ProLLaMA: A Protein Large Language Model for Multi-Task Protein Language Processing

Liuzhenghao Lv [*1] **Zongying Lin** [*1] **Hao Li** [1 2] **Yuyang Liu** [1] **Jiaxi Cui** [1] **Calvin Yu-Chian Chen** [1] **Li Yuan** [1 2] **Yonghong Tian** [1 2]

Abstract

Large Language Models (LLMs), including GPT-x and LLaMA2, have achieved remarkable performance in multiple Natural Language Processing (NLP) tasks. Under the premise that protein sequences constitute the protein language, Protein Large Language Models (ProLLMs) trained on protein corpora excel at de novo protein sequence generation. However, as of now, unlike LLMs in NLP, no ProLLM is capable of multiple tasks in the Protein Language Processing (PLP) field. This prompts us to delineate the inherent limitations in current ProLLMs: (i) the lack of natural language capabilities, (ii) insufficient instruction understanding, and (iii) high training resource demands. To address these challenges, we introduce a training framework to transform any general LLM into a ProLLM capable of handling multiple PLP tasks. Specifically, our framework utilizes low-rank adaptation and employs a two-stage training approach, and it is distinguished by its universality, low overhead, and scalability. Through training under this framework, we propose the ProLLaMA model, the first known ProLLM to handle multiple PLP tasks simultaneously. Experiments show that ProLLaMA achieves state-of-the-art results in the unconditional protein sequence generation task. In the controllable protein sequence generation task, ProLLaMA can design novel proteins with desired functionalities. In the protein property prediction task, ProLLaMA achieves nearly 100% accuracy across many categories. The latter two tasks are beyond the reach of other ProLLMs. Code is available at https://github.com/Lyu6PosHao/ProLLaMA.

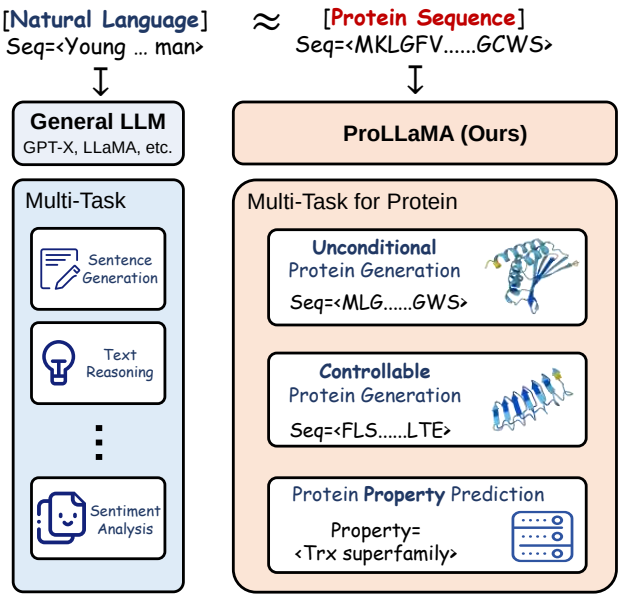


Figure 1. Analogous to general LLMs managing multiple tasks in the NLP field, our ProLLaMA can handle multiple tasks in Protein Language Processing. In contrast, other ProLLMs are limited to the single task of unconditional protein generation.

1. Introduction

Large Language Models (LLMs), like GPT-x and LLaMA2 (Bubeck et al., 2023; Touvron et al., 2023), have achieved outstanding performance in handling a wide range of Natural Language Processing (NLP) tasks (Tamkin et al., 2021; Zhao et al., 2023; Kocoń et al., 2023; Qin et al., 2023; Huang et al., 2023; Zhong et al., 2023; Bang et al., 2023), including both Natural Language Generation (NLG) and Natural Language Understanding (NLU) tasks. This surge in LLMs has extended their applications beyond traditional contexts, including their adoption in the challenging field of protein design (Notin et al., 2022; Strokach & Kim, 2022; Ferruz & Höcker, 2022).

Taking protein sequences as the protein language, researchers train models with architectures similar to current LLMs on a vast protein corpus (Strokach & Kim, 2022; Ferruz et al., 2022; Nijkamp et al., 2023; Moffat et al., 2022; Madani et al., 2023; Qin et al., 2023). These Pro-

[*]Equal contribution [1]School of Electronic and Computer Engineering, Peking University [2]Peng Cheng Laboratory. Correspondence to: Li Yuan <yuanli-ece@pku.edu.cn>, Yonghong Tian <yhtian@pku.edu.cn>.

tein LLMs (ProLLMs) have paved the way for the rapid generation of structurally plausible protein sequences, holding immense potential for biomedical and biotechnological innovations (Notin et al., 2022; Nijkamp et al., 2023). However, this progress is met with challenges, particularly in extending their capabilities beyond sequence generation.

