00001 for structural-based learning. The batch size is equal to 32. More detailed hyper-parameter setups are listed in Table 7 in the appendix.

Table 2: Results of protein function prediction. The ↑ symbol indicates that higher values are better for the corresponding metric. For each method, we employed five different random seeds to perform independent runs, reporting the average results along with their standard deviations. On each task, the best method is **bolded**, and the second best is underlined. We use "**" to denote the method that achieves statistically better results than all the other methods (through statistical tests).

Model # train/valid/test	**Fluorescence** (ρ ↑) 21446 / 5362 / 27217	**Stability** (ρ ↑) 53571 / 2512 / 12851	β-**lactamase** (ρ ↑) 4158 / 520 / 520	**Solubility** (PR-AUC ↑) 62478 / 6942 / 1999
CNN	**0.680 ± 0.001**	**0.715 ± 0.025****	**0.721 ± 0.020****	74.61 ± 0.55
CNN-RNN	0.678 ± 0.001	0.638 ± 0.021	0.695 ± 0.012	74.10 ± 1.48
Transformer	0.648 ± 0.001	0.442 ± 0.030	0.348 ± 0.002	**78.86 ± 0.46****
GCN	0.397 ± 0.002	0.392 ± 0.027	0.417 ± 0.001	69.26 ± 0.73
GAT	0.251 ± 0.002	0.165 ± 0.026	0.196 ± 0.013	62.44 ± 0.01
NeuralFP	0.413 ± 0.011	0.333 ± 0.041	0.133 ± 0.020	78.74 ± 0.24
AttentiveFP	0.260 ± 0.003	0.254 ± 0.022	0.058 ± 0.011	78.39 ± 0.19
MPNN	0.237 ± 0.020	0.110 ± 0.021	0.068 ± 0.015	62.53 ± 0.31
PAGTN	0.188 ± 0.036	0.266 ± 0.016	0.092 ± 0.018	61.33 ± 0.91
Graphormer	0.067 ± 0.002		0.103 ± 0.018	-

3.3 Experimental Setup and Implementation Details

Code Base. This library is an extension of the well-established drug discovery library, DeepPurpose [Huang et al., 2020], building upon its foundational capabilities to offer enhanced features for protein-related tasks. By leveraging the strengths of DeepPurpose, this new library provides additional tools and functionalities tailored specifically for protein science. The library is publicly available at https://anonymous.4open.science/r/DeepProtein-F8FE.

Hardware Configuration. All experiments that are mentioned in this paper were trained on a 40GB NVIDIA A40 and a 24GB NVIDIA RTX 3090. The parameters we provide have ensured the scalable training on these two types of GPUs. When running GNNs on protein localization tasks, we observed a large portion of GPU memory occupied irregularly, so we recommend cutting down the size of the number of workers from 8 to 4 or batch size from 32 to 8 or even smaller to potentially avoid GPU out-of-memory (OOM) problems.

Software Configuration. The library is implemented in Python 3.9, PyTorch 2.3.0, PyTDC 0.4.1 [Huang et al., 2021], DeepPurpose 0.1.5 [Huang et al., 2020], and RDKit 2023.9.6 [Landrum et al., 2006], scikit-learn 1.2.2 [Pedregosa et al., 2011], and DGLlife 0.3.2 [Li et al., 2021a]. Besides, wandb is included in `DeepProtein` so that researchers could observe the visualization of training curves and test results easily.

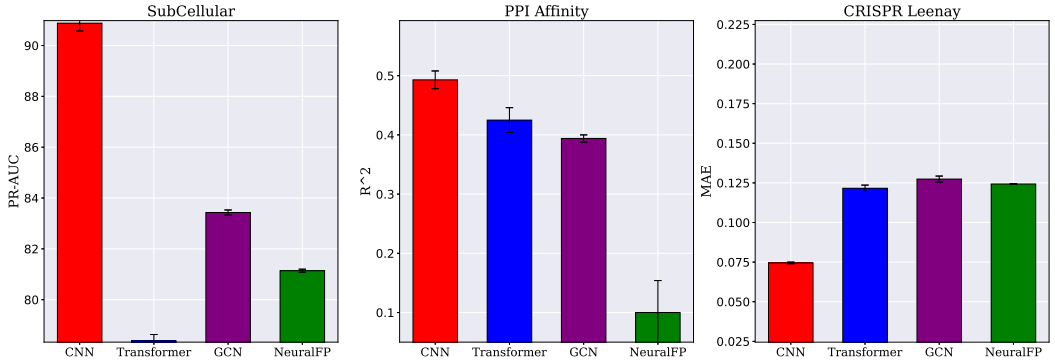

Figure 1: Performance of selected models including CNN, Transformer, GCN and NeuralFP on SubCellular, PPI Affinity and CRISPR Leenay. We observe that CNN performs significantly better than GCN.

Table 3: Results of protein localization prediction.

