PROTLLM: An Interleaved Protein-Language LLM with Protein-as-Word Pre-Training

Anonymous ACL submission

Abstract

We propose **PROTLLM**, a versatile cross-modal large language model (LLM) for both protein-centric and protein-language tasks. PROTLLM features a unique dynamic protein mounting mechanism, enabling it to handle complex inputs where the natural language text is interspersed with an arbitrary number of proteins. Besides, we propose the protein-as-word language modeling approach to train PROTLLM. By developing a specialized protein vocabulary, we equip the model with the capability to predict not just natural language but also proteins from a vast pool of candidates. Additionally, we construct a large-scale **inter**leaved **p**rotein-**t**ext dataset, named **InterPT**, for pre-training. This dataset comprehensively encompasses both (1) structured data sources like protein annotations and (2) unstructured data sources like biological research papers, thereby endowing PROTLLM with crucial knowledge for understanding proteins. We evaluate PROTLLM on classic supervised protein-centric tasks and explore its novel protein-language applications. Experimental results demonstrate that PROTLLM not only achieves superior performance against protein-specialized baselines on protein-centric tasks but also induces zero-shot and in-context learning capabilities on protein-language tasks. Our data and models will be publicly available.

1 Introduction

Understanding proteins is essential for unraveling the mysteries of life and enabling artificial intelligence systems to advance bioscience research (Wang et al., 2023a). Thanks to the development of deep learning techniques, neural network models encompass extensive protein-centric applications, such as protein-folding prediction (Jumper et al., 2021), protein-protein interaction analysis (Li et al., 2018; Su et al., 2023), function prediction (Zhang et al., 2023a), etc.

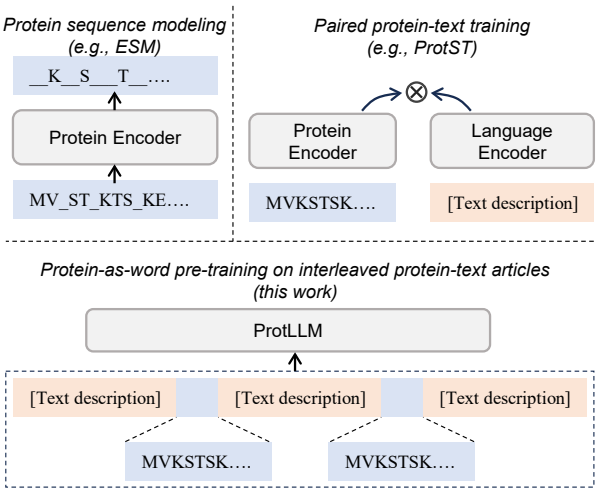


Figure 1: Unlike existing protein representation models that focus on protein-text pairs or protein-only data, PROTLLM can handle complex inputs with multiple proteins interleaved with text, thereby learning crucial knowledge from scientific papers and supporting diverse downstream tasks.

Protein representation learning methods typically employ large-scale pre-training, which learns unsupervised protein representations on massive protein sequences with masked language modeling (Rives et al., 2021), or autoregressive language modeling (Elnaggar et al., 2020). In addition to protein-centric tasks, recent studies have attempted to extend protein models to protein-language scenarios. ProtST (Xu et al., 2023b) integrates textual information into the protein encoder through multimodal pre-training on protein-text pairs, achieving zero-shot text-to-protein retrieval. Fang et al. (2023) introduces an instruction dataset tailored for the biomolecular domain and investigates how fine-tuned LLM performs on protein-domain instruction-following tasks, such as function description generation.

Despite the success of protein representation methods on specific tasks, developing a model that excels in both protein-centric and protein-language

tasks is still under-explored, facing three main challenges. Firstly, architectures are designed for particular downstream tasks, making it difficult to accommodate a wide range of tasks simultaneously. Secondly, current methods primarily derive cross-modal supervision from explicitly annotated protein-text pairs, which is not scalable to large-scale pre-training. Lastly, supporting a variable number of proteins in the input sequence introduces computational uncertainty in each training step, leading to inefficiencies during pre-training.

In this work, we propose PROTLLM, which is a versatile LLM for both protein-centric and protein-language tasks. Instead of designing for specific tasks, PROTLLM supports complex interleaved protein-text inputs and outputs, which enables our model to simultaneously handle diverse downstream tasks without re-designing task-specific architecture (see Figure 1 for illustrations). Specifically, our dynamic protein mounting mechanism enables the model to seamlessly process text interspersed with an arbitrary number of proteins. Besides, we propose protein-as-word language modeling to ensure interleaved protein-text outputs. By building a protein vocabulary, PROTLLM is trained to autoregressively predict words and proteins from their respective vocabularies.

Additionally, we present a large-scale interleaved protein-text dataset, named InterPT, for PROTLLM pre-training. InterPT is constructed from diverse data sources, consisting of both structured data such as paired protein annotation data, and unstructured data from biological research papers, which encourages PROTLLM to harness crucial knowledge from the scientific articles.

We conduct extensive experiments on a wide range of downstream tasks, ranging from classic supervised protein-centric tasks to novel protein-language applications. Experimental results demonstrate that PROTLLM outperforms specialized baselines on protein-centric tasks. PROTLLM also unlocks the in-context learning capability for protein-protein interaction prediction, and achieves zero-shot text-guided functional protein retrieval.

Our contributions are as follows:

- We propose PROTLLM, a versatile cross-modal LLM for both protein-centric and protein-language tasks. PROTLLM could process complex interleaved protein-text inputs and outputs, thereby supporting diverse tasks.
- We introduce a large-scale pre-training dataset, InterPT, interleaving proteins and text from both structured data sources and unstructured multi-protein scientific articles.
- We show that PROTLLM achieves superior results on protein-centric tasks against protein-specialized baselines, and induces zero-shot and in-context learning capabilities.

