ProtT3: Protein-to-Text Generation for Text-based Protein Understanding

Zhiyuan Liu[†] **An Zhang**[†] **Hao Fei**[†] **Enzhi Zhang**[§]
Xiang Wang[‡*] **Kenji Kawaguchi**[†] **Tat-Seng Chua**[†]
[†]National University of Singapore
[‡]University of Science and Technology of China [§]Hokkaido University
{acharkq,an.zhang3.14,xiangwang1223}@gmail.com
enzhi.zhang.n6@elms.hokudai.ac.jp, {feih, kenji,chuats}@comp.nus.edu.sg

Abstract

Language Models (LMs) excel in understanding textual descriptions of proteins, as evident in biomedical question-answering tasks. However, their capability falters with raw protein data, such as amino acid sequences, due to a deficit in pretraining on such data. Conversely, Protein Language Models (PLMs) can understand and convert protein data into high-quality representations, but struggle to process texts. To address their limitations, we introduce **ProtT3**, a framework for Protein-to-Text Generation for Text-based Protein Understanding. ProtT3 empowers an LM to understand protein sequences of amino acids by incorporating a PLM as its protein understanding module, enabling effective protein-to-text generation. This collaboration between PLM and LM is facilitated by a cross-modal projector (*i.e.,* Q-Former) that bridges the modality gap between the PLM's representation space and the LM's input space. Unlike previous studies focusing on protein property prediction and protein-text retrieval, we delve into the largely unexplored field of protein-to-text generation. To facilitate comprehensive benchmarks and promote future research, we establish quantitative evaluations for protein-text modeling tasks, including protein captioning, protein question-answering, and protein-text retrieval. Our experiments show that ProtT3 substantially surpasses current baselines, with ablation studies further highlighting the efficacy of its core components. Our code is available at https://github.com/acharkq/ProtT3.

1 Introduction

Language Models (LMs) have achieved impressive successes across diverse domains (Brown et al., 2020; Touvron et al., 2023; Liu et al., 2023c). Remarkably, the extensive biological literature in their training data has enabled LMs to excel in text-based protein understanding tasks, such as biological and medical question-answering (QA) (Taylor et al., 2022; OpenAI, 2023). These results show the potential of using LMs as the natural language interface for biomedical tasks. It further accentuates the importance of harnessing LMs to drive advancements in areas like drug discovery and protein property prediction (Kim et al., 2021; Fan et al., 2023).

Here we focus on LM's capability in text-based protein understanding, which enables textual interpretations of proteins. It can be assessed through diverse downstream tasks, particularly protein-to-text generation (Guo et al., 2023) and retrieval (Xu et al., 2023). Specifically, the retrieval task is finding an existing text description best matching a particular protein, while the generation task is delineated into

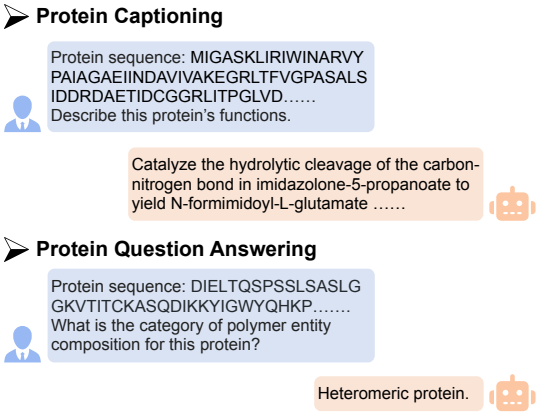


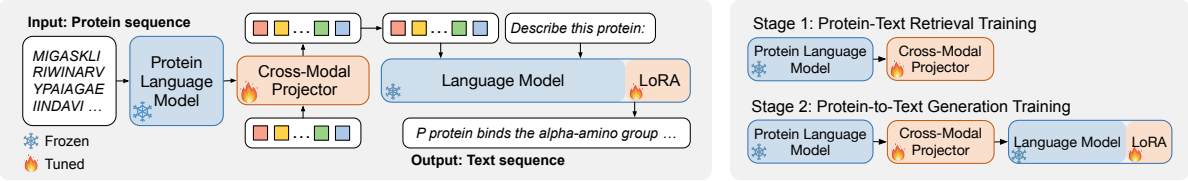
Figure 1: Examples of protein-to-text generation tasks. Proteins are represented by sequences of amino acids.

Methods	P2T Generation		Prot-Text Retrieval
	Caption	QA	
ProteinCLAP (Liu et al., 2023b)	✗	✗	?
ProtST (Xu et al., 2023)	✗	✗	?
Galactica (Taylor et al., 2022)	✓	?	✗
ProteinChat (Guo et al., 2023)	?	?	✗
Ours, ProtT3	✓	✓	✓

Table 1: Comparing the abilities of protein-text modeling methods. **?** denotes missing quantitative evaluation.

[*] Corresponding author. Xiang Wang is also affiliated with Institute of Dataspace, Hefei Comprehensive National Science Center.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5949–5966



(a) ProtT3's architecture.

(b) ProtT3's two training stages.

Figure 2: Overview of the ProtT3 framework.

two problems, as illustrated in Figure 1: 1) protein captioning, where the LM generates a description of a specific protein's functions, and 2) protein QA, where the LM answers questions about a protein. Table 1 overviews prior efforts (Liu et al., 2023b; Xu et al., 2023; Taylor et al., 2022; Guo et al., 2023) in this field. Delving into these previous studies, we identify two research gaps:

- **Lack of Exploration for Protein-to-Text Generation.** Protein-to-text generation is a conditional generation task (Keskar et al., 2019), requiring the LM to perceive proteins as the generation condition. Previous studies (Xu et al., 2023; Liu et al., 2023b) based on cross-modal contrastive learning (Radford et al., 2021) hardly interpret proteins as the direct inputs to the LM. While notably two studies, Galactica and ProteinChat, have paid certain explorations, they unfortunately appear to be quite constrained by key limitations. Specifically, Galactica (Taylor et al., 2022) incorporates only a limited set of protein sequences in its pretraining data, thereby potentially restricting its capacity for comprehensive protein understanding; ProteinChat (Guo et al., 2023) seeks to project protein representations to text space by training a linear projector, which might prove inadequate in capturing the intricate relations between proteins and texts.

- **Missing Quantitative Evaluation.** The progress in protein-text modeling is difficult to track without proper benchmarks. As Table 1 illustrates, the quantitative evaluations for these models are mostly missing, posing a challenge to further advancement in this field.

To bridge these research gaps, we propose **ProtT3**: Protein-to-Text Generation for Text-based Protein Understanding. As Figure 2a illustrates, ProtT3 empowers an LM to understand protein sequences of amino acids by incorporating a Protein Language Model (PLM) as its protein understanding module, thereby effectively conditioning the protein-to-text generation process. PLMs (Madani et al., 2023) are specialized LMs pretrained solely on protein sequences. They can generate powerful protein representations that are instrumental in presenting the proteins' 3D structures and indicating their potential properties (Chen et al., 2023). To enable the LM to understand the PLM's protein representations, ProtT3 integrates an expressive cross-modal projector – Q-Former (Li et al., 2023a) – to map protein representations into the text space of the LM. This design enables the LM to consume proteins as inputs. However, working with a large LM with billions of parameters raises a new challenge of maintaining the efficiency of downstream adaptation. Therefore, we incorporate a LoRA (Hu et al., 2022) adapter into the LM for efficient fine-tuning purposes.

To facilitate effective protein-text modeling, ProtT3 employs a two-stage training process to enhance protein-text modeling, as outlined in Figure 2b. The first stage involves protein-text retrieval training with three cross-modal tasks (Li et al., 2022a): protein-text contrasting, protein-text matching, and protein captioning. This stage not only empowers the cross-modal projector with the capability of protein-text retrieval, but also serves as a "warmup" before the second stage by encouraging the extraction of text-relevant protein features. In the second stage, we connect the cross-modal projector to the LM and conduct protein-to-text generation training.

Our key contributions are summarized as:

- We introduce ProtT3, a new framework aiming to bridge the modality gap between texts and proteins. Through a cross-modal projector, ProtT3 jointly uses a PLM for protein understanding and an LM for text processing, enabling effective protein-to-text generation.

- To set benchmarks and promote future research, we establish quantitative evaluations for protein-text modeling tasks, including protein captioning, protein QA, and protein-text retrieval. The datasets, evaluation scripts, and our pretrained

checkpoints will be made available online.