Model # train/valid/test	**Subcellular** (Acc ↑) 8,945 / 2248 / 2768	**Binary** (PR-AUC ↑) 5161 / 1727 / 1746
CNN	$\underline{50.18 \pm 1.21}$	$\underline{90.88 \pm 0.31}$
CNN-RNN	$\mathbf{52.58 \pm 0.11}$**	$\mathbf{91.43 \pm 0.45}$**
Transformer	42.63 ± 0.68	78.38 ± 0.25
GCN	47.45 ± 0.47	83.43 ± 0.10
GAT	45.14 ± 0.10	82.15 ± 0.41
NeuralFP	45.20 ± 0.49	81.14 ± 0.06
AttentiveFP	42.38 ± 1.25	80.58 ± 0.30

Table 4: Results of Protein-Protein Interaction (PPI).

Model # train/valid/test	**PPI Affinity** (R^2 ↑) 2127 / 212 / 343	**Yeast** (PR-AUC ↑) 1668 / 131 / 373	**Human PPI** (PR-AUC ↑) 6844 / 277 / 227
CNN	$\underline{0.493 \pm 0.015}$	51.93 ± 0.92	70.37 ± 1.22
CNN-RNN	$\mathbf{0.584 \pm 0.026}$**	53.28 ± 0.85	70.45 ± 2.68
Transformer	0.425 ± 0.021	53.79 ± 1.07	59.36 ± 4.00
GCN	0.394 ± 0.006	$\mathbf{58.98 \pm 0.72}$	$\mathbf{82.21 \pm 1.13}$
GAT	0.230 ± 0.001	53.72 ± 0.39	77.63 ± 3.13
NeuralFP	0.100 ± 0.054	$\underline{57.00 \pm 1.51}$	$\underline{80.11 \pm 1.25}$

Table 5: Results of epitope and paratope prediction (*residue-level classification*).

Model # train/valid/test	**IEDB** (ROC-AUC ↑) 2211 / 316 / 632	**PDB-Jespersen** (ROC-AUC ↑) 313 / 45 / 89	**SAbDab-Liberis** (ROC-AUC ↑) 716 / 102 / 205
CNN	54.03 ± 0.02	$\mathbf{74.46 \pm 0.21}$**	$\underline{90.85 \pm 0.08}$
CNN-RNN	$\underline{55.47 \pm 0.23}$	$\underline{70.10 \pm 0.97}$	$\mathbf{96.75 \pm 0.10}$**
Transformer	$\mathbf{59.79 \pm 0.06}$**	60.10 ± 0.32	64.77 ± 0.11

Table 6: Results of antibody developability prediction (TAP and SAbDab-Chen) and CRISPR repair outcome prediction (CRISPR-Leenay).

Model # train/valid/test	**TAP** (MAE ↓) 169 / 24 / 48	**SAbDab-Chen** (MAE ↓) 1686 / 241 / 482	**CRISPR-Leenay** (MAE ↓) 1065 / 152 / 304
CNN	3.217 ± 0.026	$\mathbf{0.219 \pm 0.006}$	$\mathbf{0.0745 \pm 0.0005}$
CNN-RNN	$\mathbf{0.712 \pm 0.069}$**	$\underline{0.226 \pm 0.001}$	$\underline{0.0755 \pm 0.0010}$
Transformer	3.476 ± 0.004	0.238 ± 0.012	0.1216 ± 0.0020
GCN	2.761 ± 0.155	0.326 ± 0.015	0.1274 ± 0.0019
GAT	$\underline{2.675 \pm 0.022}$	0.310 ± 0.010	0.1232 ± 0.0001
NeuralFP	3.436 ± 0.015	0.253 ± 0.011	0.1243 ± 0.0001

3.4 Results & Analysis

For each method, we used five different random seeds to conduct independent runs and reported the average results and their standard deviations. The results of protein function prediction are reported in Table 2.

Statistical Test. We also conduct statistical tests to confirm the superiority of the best-performed method compared with the second-best baseline method. The hypothesis is that the accuracies of the best method are the same as those of the baseline method. Student's T-test is used with significance level alpha as 1% to calculate the p-values. When the p-values are below the 0.05 threshold, we reject the hypothesis and accept the alternative hypothesis, i.e., the best method is statistically significant compared with the second-best method. We use "**" to denote the method that achieves statistically better results than all the other methods (pass statistical tests).

Key Observations. We summarize the following key observations as takeaways.

- Sequence-based neural architectures are powerful compared with graph-based neural architectures. Sequence-based neural architectures, such as CNN, RNN, and transformer, obtain superior performance in most protein sequence learning tasks. Specifically, in 12 out of all the 15 tasks across various protein sequence learning tasks, the sequence-based model (CNN, RNN, and transformer) takes the top-2 position.

- Among all the 12 GNN-solvable tasks (except residue-level classification), graph neural networks (GNN) obtain the best accuracy among all the methods only in two protein-protein interaction (PPI) tasks, including Yeast and Human PPI.
- Among all the graph neural networks (GNNs) across the whole 12 GNN-solvable tasks (except residue-level classification), the earliest variant, GCN [Kipf and Welling, 2016], achieves the best performance in 9 tasks.
- **Stability.** From the learning curve (Figure 6-9), we find that GNN's training curve is not stable. In contrast, the sequence-based models, including CNN, RNN, and transformer, converge more stably from the learning curve. This could be observed from Figure 6 to Figure 8. On the contrary, training is more stable, fast and accurate for GAT when it comes to TAP dataset.
- **Computational complexity**. The runtime and memory costs are reported in Figure 2, 4 and 5. We find that GNN-based models are typically computationally inefficient. The key reason behind this is that GNN utilizes molecular graph as the feature, where each atom corresponds to a node and each chemical bond corresponds to an edge. While another model, such as CNN, RNN, and transformer, uses amino acid sequences as the input feature.

