Efficiently Bridging Protein Language Model and Large Language Model with a Cross-modal Lightweight Adapter

Anonymous ACL submission

Abstract

In natural language processing and biology, large language models (LLMs) and protein language models (PLMs) have advanced significantly. Despite similarities in their organizational form, protein sequences and natural language lack direct semantic association due to domain differences. Thus, efficiently connecting LLMs and PLMs to leverage cross-field benefits and promote large model toolization remains a challenge. To bridge this gap, we propose a lightweight cross-modal adapter that aligns protein sequences with natural language representations through contrastive learning, effectively reducing modality difference, thereby bridging PLMs and LLMs and enhancing the performance of both. In the experiments, we first evaluated the performance of the PLM integrated with the adapter across multiple tasks. The experimental results show that the adapter achieved better results in many cases compared to using the PLM alone. Additionally, given the significant progress in protein-related LLMs, we further explored how the adapter can enhance this paradigm. In this experiment, we not only demonstrated that the adapter can enhance the LLM's ability to analyze protein sequences, outperforming other baseline models, but also proved the adapter's applicability in different base models.

1 Introduction

In recent years, significant breakthroughs in natural language processing, exemplified by models like ChatGPT series (OpenAI, 2023), Deepseek series (DeepSeek-AI et al., 2025) and the open-source Llama series (AI@Meta, 2024), have led to the development of powerful large language models (LLMs). These models have demonstrated impressive abilities across a wide range of fields, including natural language understanding, generation as well as tasks that go beyond traditional language processing(Zhao et al., 2023; Zhou et al., 2023; Liu et al., 2023; Dong et al., 2022; Zhang et al., 2024). Currently, large models are evolving towards multi-modal capabilities (Yin et al., 2024), typically using language models as a foundation to process different types of data, such as ChatGPT-4(OpenAI, 2023), Gemini1.5 (Reid et al., 2024), Blip-2(Li et al., 2023), Qwen-VL(Bai et al., 2023) and LLaVA(Liu et al., 2024). These multi-modal models can comprehensively handle various data types, including text, images, and audio, thereby extending the application range and capabilities of the models (Zhang et al., 2024).

At the same time, significant advancements have been made with protein language models (PLMs), exemplified by ProtBert (Brandes et al., 2022), OntoProtein (Zhang et al., 2022a), ESM1b (Rives et al., 2021) and ESM2 (Lin et al., 2022). These models have demonstrated exceptional performance in tasks such as protein structure prediction, functional analysis, and various other protein related research applications, thereby significantly advancing the field of biological sciences.(Bi et al., 2024; AI4Science and Quantum, 2023).

These foundational works demonstrate the immense potential of protein text generation, image understanding, and sequence representation, but there remains a significant opportunity to combine these models to leverage their complementary strengths for protein-related tasks. Some works both for PLM and LLM have illustrated this potential. One of the notable efforts is ProtST(Xu et al., 2023), which first fine-tunes PLMs by incorporating the knowledge from pre-trained biomedical models. Besides PLM which focus on protein sequence representation learning, there have been concurrent advancements in LLMs tailored to protein research. Protein2Text(Abdine et al., 2024) proposes a fused multi-modal encoder-decoder based protein textual description generation training framework. In this work, a protein structure encoder based on Relational Graph Convolutional

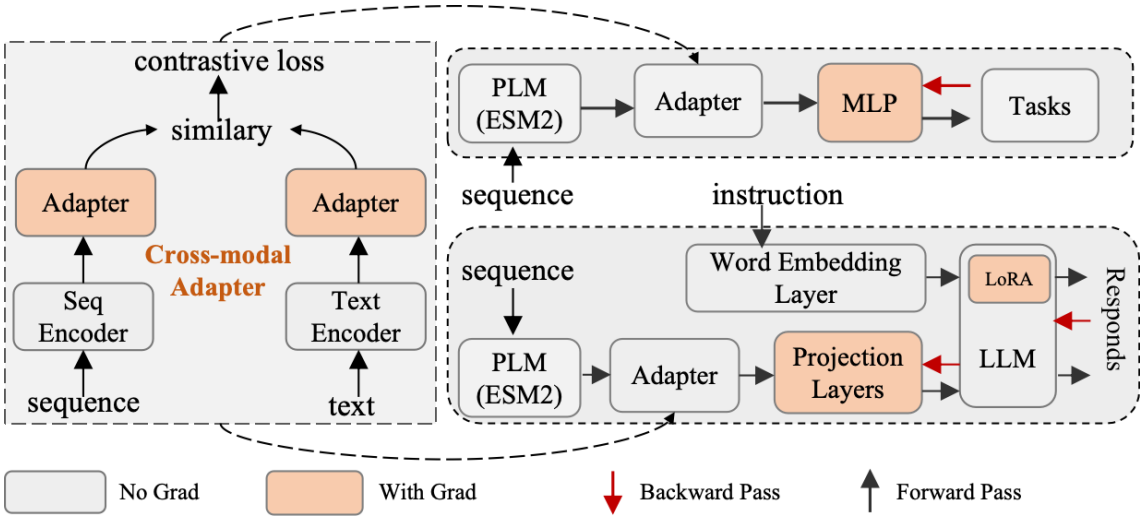


Figure 1: The framework of the cross-modal adapter

Neural Network (RGCN) and a protein sequence encoder based on ESM2 are both used for modality fusion and then generating protein's natural language function description using GPT2. InstructProtein(Wang et al., 2024) treats protein sequences as part of the natural language vocabulary and directly integrates them with natural language in the training of large language models on established knowledge graph datasets.