- ProtT3 achieves state-of-the-art performances across various tasks. For protein captioning, ProtT3 surpasses the baseline by over 10 BLEU-2 scores in the Swiss-Prot (Bairoch and Apweiler, 2000) and the ProteinKG25 (Zhang et al., 2022) datasets. For protein-text retrieval, ProtT3 outperforms baselines by over 14% in retrieval accuracy on the Swiss-Prot and ProteinKG25 datasets. Lastly, ProtT3 achieves 2.5% improvement of exact match performance for protein QA on the PDB-QA (Guo et al., 2023) dataset.

2 Related Works

Here we briefly review relevant fields of PLMs, protein-text modeling, and multi-modal LMs.

Protein Language Models (PLMs). PLMs are transformer-based LMs pretrained on large corpora of protein sequences for protein understanding and protein generation (Lin et al., 2022; Madani et al., 2023; Nijkamp et al., 2022; Chen et al., 2023; Elnaggar et al., 2021; Rives et al., 2021a; Meier et al., 2021). Similar to LMs for texts, PLMs are pretrained by the objective of masked language modeling (Lin et al., 2022) or auto-regressive modeling (Madani et al., 2023). PLMs have demonstrated promising performances on downstream tasks of 3D structure prediction and protein property prediction (Lin et al., 2022; Chen et al., 2023). However, they cannot process texts due to the absence of text in their pretraining data.

Protein-Text Modeling. The pioneering study, Galactica (Taylor et al., 2022), is an LM pretrained on a text corpus that includes a small amount of protein sequences. Its protein understanding ability can be suboptimal compared to PLMs due to insufficient protein pretraining data. To mitigate this gap, subsequent works (Xu et al., 2023; Liu et al., 2023b) jointly leverage a PLM and an LM by cross-modal contrastive learning (Radford et al., 2021). However, cross-modal contrastive learning is insufficient for protein-to-text generation. This is because protein-to-text generation is a conditional generation task. It demands the LM to understand proteins as the generation condition, which contrastive learning cannot achieve. Further, ProteinChat (Guo et al., 2023) attempts to enable an LM to understand proteins by training a linear projector between a PLM and an LM. Considering both the PLM and LM are kept frozen during training, this approach struggles to capture the nonlinear relations between proteins and texts. In a different approach, OntoProtein (Zhang et al., 2022) finetunes a PLM using a biology knowledge graph, focusing more on protein property prediction and less on text-related tasks.

Multi-modal LMs. Enabling LMs to understand another modality, such as image (Li et al., 2022a; Tsimpoukelli et al., 2021), video (Zhang et al., 2023), audio (Lyu et al., 2023), user-items (Liao et al., 2024), and molecule (Liu et al., 2023c; Shi et al., 2023; Li et al., 2024; Fang et al., 2024b; Edwards et al., 2022), has been an activate research direction. Notably, the research of multi-modal LMs is pioneered by vision-language modeling (VLM). VLM has been successfully applied for few-shot image classification, image captioning, and image QA (Li et al., 2022a; Alayrac et al., 2022). Specifically, to enable LMs to understand images, the leading VLM methods employ either 1) nonlinear and expressive cross-modal projectors (Li et al., 2023a; Alayrac et al., 2022), or 2) fine-tuned visual encoders and LMs on multi-modal datasets (Driess et al., 2023), while such studies are missing for protein-text modeling. Our work takes the initiative to explore this direction.

3 Model Architecture

Here we introduce ProtT3's three key components: a PLM for protein understanding, an LM for text processing, and a cross-modal projector to bridge the modality gap between the first two components.

Protein Language Model (PLM). We utilize ESM-2 (Lin et al., 2022) to encode protein sequences of amino acids. ESM-2 is an encoder-only transformer LM (Vaswani et al., 2017) pretrained on 60M protein sequences by masked language modeling (Devlin et al., 2019). PLMs have shown promising performances for protein folding (Lin et al., 2022), multiple sequence alignment (Rives et al., 2021b), and protein property prediction (Xu et al., 2023), demonstrating their effectiveness for capturing protein characteristics. For efficiency, we freeze ESM-2's weights in our training process.

Language Model (LM). We choose Galactica (Taylor et al., 2022) as the base LM. Galactica is a decoder-only transformer LM pretrained on a large collection of scientific papers, spanning disciplines like biology and medicine. Notably, Galactica has demonstrated a high-level understanding of protein concepts through its promising performances in biomedical QA benchmarks (Jin et al., 2019; Hendrycks et al., 2021). Furthermore, Galac-

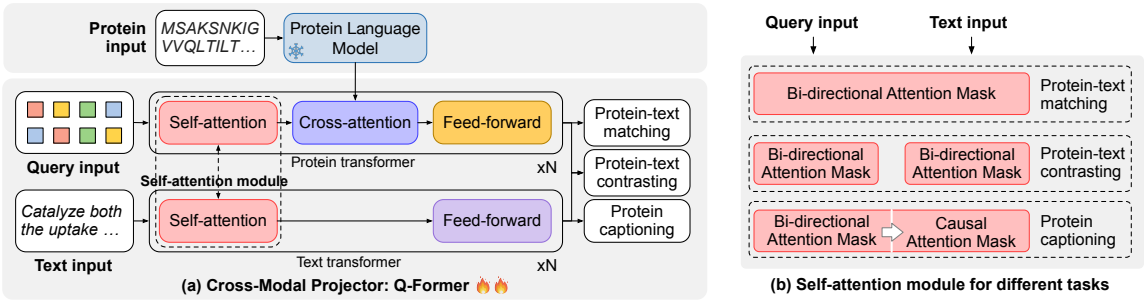


Figure 3: The training stage 1 of ProtT3. **(a):** Cross-Model Projector: Q-Former's architecture and the three training tasks. **(b):** The self-attention module uses different masking strategies for different tasks.

tica includes a small set of protein sequences in its pretraining data, and shows a capability for understanding protein sequences through the task of protein keyword prediction (Taylor et al., 2022). Therefore, we also leverage Galactica as a baseline for protein-to-text generation to ablate the effectiveness of incorporating an additional PLM for protein understanding.

Cross-Modal Projector. We employ a cross-modal projector based on Q-Former (Li et al., 2023a) to bridge the modality gap between the PLM and the LM. Q-Former has demonstrated promising performances in vision-language tasks. As Figure 3a illustrates, Q-Former consists of two transformers: one for protein encoding and another for text processing. Specifically, the protein transformer maintains N_q learnable query tokens $\{\boldsymbol{q}_i \in \mathbb{R}^d\}_{i=1}^{N_q}$ as inputs. These query tokens can interact with the PLM through the cross-attention modules, in order to extract protein features. We denote the protein transformer's output as $\mathbf{Z} \in \mathbb{R}^{N_q \times d}$, containing protein features. For text input, the text transformer adds a `[CLS]` token at the beginning, and uses the `[CLS]` token's output as the text representation. Both transformers share the self-attention to enable interactions between proteins and texts. Details are in Section 4.1.

Q-Former's weights are initialized from PubMedBERT-Abstract (Gu et al., 2022), a BERT LM pretrained on paper abstracts from the PubMed database[1]. It has shown promising performances in understanding biomedical concepts under the BLURB benchmark (Gu et al., 2022). The cross-attention module is added into the Q-Former every two layers and is randomly initialized.

4 Training Method

In this section, we introduce ProtT3's two training stages: protein-text retrieval and protein-to-text generation. The training process leverages a dataset of protein-text pairs $\mathcal{D} = \{(\boldsymbol{p}_1, \boldsymbol{t}_1), (\boldsymbol{p}_2, \boldsymbol{t}_2), ...\}$, where $\boldsymbol{p}_i$ is a protein sequence and $\boldsymbol{t}_i$ is the corresponding text sequence.

4.1 Stage 1: Protein-Text Retrieval Training

Inspired by BLIP (Li et al., 2022a, 2023a), we jointly employ three objectives for protein-text retrieval training: protein-text contrasting, protein-text matching, and protein captioning. These objectives are tailored for Q-Former's architecture, training it to extract protein features that are relevant to the text descriptions. This stage empowers the cross-modal projector with retrieval ability, and also serves as a "warmup" before the next stage.

Protein-Text Contrasting (PTC). We employ cross-modal contrastive learning (Radford et al., 2021) to align the protein representation and text representation from the Q-Former. As illustrated in Figure 3b, Q-Former's self-attention module separately processes the query tokens and text tokens without any interaction. This enforces the query tokens to extract protein features from the PLM, in order to generate protein representations that align with the corresponding text representations in contrastive learning.