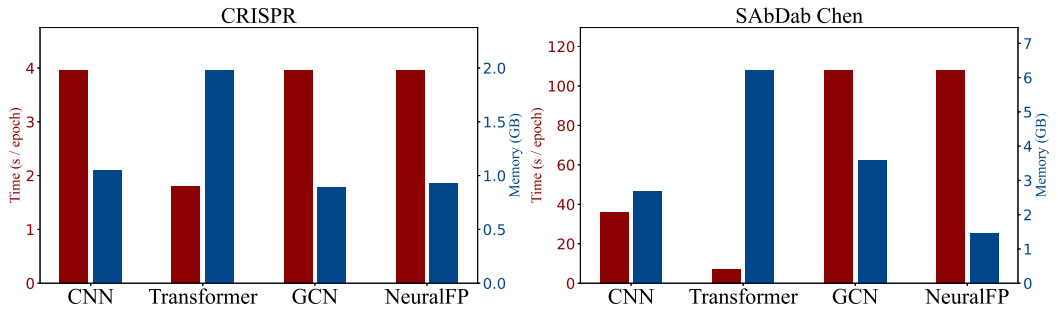

Figure 2: Empirical computational complexity on CRISPR and SAbDab Chen measured by runtime and memory.

4 Conclusion

In this paper, we have developed `DeepProtein`, which marks a significant advancement in the application of deep learning to protein science, providing researchers with a powerful and flexible tool to tackle various protein-related tasks. By integrating multiple state-of-the-art neural network architectures and offering a comprehensive benchmarking suite, `DeepProtein` empowers users to explore and optimize their models effectively. The detailed documentation and tutorials further enhance accessibility, promoting widespread adoption and reproducibility in research. As the field of proteomics continues to evolve, `DeepProtein` stands to contribute substantially to our understanding of protein functions, localization, and interactions, ultimately driving forward discoveries that can impact biotechnology and medicine.

References

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.

Amos Bairoch and Rolf Apweiler. The swiss-prot protein sequence database and its supplement trembl in 2000. *Nucleic acids research*, 28(1):45–48, 2000.

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.

Nadav Brandes, Dan Ofer, Yam Peleg, Nadav Rappoport, and Michal Linial. Proteinbert: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110, 2022.

Benson Chen, Regina Barzilay, and Tommi Jaakkola. Path-augmented graph transformer network. *arXiv preprint arXiv:1905.12712*, 2019.

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.

Jintai Chen, Yaojun Hu, Yue Wang, Yingzhou Lu, Xu Cao, Miao Lin, Hongxia Xu, Jian Wu, Cao Xiao, Jimeng Sun, et al. Trialbench: Multi-modal artificial intelligence-ready clinical trial datasets. *arXiv preprint arXiv:2407.00631*, 2024a.

Tianyi Chen, Nan Hao, Yingzhou Lu, and Capucine Van Rechem. Uncertainty quantification on clinical trial outcome prediction. *arXiv preprint arXiv:2401.03482*, 2024b.

Tianyi Chen, Nan Hao, Capucine Van Rechem, Jintai Chen, and Tianfan Fu. Uncertainty quantification and interpretability for clinical trial approval prediction. *Health Data Science*, 4:0126, 2024c.

Xingyao Chen, Thomas Dougherty, Chan Hong, Rachel Schibler, Yi Cong Zhao, Reza Sadeghi, Naim Matasci, Yi-Chieh Wu, and Ian Kerman. Predicting antibody developability from sequence using machine learning. *biorxiv*, pages 2020–06, 2020.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN Encoder–Decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Stroudsburg, PA, USA, 2014. Association for Computational Linguistics.

UniProt Consortium. Uniprot: a hub for protein information. *Nucleic acids research*, 43(D1): D204–D212, 2015.

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*, pages 2021–11, 2021.

James Dunbar, Konrad Krawczyk, Jinwoo Leem, Terry Baker, Angelika Fuchs, Guy Georges, Jiye Shi, and Charlotte M Deane. SAbDab: the structural antibody database. *Nucleic acids research*, 42(D1):D1140–D1146, 2014.

David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. *NeurIPS*, 2015.

Yi Fu, Yingzhou Lu, Yizhi Wang, Bai Zhang, Zhen Zhang, Guoqiang Yu, Chunyu Liu, Robert Clarke, David M Herrington, and Yue Wang. Ddn3. 0: Determining significant rewiring of biological network structure with differential dependency networks. *Bioinformatics*, page btae376, 2024.

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.

Wenhao Gao, Tianfan Fu, Jimeng Sun, and Connor W Coley. Sample efficiency matters: benchmarking molecular optimization. *Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*, 2022.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pages 1263–1272. PMLR, 2017.

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.

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.

Zhonghui Gu, Xiao Luo, Jiaxiao Chen, Minghua Deng, and Luhua Lai. Hierarchical graph transformer with contrastive learning for protein function prediction. *Bioinformatics*, 39(7):btad410, 2023.

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.