Although some methods have attempted to bridge PLMs and LLMs, challenges remain in developing more efficient and generalized integration strategies due to the lack of direct semantic association between protein sequences and natural language. Therefore, we propose a lightweight cross-modal adapter to bridge the PLM and LLM. Specifically, we construct the adapter using a linear projection layer and leverage contrastive learning to map the embeddings of the protein sequence encoder and the text encoder into a shared semantic space. It effectively mitigates the modality differences when integrating PLMs and LLMs. Additionally, the modular design of the adapter ensures its compatibility with various large models, enhancing the flexibility and applicability of our approach in different scenarios. Based on this, we evaluate the performance of the adapter when integrated with PLMs and explore its role in improving performance when bridging PLMs and LLMs. The experimental results show that the adapter not only improves the performance of PLMs in traditional representation tasks related to protein sequence analysis, but also enhances the performance of LLMs in the protein description generation downstream task. The contributions of this study can be summarized as follows:

- We propose a lightweight cross-modal adapter, which aligns the representation of protein sequences and text to bridge the LLMs and PLMs. This adapter can be directly applied to PLMs without fine-tuning the original model.
- The proposed lightweight cross-modal adapter enhances the original protein representation of the models on protein function prediction downstream tasks.
- The proposed lightweight cross-modal adapter enhances the performance of LLMs in the protein description generation downstream task.

2 Method

This section introduces the proposed lightweight cross-modal adapter. It aligns the feature representations of protein sequences and the semantic embedding space of texts, functioning as a bridge module between large language models (LLMs) and protein language models (PLMs). The framework is shown in Figure 1.

2.1 Adapter Module

In this paper, a protein data entry consists of a pair of protein sequences and text descriptions, represented as $P = (S, T)$. S is a protein sequence composed of n amino acids, $S = \{s_1, s_2, ..., s_n\}$, and the T is a description of the protein, $T = \{t_1, t_2, ..., t_n\}$.

We use ESM2 as the sequence encoder to convert protein sequences into high-dimensional embeddings. ESM2, a powerful protein language model, captures the intricate patterns and relationships within protein sequences, transforming them into meaningful vector representations.

$$\mathbf{z}_i^p = \frac{1}{n_i} \sum_{j=1}^{n_i} \text{ESM2}(s_{ij}) \tag{1}$$

where s_{ij} denotes the j-th amino acid of the i-th protein sequence and $\mathbf{z}_i^p$ is the corresponding high-dimensional embedding.

We use Llama3 as the text encoder to generate embeddings for text descriptions, encoding their semantic information into vector representations.

$$\mathbf{z}_i^t = \frac{1}{2} \cdot \text{mean}(\text{Llama}_{\text{first}}(t_i) + \text{Llama}_{\text{last}}(t_i)) \tag{2}$$

where t_i denotes the i-th text description, $\text{Llama}_{\text{first}}(t_i)$ and $\text{Llama}_{\text{last}}(t_i)$ represent the first and last hidden layer states of the Llama3 model for t_i, respectively, and $\mathbf{z}_i^t$ is the corresponding high-dimensional embedding.

We introduce the linear projection layer as the core component of the adapter module. This layer is tasked with aligning the embeddings of protein sequences with those of the text. The objective is to effectively map both into a unified representation space.

$$\begin{aligned} \mathbf{z}_i^{p'} &= W_p \mathbf{z}_i^p + \mathbf{b}_p \\ \mathbf{z}_i^{t'} &= W_t \mathbf{z}_i^t + \mathbf{b}_t \end{aligned} \tag{3}$$

where W_p and W_t are linear projection matrices, and $\mathbf{b}_p$ and $\mathbf{b}_t$ are bias terms.

2.2 Contrastive Sequence-text Pre-training

To achieve semantic consistency between protein sequences and textual descriptions, we introduced contrastive learning. By mapping protein sequences and textual descriptions into a shared representation space during training, the model learns how to express both modalities in this space. Although protein sequences and textual descriptions differ in form, they are represented as embeddings with similar meanings in the shared semantic space.

Contrastive learning enhances semantic consistency by minimizing the distance between positive pairs (e.g., a protein sequence and its corresponding textual description) and maximizing the distance between negative pairs (e.g., a protein sequence and an unrelated textual description). This approach reinforces the semantic similarity of positive pairs while weakening the relationship between negative pairs.

To implement contrastive learning effectively, we use the InfoNCE loss, which can be mathematically expressed as:

$$L = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp(\mathbf{z}_i^{p'} \cdot \mathbf{z}_i^{t'} / \tau)}{\sum_{j=1}^{N} \exp(\mathbf{z}_i^{p'} \cdot \mathbf{z}_j^{t'} / \tau)} \tag{4}$$

where $\mathbf{z}_i^{p'}$ and $\mathbf{z}_i^{t'}$ represent the embeddings of the positive pair for the i-th sample, and $\mathbf{z}_j^{t'}$ represents the embeddings of all other pairs (both positive and negative) in the batch except the i-th positive pair. τ is a temperature parameter that controls the sensitivity of the similarity measure.