2 Related Work

2.1 Large Language Models

The evolution of LLMs has been a cornerstone in the field of natural language processing , showcasing extraordinary capabilities across a broad spectrum of tasks (Devlin et al., 2018; Raffel et al., 2020; Brown et al., 2020; OpenAI, 2023; Touvron et al., 2023; Longpre et al., 2023; Chowdhery et al., 2022). These models, once thought to be limited to text-based tasks, have now crossed boundaries into areas traditionally dominated by human expertise, including mathematical problem-solving (Wei et al., 2022; Imani et al., 2023), drug discovery (Liang et al., 2023; Liu et al., 2023b), and complex decision making (Yu et al., 2023; Ma et al., 2023). Recent explorations further extend LLMs' expertise into the multimodal domain where they demonstrate significant promise in processing and generating content from diverse modalities (Huang et al., 2023; Zhu et al., 2023; Liu et al., 2023a; Zhao et al., 2023; Wu et al., 2023). Most of these works focus on aligning pre-trained encoders from various modalities with LLMs through instruction tuning, thus equipping LLMs to interpret multimodal inputs. In the realm of scientific research, specialized molecular LLMs have been devised for tasks like molecular property prediction (Liu et al., 2023c), captioning (Fang et al., 2023), and retrieval (Liu et al., 2023d). Despite these advances, the progress in protein understanding with LLMs lags, hindered by the scarcity of comprehensive datasets for alignment and the absence of efficient architectures to model protein-language sequences.

2.2 Protein Representation Learning

Current mainstream methods for protein understanding tasks have focused on protein representation learning. Protein language models (PLMs) (Elnaggar et al., 2020; Rives et al., 2021; Meier et al., 2021; Lin et al., 2022) have marked significant progress in the area by training the protein sequence encoders on massive protein sequence data. Protein structure encoding methods aim to

learn coarse-grained amino-acid-level representations (Gligorijević et al., 2021; Fan et al., 2022; Zhang et al., 2023a; Xu et al., 2023a) or fine-grained atom-level representations (Hermosilla et al., 2021; Jing et al., 2021; Zhang et al., 2023b). Despite the success in protein modeling, protein-related text data are left unexplored, which contains valuable supervision signals crucial for understanding proteins. To enhance protein understanding with text supervision, OntoProtein (Zhang et al., 2022a) leverages knowledge graphs, utilizing gene ontology annotations to implicitly enrich protein representation with textual information. ProtST (Xu et al., 2023b) integrates textual information into the protein encoder through multi-modal pre-training on protein-text pairs, achieving zero-shot text-to-protein retrieval.

Mol-Instruction (Fang et al., 2023) introduces a comprehensive instruction dataset specialized for biomolecules and further fine-tunes LLMs on this dataset. Similarly, InstructProtein (Wang et al., 2023b) improves the quality of instruction datasets by sampling protein-text pairs from a structured knowledge graph. This line of work focuses on aligning protein with human language using LLMs. However, a limitation of these approaches lies in their direct incorporation of protein sequences into LLMs as text, leading to suboptimal protein modeling due to the LLMs not being pre-trained on extensive protein sequence datasets. In contrast, PROTLLM provides a versatile framework that excels in both classic protein-centric tasks and novel protein-text applications.

3 Methods

In this section, we elaborate on our proposed method, PROTLLM, which is illustrated in Figure 2. Initially, we detail the model architecture in Section 3.1. Subsequently, the pre-training strategy is explained, introducing the concept of protein-as-word modeling, as outlined in Section 3.2. We then present the uniquely constructed interleaved protein-text dataset, InterPT, in Section 3.3. Lastly, we explore the application of PROTLLM on a variety of tasks in Section 3.4.

3.1 PROTLLM Framework

Model architecture PROTLLM consists of an LLM for natural language modeling, a protein encoder, and cross-modal connectors that connect the protein encoder and the LLM. We use LLaMA-7b (Touvron et al., 2023) as the backbone of PROTLLM, which is an autoregressive Transformer language model pre-trained on large-scale natural language data. To make PROTLLM understand protein sequences (i.e., sequences of amino acid tokens, which are the primary structure of proteins), we employ ProtST (Xu et al., 2023b) as the protein encoder. ProtST follows the backbone architecture of ESM-2 (Lin et al., 2022) and introduces an additional two-layer MLP projection head. Pre-trained on large-scale protein-text pairs with contrastive learning, ProtST learns protein representations that are well-aligned with text. Besides, we introduce cross-modal connectors that connect the LLM with the protein encoder, thereby enabling PROTLLM to accept multimodal inputs. Specifically, PROTLLM has two cross-modal connector layers, which are placed at the input layer and the output layer of the LLM, respectively. The input-layer connector is a trainable projection matrix and transforms the output vectors from the protein representation space to the LLM representation space. Similarly, the output-layer connector transforms the LLM output vectors back to the protein representation space. Significantly, the output-layer connector also serves as a prediction head, allowing our model to perform protein retrieval and multi-choice protein answering tasks without requiring the LLM to generate complicated protein names.

Dynamic protein mounting PROTLLM considers not only structured protein-text paired data but also free-form interleaved protein-text sequences. Although the widely used encoder-decoder architecture can handle paired data, it encounters difficulties when dealing with interleaved protein-text inputs with multiple proteins. Therefore, we propose dynamic protein mounting, which allows PROTLLM to accept an arbitrary number of proteins as either input. Specifically, given an input sequence interleaved with proteins and text,

$$\ldots[\text{text}_1]\ [\text{protein}_1]\ [\text{text}_2]\ [\text{protein}_2]\ [\text{text}_3]\ldots$$

we do not directly feed the protein sequence to the LLM, but replace sequences with mount points.

$$\ldots[\text{text}_1]\ \texttt{<PROT>}\ [\text{mount}_1]\ \texttt{</PROT>}\ [\text{text}_2]\ \ldots$$

At each mount point, we mount the protein encoder to the LLM with the cross-modal connector. Additionally, these mount points are delineated by protein tags, signaling to the LLM that it is receiving protein vector inputs at these positions, rather than text data.