Sepp Hochreiter and Jürgen Schmidhuber. Lstm can solve hard long time lag problems. *Advances in neural information processing systems*, 9, 1996.

Kexin Huang, Tianfan Fu, Lucas M Glass, Marinka Zitnik, Cao Xiao, and Jimeng Sun. Deeppurpose: a deep learning library for drug–target interaction prediction. *Bioinformatics*, 36(22-23):5545–5547, 2020.

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 therapeutics. *NeurIPS Track Datasets and Benchmarks*, 2021.

Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Artificial intelligence foundation for therapeutic science. *Nature Chemical Biology*, pages 1–4, 2022.

Martin Closter Jespersen, Bjoern Peters, Morten Nielsen, and Paolo Marcatili. Bepipred-2.0: improving sequence-based b-cell epitope prediction using conformational epitopes. *Nucleic acids research*, 45(W1):W24–W29, 2017.

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael JL Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. *arXiv preprint arXiv:2009.01411*, 2020.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

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.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2014.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *The International Conference on Learning Representations (ICLR)*, 2016.

Greg Landrum et al. Rdkit: Open-source cheminformatics, 2006.

Ryan T Leenay, Amirali Aghazadeh, Joseph Hiatt, David Tse, Theodore L Roth, Ryan Apathy, Eric Shifrut, Judd F Hultquist, Nevan Krogan, Zhenqin Wu, et al. Large dataset enables prediction of repair after crispr–cas9 editing in primary t cells. *Nature biotechnology*, 37(9):1034–1037, 2019.

Mufei Li, Jinjing Zhou, Jiajing Hu, Wenxuan Fan, Yangkang Zhang, Yaxin Gu, and George Karypis. Dgl-lifesci: An open-source toolkit for deep learning on graphs in life science. *ACS Omega*, 2021a.

Yibo Li, Jianfeng Pei, and Luhua Lai. Structure-based de novo drug design using 3d deep generative models. *Chemical science*, 12(41):13664–13675, 2021b.

Edgar Liberis, Petar Veličković, Pietro Sormanni, Michele Vendruscolo, and Pietro Liò. Parapred: antibody paratope prediction using convolutional and recurrent neural networks. *Bioinformatics*, 34(17):2944–2950, 2018.

Jason Lu. Protein folding structure prediction using reinforcement learning with application to both 2d and 3d environments. In *Proceedings of the 5th International Conference on Computer Science and Software Engineering*, pages 534–542, 2022.

Yingzhou Lu. *Multi-omics Data Integration for Identifying Disease Specific Biological Pathways*. PhD thesis, Virginia Tech, 2018.

Iain H Moal and Juan Fernández-Recio. Skempi: a structural kinetic and energetic database of mutant protein interactions and its use in empirical models. *Bioinformatics*, 28(20):2600–2607, 2012.

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.

Dimitra N Panou and Martin Reczko. Deepfoldit–a deep reinforcement learning neural network folding proteins. *arXiv preprint arXiv:2011.03442*, 2020.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Fredrik Pontén, Karin Jirström, and Matthias Uhlen. The human protein atlas—a tool for pathology. *The Journal of Pathology: A Journal of the Pathological Society of Great Britain and Ireland*, 216 (4):387–393, 2008.

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, 2019.

Matthew IJ Raybould, Claire Marks, Konrad Krawczyk, Bruck Taddese, Jaroslaw Nowak, Alan P Lewis, Alexander Bujotzek, Jiye Shi, and Charlotte M Deane. Five computational developability guidelines for therapeutic antibody profiling. *Proceedings of the National Academy of Sciences*, 116(10):4025–4030, 2019.

Gabriel J Rocklin, Tamuka M Chidyausiku, Inna Goreshnik, Alex Ford, Scott Houliston, Alexander Lemak, Lauren Carter, Rashmi Ravichandran, Vikram K Mulligan, Aaron Chevalier, et al. Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science*, 357 (6347):168–175, 2017.

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.

Emre Sevgen, Joshua Moller, Adrian Lange, John Parker, Sean Quigley, Jeff Mayer, Poonam Srivastava, Sitaram Gayatri, David Hosfield, Maria Korshunova, et al. Prot-vae: Protein transformer variational autoencoder for functional protein design. *bioRxiv*, pages 2023–01, 2023.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *The International Conference on Learning Representations (ICLR)*, 2018.

Randi Vita, Swapnil Mahajan, James A Overton, Sandeep Kumar Dhanda, Sheridan Martini, Jason R Cantrell, Daniel K Wheeler, Alessandro Sette, and Bjoern Peters. The immune epitope database (IEDB): 2018 update. *Nucleic acids research*, 47(D1):D339–D343, 2019.

Yue Wang, Yingzhou Lu, Yinlong Xu, Zihan Ma, Hongxia Xu, Bang Du, Honghao Gao, and Jian Wu. TWIN-GPT: Digital twins for clinical trials via large language model. *arXiv preprint arXiv:2404.01273*, 2024.

Chiung-Ting Wu, Sarah J Parker, Zuolin Cheng, Georgia Saylor, Jennifer E Van Eyk, Guoqiang Yu, Robert Clarke, David M Herrington, and Yue Wang. Cot: an efficient and accurate method for detecting marker genes among many subtypes. *Bioinformatics Advances*, 2(1):vbac037, 2022a.