Through iterative minimization of the InfoNCE loss during training, the adapter module effectively optimizes the parameters of the linear projection layer, resulting in the alignment of protein and text embeddings within a unified latent space. This process enhances the model's ability to comprehend and process cross-modal information by rigorously maximizing the semantic coherence of positive pairs while systematically minimizing the similarity of negative pairs.

2.3 Sampling Strategy

To ensure training stability, we integrate a fast sampling strategy, where each batch is formed by randomly selecting keys from groups and sampling one index for each key. Let the dataset be $D = \{d_1, d_2, \ldots, d_N\}$, where N is the total number of samples, and each sample d_i has an embedding $\mathbf{e}_i$.

We compute the mean of each sample embedding and convert it to a string, extracting the first 10 characters to create a key for each sample. Based on these mean strings, we create a dictionary mean_keys , where each key key_i corresponds to a list of sample indices$\{i_1, i_2, \ldots, i_k\}$ that share the same mean. During batch sampling, we randomly select keys from mean_keys and pick one sample index for each key. The total number of batches is computed as:

$$\text{total\_batches} = \begin{cases} \left\lfloor \frac{N}{\text{batch\_size}} \right\rfloor & \text{if drop\_last=True} \\ \left\lceil \frac{N}{\text{batch\_size}} \right\rceil & \text{if drop\_last=False} \end{cases} \tag{5}$$

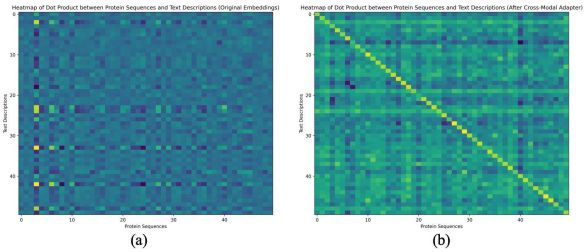
3 Experiments

In the experiments, we aimed to assess the cross-modal adapter's ability to bridge protein language model (PLM) and large language model (LLM), focusing on three key aspects: (1) measuring the correlation between aligned protein sequences and their textual descriptions; (2) Evaluate the performance of the PLM combined with adapters. (3) Evaluate whether adapters can effectively bridge PLM and LLM.

3.1 Alignment Effectiveness Analysis

We employed the ProtDescribe dataset to train the cross-modal adapter, constructed by (Xu et al., 2023), which contains 546,026 pairs of protein sequences and property descriptions. The data comes from the Swiss-Prot database (Bairoch and Apweiler, 2000), which provides annotations for various protein properties. In order to show the pretraining perfermance, we split this Dataset into Train-Valid-Test part: 436,822 pairs for the training set, 54,602 pairs for the validation set, and 54,602 pairs for the test set.

In this experiment, we evaluate the alignment performance by calculating the inner product correlation between protein sequences and their textual descriptions on ProtDescribe test dataset. The intensity of the heatmap colors reflects the strength of the correlation. As shown in Figure 2(a), the inner product heatmap of ESM2 and Llama3 embeddings displays a scattered pattern with many bright spots, indicating poor alignment and high noise. In contrast, Figure 2(b) shows a clear diagonal line after using the cross-modal adapter, indicating strong alignment and reduced noise. This result validates that the cross-modal adapter successfully maps protein sequences and their corresponding textual descriptions into the same semantic space.



(a) (b)

Figure 2: (a) Heatmap of sequence and text embeddings generated by ESM2 and Llama3. (b) Heatmap of aligned sequence and text embeddings using cross-modal adapter

3.2 Adapter Integration with PLM for Protein Representation Learning

In this experiment, we use ESM2 as the PLM, integrating it with the adapter (Adapter-ESM2) to validate its performance on protein representation learning tasks, including protein localization prediction, fitness landscape prediction, and protein function annotation.

3.2.1 Task Settings

- **Protein Localization Prediction** aims to determine the subcellular locations of proteins. In this context, we address two specific tasks from the DeepLoc dataset (Almagro Armenteros et al., 2017): subcellular localization prediction (Sub) with 10 distinct location categories, and binary localization prediction (abbreviated as Bin) with 2 location categories. We adhere to the official dataset splits for these tasks. This task involves both binary and multi-class classification, with accuracy being the metric for measuring outcomes.

- **Fitness Landscape Prediction** aims to predict the effect of residue mutations on protein fitness. We use several datasets for this purpose: the β-lactamase (abbreviated as β-lac) landscape from PEER (Xu et al., 2022), the AAV and Thermostability (Thermo) landscapes from FLIP (Dallago et al., 2021), and the Fluorescence (Flu) and Stability (Sta) landscapes from TAPE (Rao et al., 2019). For the AAV dataset, we use the "two vs many" splits, for the Thermostability dataset, we adopt the "human cell" splits, and for other tasks, we follow the default splits. This is a regression task where Spearman's ρ (Spearman's rank correlation coefficient) is used to assess the outcomes.