Formally, let $\{(\boldsymbol{p}_1, \boldsymbol{t}_1), ...(\boldsymbol{p}_B, \boldsymbol{t}_B)\}$ be a batch of protein-text pairs. We denote the protein representations as $\{\mathbf{Z}_i \in \mathbb{R}^{N_q \times d}\}_{i=1}^{B}$, where $\mathbf{Z}_{ij} \in \mathbb{R}^d$ is the reprentation of the protein $\boldsymbol{p}_i$'s j-th query token; and denote text $\boldsymbol{t}_i$'s representation as $\boldsymbol{m}_i \in \mathbb{R}^d$, which is the `[CLS]` token's output. Protein-text similarity is measured by the maximum similarity between $\boldsymbol{m}$ and each row of $\mathbf{Z}$. The contrastive learning loss $\mathcal{L}_{\text{PTC}}$ can then be written as:

$$\mathcal{L}_{\text{p2t}} = \frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(\max_k \cos(\mathbf{Z}_{ik}, \boldsymbol{m}_i)/\tau)}{\sum_{j=1}^{B} \exp(\max_k \cos(\mathbf{Z}_{ik}, \boldsymbol{m}_j)/\tau)},$$

$$\mathcal{L}_{\text{t2p}} = \frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(\max_k \cos(\mathbf{Z}_{ik}, \boldsymbol{m}_i)/\tau)}{\sum_{j=1}^{B} \exp(\max_k \cos(\mathbf{Z}_{jk}, \boldsymbol{m}_i)/\tau)},$$

$$\mathcal{L}_{\text{PTC}} = -\mathcal{L}_{\text{p2t}} - \mathcal{L}_{\text{t2p}}, \quad (1)$$

[1] https://pubmed.ncbi.nlm.nih.gov/

where $\cos(\cdot,\cdot)$ is the cosine similarity; Temperature τ is empirically set to 0.1.

Protein-Text Matching (PTM). PTM is a binary classification objective aiming to discriminate whether a protein-text pair matches or not. Unlike PTC which computes protein-text similarity by applying cosine similarity on their output representations, PTM can obtain more fine-grained protein-text similarity. As illustrated in Figure 3b, PTM feeds the query tokens and text tokens into the same self-attention module, allowing them to interact at the Q-Former's early layers. In this way, the query tokens can capture information on both proteins and texts. The mean pooling of query tokens' representations is then fed into a linear classifier for PTM prediction. During training, for each positive protein-text pair $(\boldsymbol{p}_i, \boldsymbol{t}_i)$, we randomly sample two negative pairs: $(\boldsymbol{p}_i, \boldsymbol{t}_{i'})$ and $(\boldsymbol{p}_{i''}, \boldsymbol{t}_i)$. Let $f(\boldsymbol{p}, \boldsymbol{t}) \in [0, 1]$ be the Q-Former's predicted probability that $(\boldsymbol{p}, \boldsymbol{t})$ is a matched pair, the PTM loss function $\mathcal{L}_{\text{PTM}}$ can be written as:

$$\mathcal{L}_{\text{PTC}} = \frac{1}{B}\sum_{i=1}^{B}[-\log f(\boldsymbol{p}_i, \boldsymbol{t}_i) + \log f(\boldsymbol{p}_{i'}, \boldsymbol{t}_i) + \log f(\boldsymbol{p}_i, \boldsymbol{t}_{i''})]. \quad (2)$$

In experiments, we find that PTM surpasses PTC for protein-text retrieval. However, PTM incurs a higher computational cost for encoding every protein-text pair. To balance performance and speed, we use PTC to obtain the top k ranked candidates, and then use PTM for re-ranking.

Protein Captioning (PCap). Protein captioning trains the Q-Former to generate text descriptions of given proteins. As shown in Figure 3b, we apply a special masking strategy for this objective: 1) Bi-directional attention mask is applied to query tokens, enabling them to interact with each other but not text tokens; 2) Causal attention mask is used for text tokens, allowing them to attend query tokens and the preceding text tokens, but not the following text tokens. This design ensures the text tokens extract protein features exclusively from the query tokens, because they cannot directly interact with the proteins. Meanwhile, the query tokens are enforced to extract protein features through cross-attentions, satisfying the informational needs of protein captioning. Let $P(\boldsymbol{t}|\boldsymbol{p})$ be the probability that Q-Former generate text $\boldsymbol{t}$ given protein $\boldsymbol{p}$. The protein captioning loss $\mathcal{L}_{\text{PCap}}$ can be written as:

$$\mathcal{L}_{\text{PCap}} = -\frac{1}{B}\sum_{i=1}^{B}\log P(\boldsymbol{t}_i|\boldsymbol{p}_i). \quad (3)$$

	Size	Avg Protein Len	Avg Text Len
Train	430595	336	48
Valid	10000	358	59
Test	10000	357	60

(a) Swiss-Prot dataset of proteins and their text descriptions.

	Size	Avg Protein Len	Avg Text Len
Train	422315	338	101
Valid	10000	360	104
Test	10000	360	107

(b) ProteinKG25 dataset of proteins and their text descriptions.

	#Protein	#QA Pair	Avg Protein Len	Avg Text Len
Train	114690	3359693	291	10
Valid	10000	291795	257	10
Test	10000	291779	259	10

(c) PDB-QA dataset. Each protein is acompanied by multiple QA pairs, therefore proteins and QA pairs have different numbers.

Table 2: Statistics of our used protein-text datasets. Text lengths are counted by splitting at spaces.

4.2 Stage 2: Protein-to-Text Generation Training

In this stage, we train ProtT3 for protein-to-text generation. As Figure 2a illustrates, we connect the cross-modal projector to the LM, feeding the protein representations $\mathbf{Z}$ into the LM, so as to condition the text generation process by protein information. Note that, we use a linear layer to project $\mathbf{Z}$ to the same dimension of the LM's input.

We train ProtT3 for each generation dataset separately, and append different text prompts after the protein representations to further control the generation process. For example, we use the text prompt of "*Describe this protein's function*" for protein captioning, and "*Question: Does this protein contain polymer entities? Answer:*" for protein QA. For training, we use the same loss as the protein captioning task in the previous section.

For training efficiency, we incorporate Low-Rank Adaptation (LoRA) (Hu et al., 2022) adapters into the LM. LoRA adds pairs of trainable rank decomposition matrices into the selected weights of the LM. For example, LoRA modifies a pretrained weight $\mathbf{W}_0 \in \mathbb{R}^{d_1 \times d_2}$ by adding a pair of matrices $\mathbf{B} \in \mathbb{R}^{d_1 \times r}$ and $\mathbf{A} \in \mathbb{R}^{r \times d_2}$:

$$\mathbf{W} = \mathbf{W}_0 + \mathbf{BA}, \quad (4)$$

QA Pair	Type
Q: Does this protein contain polymer entities? A: Yes	String structure/property
Q: What are the software programs reported in connection with the production of this protein? A: CNS, REFMAC, SCALEPACK	String supplementary information
Q: How many polymer monomers does this protein have? A: 891	Number structure/property
Q: When is this protein first published? A: 2005	Number supplementary information

Table 3: Sampled QA pairs from the PDB-QA dataset. The sampled protein's PDB id is 1xf2.

Model	Base LM	Exact Match	BLEU-2	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L	METEOR
LoRA fine-tune	$\text{Galac}_{1.3B}$	13.49	42.48	37.79	44.42	34.73	42.46	48.27
ProtT3 w/ MLP Proj.	$\text{Galac}_{1.3B}$	20.60	48.96	44.59	57.28	50.17	56.89	57.30
ProtT3 w/o stage 1	$\text{Galac}_{1.3B}$	22.88	50.21	46.76	58.64	51.63	57.17	58.62
ProtT3	$\text{Galac}_{1.3B}$	**25.74**	**55.03**	**51.47**	**63.67**	**56.59**	**62.16**	**63.63**

(a) Performances (%) on the Swiss-Prot (Bairoch and Apweiler, 2000) dataset.

Model	Base LM	Exact Match	BLEU-2	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L	METEOR
LoRA fine-tune	$\text{Galac}_{1.3B}$	2.67	64.97	56.96	68.20	59.71	62.51	65.17
ProtT3 w/ MLP Proj.	$\text{Galac}_{1.3B}$	4.07	71.88	63.30	74.15	66.59	68.05	72.39
ProtT3 w/o stage 1	$\text{Galac}_{1.3B}$	4.85	72.33	64.79	75.67	68.21	69.34	73.00
ProtT3	$\text{Galac}_{1.3B}$	**5.48**	**76.53**	**68.67**	**78.29**	**70.50**	**71.40**	**76.76**

(b) Performances (%) on the ProteinKG25 (Zhang et al., 2022) dataset.