Analogous to NLP, tasks related to protein language can be viewed as Protein Language Processing (PLP) (Bepler & Berger, 2021; Ofer et al., 2021). It is evident that current ProLLMs primarily focus on a single task in PLP, not covering multiple tasks like LLMs do in NLP (Ferruz et al., 2022; Madani et al., 2023). These limitations prompt the need for innovative solutions to unleash the full potential of ProLLMs. Developing a multi-tasking ProLLM would be highly beneficial for protein engineering and understanding the protein fitness landscape (Wright et al., 1932; Pan & Kortemme, 2021; Ren et al., 2022; Song & Li, 2023), but three main challenges must be considered:
(i) Necessity of Natural Language: Protein language is not fully sufficient for PLP tasks, meaning it cannot fully represent all components of a task (the user instruction and the expected output) (Xu et al., 2023; Wang et al., 2023). The instruction and the output require a language beyond protein language (typically, natural language) for representation, which current ProLLMs lack.
(ii) Instruction Following: To possess multi-tasking capabilities, models must execute tasks following user instructions (Zeng et al., 2023; Liu et al., 2023b; Zhou et al., 2023). However, current ProLLMs are unable to follow instructions.
(iii) Training Resource Consumption: Substantial training resources are needed for models to learn natural language, protein language, and user instructions (Cui et al., 2023), which can sometimes be unaffordable.

To address the challenges, we propose a two-stage training framework to achieve a ProLLM for multi-task PLP. In the first stage, we leverage a pre-trained general LLM like LLaMA2 to continually learn the protein language while maintaining the natural language knowledge of the model. In the second stage, the model is further trained on a multi-task PLP dataset through instruction tuning to equip the model with instruction following capabilities and multi-task capabilities of PLP. Notably, during both stages, we adopt Low-Rank Adaptation (LoRA) (Hu et al., 2021), which prevents catastrophic forgetting in the first stage, enhances scalability in the second stage and reduces the training cost.

Using the training framework, we develop ProLLaMA, a model capable of multi-tasks for PLP, distinguishing it from all other ProLLMs. Through a series of experiments, we demonstrate the multitasking capabilities of our ProLLaMA. Specifically, in unconditional protein generation, ProLLaMA outperforms current ProLLMs and other sequence-generative models on common metrics such as pLDDT and TM-score. In controllable protein generation, based on a user-provided textual description of the desired protein functionalities, ProLLaMA generates novel proteins from scratch with functionalities as competent as natural proteins, such as SAM-MT and Trx. For protein property prediction, ProLLaMA achieves an average accuracy of 72% on the test dataset and nearly 100% accuracy in many subcategories. In summary, the contributions of our research are as follows:

- We propose a training framework with scalability and efficiency that enables any general LLM to be trained as a proficient model for multiple tasks in Protein Language Processing.

- Our ProLLaMA is the first model to our knowledge capable of simultaneously handling multiple PLP tasks.

- Experiments show that our ProLLaMA not only handles PLP tasks beyond the reach of existing ProLLMs but also achieves state-of-the-art results in the protein generation task where current ProLLMs are active.

2. Preliminaries

2.1. Necessity of Natural Language

As aforementioned, given the similarities between protein sequences and natural language, tasks related to protein sequences can be considered as Protein Language Processing (PLP), analogous to NLP. However, we observe a fundamental difference between protein language and natural language: natural language is complete for NLP tasks, whereas protein language is not complete for PLP tasks. Specifically, natural language can represent all components of a task (i.e., the input instructions and the expected output) (Liu et al., 2023a). For example, in a sentiment analysis task, the instruction and output can be expressed as “Analyze the sentiment contained in this sentence: I am happy” and “The sentiment is positive” respectively, all in natural language. However, for PLP tasks, the instructions and outputs cannot be fully represented in protein language. For instance, in the protein property prediction task, the task instruction could be ”Predict the property of this protein: MAFCF...FEV”, with the expected output being ”The property is Trx superfamily.” It is evident that both the instructions and outputs of tasks require assistance from a language beyond protein language, in this case, natural language, for representation.

Therefore, multi-task ProLLMs must possess a certain level of natural language ability, especially as more textual descriptions of proteins become available (Xu et al., 2023).