Chiung-Ting Wu, Minjie Shen, Dongping Du, Zuolin Cheng, Sarah J Parker, Yingzhou Lu, Jennifer E Van Eyk, Guoqiang Yu, Robert Clarke, David M Herrington, et al. Cosbin: cosine score-based iterative normalization of biologically diverse samples. *Bioinformatics Advances*, 2(1):vbac076, 2022b.

Yue Wu, BENJAMIN W EHLERT, DALIA PERELMAN, HEYJUN PARK, AHMED A METWALLY, YINGZHOU LU, ALESSANDRA CELLI, CAROLINE BEJIKIAN, TRACEY MCLAUGHLIN, and MICHAEL SNYDER. 1596-p: Personalized glycemic response to carbohydrates and associated physiological signatures in multiomics. *Diabetes*, 73(Supplement_1), 2024.

Chunqiu Xia, Shi-Hao Feng, Ying Xia, Xiaoyong Pan, and Hong-Bin Shen. Leveraging scaffold information to predict protein–ligand binding affinity with an empirical graph neural network. *Briefings in Bioinformatics*, 24(1), 2023.

Zhaoping Xiong, Dingyan Wang, Xiaohong Liu, Feisheng Zhong, Xiaozhe Wan, Xutong Li, Zhaojun Li, Xiaomin Luo, Kaixian Chen, Hualiang Jiang, et al. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *Journal of Medicinal Chemistry*, 63(16):8749–8760, 2019.

Bohao Xu, Yingzhou Lu, Chenhao Li, Ling Yue, Xiao Wang, Nan Hao, Tianfan Fu, and Jim Chen. Smiles-mamba: Chemical mamba foundation models for drug admet prediction. *arXiv preprint arXiv:2408.05696*, 2024.

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.

Steven Yi, Adam Yee, John Harmon, Frank Meng, and Saurabh Hinduja. Enhance wound healing monitoring through a thermal imaging based smartphone app. In *Medical imaging 2018: Imaging informatics for healthcare, research, and applications*, volume 10579, pages 438–441. SPIE, 2018.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34, 2021.

Qianmu Yuan, Sheng Chen, Jiahua Rao, Shuangjia Zheng, Huiying Zhao, and Yuedong Yang. Alphafold2-aware protein–dna binding site prediction using graph transformer. *Briefings in Bioinformatics*, 23(2):bbab564, 2022.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in neural information processing systems*, 32, 2019.

Bai Zhang, Yi Fu, Yingzhou Lu, Zhen Zhang, Robert Clarke, Jennifer E Van Eyk, David M Herrington, and Yue Wang. DDN2.0: R and python packages for differential dependency network analysis of biological systems. *bioRxiv*, pages 2021–04, 2021.

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.

.1 Evaluation Metrics

In this section, we describe the basic evaluation metrics for both classification and regression tasks.

Classification metrics. Most classification tasks are binary classification, except subcellular prediction in protein localization prediction, which is a 10-category classification problem, where we use **accuracy (acc)** (the fraction of correctly predicted/classified samples) as the evaluation metric. In binary classification, there are four kinds of test data points based on their ground truth and the model's prediction,

1. positive sample and is correctly predicted as positive, known as *True Positive (TP)*;
2. negative samples and is wrongly predicted as positive samples, known as *False Positive (FP)*;
3. negative samples and is correctly predicted as negative samples, known as *True Negative (TN)*;
4. positive samples and is wrongly predicted as negative samples, known as *False Negative (FN)*.

- **Precision**. The precision is the performance of a classifier on the samples that are predicted as positive. It is formally defined as precision $= TP/(TP + FP)$.
- **Recall**. The recall score measures the performance of the classifier to find all the positive samples. It is formally defined as recall $= TP/(TP + FN)$.
- **PR-AUC** (Precision-Recall Area Under Curve). The area under the Precision-Recall curve summarizes the trade-off between the true positive rate and the positive predictive value for a predictive model using different probability thresholds.
- **ROC-AUC** Area Under the Receiver Operating Characteristic Curve summarizes the trade-off between the true positive rate and the false positive rate for a predictive model using different probability thresholds. ROC-AUC is also known as the Area Under the Receiver Operating Characteristic curve (AUROC) in some literature.

For all these metrics, the numerical values range from 0 to 1, a higher value represents better performance.

Regression metrics. In the regression task, both ground truth and prediction are continuous values.

- **Mean Squared Error (MSE)** measures the average of the squares of the difference between the forecasted value and the actual value. It is defined as MSE $= \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$, where N is the size of the test set; y_i and $\hat{y}_i$ denote the ground truth and predicted score of the i-th data sample in the test set, respectively. MSE value ranges from 0 to positive infinity. A lower MSE value indicates better performance.
- **Mean Absolute Error (MAE)** measures the absolute value of the difference between the predicted value and the actual value. It is defined as MAE $= \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$, where N is the size of the test set; y_i and $\hat{y}_i$ denote the ground truth and predicted score of the i-th data sample in the test set, respectively. MAE value ranges from 0 to positive infinity. It emphasizes the ranking order of the prediction instead of the absolute value. A lower MAE value indicates better performance.
- **Spearman rank correlation** (ρ), also known as Spearman's ρ, is a nonparametric statistical test that measures the association between two ranked variables. A higher ρ value indicates better performance.
- **R-squared (R^2) score** is defined as the proportion of the variation in the dependent variable that is predictable from the independent variable(s). It is also known as the coefficient of determination in statistics. Higher R^2 scores indicate better performance.