- **Protein Function Annotation** aims to assign multiple functional labels to a protein. We utilize two standard benchmarks proposed by DeepFRI (Gligorijević et al., 2021): Enzyme Commission (EC) number prediction and Gene Ontology (GO) term prediction. The GO benchmark is further divided into three branches: molecular function (abbreviated as GO-MF), biological process (abbreviated as GO-BP), and cellular component (abbreviated as GO-CC). Following (Zhang et al., 2022b), we use dataset splits with a 95%

Model	Loc. pred. (Acc%)		Fitness pred. (Spearman's ρ)					
	Bin	Sub	β-lac	AAV	Thermo	Flu	Sta	Mean-ρ
ResNet	78.99	52.30	0.152	**0.739**	0.528	0.636	0.126	0.436
LSTM	88.11	62.98	0.139	0.125	0.564	0.494	0.533	0.371
Transformer	75.74	56.02	0.261	0.681	0.545	**0.643**	0.649	0.556
ProtBert	81.54	59.44	0.616	0.209	0.562	0.339	0.697	0.485
OntoProtein	84.87	68.34	0.471	0.217	0.605	0.432	0.688	0.483
ESM1b	91.61	79.82	0.528	0.454	0.674	0.430	0.750	0.567
ESM2	91.32	80.84	0.559	0.374	0.677	0.456	0.746	0.562
ProtST-ESM2	92.52	**83.39**	0.586	0.398	0.681	0.499	0.776	0.584
Adapter-ESM2	**92.82**	82.10	**0.715**	0.426	**0.711**	0.570	**0.786**	**0.642**

Table 1: Results on protein localization and fitness landscape prediction.

Model	EC		GO-BP		GO-MF		GO-CC	
	AUPR	$F_{\max}$	AUPR	$F_{\max}$	AUPR	$F_{\max}$	AUPR	$F_{\max}$
ResNet	0.137	0.145	0.166	0.280	0.267	0.266	0.261	0.403
LSTM	0.032	0.082	0.130	0.248	0.100	0.166	0.150	0.320
Transformer	0.187	0.219	0.135	0.257	0.172	0.240	0.170	0.380
ProtBert	0.859	0.838	0.188	0.279	0.464	0.456	0.234	0.408
OntoProtein	0.854	0.841	0.284	0.436	0.603	0.631	0.300	0.441
ESM1b	0.884	0.869	0.332	0.452	0.630	0.659	0.324	0.477
ESM2	0.888	0.874	0.340	0.472	0.643	0.662	0.350	0.472
ProtST-ESM2	0.898	**0.878**	0.342	0.482	0.647	0.668	0.364	0.487
Adapter-ESM2	**0.901**	0.866	**0.367**	**0.490**	**0.676**	**0.669**	**0.386**	**0.503**

Table 2: Results on protein function annotation.

sequence identity cutoff for both EC and GO tasks. This is a multi-label classification task measured by Area Under the Precision-Recall Curve (AUPR) and F_{max}.

3.2.2 Baselines

We compare three categories of models: Protein sequence encoders trained from scratch: ResNet, LSTM and Transformer.(Rao et al., 2019). Four advanced protein language models (PLMs): ProtBert (Brandes et al., 2022), OntoProtein (Zhang et al., 2022a), ESM1b (Rives et al., 2021) and ESM2 (Lin et al., 2022). PLMs enhanced with biomedical texts through the ProtST framework (Xu et al., 2023), specifically using ProtST-ESM2 for comparison.

3.2.3 Results

The results for localization and fitness prediction are shown in Table 1, and those for function annotation are in Table 2. As illustrated in these tables, we observe the following:

Protein Localization Prediction (Loc. pred.): as shown in Table 1, Adapter-ESM2 (ESM2 with the cross-modal adapter) achieves accuracy of 92.82% and 82.10% in localization prediction task. Although it does not surpass ProtST-ESM2 in the subcellular localization task, it still outperforms traditional methods and other pre-trained language models (PLMs), especially with improvements of 1.5% and 1.26% over the base model ESM2.

Fitness Landscape Prediction (Fitness pred.): as shown in Table 1, we can find that Adapter-ESM2 achieves Spearman's correlation coefficients of 0.715, 0.426, 0.711, 0.570 and 0.786 on the five subtasks. Compared to the base model ESM2, Adapter-ESM2 achieved improvements of 0.156, 0.052, 0.034, 0.114 and 0.040. Additionally, compared to ProtST-ESM2, Adapter-ESM2 still achieved better performance in the Fitness Landscape Prediction task.

Protein Function Annotation (AUPR and F_{max}): as shown in Table 2, Adapter-ESM2 excels in four functional annotation tasks, achieving the highest AUPR (0.367, 0.676, and 0.386) and F_{max} (0.490, 0.669, and 0.503) scores in three of the tasks (GO-BP, GO-MF, and GO-CC), surpassing other models. The only exception is the EC task, where the F_{max} scores of ESM2 and ProST-ESM2 are 0.874 and 0.878, respectively, slightly higher than Adapter-ESM2's 0.866.

Based on the results of the downstream tasks experiments, we can observe that the cross-modal adapter not only successfully preserves the key biological characteristics of protein sequences but also significantly enhances these characteristics

through alignment with text. This indicates that the cross-modal adapter can retain the intrinsic properties of protein sequences while further improving their predictive capabilities across various bioinformatics tasks. Considering the low computation requirement of Cross-modal adapter training, this experiment highlight the efficiency of the Adapter approach in integrating textual semantics and protein representations.