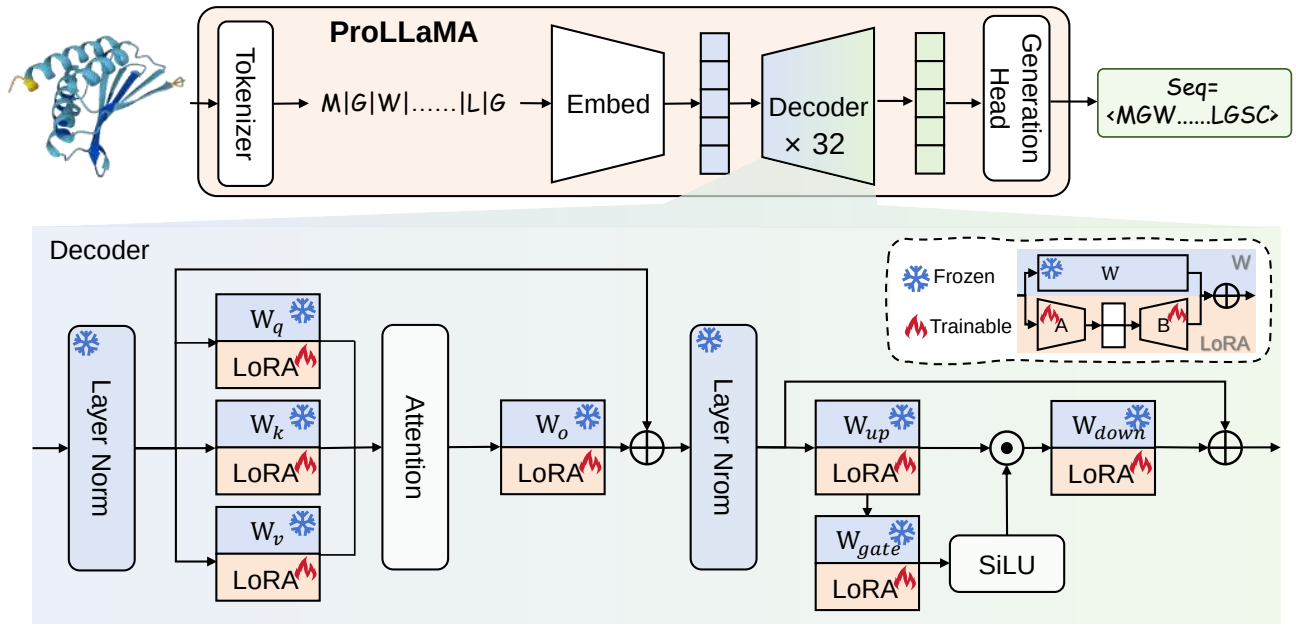


Figure 2. **The overall architecture of our ProLLaMA.** We add low-rank adapters (LoRA) to certain weights. During the training process, we freeze these weights and other parameters, focusing solely on training LoRA.

2.2. Causal Language Modeling

Causal Language Modeling (CLM) is the objective of training LLMs. Given a token sequence $\boldsymbol{x} = \{x_0, x_1, \ldots, x_{n-1}\}$ that is subject to training. CLM can be conceptualized as predicting the i-th token based on the preceding $i-1$ tokens (Bengio et al., 2000). Therefore, the optimization objective when training LLM using CLM is formulated as:

$$\mathcal{L}(\Theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[-\sum_i \log p(x_i|x_0, x_1, \ldots, x_{i-1}; \Theta)\right] \quad (1)$$

where $\mathcal{L}$ denotes the loss function, Θ denotes the trainable parameters of the model, and $\mathcal{D}$ is the training dataset.

2.3. Low Rank Adaptation

Low-Rank Adaptation (LoRA) (Hu et al., 2021) is a parameter-efficient technique for fine-tuning LLMs. Due to the immense parameter size of LLMs, full-parameter fine-tuning could be impractical sometimes. LoRA circumvents this by freezing the original parameters of LLMs and introducing additional trainable low-rank adapters. It achieves fine-tuning with a significantly smaller number of trainable parameters, yielding results comparable to full-parameter fine-tuning.

Theoretically, fine-tuning can be conceptualized as a process of finding the change in parameters (Ding et al., 2023). Let the original parameters of the model be denoted as W_0, and the parameters after fine-tuning as W. The objective of fine-tuning is to find $\Delta W = W - W_0$. Hypothesizing ΔW is of low rank (Aghajanyan et al., 2021), denoted as r, it can be decomposed into two low-rank matrices, $\Delta W = AB$. This leads to the following equation:

$$W = W_0 + AB \quad (2)$$

where $W, W_0 \in \mathbb{R}^{d\times h}$, $A \in \mathbb{R}^{d\times r}$, and $B \in \mathbb{R}^{r\times h}$. And the fine-tuning objective is transformed from finding ΔW to finding A and B:

$$\min_{\Delta W} \mathcal{L}(\Delta W) \rightarrow \min_{A,B} \mathcal{L}(A, B) \quad (3)$$

Consequently, the quantity of parameters involved in training is reduced from dh to $r(d+h)$. Given the low-rank hypothesis, where $r \ll d$ and $r \ll h$, this reduction in the number of trainable parameters is quite significant.

During training, the original parameters of the model are frozen, with only the low-rank adapters trainable. After training, the newly acquired knowledge is stored in the low-rank adapters, namely A and B. Additionally, A and B can be integrated into the original model using Equation 2. Although this integration means the low-rank adapters are no longer plug-and-play, it ensures that the architecture of the post-training model remains consistent with that of the original model. Finally, LoRA prevents catastrophic forgetting of the original knowledge, as the newly learned knowledge has a lower rank than the original knowledge.