A Hyperparameter Settings

In table 7, we have listed a common settings of hyperparameter used in this library. In terms of learning rate (lr), a higher learning rate which is equal to 0.0001 for graph neural networks would lead to failure in training. For Subcellular and its binary version, a training epoch of 60 is enough for convergence. For small-scale protein datasets such as IEDB [Yi et al., 2018], PDB-Jespersen, and SAbDab-Liberis, a larger learning rate of 0.001 also leads to convergence and the same performance when using CNN, CNN-RNN, and Transformer. For TAP, SAbDab-Chen and CRISPR-Leenay, larger learning rate of 0.0001 is suggested when training graph neural networks.

Table 7: Default Model Configurations for Protein Sequence Learning.

Model	lr	dropout	activation	# heads	# layers	hidden dim	pooling	batch size	# epochs	norm
CNN	10^{-4}	0.1	ReLU	-	3	256	MaxPool1d	32	100	-
CNN-GRU	10^{-4}	0.1	ReLU	-	2	64	-	32	100	-
Transformer	10^{-4}	0.1	ReLU	4	2	64	-	32	100	LayerNorm
GCN	10^{-5}	0.1	ReLU	-	3	64	Weighted Sum + Max	32	100	BatchNorm
GAT	10^{-5}	0.1	ReLU	-	3	64	Weighted Sum + Max	32	100	-
NeuralFP	10^{-5}	0.1	ReLU	-	3	64	Sum + Max	32	100	BatchNorm
AttentiveFP	10^{-5}	0.1	ReLU	-	3	64	AttentiveFPReadout	32	100	-
MPNN	10^{-5}	0.1	ReLU	-	6	64	Sum + Max	32	100	-
PAGTN	10^{-5}	0.1	LeakyReLU	-	5	64	Weighted Sum + Max	32	100	-
Graphormer	10^{-5}	0.1	ReLU	8	1	64	MaxPooling	32	100	LayerNorm

B More results of performance on protein sequence learning

In this section, we show three more model performance on Fluorescence, Beta-lactamase and Human PPI dataset.

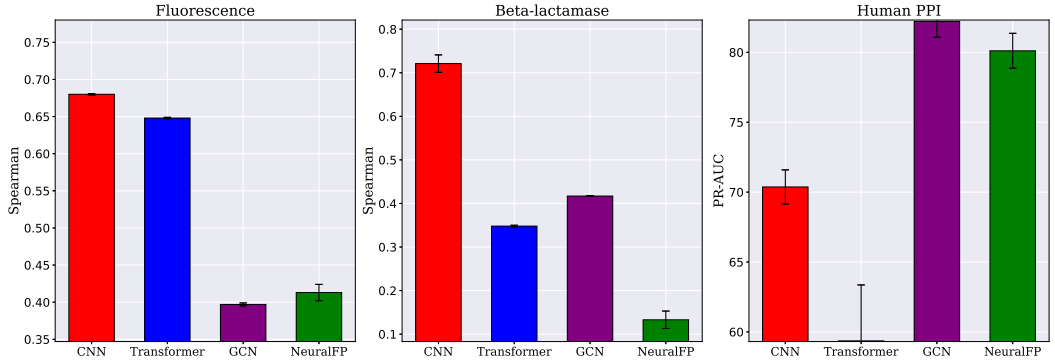

Figure 3: Performance of selected models including CNN, Transformer, GCN and NeuralFP on Fluorescence, Beta-lactamase and Human PPI. We observe that CNN performs significantly better than GCN on protein function prediction tasks. However, GCN dominates on large-scale PPI datasets such as the Human PPI dataset.

C More results of empirical memory and runtime complexity

In this section, we show four more empirical complexity results on Fluorescence, Beta-lactamase, SubCellular Binary and Yeast PPI. In Figure 4, the same pattern is found as Figure 2: Transformer took up a large amount of memory to train while the training speed is reasonably fast. Training GNN is slow since the DGL framework computes the message passing on the CPU. CNNs are both fast and memory efficient to train. A different case is found in Figure 5 where for SubCellular Binary and Yeast PPI, GCN is the slowest to train with occupying a large amount of GPU memory space.

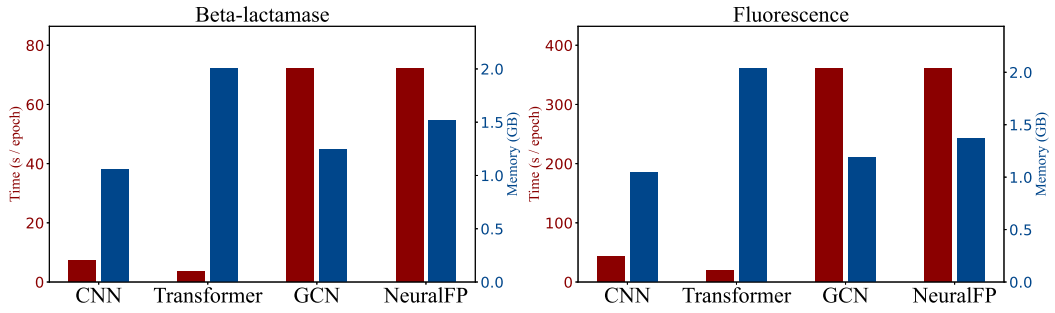

Figure 4: Empirical computational complexity on CRISPR and SAbDab Liberis measured by runtime and memory.