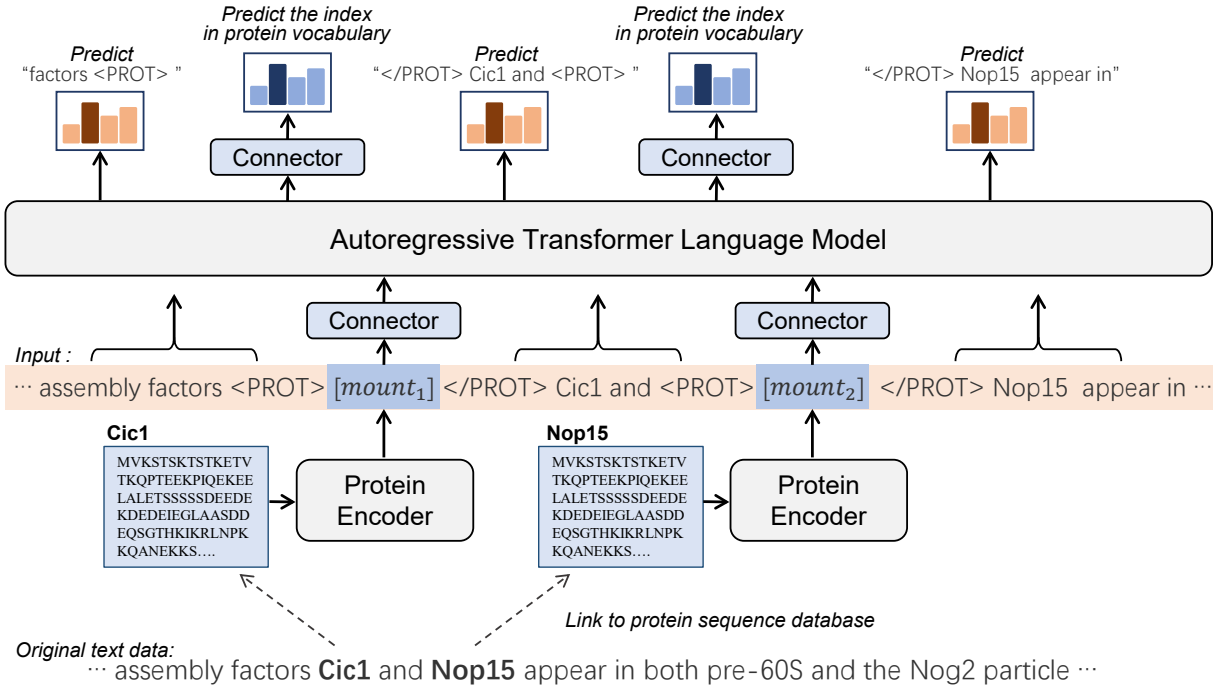


Figure 2: An overview of PROTLLM. The architecture of PROTLLM consists of an autoregressive transformer, a protein encoder, and cross-modal connectors. With dynamic protein mounting, PROTLLM adeptly handles free-form interleaved protein-text sequences with an arbitrary number of proteins in the input. PROTLLM is pre-trained with protein-as-word language modeling that unifies word and protein prediction by constructing a protein vocabulary.

3.2 PROTLLM Pre-Training

Protein-as-word language modeling We introduce the protein-as-word language modeling training objective, which unifies protein prediction and word prediction as an autoregressive language modeling task. Consider an input sequence interleaved with n tokens $[x_1, x_2, ..., x_n]$, where the i-th token x_i represents either a natural language token or a protein. The protein-as-word language modeling object is to maximize the likelihood:

$$\arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log p(x_i | x_{<i}; \boldsymbol{\theta}), \tag{1}$$

where $p(x_i|x_{<i};\boldsymbol{\theta})$ is a categorical probability distribution over a natural language vocabulary when predicting natural words, or a protein vocabulary when predicting proteins. The probability is computed by

$$p(x_i|x_{<i};\boldsymbol{\theta}) = \begin{cases} \text{softmax}(\boldsymbol{h}_i^\top \boldsymbol{e}_j)_j, j \in \mathcal{V} & \textit{if } x_i \textit{ is word} \\ \text{softmax}(\boldsymbol{h}_i^\top \boldsymbol{W} \boldsymbol{v}_k)_k, k \in \mathcal{V}_\text{p} & \textit{if } x_i \textit{ is protein} \end{cases} \tag{2}$$

where h_i is the last-layer LLM hidden states of x_i; $\boldsymbol{e}_j$ is the word embedding of the word j from the natural language vocabulary $\mathcal{V}$; $\boldsymbol{W}$ stands for the output connector matrix, and $\boldsymbol{v}_k$ is the protein embeddings of the protein k from the protein vocabulary $\mathcal{V}_\text{p}$. To construct the protein vocabulary, we collect all protein sequences in the training data. We then filter out proteins present in the downstream test sets to prevent data leakage. Finally, we compile a vocabulary consisting of the $1,076,781$ proteins.

Pre-training acceleration with protein cache Although our dynamic protein mounting design introduces flexibility for the input format, it also introduces computational uncertainty into the pre-training process, i.e., the computational cost of each step can vary significantly with the number of input proteins. Consequently, the throughput is limited by the worst case, leading to markedly reduced training efficiency. To accelerate the pre-training, we build a protein cache where we store all the pre-computed protein vectors encoded by the protein encoder. With the protein cache, we eliminate the heavy computational cost of the protein encoder, thereby accelerating the pre-training procedure with stable throughput. Besides, we utilize LoRA (Hu et al., 2022) for efficient training.