Table 4: Protein captioning performances on Swiss-Prot and ProteinKG25. **Bold** indicates the best performance.

where $\mathbf{W}_0$ is kept frozen and only the newly added $\mathbf{BA}$ is tuned. By using a small rank $r \ll \min(d_1, d_2)$, LoRA can adapt an LM to a new task while requiring little memory for storing gradients. This method has shown comparable performances to full-parameter fine-tuning (Hu et al., 2022).

5 Experiments

We begin by introducing the datasets of protein-text pairs, followed by the experimental results. Details on experimental settings, such as hyperparameters and baseline implementations, are provided in Appendix C. In the experiments, ProtT3 utilizes $\text{Galactica}_{1.3B}$ as the base LM and ESM-2_{150M} as the PLM.

5.1 Protein-Text Dataset Collection

In this section, we detail the protein-text pair datasets and the data processing procedure employed in our study. For all the used datasets, we discard protein sequences longer than 1022 tokens, and carefully split the datasets to ensure no overlap between train/valid/test sets. Dataset statistics are available in Table 2, and details are in Appendix A.

Swiss-Prot (Bairoch and Apweiler, 2000) is a protein sequence database with text annotations. We process the dataset following (Xu et al., 2023), but excluding protein names from the text annotations to prevent information leakage. The resulting text descriptions concatenate the annotations of protein functions, locations, and families.

ProteinKG25 (Zhang et al., 2022) is a knowledge graph derived from the Gene Ontology (Aleksander et al., 2023) database. We transform its triples into free texts by first aggregating the triples of the same protein, and then filling the protein information into a pre-defined text template.

PDB-QA (Guo et al., 2023) is a protein single-turn QA dataset derived from RCSB PDB[2]. It includes 30 question templates about proteins' structures, properties, and supplementary information. As shown in Table 3, to enable a fine-grained evaluation, we categorize the questions into four types, based on the answer's format (string or number) and content focus (structure/property or supplementary information). It is worth noting that supplementary information is hard to predict given the protein sequence alone.

Training Pipeline. In training stage 1, we train ProtT3 on the combination of the Swiss-Prot and ProteinKG25 datasets for protein-text retrieval. In training stage 2, we load the checkpoint from stage 1 and conduct separate fine-tuning on the three datasets for protein-to-text generation tasks.

[2] https://www.rcsb.org

	Retrieval in batch				Retrieval in test set			
	P2T		T2P		P2T		T2P	
Model	Acc	R@20	Acc	R@20	Acc	R@20	Acc	R@20
ProtST	81.4	99.4	79.6	99.4	15.9	56.6	17.8	63.5
ProteinCLAP	95.5	99.3	95.7	99.3	44.1	94.2	46.6	94.1
ProtT3 w/o PCap	97.2	**99.9**	97.1	**99.9**	66.4	95.3	66.2	95.2
ProtT3 w/o PTM	96.9	99.5	96.7	99.5	66.0	95.2	66.5	94.9
ProtT3	**97.7**	**99.9**	**97.5**	**99.9**	**68.3**	**96.0**	**68.1**	**95.8**

(a) Performances (%) on the Swiss-Prot (Bairoch and Apweiler, 2000) dataset.

	Retrieval in batch				Retrieval in test set			
	P2T		T2P		P2T		T2P	
Model	Acc	R@20	Acc	R@20	Acc	R@20	Acc	R@20
ProtST	70.8	98.5	70.9	98.2	5.5	41.6	5.8	43.3
ProteinCLAP	93.2	99.2	93.2	99.3	39.0	89.4	39.3	89.7
ProtT3 w/o PCap	**95.1**	99.8	95.0	**99.9**	53.4	91.2	53.0	91.2
ProtT3 w/o PTM	94.8	99.4	94.7	99.3	53.8	91.3	54.1	91.3
ProtT3	**95.1**	**99.9**	**95.3**	**99.9**	**55.8**	**91.7**	**55.6**	**91.7**

(b) Performances (%) on the ProteinKG25 (Zhang et al., 2022) dataset.

Table 5: Protein-text retrieval performances. **Bold** indicates the best performance and underline indicates the second best performance. We report performances of protein-to-text retrieval (P2T) and text-to-protein (T2P) retrieval.

5.2 Protein Captioning

We evaluate protein captioning performances on the Swiss-Prot and ProteinKG25 datasets. Following (Edwards et al., 2022), we use the evaluation metrics of BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005). Additionally, we report the percentage that the prediction exactly matches the ground truth annotation. For comparison and ablation study purposes, we report the performances of LoRA fine-tuned Galactica$_{1.3B}$, and ProtT3's two variants: 1) ProtT3 w/ MLP Proj., which replaces ProtT3's cross-modal projector by an MLP, following (Liu et al., 2023a); and 2) ProtT3 w/o stage 1, which skips ProtT3's training stage 1. We do not compare with ProteinChat (Guo et al., 2023) because it requires 3D protein structures, which are unavailable for these two datasets.

Table 4 presents the results. We observe the following: 1) ProtT3 and its variants substantially outperform the LoRA fine-tuned Galactica$_{1.3B}$, with ProtT3 showing a 10-point improvement in BLEU-2 scores. This underscores the significance of incorporating a PLM for protein understanding and ProtT3's effectiveness in undertanding protein inputs. 2) ProtT3 consistently surpasses its two variants on both datasets, highlighting the advantages of using a Q-Former projector and training stage 1.

5.3 Protein-Text Retrieval

We evaluate protein-text retrieval performances on the Swiss-Prot and ProteinKG25 datasets. Following (Su et al., 2022), our evaluation includes protein-text retrieval in a batch of 64 random samples and retrieval in the entire test set. We use Accuracy and Recall@20 as the evaluation metrics. For retrieval, ProtT3 first retrieves the top 128 candidates by PTC, and then uses PTM for re-ranking. For comparison, we employ ProtST (Xu et al., 2023) and ProteinCLAP (Liu et al., 2023b) as baselines, and present ProtT3 w/o PTM and ProtT3 w/o PCap for ablation studies.

Table 5 shows the results. We observe that: 1) ProtT3 outperforms baselines by over 14% accuracy for retrieval in the test set, highlighting its capability in aligning proteins with corresponding text descriptions. 2) PTM improves ProtT3's retrieval accuracy in test set by $1 \sim 2\%$ on both datasets. This is because PTM allows protein and text information to interact at the Q-Former's early layers, achieving more fine-grained protein-

		String		Number		
Model	Base LM	SP	SI	SP	SI	Overall
LoRA ft	Galac$_{1.3B}$	82.2	65.7	45.5	38.0	62.5
LoRA ft, Q-only	Galac$_{1.3B}$	76.0	66.4	44.7	**40.4**	60.2
ProteinChat	Vicuna$_{13B}$	7.2	10.3	28.9	19.2	15.5
ProtT3 w/o stage 1	Galac$_{1.3B}$	84.3	67.8	46.2	39.0	63.8
ProtT3	Galac$_{1.3B}$	**85.4**	**69.8**	**47.2**	39.4	**65.0**

Table 6: Exact match performances (%) for protein QA on the PDB-QA dataset (Guo et al., 2023). We categorize the QA pairs by their types. SP stands for structure/property, SI stands for supplementary information, and ft stands for fine-tuning. ProteinChat is evaluated using a checkpoint shared by the authors.

	Retrieval in batch				Retrieval in test set			
	P2T (%)		T2P (%)		P2T (%)		T2P (%)	
PLM	Acc	R@20	Acc	R@20	Acc	R@20	Acc	R@20
ESM-2_{8M}	96.6	**99.9**	96.2	99.8	64.8	94.1	64.4	93.7
ESM-2_{35M}	96.8	**99.9**	96.7	**99.9**	66.4	94.7	66.1	94.6
ESM-2_{150M}	**97.7**	**99.9**	**97.5**	**99.9**	**68.3**	**96.0**	**68.1**	**95.8**

(a) Performances (%) on the Swiss-Prot (Bairoch and Apweiler, 2000) dataset.

	Retrieval in batch				Retrieval in test set			
	P2T (%)		T2P (%)		P2T (%)		T2P (%)	
PLM	Acc	R@20	Acc	R@20	Acc	R@20	Acc	R@20
ESM-2_{8M}	93.1	99.8	93.5	99.7	51.9	88.6	51.7	89.0
ESM-2_{35M}	94.0	99.8	94.3	99.8	54.1	89.8	54.0	90.1
ESM-2_{150M}	**95.1**	**99.9**	**95.3**	**99.9**	**55.8**	**91.7**	**55.6**	**91.7**

(b) Performances (%) on the ProteinKG25 (Zhang et al., 2022) dataset.