3. Methods

The overview of the architecture of our ProLLaMA is shown in Figure 2, and the overview of our training framework is shown in Figure 3. In Section 3.1, we show how the

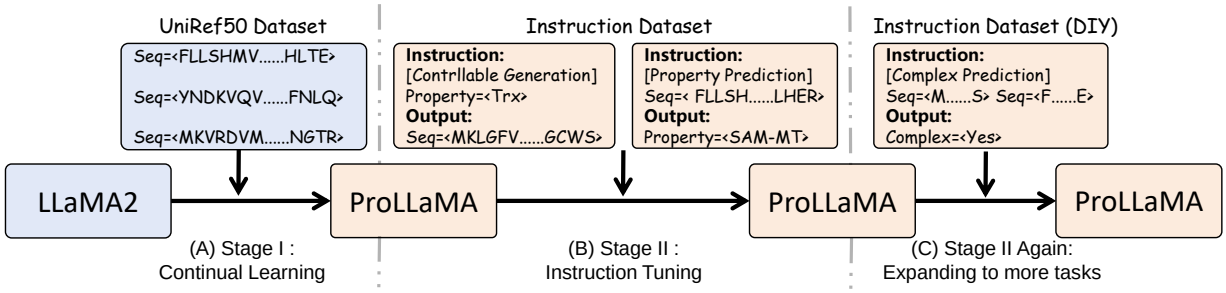


Figure 3. **The overall training framework of our ProLLaMA.** (A-B): Training is divided into two stages. (A): By continual learning on UniRef50, LLaMA2 learns protein language, resulting in ProLLaMA. (B) By instruction tuning on the instruction dataset, ProLLaMA is capable of multi-tasking. (C) By instruction tuning again on your own dataset, ProLLaMA can be expanded to more tasks.

LLaMA2 model learns protein language through continual learning, resulting in ProLLaMA. In Section 3.2, we show the integration of various tasks into ProLLaMA through instruction tuning. In Section 3.3, we show more tasks can be integrated into ProLLaMA. These three sections correspond to Figure 3(A-C).

3.1. Continual Learning on Protein Language

As mentioned in Section 1, current ProLLMs are lack of natural language abilities, which hinders multi-task capabilities. To solve this problem, we propose leveraging a pre-trained LLaMA2 to perform continual learning on protein language, as shown in Figure 3(A). This approach is analogous to humans learning a foreign language, where the model learns protein language while retaining its original natural language abilities.

Specifically, we construct a dataset based on UniRef50 (Suzek et al., 2015). We preprocess each protein sequence with specific prefixes and suffixes. This standardized format aids LLaMA2 in distinguishing the new protein language from its existing natural language knowledge, thereby reducing confusion.

We add low-rank adapters (LoRA) into LLaMA2. To be specific, in each Decoder layer of LLaMA2, we add LoRA to certain weights including W_q, W_k, W_v, W_o, W_{up}, W_{gate} and W_{down}. The original parameters of LLaMA2 are frozen, enabling only LoRA to be trained. Due to the significant differences between protein language and natural language, we choose a relatively high rank for LoRA, which helps the model learn protein sequences more effectively and prevents under-fitting. We include both the *Embed* and *Generation Head* layers in training. This is based on the premise that a token may have different meanings in protein sequences and natural languages, requiring distinct embeddings for the same token.

We train the model with Equation 1 as the target on the aforementioned dataset, resulting in ProLLaMA. Benefiting from LoRA, we train only 10% of the parameters, in contrast to full-parameter training, which significantly reduces training costs. Additionally, as the remaining parameters are not involved in training, the inherent natural language abilities of the model are preserved.

In summary, by enabling a pre-trained LLaMA2 to continually learn protein language and utilizing LoRA, we have developed ProLLaMA, a model that comprehends both protein sequence and natural language. Consequently, we have addressed the problem mentioned in the introduction: the lack of natural language abilities and excessive consumption of training resources.

3.2. Performing Various Tasks

As mentioned in Section 1, current ProLLMs are unable to perform multiple tasks based on user instructions. To solve this problem, we perform instruction tuning on the ProLLaMA obtained from the previous section, as shown in Figure 3(B).

To be specific, we construct a multi-task dataset, where each item represents a task sample. A task consists of two parts: the instruction and the output. Training conducted on this dataset can be denoted as follows:

$$\mathcal{L}(\Theta) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{u}\sim\mathcal{D}}\left[-\log\ p(\boldsymbol{x}|\boldsymbol{u};\Theta)\right] \tag{4}$$

Here, Θ denotes the parameters to be optimized, $\mathcal{L}$ the loss function, and $\mathcal{D}$ the dataset. $\boldsymbol{u}$ denotes the instruction of a task. $\boldsymbol{x} = \{x_0, x_1, \ldots, x_{n-1}\}$ denotes the output of a task, where x_i is the i-th token of the output.