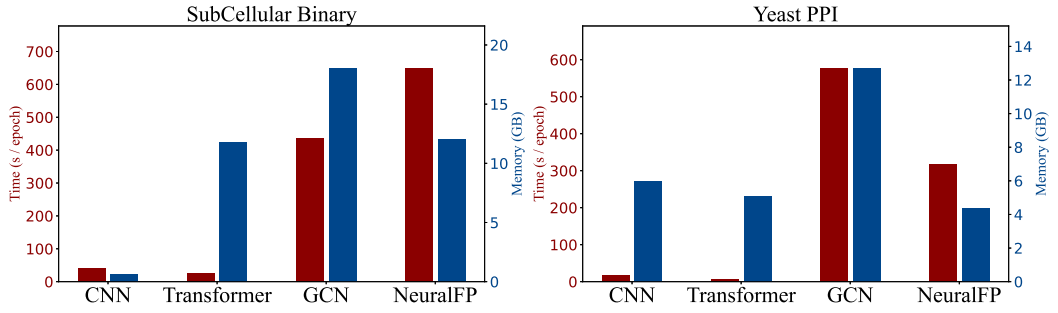

Figure 5: Empirical computational complexity on SubCellular Binary and Yeast PPI measured by runtime and memory.

D Training Curves of `DeepProtein`

In this section, we plot the metrics (Accuracy or R2) and training loss for SubCellular, PPI Affinity, SAbDab-Chen and TAP.