Data Source	Data Type	Size
PubMed	Multi-protein articles	165,206
UniProt	Annotations	64,634
STRING	Annotations	25,682
Mol-Instructions	Instruction-following data	173,973

Table 1: Category and statistics of InterPT components.

3.3 InterPT: Interleaving Protein-Text Data

We propose a large-scale **inter**leaved **p**rotein-**t**ext multimodal dataset, named **InterPT**, to pre-train PROTLLM with comprehensive protein-related knowledge. This dataset encompasses three types of data sources, i.e., multi-protein scientific articles, protein-annotation pairs, and protein instruction-following data. The statistics of each component are listed in Table 1.

Multi-protein scientific articles Multi-protein scientific articles describe complex relationships among different proteins found in biological research, where each sample could contain multiple proteins. Unlike data presented in structured formats such as pairs or knowledge graphs, these articles offer detailed insights in unstructured natural language. Guided by the recording in the STRING database (Mering et al., 2003) of multi-protein interactions and the scientific articles supporting them, we retrieve all involved articles from the PubMed database (Canese and Weis, 2013), specifically selecting instances where multiple proteins co-occur within the same paragraph. All proteins in these paragraphs are linked to the UniProt database (Consortium, 2015) for their amino acid sequences. Finally, we collect 165K interleaved protein-text sequences from PubMed articles.

Protein-annotation pairs This data maps individual proteins to their textual annotations such as function descriptions. We integrate two data sources, i.e., the UniProt database (Consortium, 2015) and the STRING database (Mering et al., 2003), adding up to 90K protein-annotation pairs. Given such a pair, we utilize it for two tasks, i.e., protein-to-text prediction and text-to-protein prediction, with the probability of 0.8 and 0.2, respectively. Besides, during pre-training, we interleave the data into longer sequences by concatenating multiple pairs into a single sequence, which has two advantages: (1) this operation can bridge the data length gap across different data sources and reduce the number of padding tokens, leading to higher training efficiency; (2) training multiple pairs in a single sequence encourages the model to obtain in-context learning capabilities (Gu et al., 2023).

Protein instruction-following data This data is in the instruction-following style (Ouyang et al., 2022), typically requiring the model to generate open-ended text given a protein and an instruction (Fang et al., 2023). We select the data items of proteins from the Mol-Instructions dataset (Fang et al., 2023) and include them into InterPT. Similar to the processing of protein-annotation pairs, we also concatenate multiple instruction-following data into a single pre-training example, so as to improve training efficiency and acquire in-context learning capabilities.

3.4 Applying PROTLLM to Diverse Tasks

Supervised fine-tuning The best practice for adapting PROTLLM to downstream tasks is supervised fine-tuning when training data are available. Since PROTLLM supports flexible input and output formats, we can simply transform the downstream task data into an interleaved format and directly perform protein-as-word language modeling for supervised fine-tuning. The input and output prompt format for each downstream task can be found in the Appendix A. During fine-tuning, we also apply the LoRA adapter to the LLM for efficient fine-tuning while preventing the model from overfitting several proteins in the training set.

In-context learning In-context learning is a promising capability of LLM, which can adapt the LLM to specific tasks with a few examples without training the model. PROTLLM can achieve in-context learning by pretending a few demonstration examples. To the best of our knowledge, PROTLLM is the first protein-language LLM that is capable of in-context learning.

Instruction-following protein retrieval For another interesting application, PROTLLM can be programmed to execute protein retrieval with customized requirements by following instructions. In Section 4.3, we show that PROTLLM can well retrieve functional proteins based only on function descriptions, and it can be further improved by prepending a one-shot demonstration.

4 Experiments

We evaluate PROTLLM on three types of downstream tasks: (1) protein-centric tasks, which include supervised fine-tuning on conventional

Model	Pre-training		EC		GO-BP		GO-MF		GO-CC		PPI
	Protein	Text	AUPR	F_{max}	AUPR	F_{max}	AUPR	F_{max}	AUPR	F_{max}	ACC
DeepFRI	✓	✗	0.546	0.631	0.282	0.399	0.462	0.465	0.363	0.460	-
GearNet	✓	✗	0.892	0.874	0.292	0.490	0.596	0.650	0.226	0.486	73.86
ProtBert	✓	✗	0.859	0.838	0.188	0.279	0.464	0.456	0.234	0.408	77.32
ESM-1b	✓	✗	0.884	0.869	0.332	0.452	0.630	0.659	0.324	0.477	82.22
ESM-2	✓	✗	0.888	0.874	0.340	0.472	0.643	0.662	0.350	0.472	86.90
OntoProtein	✓	✓	0.854	0.841	0.284	0.436	0.603	0.631	0.300	0.441	70.42
ProtST	✓	✓	**0.898**	**0.878**	0.342	0.482	0.647	**0.668**	0.364	0.487	88.19
PROTLLM	✓	✓	0.874	0.860	**0.349**	**0.503**	**0.652**	**0.668**	**0.469**	**0.596**	**89.87**

Table 2: Comparative benchmark results on protein-centric tasks. We use AUPR and F_{max} on EC and GO prediction and accuracy (%) on PPI prediction. Bold figures denote the best performance. '-' indicates not applicable.

benchmarks for protein understanding; (2) protein-text in-context learning, where we show the unique ability of PROTLLM by in-context learning on protein-protein interaction prediction; (3) text-guided functional protein retrieval, where we conduct a real-world enzyme mining task as a proof-of-concept study to validate the retrieval capability of PROTLLM. We present detailed hyperparameters, and prompt templates for pre-training and fine-tuning in Appendix A.