Since ProLLaMA employs causal language modeling, we need to combine Equation 4 with Equation 1:

$$\mathcal{L}(\Theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[-\sum_i \log p(x_i|\boldsymbol{u}, x_0, x_1, \ldots, x_{i-1};\Theta)\right] \tag{5}$$

Equation 5 is the optimization objective for instruction tuning of ProLLaMA. $\boldsymbol{u}$ is not involved in the loss calculation, whereas $\boldsymbol{x}$ is. This is because the latter is the output part, where ensuring its quality of generation is crucial. The former, $\boldsymbol{u}$, as an instruction signal, only needs to be understood by the model and does not require generation. In the instruc-

tion tuning stage, we exclusively train LoRA at a lower rank than specified in Section 3.1.

In summary, through instruction tuning, we have made ProLLaMA capable of following instructions and performing multiple tasks. In contrast, other ProLLMs remain limited to the single task of protein sequence generation. Consequently, we have addressed the problems mentioned in the introduction: the lack of instruction following and the lack of multi-task capabilities.

3.3. Expanding to More Tasks

In this section, we demonstrate that benefiting from the scalability of the training framework, ProLLaMA can be easily extended to a broader range of tasks as shown in Figure 3(C).

To be specific, to adapt ProLLaMA to more tasks, researchers must customize the instruction dataset and then perform instruction tuning on ProLLaMA again with this dataset. ProLLaMA can be either the one obtained in the first stage or the one obtained in the second stage. Benefiting from LoRA, such instruction tuning requires only minimal training resources. An instance is shown in Figure 3(C). The new task in this instance is protein complex prediction, where the model must ascertain whether two input proteins can form a complex. In this task, the instruction $\boldsymbol{u}$ comprises a sentence describing the user intent and two protein sequences, while the output $\boldsymbol{x}$ should be a judgment of Yes or No. In summary, our framework and model are flexible and easily extensible. With collaborative efforts from other researchers, ProLLaMA has the potential to become more powerful.

4. Experiments

We introduce the experiment setup in Section 4.1. And we evaluate the unconditional protein generation task in Section 4.2, the controllable protein generation task in Section 4.3, the protein property prediction task in Section 4.4, and the natural language ability in Section 4.5.

4.1. Experiment Setup

Training Settings: For continual learning, the LoRA rank is set to 128, employing the AdamW optimizer alongside a cosine annealing scheduler with warm-up. The peak learning rate stands at 0.05, with a total of one training epoch. For instruction tuning, the LoRA rank is set to 64 with two training epochs, and all other settings remain consistent with the continual learning setup.

Training Datasets: For continual learning, the dataset used consists of protein sequences from UniRef50 (Suzek et al., 2015). For instruction tuning, the instruction dataset is constructed using protein sequences from UniRef50 and protein property texts from InterPro (Paysan-Lafosse et al., 2023).

Evaluation Settings: Unconditional protein generation involves generating protein sequences without specific instructions. Controllable protein generation involves generating desired protein sequences based on instructions that specify the required functionalities. Property prediction involves predicting protein properties based on instructions, which include the protein sequences to be predicted.

Evaluation Metrics: We use the following metrics to evaluate the generated protein sequences. The pLDDT (Jumper et al., 2021) is used to measure whether sequences are structurally plausible. Self-Consistency Perplexity (SC-Perp) (Alamdari et al., 2023) serves as an additional metric of plausible structures since pLDDT falls short in dealing with intrinsically disordered regions (IDRs) (Davey, 2019). TM-score (Zhang & Skolnick, 2004) reflects the structural similarity between generated sequences and known ones in AFDB (Varadi et al., 2022) and PDB (Berman et al., 2002). RMSD also reflects the structural similarity from the perspective of atomic distance. Homologous probability (H-Prob) reflects the probability that the generated protein is homologous to a known one. Seq-Ident reflects the sequence similarity between generated sequences and known ones.

4.2. Unconditional Protein Generation

We compare our model with other state-of-the-art models in the protein sequence generation field. These models encompass a wide range of architectures and are trained exclusively on protein sequences. Table 3.2 shows the results. Our ProLLaMA is optimal on pLDDT, TM-score, and RMSD and is suboptimal on SC-Perp. This indicates that ProLLaMA, through its training on protein sequence data, can generate structurally plausible proteins. Additionally, it is important to note that ProLLaMA-generated proteins exhibit a mean and standard deviation for pLDDT and SC-Perp of 66.49±12.61 and 3.10±0.65, respectively. These values are comparable to those of natural proteins as reported by Alamdari et al. (2023), which are 68.25±17.85 and 3.09±0.63, respectively.

De novo design of long and structurally plausible protein sequences is highly challenging (Ferruz et al., 2022), yet our ProLLaMA excels in meeting this challenge. Figure 4(A) shows that as the sequence length increases, the pLDDT of proteins generated by ProLLaMA does not decrease but rather increases. In contrast, the pLDDT of proteins generated by ESM2 significantly decreases as the length increases. This indicates that ProLLaMA is able to capture long-range dependencies between amino acids, while ESM2 faces difficulties in the de novo design of long protein sequences. Figure 4(B) and Figure 4(C) also support this point. Figure 4(B)

Table 1. **Comparison of proteins generated by different models.** Our ProLLaMA achieves the best performance on pLDDT, TM-score, and RMSD metrics, and is second-best in SC-Perp, demonstrating ProLLaMA excels in de novo protein design.