3.3 Adapter-based Bridging of PLM and LLM for Protein Description Generation

In this experiment, we employ the adapter as the bridging module of ESM2 and a large language model (LLM) and investigate the potential benefits of the cross-modal adapter in protein description generation tasks. Our motivation is twofold: First, as introduced in previous sections, recent studies have demonstrated the feasibility of using auto-regressive natural language models to tackle protein-related tasks. We aim to explore whether the cross-modal adapter, which integrates protein sequences with textual descriptions, can further enhance this learning paradigm. Second, several works in visual multi-modal language models suggest that an encoder with superior generalization capability often serves as a catalyst for improved generative performance. Hence, we hypothesize that evaluating the generalization ability of the cross-modal adapter should include assessing its contributions to auto-regressive language model-based text generation.

3.3.1 Dataset

Instruction: Can you provide the functional description of the following protein sequence?

Input:MVKILKPGKVALITRGRFAGKKVVILQAIDQGSKSHPFGHAVV AGVERYPLKVTKSMGAKRIARRSRVKPFIKVVNYNHLMPTRYALEL DNLKGLITADTFKEPTQRSAARKTVKKTFEEKYQSGKSAWFFTPLRF

Ground Truth: Component of the ribosome, a large ribonucleoprotein complex responsible for the synthesis of proteins in the cell. The small ribosomal subunit binds messenger RNAs and translates the encoded message by selecting cognate aminoacyl transfer RNA molecules. The large subunit contains the ribosomal catalytic site termed the peptidyl transferase center, which catalyzes the formation of peptide bonds, thereby polymerizing the amino acids delivered by tRNAs into a polypeptide chain.

Figure 3: One entry of Protein description downstream dataset.

Protein Description Generation is a text-generation task aiming to generate protein's function generation based protein's information such as sequence and structure. In this task, our dataset is constructed based on the SwissProt (UniProtKB 2022-04 release), a high-quality curated protein knowledge base containing 256,690 different protein sequences. We set same dataset split ensuring a maximum sequence identity of 40% across splits with Prot2Text(Abdine et al., 2024), ensuring a maximum sequence identity of 40% across splits. Although SwissProt provides high-quality textual descriptions, we still need employ instruction expansion techniques to generate diverse rephrasings of prompts that has the same meaning with "Can you provide a detailed summary of this protein's function?". This process facilitates the construction of a supervised fine tuning(SFT) dataset by forming structured triples of protein sequences, protein descriptions, and the constructed instructions. (A data entry case is provided in Figure 3)

3.3.2 Downstream Task Settings

Experiment Setup: As shown in Figure 1, in order to let the adapter bridge the PLM and LLM, we use Adapter-ESM2 to connect multi-modal large language model and fine-tune the MLLM based on the SwissProt database to evaluate the impact of adapter integration on the model's ability to understand and process protein sequences. This fine-tuning employed Low-Rank Adaptation (LoRA) technology(Hu et al., 2021), by concatenating the input as $[< protein >]$ so that the model's response is grounded on both textual instructions and protein sequence inputs. For training pipeline, We employ a 2-stage training framework, which includes a pretraining-stage for a projection layer and a supervised finetuning stage using lora. On the first stage the parameters of language model are frozen, only the projection layer that converts adapter's output into language model's embedding space is trained. On the second stage, both the projection layer and the language model's parameters are trainable.

The experiment was conducted on 4 NVIDIA V100 GPUs. In order to show the applicability of the cross-modal adapter, we selected the different large language model including Llama3 and Galactica(Taylor et al., 2022) as the base model. For more training details and hyperparameter settings of the 2-stage pipeline, please refer to Appendix A.

3.3.3 Baselines and Evaluation Metrics

Since the same protein sequence split setting is applied, two models introduced by Prot2Text, including ESM2Text and Prot2Text, are considered

Model	Required		Evaluation Metrics				
	Sequence	Structure	ROUGE-1 ↑	ROUGE-2 ↑	ROUGE-L ↑	BERT SCORE ↑	BLEU ↑
$\text{Prot2Text}_{\text{BASE}}$ (Abdine et al., 2024)	✓	✓	0.5059	0.4217	0.4849	0.8430	0.3511
$\text{Prot2Text}_{\text{MEDIUM}}$ (Abdine et al., 2024)	✓	✓	0.5213	0.4417	0.5004	0.8483	0.3651
$\text{Prot2Text}_{\text{LARGE}}$ (Abdine et al., 2024)	✓	✓	0.5368	0.4560	0.5140	0.8520	0.3629
ESM2Text (Abdine et al., 2024)	✓	✗	0.4746	0.3918	0.4531	0.8321	0.3211
Llama3-8B Finetuning	✓	✗	0.1907	0.0876	0.1693	0.7182	0.0637
BioMedGPT (Luo et al., 2023)	✓	✗	0.3518	0.2355	0.3283	0.7961	0.1062
Adapter-ESM2-Galactica-125M	✓	✗	0.5308	0.4498	0.5100	0.8535	0.3429
Adapter-ESM2-Galactica-1.6B	✓	✗	0.5633	**0.4923**	0.5441	**0.8644**	**0.4117**
Adapter-ESM2-Llama3-8B	✓	✗	**0.5643**	0.4894	**0.5445**	0.8617	0.4084