4.1 Protein-Centric Tasks

Setup Following the settings in PEER benchmark (Xu et al., 2022), we adopt three standard tasks in protein understanding to validate our method. **Enzyme Commission (EC) number prediction** (Gligorijević et al., 2021) aims to predict all possible EC numbers of a protein simultaneously, reflecting the chemical reactions it catalyzes. **Gene Ontology (GO) term prediction** (Gligorijević et al., 2021) extends as a multi-label classification task, seeking to predict whether a protein belongs to specific GO terms. The GO benchmark is categorized into three branches, namely biological process (BP), molecular function (MF), and cellular component (CC). **Protein-Protein Interaction (PPI) prediction** aims to determine whether two given proteins interact or not. We adopt the human PPI dataset (Pan et al., 2010) for experiments.

To evaluate performances on multi-label classification tasks including EC and GO prediction, we report pair-centric area under precision-recall curve (AUPR) values and F_{max}, a widely used metric in the CAFA challenges (Radivojac et al., 2013). PPI prediction results are evaluated by mean accuracy. These metrics require the soft probability of each target label. To achieve this, we initially extract the probabilities of "Yes" for the positive label and "No" for the negative label, respectively. Then, these probabilities are normalized via the softmax function to get the final predicted probabilities.

Baselines We compare PROTLLM with seven existing protein representation learning methods. As shown in Table 2, these methods can be categorized into two distinct categories: **protein-only approaches** and **protein-text learning approaches**. The former encompasses sequence-based models including ProtBert (Elnaggar et al., 2020), ESM-1b (Rives et al., 2021), and ESM-2 (Lin et al., 2022), which are pre-trained using extensive collections of protein sequences, alongside structure-based models, such as DeepFRI (Gligorijević et al., 2021), and GearNet (Zhang et al., 2022b). The latter, protein-text learning approaches, includes OntoProtein (Zhang et al., 2022a), and ProtST (Xu et al., 2023b).

Note that Mol-Instructions (Fang et al., 2023) and InstructProtein (Wang et al., 2023b) also belong to the protein-text learning approaches. However, their methods directly take protein sequences as human language and tokenize the data using byte-pair encoding. Contrasting with the protein-as-word strategy in PROTLLM, this exponentially increases the context length, rendering the evaluation on tasks with extensive label sets, like EC and GO prediction, or those requiring multiple protein inputs, such as PPI prediction, impractical for their approaches.

Results The results are shown in Table 2. PROTLLM consistently shows competitive or even superior performance compared to both protein-

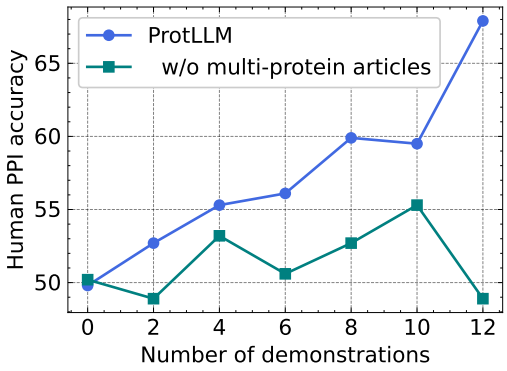


Figure 3: In-context learning results on human PPI.

Reactant	Method	Pool Size: 500			Pool Size: 1000		
		Top-10	Top-20	Top-50	Top-10	Top-20	Top-50
IsoC5	Zero-shot	0.40	0.40	0.40	0.33	0.33	0.33
	In-context	**0.60**	**0.80**	**0.80**	**0.50**	**0.67**	**0.67**
C3	Zero-shot	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
	In-context	**1.0**	**1.0**	**1.0**	0.67	0.67	0.67
C5	Zero-shot	0.40	0.40	0.40	0.25	0.25	0.25
	In-context	**0.60**	**0.80**	**0.80**	**0.38**	**0.50**	**0.50**
C8	Zero-shot	0.33	0.33	0.50	0.22	0.22	0.33
	In-context	**0.83**	**0.83**	**0.83**	**0.56**	**0.56**	**0.56**

Table 3: Performance comparisons between zero-shot retrieval and in-context learning on enzyme mining. **Top-10, 20 and 50 Recall** are reported.

only and protein-text approaches across all benchmarks, indicating the effectiveness of our proposed framework on conventional close-ended protein understanding tasks. Remarkably, PROTLLM obtain 0.596 F_{max} and 0.469 AUPR on GO-CC, which outperforms ProtST by a large margin. As depicted in Section 3.1, PROTLLM directly uses pre-trained ProtST as the protein encoder, with the key difference lying in our LLM decoder and pre-training stage for alignment. By comparing PROTLLM with ProtST, the overall improvements strongly highlight the potential benefits of incorporating richer protein-text information and scaling the size of the language model.

Moreover, PROTLLM also outperforms two structure-based models on GO and PPI prediction despite we only leverage sequence information during training. Protein structures encode rich information and have direct relations to their functions. This opens up a promising direction to further incorporate protein structure into our framework, which we leave for future work.

4.2 Unlocking In-Context Learning

In-context learning is the capability that rapidly adapts the model to specific tasks using only a few annotated demonstration examples, which is originally found in autoregressive language models (Brown et al., 2020) and then is extended to visual language models (Alayrac et al., 2022). In this section, we investigate whether PROTLLM can achieve in-context learning on the human protein-protein interaction (PPI) prediction task.