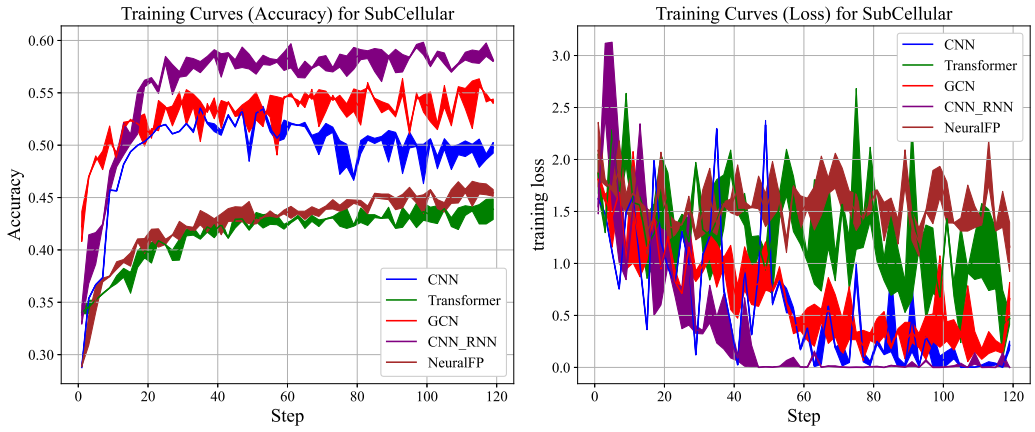

Figure 6: SubCellular

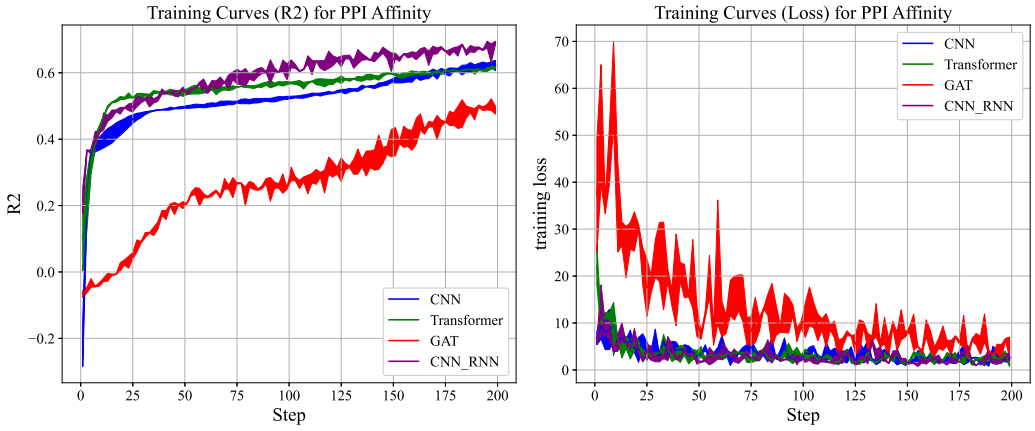

Figure 7: PPI Affinity

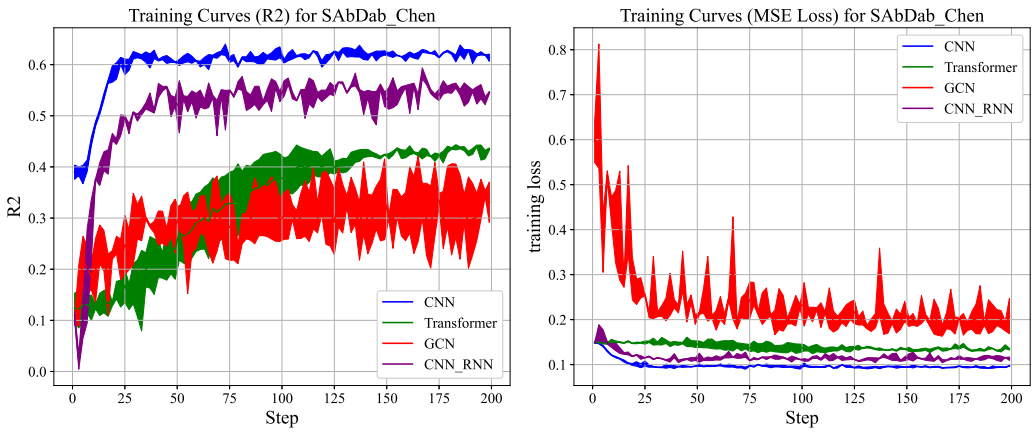

Figure 8: SAbDab_Chen

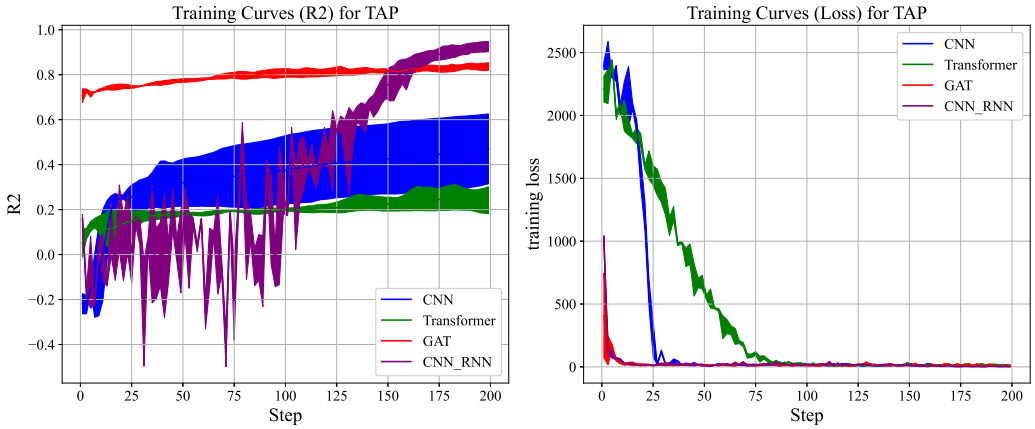

Figure 9: TAP

E Few Lines of Codes

Protein Function Prediction We show an example of CNN on Beta_lactamase with `DeepProtein`. More case studies can be found in the GitHub repository of `DeepProtein`. We annotate the code to make researchers better understand the function of code base, which is simpler and clearer than most of the protein learning libraries available.

```
### package import
import os, sys, argparse, torch, wandb

### Our library DeepProtein
from DeepProtein.dataset import *
import DeepProtein.utils as utils
import DeepProtein.ProteinPred as models

### Load Beta lactamase dataset
path = os.getcwd()
train_beta = Beta_lactamase(path + '/DeepProtein/data', 'train')
valid_beta = Beta_lactamase(path + '/DeepProtein/data', 'valid')
test_beta = Beta_lactamase(path + '/DeepProtein/data', 'test')

train_protein_processed, train_target, train_protein_idx = collate_fn(train_beta)
```

```
valid_protein_processed, valid_target, valid_protein_idx = collate_fn(valid_beta)
test_protein_processed, test_target, test_protein_idx = collate_fn(test_beta)

### Train Valid Test Split
target_encoding = 'CNN'
train, _, _ = utils.data_process(X_target=train_protein_processed, y=train_target,
    target_encoding=target_encoding, split_method='random', frac=[0.99998, 1e-5, 1e-5])

_, val, _ = utils.data_process(X_target=valid_protein_processed, y=valid_target,
    target_encoding=target_encoding, split_method='random', frac=[1e-5, 0.99998, 1e-5])

_, _, test = utils.data_process(X_target=test_protein_processed, y=test_target,
    target_encoding=target_encoding,split_method='random', frac=[1e-5, 1e-5, 0.99998])

### Load configuration for model
config = generate_config(target_encoding=target_encoding,
                         cls_hidden_dims=[1024, 1024],
                         train_epoch=20,
                         LR=0.0001,
                         batch_size=32,
                         )
config['multi'] = False
torch.manual_seed(args.seed)
model = models.model_initialize(**config)

### Train our model
model.train(train, val, test, compute_pos_enc = False)
```

If we use argparse in Python, this could be easily done in one line, where

```
python beta.py  --target_encoding CNN
                --seed 7
                --wandb_proj DeepPurposePP
                --lr 0.0001
                --epochs 100
```

Note that if you want to observe the result offline without wandb, then set

```
    wandb.init(mode='offline', name = "your_project")
```

inside the training file.

F Tutorial

Demos could be found under DEMOS folder in `DeepProtein`.