Architecture	Method	pLDDT↑	SC-Perp↓	AFDB		PDB	
				TM-score↑	RMSD↓	TM-score↑	RMSD↓
CNN	CARP (Alamdari et al., 2023)	34.40±14.43	4.05±0.52	0.28	19.38	0.38	8.95
	LRAR (Alamdari et al., 2023)	49.13±15.50	3.59±0.54	0.40	14.47	0.43	9.47
AutoEncoder	ESM-1b (Rives et al., 2021)	59.57±15.36	3.47±0.68	0.34	20.88	0.44	8.59
	ESM-2 (Lin et al., 2023)	51.16±15.52	3.58±0.69	0.20	35.70	0.41	9.57
Diffusion	EvoDiff (Alamdari et al., 2023)	44.29±14.51	3.71±0.52	0.32	21.02	0.41	10.11
LLM	ProtGPT2 (Ferruz et al., 2022)	56.32±16.05	3.27±0.59	0.44	12.60	0.43	9.19
	ProGen2 (Nijkamp et al., 2023)	61.07±18.45	**2.90±0.71**	0.43	15.52	0.44	11.02
	ProLLaMA (ours)	**66.49±12.61**	3.10±0.65	**0.49**	**9.50**	**0.48**	**7.63**

Table 2. **Controllable generation on four different instructions.** Given SAM-MT, TPHD, Trx, and CheY as instructions, ProLLaMA generates proteins with the desired functionalities. High values of TM-score and H-Prob indicate the generated proteins meet the instructions, in contrast to the uncontrollable generation of other models.

Method	SAM-MT		TPHD		Trx		CheY	
	TM-score↑	H-Prob↑	TM-score↑	H-Prob↑	TM-score↑	H-Prob↑	TM-score↑	H-Prob↑
ESM-1b	0.58	0.37	0.55	0.48	0.61	0.37	0.63	0.27
ESM-2	0.52	0.26	0.51	0.25	0.53	0.30	0.57	0.18
EvoDiff	0.46	1.17	0.42	1.80	0.42	1.10	0.46	1.43
ProtGPT2	0.45	3.86	0.43	4.62	0.44	2.53	0.45	4.86
ProGen2	0.44	1.90	0.45	2.49	0.43	2.44	0.44	2.13
ProLLaMA (ours)	**0.71**	**98.13**	**0.82**	**100.00**	**0.93**	**99.96**	**0.81**	**100.00**

shows that as the length increases, SC-Perp of ProLLaMA slightly increases and then continues to decrease, while SC-Perp of ESM2 does the opposite. Figure 4(C) shows that as the length increases, the TM-score of ESM2 gradually decreases from above 0.6 to less than 0.2. In contrast, the TM-score of ProLLaMA remains above 0.5 for nearly all lengths, even exceeding 0.8 when the length is greater than 350. These demonstrate the robust sequence generation capability of ProLLaMA, especially in generating longer sequences.

4.3. Controllable Protein Generation

To instruct ProLLaMA in controllable protein generation, we utilize four superfamily descriptions as instructions respectively: the S-adenosyl-L-methionine-dependent methyltransferase superfamily (SAM-MT), the Tetratricopeptide-like helical domain superfamily (TPHD), the Thioredoxin-like superfamily (Trx), and the CheY-like superfamily (CheY). For each superfamily, ProLLaMA generates 100 protein sequences. Additionally, we use 100 natural proteins from each of the four superfamilies as benchmarks for comparison. We employ Foldseek to compare the generated proteins with the natural ones. The results shown in Table 2 demonstrate that ProLLaMA can generate desired protein sequences based on instructions that specify the required functionalities, confirming the capability for controllable generation. For SAM-MT, the TM-scores of our generated sequences exceed 0.7; for TPHD and CheY, they are over 0.8; and for Trx, they surpass 0.9. The high TM-score indicates that the structures of the generated proteins closely resemble those of natural proteins in the same superfamily, implying functional similarity. For SAM-MT, TPHD, Trx, and CheY, all of the H-prob values are close to or even equal to 100%, indicating that the generated proteins are homologous to natural proteins and belong to the same superfamily. In summary, these provide strong evidence that the protein generation of ProLLaMA is controllable under instructions. In contrast, other models exhibit low TM-score and very low H-Prob due to their uncontrollable generation.