Table 7: Ablation studies of PLMs for molecule-text retrieval.

Base LM	Exact Match	BLEU-2	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L	METEOR
Phi-$1.5_{1.3B}$	1.13	71.71	63.65	68.62	60.29	61.52	69.28
$Galac_{1.3B}$	**5.48**	**76.53**	**68.67**	**78.29**	**70.50**	**71.40**	**76.76**

Table 8: Ablation studies of textual LMs for protein captioning on the ProteinKG25 (Zhang et al., 2022) dataset.

text similarity measurements. 3) PCap improves ProtT3's retrieval accuracy by $\sim 2\%$. This is because PCap encourages the query tokens to extract protein information most relevant to the text input, therefore aiding in protein-text alignment.

5.4 Protein Question-Answering

We evaluate protein QA performances on the PDB-QA dataset. Considering that the answers in PDB-QA typically consist of $1 \sim 2$ words, we select exact match as the evaluation metric. For comparison, we employ ProteinChat (Guo et al., 2023) and LoRA fine-tuned $Galactica_{1.3B}$ as baselines. We also assess a $Galactica_{1.3B}$ that consumes only questions without proteins during training and prediction (*i.e.,* LoRA ft, Q-only). This baseline measures the dataset's uni-modal bias (Cadène et al., 2019): the proportion of questions that can be correctly answered without looking at proteins.

The results are shown in Table 6. We observe that: 1) ProtT3 surpasses baselines by 2.5% overall, and consistently outperforms them in predicting protein structures and properties. This demonstrates ProtT3's multi-modal understanding ability for the proteins and the textual questions. 2) The baseline consuming question only (*i.e.,* LoRA ft, Q-only) presents comparable performance, indicating that the dataset has a substantial uni-modal bias. This finding suggests an opportunity to explore debias techniques (Mahabadi et al., 2020) in future studies. 3) ProteinChat underperforms other methods, possibly because it only trains a linear layer between the frozen PLM and LM. The linear layer is insufficient to map the protein representations into the text space of the LM, and adapt the LM to QA. 4) Models perform worse on questions about numbers or supplementary information. This observation suggests the potential benefit of augmenting these models with external tools (Schick et al., 2023) like calculators and search engines.

5.5 Ablation Studies on Pretrained Models

Here we conduct ablation studies on the pretrained models used in our method. Specifically, we ablate the impact of different PLMs in training stage 1, and ablate different LMs in training stage 2.

Ablating PLMs. In training stage 1, we replace the ESM-2_{150M} protein encoder with its smaller variants, namely ESM-2_{8M} and ESM-2_{35M}, to evaluate their performances for protein-text retrieval. As shown in Table 7, we can observe that retrieval performance increases monotonically with model size, a trend consistent with previous observations in the LM domain.

Ablating LMs. In training stage 2, we replace $Galactica_{1.3B}$ to Phi-$1.5_{1.3B}$ (Li et al., 2023c). Unlike Galactica, Phi-$1.5_{1.3B}$ is pretrained on general domain data but not focusing on scientific liter-

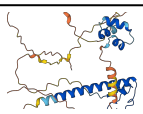
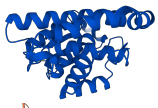
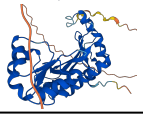
UniProt ID		Ground truth	ProtT3	Galactica
Q8CFG4		SUBCELLULAR LOCATION: Nucleus. SIMILARITY: Belongs to the DMRT family.	FUNCTION: May be involved in sexual development. SUBCELLULAR LOCATION: Nucleus. SIMILARITY: Belongs to the DMRT family.	FUNCTION: Orphan nuclear receptor. SUBCELLU-LAR LOCATION: Nucleus. SIMILARITY: Belongs to the nuclear hormone receptor family.
P46427		FUNCTION: Appears to play a central role in the parasite detoxification system. SIMILARITY: Belongs to the GST superfamily. Pi family.	FUNCTION: Conjugation of reduced glutathion-e to a wide number of exogenous and endog-enous hydrophobic electrophiles. SIMILARITY: Belongs to the GST superfamily. Pi family.	FUNCTION: Conjugation of reduced glutathione to a wide number of exogenous and endogenous hydrophobic electrophiles. SIMILARITY: Belongs to the GST superfamily. Theta family.
Q9LY66		FUNCTION: This protein binds directly to 23S ribosomal RNA. SUBCELLULAR LOCATION: Plastid, chloroplast. SIMILARITY: Belongs to the universal ribosomal protein uL1 family.	FUNCTION: Binds directly to 23S rRNA. SUBCELLULAR LOCATION: Plastid, chloroplast. SIMILARITY: Belongs to the universal ribosomal protein uL1 family.	SUBCELLULAR LOCATION: Plastid, chloroplast. SIMILARITY: Belongs to the universal ribosomal protein uL1 family.

Figure 4: Protein captioning examples from Swiss-Prot. We highlight sentences that exactly match the ground truth. Figures of protein structures are generated by AlphaFold2 (Jumper et al., 2021).

QA Pair	ProtT3	Galactica	ProteinChat
Q: What are the terms characterizing the protein? A:TRANSPORT PROTEIN.	TRANSPORT PROTEIN.	TRANSPOR TPROTEIN.	1.
Q: What are the bound nonpolymer components for this protein A: MG.	MG.	MG.	2F2, ATP.
Q: How many assemblies does this protein have? A: 2.	1.	0.	0.

Figure 5: Examples of protein QA results in the PDB-QA dataset. We highlight the correct predictions.

ature. Table 8 shows protein captioning performance on the ProteinKG25 (Zhang et al., 2022) dataset. We can observe that Galactica$_{1.3B}$ significantly outperforms Phi-1.5$_{1.3B}$ for protein captioning, although they have similar sizes. We attribute this performance gap to their pretraining corpus, with Galactica$_{1.3B}$ performing better for pretraining on more scientific literature.

5.6 Examples of Protein-to-Text Generation

Figure 4 shows three examples of protein captioning. In the first example, ProtT3's caption is more accurate by correctly identifying the DMRT family, while Galactica does not. In the second example, both models fail to identify the protein's function. Nevertheless, ProtT3's prediction regarding the protein family is more accurate. The third example shows that both models successfully predict the subcellular location and protein family. ProtT3 goes a step further by predicting the protein's function, which is closer to the ground truth description.

Figure 5 shows three examples of protein QA. We can observe that both ProtT3 and Galactica answer the first two questions about protein property/structure correctly, and fail on the third question, which requires a numerical answer. On the other hand, ProteinChat struggles with all three questions, failing to answer each of them.

6 Conclusion and Future Works

In this work, we propose ProtT3, a new protein-text modeling framework. ProtT3 aims to facilitate text-based protein understanding via protein-to-text generation and protein-text retrieval. To achieve this, ProtT3 integrates a PLM into an LM to enhance the LM's protein understanding ability. A cross-modal projector enables this integration by bridging the modality gap between the two modules. To promote future research, we set benchmarks for protein-text modeling tasks, including protein captioning, protein QA, and protein-text retrieval, where ProtT3 significantly outperforms existing baselines.

Looking ahead, we plan to explore enabling LMs to understand 3D protein structures (Li et al., 2024) and apply this understanding to more tasks of drug discovery, property prediction (Liu et al., 2023d; Li et al., 2022b, 2023b), molecule generation (Luo et al., 2024), and OOD generalization (Fang et al., 2023, 2024a).

Limitations

On Representing Proteins by 1D Sequences. Proteins can be represented by either 1D sequences of amino acids or 3D coordinates of their atoms. In this study, we represent proteins by 1D sequences considering their abundance compared to 3D structures. As of 2023, the Protein Data Bank contains 220K 3D structures [3], whereas the UniProt database houses 227M 1D sequences (Consortium, 2023). This abundance of 1D sequences enables us to collect a larger dataset. In Section 5.1, we collect a total of 1M protein-text pairs, a significant increase from the 143K pairs in the earlier work (Guo et al., 2023) using 3D structures.

[3] https://www.wwpdb.org/stats/deposition

While it is feasible to predict the 3D structures of protein sequences by running protein folding algorithms (Jumper et al., 2021; Lin et al., 2022), this prediction process is notably time-consuming. For example, predicting the 3D structures of the 440k protein sequences in the ProteinKG25 (Zhang et al., 2022) dataset would take approximately four months when using an A100 80GB GPU. Due to this computational demand, we choose to focus on 1D sequences, leaving the exploration of 3D structures for future research.