Setup We directly evaluate the pre-trained PROTLLM model on the human PPI task without updating any parameters. For the k-shot in-context learning, we randomly sample k examples from the validation set as the demonstrations and prepend them to each test sequence. Both the demonstration example and test example are prompted with the same template. For example, a one-shot prompted input is as follows:

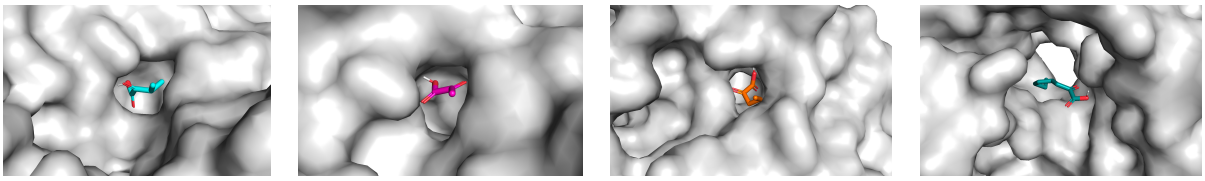
```
Do <PROT> [mount_1] </PROT> and <PROT> [mount_2] </PROT>
interact with each other? Yes\n Do <PROT> [mount_3]
</PROT> and <PROT> [mount_4] </PROT> interact with each
other?
```

The protein sequences of demonstration and test examples are first encoded by the protein encoder and then fed to the language model at each mount point. The final answer is predicted by selecting the verbalizer, i.e., "Yes" or "No", with the higher probability. Besides, to understand how multi-protein pre-training data from scientific articles improves PROTLLM, we also evaluate a variant of our model by removing the multi-protein scientific articles from the pre-training corpora.

Results Figure 3 presents the in-context learning performance on human PPI with varying numbers of demonstration examples. Our model consistently achieves higher PPI accuracy with an increasing number of demonstration examples, demonstrating its effective in-context learning capability for protein-centric tasks. In comparison, the model performs drastically worse upon removing the multi-protein scientific articles, and fails to learn in context with the 2, 6, and 12 demonstrations. We believe that the in-context learning capability of our model could empower biologists to apply it to specialized tasks that lack annotated data, using minimal examples. Our experiments on enzyme mining illustrate a tangible application of in-context learning, as detailed in Section 4.3.

4.3 Text-Guided Functional Protein Retrieval

Setup This experiment aims to study the capability of PROTLLM to retrieve functional proteins



(a) Lead Enzyme for IsoC5:
- K_{cat}/K_M: 1300 ($mol^{-1}s^{-1}$)
- K_{cat}: 7.8 (s^{-1})
- **Vina Energy:** -7.8 (kcal/mol)

(b) Lead Enzyme for C3:
- K_{cat}/K_M: 8200 ($mol^{-1}s^{-1}$)
- K_{cat}: 46.7 (s^{-1})
- **Vina Energy:** -6.8 (kcal/mol)

(c) Lead Enzyme for C5:
- K_{cat}/K_M: 1700 ($mol^{-1}s^{-1}$)
- K_{cat}: 4.7 (s^{-1})
- **Vina Energy:** -7.3 (kcal/mol)

(d) Lead Enzyme for C8:
- K_{cat}/K_M: 17000 ($mol^{-1}s^{-1}$)
- K_{cat}: 10.1 (s^{-1})
- **Vina Energy:** -7.0 (kcal/mol)

Figure 4: Top-1 enzyme mining results based on PROTLLM retrieval and AutoDock Vina post-screening. K_{cat}/K_M and K_{cat} measure enzyme activity (*higher the better*). Vina energy measures binding affinity (*lower the better*).

based on text prompts and demonstrations. For this purpose, we apply PROTLLM to enzyme mining, which is a critical stage in enzyme and metabolic engineering pipelines. In this experiment, we evaluate our model on mining carboxylate reductases that transform various ketoacids into their corresponding aldehydes. Four ketoacid reactants, i.e., 2-ketoisovaleric acid (IsoC5), pyruvic acid (C3), 2-ketovaleric acid (C5), and 2-ketooctanoic acid (C8), studied in Mak et al. (2015) are employed for evaluation.

Using a reported enzyme for IsoC5, ketoisovalerate decarboxylase (KIVD) (De La Plaza et al., 2004), as the query, we first search for a pool of enzyme candidates by BLASTp (McGinnis and Madden, 2004), where the pools with the size of 500 and 1000 are respectively tested. We then leverage PROTLLM to retrieve active enzymes from the pool for each reactant in two modes. In the **zero-shot retrieval** setting, given the prompt:

```
Identify the enzymes: {Reactant} → Isobutanal. <PROT>
```

describing the reaction from reactant (IsoC5, C3, C5 or C8) to product, PROTLLM generates a protein embedding at the token <PROT>. Then, we encode all the candidate enzymes as embeddings with the protein encoder. Finally, we utilize this embedding to rank all enzyme candidates by comparing embedding similarity. For **in-context learning**, we further add a one-shot demonstration of carboxylate reductase before the prompt above to facilitate enzyme mining. The demonstration is:

```
Indentify the enzymes: Oxidizes aldehydes to the
corresponding carboxylic acids with a preference for
aromatic aldehydes. <PROT> [mount] </PROT>
```

where the "[mount]" token is represented by the protein embedding of PaoC (Neumann et al., 2009), a typical carboxylate reductase.

Results In Table 3, we report the recall of active enzymes found in Mak et al. (2015) at top 10, 20, and 50 ranked candidates. It is observed that in-context learning outperforms zero-shot retrieval on 18 out of 24 metrics, which verifies that PROTLLM can learn from a few demonstrations and improve its enzyme mining performance based on such knowledge. To study the top-ranked enzymes by PROTLLM more in depth, we employ AutoDock Vina (Trott and Olson, 2010) to further screen the top-20 enzymes found by in-context learning and pick the one with the lowest Vina energy for visualization. As shown in Figure 4, the lead enzymes selected in this way are all with good properties, possessing high enzyme activity (i.e., high K_{cat}/K_M and K_{cat} values measured by Mak et al. (2015)) and low binding energy measured by AutoDock Vina. These results altogether prove the effectiveness of PROTLLM on enzyme mining.