Table 3: **Evaluation results of protein description generation task**

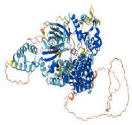
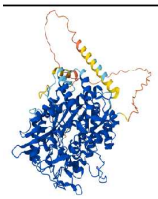
Protein	Ground_Truth	Adapter-ESM2-Llama3-8B	$\text{Prot2Text}_{\text{LARGE}}$	BioMedGPT
 Q8NG08	5'-3' DNA helicase involved in DNA damage response by acting as an inhibitor of DNA end resection. Recruitment to single-stranded DNA (ssDNA) following DNA damage leads to inhibit the nucleases catalyzing resection, such as EXO1, BLM and DNA2, possibly via the 5'-3' ssDNA translocase activity of HELB. As cells approach S phase, DNA end resection is promoted by the nuclear export of HELB following phosphorylation. Acts independently of TP53BP1. Unwinds duplex DNA with 5'-3' polarity. Has single-strand DNA dependent ATPase and DNA helicase activities. Prefers ATP and dATP as substrates. During S phase, may facilitate cellular recovery from replication stress.	5'-3' DNA helicase involved in DNA damage response by acting as an inhibitor of DNA end resection. Recruitment to single-stranded DNA (ssDNA) following DNA damage leads to inhibit the nucleases catalyzing resection, such as EXO1, BLM and DNA2, possibly via the 5'-3' ssDNA translocase activity of HELB. As cells approach S phase, DNA end resection is promoted by the nuclear export of HELB following phosphorylation. Acts independently of TP53BP1. Unwinds double-stranded DNA (dsDNA) with 5'-3' polarity. Has single-strand DNA (ssDNA)-dependent ATPase and DNA helicase activities. Prefers ATP and dATP as substrates. During S phase, may facilitate cellular recovery from replication stress.	DNA-dependent ATPase and 5'-3' DNA helicase required for the maintenance of genome stability. Involved in various processes such as transcription-coupled nucleotide excision repair, mitotic spindle assembly, DNA damage response and DNA repair. During DNA double-strand break (DSB) repair, involved in limiting the number of single-stranded DSBs through DNA end resection by promoting the recruitment of RAD51 to sites of DNA damage. In addition to DNA end reannealing, also promotes the resumption of DNA synthesis from the new 3'-tail end of the existing DNA ends. Acts as a regulator of telomerase by inhibiting telomeric silencing: binds to telomeromyosin, and is required for...	ATP-dependent helicase involved in DNA replication and repair. May also be involved in telomere maintenance.
 Q9W3K5	Catalyzes the ATP-dependent ligation of L-glutamate and L-cysteine and participates in the first and rate-limiting step in glutathione biosynthesis.	Catalyzes the ATP-dependent ligation of L-glutamate and L-cysteine and participates in the first and rate-limiting step in glutathione biosynthesis.	Catalyzes the specific cleavage of the interchain peptide bond of (S)-3-methylglutamate in the glutathione (GSH) residue of (S)-3-methylglutamate. S-3-methylglutamate is a universal solute responsible for many trace modifications in vertebrate skeletal muscle, including increased expression of the nitrogen-regulated genes nhr-80 and nhr-86, and increased expression of the potassium channel protein stp-1. Able to catalyze both uniport and transfer-on incorporation in vivo, with a maximal efficiency (in vitro).	Catalyzes the ATP-dependent conjugation of glutamate to the ammonia-forming compound glutamine, which is then released into the cytoplasm and used for protein synthesis.

Figure 4: Case Study: A comparision of Adapter-ESM2-Llama3's generation with Ground Truth and other models. Sentences marked in green in the description represent generated content part that has a **perfect match** with the ground truth. Blue indicates a rough match, meaning the predicted results may have **ambiguities or conceptual generalizations** compared with the ground truth. Red represents that the predicted results have **no relation to the ground truth or even contain some fatucal errors**.

as baselines. Additionally, BioMedGPT(Luo et al., 2023), another protein MLLM, is included in the comparison. BioMedGPT directly employs ESM2-3B as its encoder and utilizes BioMedGPT-LM as its base model. For evaluation metrics, we choose ROUGE(Lin, 2004), BERT-Score(Zhang et al., 2020) and BLEU(Papineni et al., 2002) to evaluate the generation performance of our models. ROUGE measures the overlap of n-grams, word sequences, and longest common subsequences between the generated and reference texts. BLEU quantifies n-gram precision by comparing generated outputs to reference texts. BERT-Score, which computes similarity using contextual embeddings, provides a more nuanced evaluation of semantic alignment; in our experiments, we employ BioBERTLARGE-cased v1.1 (Chakraborty et al., 2020) to calculate BERT-Score, leveraging its domain-specific understanding for protein function text.

3.3.4 Results

To validate the enhancement ability of adapter serving as language model's modality module and evaluate the protein sequence understanding ability, we train three sequence based models based on Llama3-8B, Galactica-125M, Adapter-ESM2-Galactica-1.6B. We call them Adapter-ESM2-Llama3-8B, Adapter-ESM2-Galactica-125M and Adapter-ESM2-Galactica-1.6B. Indeed, Prot2Texts need structure information as additional input so we use protein AlphaFoldDB ID as input. For sequence based models relatively, we uniformly use same system prompts(if required) and instruction 'What is the functional description of this protein?'. This instruction and protein sequence are provided as model inputs.