Additionally, using natural proteins as a benchmark, we assess pLDDT, SC-Perp, and TM-score of proteins generated by ProLLaMA. Figure 4(D) shows that for CheY, TPHD,

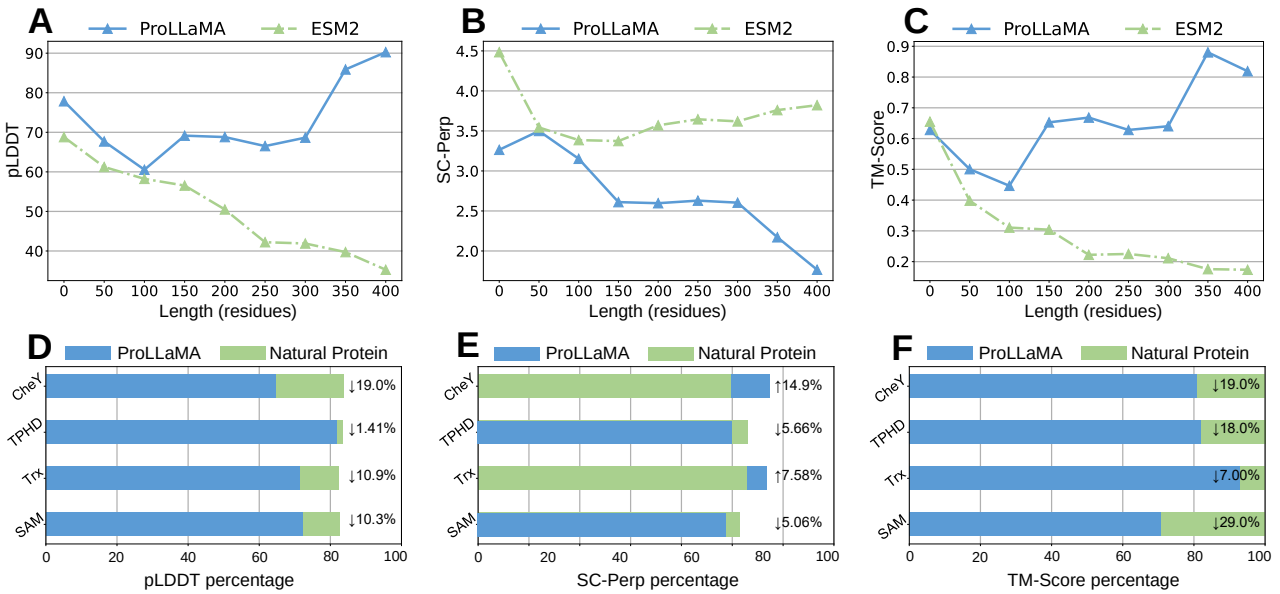


Figure 4. **Contrast experiment of ProLLaMA.** (A-C): Compared to the baseline (ESM2), ProLLaMA maintains a high, and even higher, quality of generated proteins as their length increases. (D-F): For the four superfamilies of CheY, TPHD, Trx, and SAM-MT (SAM), the proteins generated by ProLLaMA are comparable to natural proteins.

Table 3. **Protein property prediction.** Our ProLLaMA achieves nearly 100% accuracy for many superfamilies. Here are 10 of them.

	OBFD	UPF0145	NACD	U3S	CCHC	Kazal	SAM-MT	TPHD	Trx	CheY
Accuracy	100.00%	100.00%	100.00%	100.00%	95.24%	100.00%	93.67%	90.84%	94.17%	100.00%

Trx, and SAM-MT, the average pLDDT of generated proteins is only 19.0%, 1.41%, 10.9%, and 10.3% lower than that of natural proteins, respectively. Figure 4(E) shows that for TPHD and SAM-MT, the average SC-Perp is 5.66% and 5.06% lower; for Trx and CheY, the average SC-Perp is 7.58% and 14.92% higher. Figure 4(F) visualizes the TM-score, with scores near the maximum indicating a high degree of structural similarity. These findings indicate that proteins generated by ProLLaMA are comparable to their natural counterparts in the same superfamily. Given that averages are considered, there are instances where generated proteins outperform natural proteins.

In Figure 5, we present actual examples of proteins generated by ProLLaMA (depicted in blue) alongside the most structurally similar natural proteins from PDB (depicted in yellow). The four proteins are generated under the instructions of SAM-MT, TPHD, Trx, and CheY. We find that the four natural proteins also belong to SAM-MT, TPHD, Trx, and CheY, respectively, according to information from InterPro. This implies functional similarity between the generated proteins and natural ones, which further validates the effectiveness of controllable generation. The significant overlap in the 3D structures and the high TM-score confirm structural similarity. The lower Seq-Ident scores indicate sequence diversity. In summary, through controllable protein generation, ProLLaMA is capable of generating desired proteins with functions and structures similar to natural proteins, yet with novel sequences.

4.4. Property Prediction

We use the test dataset to evaluate whether ProLLaMA can predict the superfamily to which a given protein belongs. The test dataset consists of 10,000 samples. One protein may belong to two or more superfamilies simultaneously. Therefore, for the i-th protein in the test dataset, we denote its true superfamily text description as the set F_i and the prediction by ProLLaMA as the set F_i'. We calculate the prediction accuracy using:

$$\text{Acc} = \frac{\sum_{i=1}^{N} |F_i \cap F_i'|}{\sum_{i=1}^{N} |F_i'|} \tag{6}$$

where $|X|$ denotes the number of elements in set X, $\cap$ denotes the intersection of two sets, and N denotes the number of samples in the test dataset.