5 Conclusion

In this paper, we present PROTLLM, a versatile LLM designed to tackle both protein-centric and protein-language tasks. Through dynamic protein mounting and protein-as-word modeling, PROTLLM adeptly handles complex interleaved protein-text data, seamlessly unifying a wide array of protein tasks via a natural language interface. Besides, we construct a large-scale protein-language pre-training dataset, called InterPT, which encourages the model to learn from diverse data sources ranging from structured paired data to unstructured multi-protein scientific articles. Extensive experiments demonstrate that PROTLLM not only achieves competitive performance against specialized baselines across standard protein-centric benchmarks but also paves the way for exploring novel protein-language applications.

Limitations

In this paper, we primarily focus on sequence modeling for protein understanding. Nonetheless, PROTLLM is a general interface for the inputs in other modalities. Future research could further extend PROTLLM to additional modalities, such as protein structures and molecular graphs, by incorporating modality-specific encoders. Besides, we would like to explore more novel applications of PROTLLM such as scientific discovery.

Ethics Statement

This work complies with the ACL Code of Ethics. We declare that there are no ethical issues in this paper, to the best of our knowledge.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Kathi Canese and Sarah Weis. 2013. Pubmed: the bibliographic database. *The NCBI handbook*, 2(1).

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

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

Marta De La Plaza, Pilar Fernández de Palencia, Carmen Peláez, and Teresa Requena. 2004. Biochemical and molecular characterization of α-ketoisovalerate decarboxylase, an enzyme involved in the formation of aldehydes from amino acids by lactococcus lactis. *FEMS microbiology letters*, 238(2):367–374.

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

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rihawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. 2020. Prottrans: towards cracking the language of life's code through self-supervised deep learning and high performance computing. *arXiv preprint arXiv:2007.06225*.

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

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. 2023. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. *arXiv preprint arXiv:2306.08018*.

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. 2021. Structure-based protein function prediction using graph convolutional networks. *Nature communications*, 12(1):3168.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Pre-training to learn in context. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4849–4870, Toronto, Canada. Association for Computational Linguistics.

Pedro Hermosilla, Marco Schäfer, Matěj Lang, Gloria Fackelmann, Pere Pau Vázquez, Barbora Kozlíková, Michael Krone, Tobias Ritschel, and Timo Ropinski. 2021. Intrinsic-extrinsic convolution and pooling for learning on 3d protein structures. *International Conference on Learning Representations*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei

Cui, Owais Khan Mohammed, Qiang Liu, et al. 2023. Language is not all you need: Aligning perception with language models. *arXiv preprint arXiv:2302.14045*.

Shima Imani, Liang Du, and Harsh Shrivastava. 2023. Mathprompter: Mathematical reasoning using large language models. *arXiv preprint arXiv:2303.05398*.

Bowen Jing, Stephan Eismann, Pratham N. Soni, and Ron O. Dror. 2021. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*.

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

Hang Li, Xiu-Jun Gong, Hua Yu, and Chang Zhou. 2018. Deep neural network based predictions of protein interactions using primary sequences. *Molecules*, 23(8):1923.

Youwei Liang, Ruiyi Zhang, Li Zhang, and Pengtao Xie. 2023. Drugchat: towards enabling chatgpt-like capabilities on drug molecule graphs. *arXiv preprint arXiv:2309.03907*.

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. 2022. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*.

Shengchao Liu, Jiongxiao Wang, Yijin Yang, Chengpeng Wang, Ling Liu, Hongyu Guo, and Chaowei Xiao. 2023b. Chatgpt-powered conversational drug editing using retrieval and domain feedback. *arXiv preprint arXiv:2305.18090*.

Zequn Liu, Wei Zhang, Yingce Xia, Lijun Wu, Shufang Xie, Tao Qin, Ming Zhang, and Tie-Yan Liu. 2023c. Molxpt: Wrapping molecules with text for generative pre-training. *arXiv preprint arXiv:2305.10688*.

Zhiyuan Liu, Sihang Li, Yanchen Luo, Hao Fei, Yixin Cao, Kenji Kawaguchi, Xiang Wang, and Tat-Seng Chua. 2023d. Molca: Molecular graph-language modeling with cross-modal projector and uni-modal adapter. *arXiv preprint arXiv:2310.12798*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Eureka: Human-level reward design via coding large language models. *arXiv preprint arXiv: Arxiv-2310.12931*.

Wai Shun Mak, Stephen Tran, Ryan Marcheschi, Steve Bertolani, James Thompson, David Baker, James C Liao, and Justin B Siegel. 2015. Integrative genomic mining for enzyme function to enable engineering of a non-natural biosynthetic pathway. *Nature communications*, 6(1):10005.

Scott McGinnis and Thomas L Madden. 2004. Blast: at the core of a powerful and diverse set of sequence analysis tools. *Nucleic acids research*, 32(suppl_2):W20–W25.

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

Christian von Mering, Martijn Huynen, Daniel Jaeggi, Steffen Schmidt, Peer Bork, and Berend Snel. 2003. String: a database of predicted functional associations between proteins. *Nucleic acids research*, 31(1):258–261.

Meina Neumann, Gerd Mittelstädt, Chantal Iobbi-Nivol, Miguel Saggu, Friedhelm Lendzian, Peter Hildebrandt, and Silke Leimkühler. 2009. A periplasmic aldehyde oxidoreductase represents the first molybdopterin cytosine dinucleotide cofactor containing molybdo-flavoenzyme from escherichia coli. *The FEBS journal*, 276(10):2762–2774.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Xiao-Yong Pan, Ya-Nan Zhang, and Hong-Bin Shen. 2010. 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.