Model	ROUGE-1 ↑	ROUGE-2 ↑	ROUGE-L ↑	BERT SCORE ↑	BLEU ↑
Gala-125M w.o Adapter	0.4896	0.4086	0.4711	0.8430	0.3099
Gala-125M with ProtST (Xu et al., 2023)	0.5078	0.4296	0.4896	0.8490	0.3283
Gala-125M with Adapter	**0.5308**	**0.4498**	**0.5100**	**0.8535**	**0.3429**

Table 4: Ablation Study of Adapter Module. **Gala-125M w.o Adapter** refers that we directly use ESM2-650M's embedding without adapter and keep the training strategy and other settings same. Gala-125M with ProtST refers we load the finetuned ESM2 from ProtST.

As shown in Figure 3, it is easy to find that simply considering sequence as a part of natural language model and directly tuning Llama3 fails to achieve generalization ability under the condition of strictly controlling protein sequence similarity split. The results show that the most lightweight Adapter-ESM2-Galactica-125M achieves competitive results compared with $\text{Prot2Text}_{\text{MEDIUM}}$ and $\text{Prot2Text}_{\text{LARGE}}$. Without structure information, Adapter-ESM2-Llama3-8B outperforms all sequence baselines and sequence&structure baselines. Furthermore, Adapter-ESM2-Galactica-1.6B achieved even better performance than the base model Llama3-8B, maybe benefiting from its rich pretraining knowledge in the biomedical field. These results demonstrate the outperforming performance of the proposed adapter in bridging PLM and LLM.

We also provides two cases shown on Figure 4 from test dataset and compared Adapter-ESM2-Llama3-8B's generation with both ground truth from Swiss-prot and compared models' generation including Prot2Text and BioMedGPT. In the first case we select protein Q8NG08, a DNA helicase B which sequence consists of 1087 Amino acids. Adapter-ESM2-Llama3 excellently generate all function entries compared with the ground truth. While we also employed Prot2Text and BioMedGPT, they failed to give more exact answers even though Prot2Text generates a related topic like 'DNA double-strand break repair' and BioMedGPT predicts a vague answer involved with DNA replication and repair. Similar situation for case 2 of protein Q9W3K5, a Glutamate–cysteine ligase, Prot2Text provides exactly wrong function prediction even provided with structure information and BioMedGPT generates a relevant description but lack of comprehensiveness and exactitude compared with Adapter-ESM2-Llama3-8B.

To further validate the effectiveness of the proposed cross-modal adapter, we conduct an ablation study by removing the adapter and directly using ESM2 to process the protein sequences, or by replacing the adapter with ProtST's ESM2 module. The results in Table 4 show a noticeable decline in performance across all metrics when either ESM2 or ProtST's ESM2 is used. This indicates that directly using ESM2, without the adapter, results in lower performance. Additionally, fine-tuning ESM2 with textual information truly improves the performance of the bridged multi-modal large language model (MLLM), but not as efficiently as using the lightweight cross-modal adapter.

4 Conclusions and Future Work

In this paper, we present a lightweight cross-modal adapter that effectively bridges the gap between protein language models (PLMs) and large language models (LLMs). By embedding protein sequences and their corresponding textual descriptions into a unified semantic space, the adapter facilitates seamless integration between these two distinct modalities. The modular design of the adapter ensures compatibility with various large models, enhancing its applicability across different scenarios. This study highlights the potential of cross-modal adapters in both protein sequence representation learning and advancing the toolization of large models, enabling more effective utilization of both biological and natural language data. Future work will explore the integration of additional protein data modalities, such as structural information, with large models. This integration aims to further improve the generalization ability and applicability for large models, thereby advancing their use in both biology and natural language processing.

5 Limitations

In this paper, we focus on an efficient method that could bridge the gap between LLM and PLM, while only considering protein sequence-level representation. It remains unclear whether fine-grained amino acid or residue-level representations can be effectively enhanced using the adapter approach with text-labeled information. Another limitation is that we evaluate the adapter's performance only in protein description generation tasks. In future work, we plan to investigate whether the adapter can benefit multi-modal large language models in other protein-related tasks, particularly for complex annotation scenarios.

References

Hadi Abdine, Michail Chatzianastasis, Costas Bouyioukos, and Michalis Vazirgiannis. 2024. Prot2text: Multimodal protein's function generation with gnns and transformers. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(10):10757–10765.

Microsoft Research AI4Science and Microsoft Azure Quantum. 2023. The impact of large language models on scientific discovery: a preliminary study using gpt-4. *Preprint*, arXiv:2311.07361.

AI@Meta. 2024. Llama 3 model card.

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

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.

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

Zhenyu Bi, Sajib Acharjee Dip, Daniel Hajialigol, Sindhura Kommu, Hanwen Liu, Meng Lu, and Xuan Wang. 2024. Ai for biomedicine in the era of large language models. *Preprint*, arXiv:2403.15673.

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

Souradip Chakraborty, Ekaba Bisong, Shweta Bhatt, Thomas Wagner, Riley Elliott, and Francesco Mosconi. 2020. BioMedBERT: A pre-trained biomedical language model for QA and IR. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 669–679, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Christian Dallago, Jody Mou, Kadina E Johnston, Bruce J Wittmann, Nicholas Bhattacharya, Samuel Goldman, Ali Madani, and Kevin K Yang. 2021. Flip: Benchmark tasks in fitness landscape inference for proteins. *bioRxiv*, pages 2021–11.