On Dynamic Protein Structures. Proteins have dynamic structures, and natural language is limited in fully depicting protein dynamics. Nonetheless, similar to video-to-text generation (Aafaq et al., 2020), where language partially describes dynamic videos, our work on protein-to-text generation is a crucial step towards encapsulating complex biological phenomena in a human-readable form.

Potential Ethics Impact

In this study, the proposed method and dataset focus on proteins and their properties, and include no human subjects. Consequently, we believe this study presents no direct ethical concerns. However, the proposed LMs can be abused to generate biased or toxic information, and can generate inaccurate description of protein properties and structures. Therefore, the ethical implications of our work align with those common to LM research.

Acknowledgement

This research is supported by the National Science and Technology Major Project (2023ZD0121102), National Natural Science Foundation of China (92270114). This research is partially supported by the National Research Foundation Singapore under the AI Singapore Programme (AISG Award No: AISG2-TC-2023-010-SGIL), the Singapore Ministry of Education Academic Research Fund Tier 1 (Award No: T1 251RES2207) and the Google Cloud Research Credits program with the award (Q4MJ-YH1K-3MVX-FP6Q). This research is supported by NExT Research Center.

References

Nayyer Aafaq, Ajmal Mian, Wei Liu, Syed Zulqarnain Gilani, and Mubarak Shah. 2020. Video description: A survey of methods, datasets, and evaluation metrics. *ACM Comput. Surv.*, 52(6):115:1–115:37.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. 2022. Flamingo: a visual language model for few-shot learning. In *NeurIPS*.

Suzi A Aleksander, James Balhoff, Seth Carbon, J Michael Cherry, Harold J Drabkin, Dustin Ebert, Marc Feuermann, Pascale Gaudet, Nomi L Harris, et al. 2023. The gene ontology knowledgebase in 2023. *Genetics*, page iyad031.

Amos Bairoch and Rolf Apweiler. 2000. The SWISS-PROT protein sequence database and its supplement trembl in 2000. *Nucleic Acids Res.*, 28(1):45–48.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *IEEvaluation@ACL*, pages 65–72. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. In *EMNLP/IJCNLP (1)*, pages 3613–3618. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*.

Rémi Cadène, Corentin Dancette, Hédi Ben-Younes, Matthieu Cord, and Devi Parikh. 2019. Rubi: Reducing unimodal biases for visual question answering. In *NeurIPS*, pages 839–850.

Bo Chen, Xingyi Cheng, Yangli-ao Geng, Shen Li, Xin Zeng, Boyan Wang, Jing Gong, Chiming Liu, Aohan Zeng, Yuxiao Dong, Jie Tang, and Le Song. 2023. xtrimopglm: Unified 100b-scale pre-trained transformer for deciphering the language of protein. *BioRxiv*.

The UniProt Consortium. 2023. Uniprot: the universal protein knowledgebase in 2023. *Nucleic Acids Research*, 51(D1):D523–D531.

Tri Dao. 2023. FlashAttention-2: Faster attention with better parallelism and work partitioning.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186. Association for Computational Linguistics.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. 2022. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *arXiv preprint arXiv:2203.06904*.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. 2023. Palm-e: An embodied multimodal language model. In *ICML*.

Carl Edwards, Tuan Manh Lai, Kevin Ros, Garrett Honke, Kyunghyun Cho, and Heng Ji. 2022. Translation between molecules and natural language. In *EMNLP*, pages 375–413. Association for Computational Linguistics.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Wang Yu, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. 2021. Prottrans: Towards cracking the language of lifes code through self-supervised deep learning and high performance computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

Hehe Fan, Zhangyang Wang, Yi Yang, and Mohan S. Kankanhalli. 2023. Continuous-discrete convolution for geometry-sequence modeling in proteins. In *ICLR*. OpenReview.net.

Junfeng Fang, Xinglin Li, Yongduo Sui, Yuan Gao, Guibin Zhang, Kun Wang, Xiang Wang, and Xiangnan He. 2024a. Exgc: Bridging efficiency and explainability in graph condensation. In *WWW*. ACM.

Junfeng Fang, Wei Liu, Yuan Gao, Zemin Liu, An Zhang, Xiang Wang, and Xiangnan He. 2023. Evaluating post-hoc explanations for graph neural networks via robustness analysis. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Junfeng Fang, Shuai Zhang, Chang Wu, Zhiyuan Liu, Sihang Li, Kun Wang, Wenjie Du, Xiang Wang, and Xiangnan He. 2024b. Moltc: Towards molecular relational modeling in language models.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2022. Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Heal.*, 3(1):2:1–2:23.

Han Guo, Mingjia Huo, Ruiyi Zhang, and Pengtao Xie. 2023. Proteinchat: Towards achieving chatgpt-like functionalities on protein 3d structures.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *ICLR*. OpenReview.net.

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 *ICLR*. OpenReview.net.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *EMNLP/IJCNLP (1)*, pages 2567–2577. Association for Computational Linguistics.

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.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Jintae Kim, Sera Park, Dongbo Min, and Wankyu Kim. 2021. Comprehensive survey of recent drug discovery using deep learning. *International Journal of Molecular Sciences*, 22(18):9983.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. 2023a. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. *CoRR*, abs/2301.12597.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. 2022a. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900. PMLR.

Sihang Li, Zhiyuan Liu, Yanchen Luo, Xiang Wang, Xiangnan He, Kenji Kawaguchi, Tat-Seng Chua, and Qi Tian. 2024. Towards 3d molecule-text interpretation in language models. In *ICLR*.

Sihang Li, Yanchen Luo, An Zhang, Xiang Wang, Xiangnan He, and Tat-Seng Chua. 2023b. Self-attentive rationalization for graph contrastive learning.

Sihang Li, Xiang Wang, An Zhang, Yingxin Wu, Xiangnan He, and Tat-Seng Chua. 2022b. Let invariant rationale discovery inspire graph contrastive learning. In *International conference on machine learning*, pages 13052–13065. PMLR.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023c. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*.

Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. 2024. Llara: Large language-recommendation assistant.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

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, Yuheng Li, and Yong Jae Lee. 2023a. Improved baselines with visual instruction tuning. *CoRR*, abs/2310.03744.

Shengchao Liu, Yutao Zhu, Jiarui Lu, Zhao Xu, Weili Nie, Anthony Gitter, Chaowei Xiao, Jian Tang, Hongyu Guo, and Anima Anandkumar. 2023b. A text-guided protein design framework. *arXiv preprint arXiv:2302.04611*.

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

Zhiyuan Liu, Yaorui Shi, An Zhang, Enzhi Zhang, Kenji Kawaguchi, Xiang Wang, and Tat-Seng Chua. 2023d. Rethinking tokenizer and decoder in masked graph modeling for molecules. In *NeurIPS*.

Yanchen Luo, Sihang Li, Zhiyuan Liu, Jiancan Wu, Zhengyi Yang, Xiangnan He, Xiang Wang, and Qi Tian. 2024. Text-guided diffusion model for 3d molecule generation.

Chenyang Lyu, Minghao Wu, Longyue Wang, Xinting Huang, Bingshuai Liu, Zefeng Du, Shuming Shi, and Zhaopeng Tu. 2023. Macaw-llm: Multi-modal language modeling with image, audio, video, and text integration. *arXiv*.

Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos Jr, Caiming Xiong, Zachary Z Sun, Richard Socher, et al. 2023. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, pages 1–8.

Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. 2020. End-to-end bias mitigation by modelling biases in corpora. In *ACL*, pages 8706–8716. Association for Computational Linguistics.

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. In *NeurIPS*, pages 29287–29303.

Erik Nijkamp, Jeffrey Ruffolo, Eli N. Weinstein, Nikhil Naik, and Ali Madani. 2022. Progen2: Exploring the boundaries of protein language models. *arXiv*.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318. ACL.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. 2021a. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci. USA*, 118(15):e2016239118.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. 2021b. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci. USA*, page e2016239118.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *CoRR*, abs/2302.04761.

Yaorui Shi, An Zhang, Enzhi Zhang, Zhiyuan Liu, and Xiang Wang. 2023. ReLM: Leveraging language models for enhanced chemical reaction prediction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5506–5520.

Bing Su, Dazhao Du, Zhao Yang, Yujie Zhou, Jiangmeng Li, Anyi Rao, Hao Sun, Zhiwu Lu, and Ji-Rong Wen. 2022. A molecular multimodal foundation model associating molecule graphs with natural language. *CoRR*, abs/2209.05481.