Predrag Radivojac, Wyatt T Clark, Tal Ronnen Oron, Alexandra M Schnoes, Tobias Wittkop, Artem Sokolov, Kiley Graim, Christopher Funk, Karin Verspoor, Asa Ben-Hur, et al. 2013. A large-scale evaluation of computational protein function prediction. *Nature methods*, 10(3):221–227.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*.

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

Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. 2023. Saprot: Protein language modeling with structure-aware vocabulary. *bioRxiv*, pages 2023–10.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Oleg Trott and Arthur J Olson. 2010. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461.

Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, et al. 2023a. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60.

Zeyuan Wang, Qiang Zhang, Keyan Ding, Ming Qin, Xiang Zhuang, Xiaotong Li, and Huajun Chen. 2023b. Instructprotein: Aligning human and protein language via knowledge instruction. *arXiv preprint arXiv:2310.03269*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*.

Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. 2023. Next-gpt: Any-to-any multimodal llm. *arXiv preprint arXiv:2309.05519*.

Minghao Xu, Yuanfan Guo, Yi Xu, Jian Tang, Xinlei Chen, and Yuandong Tian. 2023a. Eurnet: Efficient multi-range relational modeling of protein structure. In *ICLR 2023 - Machine Learning for Drug Discovery workshop*.

Minghao Xu, Xinyu Yuan, Santiago Miret, and Jian Tang. 2023b. Protst: Multi-modality learning of protein sequences and biomedical texts.

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

Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montse Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasenclever, Jan Humplik, et al. 2023. Language to rewards for robotic skill synthesis. *arXiv preprint arXiv:2306.08647*.

Ningyu Zhang, Zhen Bi, Xiaozhuan Liang, Siyuan Cheng, Haosen Hong, Shumin Deng, Qiang Zhang, Jiazhang Lian, and Huajun Chen. 2022a. Ontoprotein: Protein pretraining with gene ontology embedding. In *International Conference on Learning Representations*.

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

Zuobai Zhang, Minghao Xu, Arian Rokkum Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. 2023a. Protein representation learning by geometric structure pretraining. In *The Eleventh International Conference on Learning Representations*.

Zuobai Zhang, Minghao Xu, Aurelie Lozano, Vijil Chenthamarakshan, Payel Das, and Jian Tang. 2023b. Pre-training protein encoder via siamese sequence-structure diffusion trajectory prediction. In *Annual Conference on Neural Information Processing Systems*.

Jianan Zhao, Le Zhuo, Yikang Shen, Meng Qu, Kai Liu, Michael Bronstein, Zhaocheng Zhu, and Jian Tang. 2023. Graphtext: Graph reasoning in text space. *arXiv preprint arXiv:2310.01089*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

A Experimental Details

Hyperparameter	
Batch size	256
Sequence length	512
Training steps	10K
Optimizer	AdamW
Adam β	(0.9, 0.999)
Adam ϵ	1×10^{-6}
Learning rate	2×10^{-4}
Learning rate schedule	Cosine decay
Warmup ratio	0.03
Weight decay	0
LoRA r	32
LoRA a	64
LoRA dropout	0.1
LoRA modules	All linear modules

Table 4: Pre-training hyperparameters of PROTLLM.

Pre-training We list the detailed pre-training hyperparameters of PROTLLM in Table 4. PROTLLM is pre-trained on 4 NVIDIA A100 GPUs for $10,000$ steps with batch size 256 on the InterPT dataset described in Table 1. We adopt LoRA for efficient training, applying LoRA to all linear models of LLaMA, including `[down_proj, up_proj, q_proj, v_proj, k_proj, o_proj, gate_proj]`. Notice that only the LoRA weights and the cross-modal connector modules are updated during training.

Fine-tuning PROTLLM is further fine-tuned on various downstream tasks including EC number prediction, GO term prediction, and PPI prediction. Table 5 presents the fine-tuning hyperparameters. We apply LoRA for efficient tuning of the language model weights. The weights of the protein encoder are frozen for the PPI task, and updated for the other tasks, with a learning rate of 2×10^{-5}. The handcrafted prompt templates for each task are shown in Table 6. At each `[mount]` position, we encode the protein sequences with the protein encoder and feed the resulting protein embedding to the language model. For multilabel classification tasks, i.e., GO and EC, we convert the tasks to binary classification tasks for each label. We fill `[name]` with the label name, and fill `[description]` with text descriptions associated with this label. Besides, we utilize a resampling strategy during fine-tuning to ensure a uniform distribution of positive and negative labels.

Hyperparameter	EC	GO-BP	GO-MF	GO-CC	PPI
Batch size	128	128	128	128	16
Training steps	50K	50K	50K	10K	10K
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW
Learning rate	2×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-4}
Learning rate schedule	Cosine decay	Cosine decay	Cosine decay	Cosine decay	Cosine decay
Warmup ratio	0.03	0.03	0.03	0.03	0
Weight decay	0	0	0	0	0
LoRA r	128	128	128	128	32
LoRA a	256	256	256	256	64
LoRA dropout	0.1	0.1	0.1	0.1	0.1
LoRA modules	All linear	All linear	All linear	All linear	All linear
Update protein encoder	Yes	Yes	Yes	Yes	No

Table 5: Fine-tuning hyperparameters of PROTLLM on various downstream tasks.

Task	Prompt template	Verbalizer
GO	`<PROT> [mount] </PROT> Does the protein belong to [name], which is [description]?`	Yes/No
EC	`<PROT> [mount] </PROT> Does the protein catalyze [name], which is [description]?`	Yes/No
PPI	Do <PROT> [mount$_1$] </PROT> and <PROT> [mount$_2$] </PROT> interact with each other?	Yes/No

Table 6: Prompt templates for each task.