DeepSeek-AI, Daya Guo, Dejian Yang, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Qi Dong, Lei Li, Dawei Dai, Cheng Zheng, Zhongli Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

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.

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

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

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*, 2022:500902.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Yizhen Luo, Jiahuan Zhang, Siqi Fan, Kai Yang, Yushuai Wu, Mu Qiao, and Zaiqing Nie. 2023.

Biomedgpt: Open multimodal generative pre-trained transformer for biomedicine. *Preprint*, arXiv:2308.09442.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

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):e2016239118.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *Preprint*, arXiv:2211.09085.

Zeyuan Wang, Qiang Zhang, Keyan Ding, Ming Qin, Xiang Zhuang, Xiaotong Li, and Huajun Chen. 2024. InstructProtein: Aligning human and protein language via knowledge instruction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1114–1136, Bangkok, Thailand. Association for Computational Linguistics.

Minghao Xu, Xinyu Yuan, Santiago Miret, and Jian Tang. 2023. Protst: Multi-modality learning of protein sequences and biomedical texts. In *International Conference on Machine Learning*, pages 38749–38767. PMLR.

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.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2024. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*.

Duzhen Zhang, Yahan Yu, Chenxing Li, Jiahua Dong, Dan Su, Chenhui Chu, and Dong Yu. 2024. Mm-llms: Recent advances in multimodal large language models. *arXiv preprint arXiv:2401.13601*.

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*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

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*.

Wayne Xin Zhao, Ke Zhou, Ji-Rong Li, Tianyu Tang, Xin Wang, Yulan Hou, Yongjun Min, Beichen Zhang, Junjie Zhang, Zhicheng Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Chunting Zhou, Qian Li, Cheng Li, Jing Yu, Yu Liu, Guang Wang, Kai Zhang, Cong Ji, Qiu Yan, Licheng He, et al. 2023. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *arXiv preprint arXiv:2302.09419*.

A Appendix

A.1 Training and Experiment Details

In training the cross-modal adapter, we used an Adam optimizer with a constant learning rate of $1.0 * 10^{-4}$ and a weight decay of 0.001. The model was trained for 100 epochs on a single NVIDIA A100 GPU, with a batch size of 128. Total training costs 1.5 GPU hours on a single NVIDIA A100 GPU.

In the training stage of protein description generation, we employ a 2-stage training pipeline. All experiments are constructed on 4 NVIDIA V100 GPU. For all models equipped with cross-modal adapter, we select a 'mlp2x_gelu' network as the projection layer that needs to convert cross-modal adpater's output into LLMs' token representation. We use a multi-head mlp layers which refers the final layer of the projector needs to convert cross-modal adapter's output vector into n tokens. We set $n = 8$ for all models including Adapter-ESM2 based models and ESM2/ProtST-ESM2-based models. In order to reduce the GPU memory cost and improve computational efficiency, we prepare ESM2 embedding before training stage by storaging them into jsonl file. Our practice shows that it can avoid esm uses up all GPU memory when facing long protein sequences. To minimize the impact of other factors and ensure a fair comparison, all models using same scale base llm model will only differ from encoders or adapter removal, while keeping all other parameters consistent. Details training hyperparameters of four different base models of protein description generation downstream task are shown on Table 5, Table 6 and Table 7.

In the LLM inference phase, to ensure performance consistency with the baseline model as described in the original paper, all our inference results are based on the parameter settings from the original paper. For the models of different scales and architectures that we trained, we use a consistent set of inference parameters to ensure a fair comparison: temperature $= 0.7$, top_p $= 0.8$, and num_beams $= 3$. For the 125M model, max_new_tokens is set to 512, while for the 1.6B, 8B models, max_new_tokens is set to 256.

Hyperparameter	Stage 1	Stage 2
Batch Size	256	256
Base LLM LR	✗	1e-3
Switch Projector LR	1e-3	1e-5
Weight Decay	0.0	0.0
Epochs	3	30
LR_Schedule	Constant	Warming Up
Warming Up Ratio	✗	0.006
Lora_r	✗	64
Lora_alpha	✗	16
Lora_dropout	✗	0.05
Lora_bias	✗	none
Model Max Length	512	512

Table 5: **Galactica-125M's hyperparameter settings serving as Base Model**

Hyperparameter	Stage 1	Stage 2
Batch Size	192	96
Base LLM LR	✗	1e-3
Switch Projector LR	1e-3	1e-5
Weight Decay	0.0	0.0
Epochs	3	15
LR_Schedule	Constant	Warming Up
Warming Up Ratio	✗	0.006
Lora_r	✗	64
Lora_alpha	✗	16
Lora_dropout	✗	0.05
Lora_bias	✗	none
Model Max Length	256	256

Table 6: **Galactica-1.6B's hyperparameter settings serving as Base Model**

Hyperparameter	Stage 1	Stage 2
Batch Size	32	24
Base LLM LR	✗	1e-3
Switch Projector LR	1e-3	1e-5
Weight Decay	0.0	0.0
Epochs	3	10
LR_Schedule	Constant	Warming Up
Warming Up Ratio	✗	0.006
Lora_r	✗	64
Lora_alpha	✗	16
Lora_dropout	✗	0.05
Lora_bias	✗	none
Model Max Length	256	256

Table 7: **Llama3-8B hyperparameter settings serving as Base Model**