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. *CoRR*, abs/2211.09085.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. 2021. Multimodal few-shot learning with frozen language models. In *NeurIPS*, pages 200–212.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

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

Hang Zhang, Xin Li, and Lidong Bing. 2023. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv*.

Ningyu Zhang, Zhen Bi, Xiaozhuan Liang, Siyuan Cheng, Haosen Hong, Shumin Deng, Qiang Zhang, Jiazhang Lian, and Huajun Chen. 2022. Ontoprotein: Protein pretraining with gene ontology embedding. In *ICLR*. OpenReview.net.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

A Details of Protein-Text Dataset Collection

Swiss-Prot (Bairoch and Apweiler, 2000). Dataset samples are provided in Table 11. We largely follow the instructions in (Xu et al., 2023) for data processing. We first download the lastest dump of Swiss-Prot from the UniProt website[4], and then select three annotation fields: *Function*, which describes protein's roles such as catalysis and transport; *Subcellular Location*, indicating where a protein is typically found within a cell; *Similarity*, which details the protein's family. These three annotation fields are then concatenated to form the text description of proteins. Different from (Xu et al., 2023), we do not include the annotation of *Protein Name* in the text description. This is because *Protein Name* can be potential information leakage for the protein-text retrieval task.

ProteinKG25 (Zhang et al., 2022). Dataset samples are shown in Table 13. ProteinKG25 is originally a knowledge graph derived from the Gene Ontology database (Aleksander et al., 2023). ProteinKG25 is stored as triples in the format of (protein sequence, relation, property), such as *(protein 1, enables, structural constituent of ribosome)*, *(protein 2, is involved in metabolic process, rRNA processing)*, and *(protein 3, is located in, cytoplasm)*. To synthetic text descriptions using such triples, we perform the following steps:

- Group triples by protein sequences. Each resulting group has triples of the same protein.
- Further group triples by relation types. Each resulting group has triples of the same protein and the same relation type. For example, a group can include triples of *(protein 1, is involved in lipid metabolic process, fatty acid biosynthetic process)*, *(protein 1, is involved in lipid metabolic process, lipid catabolic process)*, and *(protein 1, is involved in lipid metabolic process, fatty acid metabolic process)*.
- Transform the triples in each group into a sentence by filling the information into a text template. For example, the three triples in the previous step will be transformed into: *This protein is involved in the following lipid metabolic process: fatty acid biosynthetic process, lipid catabolic process, and fatty acid metabolic process.*
- Concatenate the sentences of the same protein together to form the final text description.

Note that, we do not include the protein sequence information in the text description to avoid information leakage. The Python script for data processing and the text templates will be made available online to facilitate future research.

PDB-QA (Guo et al., 2023). The categorization of all 30 question templates is detailed in Table 14. The original PDB-QA dataset provides only 3D structures of proteins. To retrieve their 1D sequences, we call the web API of the RCSB PDB website[5] using the proteins' PDB IDs.

B Error Analysis

Questions of Structures/Properties. Table 9 presents the statistics of sampled questions in this category. We can observe a strong correlation between accuracy and the number of candidate answers. This observation is intuitive, as questions become more challenging with an expanded answer pool. Further, questions that have a large number of candidate answers are those that require numerical outputs. This highlights the potential of improving the ability of counting substructures and processing numbers in future studies.

Questions of Supplementary Information. Table 10 shows the sampled questions of supplementary information and the candidate answers. We can observe that the questions and answers are not directly relevant to protein structures or functions. Therefore, we infer that the success of these questions often relies on exploiting the dataset's inherent biases rather than understanding protein structures and properties. This realization prompts a concern that the model's performance could significantly diminish when applied to datasets without similar biases, indicating a potential area for improvement using retrieval augmentation.

C Experimental Details

Following MolT5 (Edwards et al., 2022) and ProteinCLAP (Liu et al., 2023b), we use a single same random seed for experiments. Using multiple random seeds is cost-prohibitive for tuning large LMs.

The number of parameters of ProtT3 and key-baseline are summarized in Table 12. All the experiments are conducted on either two A100-40GB GPUs or four V100-32GB GPUs. We have used

[4] https://www.uniprot.org/

[5] https://www.rcsb.org/

Question	Accuracy	Number of candidate answers
Does this protein contain RNA polymer entities?	0.9871	2
Does this protein contain DNA polymer entities?	0.9790	2
How many nucleic acid polymer entities(DNA or RNA) does this protein have?	0.9644	7
Does this protein contain branched entities?	0.9509	2
What is the polymer entity type for this protein?	0.9192	4
How many model structures deposited for this protein?	0.9036	39
What are the software programs reported in connection with the production of this protein?	0.0733	3611
What is the molecular mass (KDa) of polymer and non-polymer entities (exclusive of solvent) for this protein?	0.0037	7367
How many heavy atom coordinates records does this protein have?	0.0010	6550

Table 9: Sampled questions of structures/properties and the corresponding accuracies and numbers of candidate answers.

Question	Candidate answers
How many model structures deposited for this protein?	{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ...}
What experimental method(s) were used to determine the structure of this protein?	{'NMR', 'X-ray', 'EM', 'Other', 'Multiple methods', 'Neutron'}
When is this protein first published?	{1975, 1976, 1977, 1979, 1980, 1981, 1982, 1983, 1984, 1985 ...}

Table 10: Sampled questions of supplementary information and the candidate answers.

Entry Name	Description
ENO1_CHLTE	FUNCTION: Catalyzes the reversible conversion of 2-phosphoglycerate into phosphoenolpyruvate. It is essential for the degradation of carbohydrates via glycolysis. SUBCELLULAR LOCATION: Cytoplasm Secreted Cell surface Note=Fractions of enolase are present in both the cytoplasm and on the cell surface. The export of enolase possibly depends on the covalent binding to the substrate; once secreted, it remains attached to the cell surface. SIMILARITY: Belongs to the enolase family.
6PGD_AGGAC	FUNCTION: Catalyzes the oxidative decarboxylation of 6-phosphogluconate to ribulose 5-phosphate and CO(2), with concomitant reduction of NADP to NADPH. SIMILARITY: Belongs to the 6-phosphogluconate dehydrogenase family.
RIMP_ACAM1	FUNCTION: Required for maturation of 30S ribosomal subunits. SUBCELLULAR LOCATION: Cytoplasm. SIMILARITY: Belongs to the RimP family.

Table 11: Examples of protein descriptions in the Swiss-Prot dataset. Entry name is the protein index in the original database.

FlashAttention-2 (Dao, 2023) to speedup the ESM-2 and Galactica model.

C.1 Experimental Details of ProtT3

Hyperparameters. The training stage 1 has 50 epochs and stage 2 has 10 epochs. The batch size is 128 for both stage 1 and stage 2. Q-Former has 8 query tokens ($N_q = 8$). The optimizer is configured following (Li et al., 2023a). We use an AdamW optimizer and a learning rate scheduler that is a combination of linear warmup and cosine decay. The peak learning rate is $1e-4$ and the warmup has 1000 steps. Weight-decay is set to 0.05. For all experiments that involve fine-tuning the $\text{Galactica}_{1.3\text{B}}$ LM, we set LoRA's rank r to 8 and apply LoRA to Galactica's weights of `[q_proj, v_proj, k_proj, out_proj, fc1, fc2]`. This configuration yields LoRA adapters with 7M parameters, constituting merely 0.54% of the parameters in the $\text{Galactica}_{1.3\text{B}}$. LoRA is implemented using the OpenDelta (Ding et al., 2022) library. For the text generation training, we truncate all the texts to at most 128 tokens.

Case Studies. The examples in Figure 4 are sampled from the predictions of models in Table 4a. The examples in Figure 5 are sampled from the predictions of models in Table 4b. Specifically, the Galactica model in Figure 4 and Figure 5 refers to

Retrieval Task		Generation Task	
Model	#Parameters	Model	#Parameters
ProtST	786M	ProteinChat	13.6B
ProteinCLAP	530M	$\text{Galactica}_{1.3\text{B}}$	1.3B
ProtT3 stage 1	326M	ProtT3 stage 2	1.6B

Table 12: Number of parameters of ProtT3 and key baselines.

the LoRA fine-tuned Galactica on the corresponding dataset.

C.2 Experimental Details of Baselines and ProtT3 Variants.

LoRA fine-tune $\text{Galactica}_{1.3\text{B}}$. We use LoRA fine-tune $\text{Galactica}_{1.3\text{B}}$ as a baseline to verify the effectiveness of using a PLM for protein understanding. The hyperparameters are the same as ProtT3: they use the same optimizer, learning rate, batch size, epochs, and LoRA configuration. This baseline is applied directly for training stage 2 without training stage 1 because of its incapability of protein-text retrieval.

ProtT3 w/ MLP Proj. This ProtT3 variant replaces ProtT3's Q-Former projector with a two-layer MLP following (Liu et al., 2023a). Hyperparameters are the same as ProtT3. This variant includes training stage 2 without stage 1, because the MLP projector cannot process multi-modal inputs, therefore cannot be applied in stage 1.

ProtT3 w/o Stage 1. This ProtT3 variant is to verify the effectiveness of training stage 1 for downstream generation tasks. Hyperparameters are the same as ProtT3.

ProtT3 w/o PTM. This is a variant of ProtT3's stage 1 model that removes the PTM objective function. Without the PTM objective, ProtT3 relies on retrieval ability of the PTC objective for protein-text retrieval.

ProtT3 w/o PCap. This is a variant of ProtT3's stage 1 model that removes the PCap objective function.

ProtST (Xu et al., 2023). We re-implement their method on our datasets, using their released codes and the original hyperparameters in their paper. Specifically, we have implemented their ProtST-ESM-2 model and their multi-modal pretraining stage. Following their original paper, we adopt $\text{ESM-2}_{650\text{M}}$ as the protein encoder and adopt $\text{PubMedBERT-abs}_{109\text{M}}$ (Gu et al., 2022) as the text encoder. The model is trained on the combination of the Swiss-Prot and ProteinKG25 datasets, which is the same as ProtT3's training stage 1. The pretrained model is then applied for protein-text retrieval directly.

ProteinCLAP (Liu et al., 2023b). We re-implement their method from scratch for their source codes are not publicly available. Our implementation largely follows the description in their paper: using SciBERT (Beltagy et al., 2019) for text encoding and ProtBERT (Elnaggar et al., 2021) for protein encoding, learning rate $1e-5$. We have trained the model on the combination of the Swiss-Prot and ProteinKG25 datasets for 50 epochs, which is the same as ProtT3's training stage 1. The pretrained model is then applied for protein-text retrieval.

ProteinChat. ProteinChat (Guo et al., 2023) is developed using the base LM of $\text{Vicuna}_{13\text{B}}$ (Zheng et al., 2023). We did not evaluate ProteinChat for the Swiss-Prot and ProteinKG25 datasets because these two datasets do not include proteins' 3D structures, which is required by ProteinChat's protein encoder. For the PDB-QA dataset, we evaluate a checkpoint shared by the authors, which is the checkpoint used in their original paper.

D Further Discussions

D.1 Intuitions on Q-Former's Capabilities

For clarity, we briefly summarize Q-Former's capability in the two stages:

Stage 1: Protein-Text Retrieval Training.

- **Protein-Text Retrieval with Protein-Text Contrasting (PTC) and Protein-Text Matching (PTM).** Both of these two pretraining tasks are designed for retrieval: PTC is faster for using cosine similarity to model sequence-level similarity between proteins and texts; PTM is slower but more accurate for using self-attention to model the token-level similarity. The other pretraining task, Protein Captioning, although cannot be applied for retrieval directly, improves Q-Former's representation learning ability, so as to improve the retrieval performance.

- **Warmup before stage 2.** The three pretraining

Protein ID	Description
B1VNN9	This protein is involved in the following process: nitrogen compound metabolic process. This protein is located in the following component: cytoplasm. This protein is involved in metabolic process: nitrogen compound metabolic process. This protein enables metal ion binding: nickel cation binding. This protein enables the following function: nickel cation binding.
Q01644	This protein is part of the following component: cellular component. This protein is involved in the following process: multicellular organism development, cell differentiation, sperm axoneme assembly, and spermatogenesis. This protein enables the following function: molecular function.
Q32AB9	This protein is involved in metabolic process: tricarboxylic acid cycle, carbon fixation, and oxaloacetate metabolic process. This protein enables metal ion binding: magnesium ion binding. This protein enables catalytic activity: hosphoenolpyruvate carboxylase activity, catalytic activity, and lyase activity. This protein enables the following functions: lyase activity, magnesium ion binding, catalytic activity, and phosphoenolpyruvate carboxylase activity.

Table 13: Examples of protein descriptions in the ProteinKG25 dataset. Protein ID is the protein index in the original dataset.

Question	Type	Question	Type
What is the nucleic acid polymer entity type for this protein?	String structure/property	How many heavy atom coordinates records does this protein have?	Number supplementary information
When is this protein first published?	Number supplementary information	How many intermolecular metalic bonds does this protein have?	Number structure/property
How many polymer monomers does this protein have?	Number structure/property	Does this protein have hybrid nucleic acid polymer entities?	String structure/property
How many assemblies does this protein have?	Number structure/property	What is the molecular mass (KDa) of polymer and non-polymer entities (exclusive of solvent) for this protein?	Number structure/property
How many heavy solvent atom coordinates records does this protein have?	Number supplementary information	Is this protein determined by experimental or computational methods?	String supplementary information
Does this protein have cis-peptide linkages?	String structure/property	What is the radiation wavelength in angstroms for this protein?	Number structure/property
Does this protein contain branched entities?	String structure/property	Does this protein have unmodeled polymer monomers?	String supplementary information
How many entities does this protein have?	Number structure/property	How many intermolecular covalent bonds does this protein have?	Number structure/property
Does this protein contain solvent entities?	String structure/property	How many model structures deposited for this protein?	Number supplementary information
What is the polymer entity type for this protein?	String structure/property	What are the software programs reported in connection with the production of this protein?	String supplementary information
How many nucleic acid polymer entities (DNA or RNA) does this protein have?	Number structure/property	How many hydrogen atom coordinates records does this protein have?	Number supplementary information
What is the polymer entity composition for this protein?	String structure/property	What experimental method(s) were used to determine the structure of this protein?	String supplementary information
Does this protein contain DNA polymer entities?	String structure/property	Does this protein contain polymer entities?	String structure/property
Does this protein contain non-polymer entities?	String structure/property	Does this protein contain RNA polymer entities?	String structure/property
What are the terms characterizing the protein?	String structure/property	What are the bound nonpolymer components for this protein	String structure/property

Table 14: Categorization of the 30 questions in the PDB-QA dataset (Guo et al., 2023).

tasks pretrain Q-Former to have multi-modal representation ability of texts and proteins, acting as a warmup training before stage 2.

Stage 2: Protein-to-Text Generation Training.

- **Nonlinear Mapping from Protein Space to Text Space.** Q-Former maps the protein encoder's output to the input space of an LM. This innovative approach allows the LM to leverage pre-existing knowledge from the protein encoder, facilitating a deeper understanding of protein structures and functions.

D.2 Discussion on Linear Mapping

Proteins and texts inherently exist within high-dimensional spaces, exhibiting intricate and non-linear semantic relationships. The utilization of a mere linear layer to map between protein and text domains often proves insufficient, potentially resulting in a loss of critical structural details of proteins or inaccuracies in text representatio. This claim is supported by the observation in Table 6, where a baseline using a linear projector significantly underpforms the others.

It is important to note, however, that applying a linear layer between the protein encoder and the LM does not necessarily produce a strictly linear mapping from proteins to texts. Should either the protein encoder or the LM undergo fine-tuning — as opposed to remaining frozen — their weights can adapt to facilitate a nonlinear mapping between protein and text. Thus, our critique is not directed at the use of a linear layer per se, but rather at its employment in conjunction with frozen protein encoders and LMs.

E Discussion on Licensing

Pretrained Models and Codes. The Galactica (Taylor et al., 2022) model is under the CC BY-NC 4.0 license, ESM-2 (Lin et al., 2022) under the MIT License, and Sci-BERT (Beltagy et al., 2019) under the Apache License. Collectively, these licenses permit non-commercial use and redistribution of the models.

Datasets. The Swiss-Prot (Bairoch and Apweiler, 2000) dataset is distributed under the CC BY 4.0 License. ProteinKG25 (Zhang et al., 2022) dataset is released under the MIT License, its original source GeneOntology (Aleksander et al., 2023) is under the CC BY 4.0 License. The PDB-QA (Guo et al., 2023) dataset is released under the BSD 3-Clause License, its original source RCSB-PDB is under the CC0 1.0 Universal License. These licenses collectively allow the use and redistribution of the datasets.

Our Licensing Approach. In line with the licensing terms of the codes, models, and datasets we utilized, we will release our pretrained models and codes under the CC BY-NC 4.0 License. For our datasets, we will apply the CC BY 4.0 License.