Although ProLLaMA actually performs a classification task here, it is more complex than typical ones. The key difference is that typical classification tasks require models to output a fixed label, often in one-hot encoding. In contrast, ProLLaMA outputs a full textual description of the result,

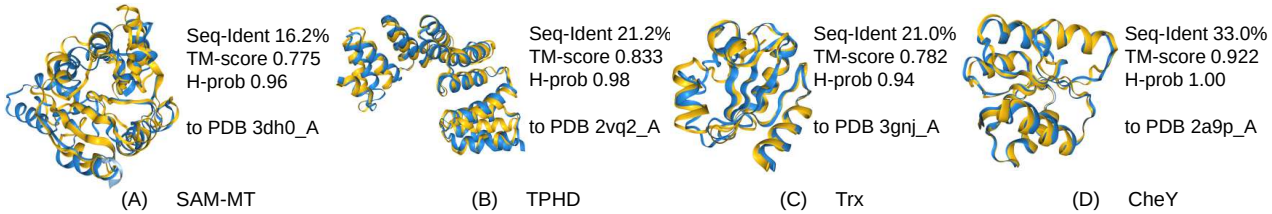


Figure 5. **Protein visualization.** Cases of controllable generation by ProLLaMA using SAM-MT, TPHD, Trx, and CheY as instructions. Blue is generated proteins, and yellow is natural. They are similar in structure and function but different in sequence.

Table 4. **Comparison of the natural language ability.** Vocab represents whether the vocabulary of the model supports natural language. Generation represents the sentence generation.

Type	Model	Vocab	Generation	QA
GeneralLLM	LLaMA2	✓	45%	44%
ProLLM	ProGen2	✗	-	-
	ProtGPT2	✓	0%	-
	ProLLaMA	✓	**26%**	**33%**

significantly increasing the challenge. Even so, ProLLaMA still achieves an average accuracy of 72% on the test dataset. In addition, ProLLaMA achieves considerably high accuracy in many specific superfamily categories. Table 3 lists ten of them, showing that the accuracy exceeds 90% in all the ten superfamilies and reaches 100% in OBFD, UPF0145, NACD, U3S, Kazal, and CheY.

4.5. Natural Language Ability

We evaluate the natural language ability of ProLLaMA and other models. As shown in Table 4, the vocabulary of ProGen2 only includes uppercase letters representing amino acids, failing to cover all English words and phrases. As for the sentence generation task on Wikipedia text, the accuracy of ProtGPT2 is 0%, indicating that it lacks natural language abilities. The accuracy of ProLLaMA is 26%, compared to 45% of LLaMA2. As for the QA task, the accuracy of ProLLaMA is 33%, compared to 44% of LLaMA2. These suggest that ProLLaMA has natural language abilities, albeit at a reduced level compared to LLaMA2.

5. Related Work

Protein Language Models

ProLLMs aforementioned are a subset of Protein Language Models (PLMs). Recognizing the similarity between natural language sequences and protein sequences, many algorithms from NLP have been applied to protein sequence data (Yang et al., 2018; Alley et al., 2019; Rao et al., 2021; Elnaggar et al., 2021). This has led to the development of PLMs, which are broadly categorized into two types (Ferruz & Höcker, 2022; Zheng et al., 2023).

One type adopts the decoder-only architecture and Causal Language Modeling (CLM) (Bengio et al., 2000; Vaswani et al., 2017), similar to general LLMs. Therefore, we refer to it as the Protein Large Language Model (ProLLM). ProLLMs primarily concentrate on de novo protein sequence generation (Moffat et al., 2022; Ferruz et al., 2022; Madani et al., 2023; Nijkamp et al., 2023), with a minority also focusing on fitness prediction (Notin et al., 2022). The other type adopts the encoder-only architecture and Masked Language Modeling (MLM) (Devlin et al., 2018; Meier et al., 2021; Rives et al., 2021; Brandes et al., 2022; Lin et al., 2023). They excel in protein representation learning, with the learned representations being applied to downstream tasks such as property prediction (Xu et al., 2023). However, they face challenges in de novo protein design.

Our ProLLaMA is capable of multitasking, excelling in tasks that both of the above types specialize in, surpassing existing ProLLMs and even PLMs.

Training LLMs

There is a well-established procedure for training LLMs (Zhao et al., 2023). Initially, models undergo pretraining on large-scale, unlabeled corpora to grasp grammatical and semantic principles (Min et al., 2023). Then, models are subjected to instruction tuning using an instruction dataset, enabling models to understand instructions and perform various tasks (Zhang et al., 2023). However, current ProLLMs cannot undergo instruction tuning due to the lack of natural language abilities.

Various parameter-efficient techniques have been proposed to accelerate training and conserve memory (He et al., 2021; Li & Liang, 2021; Hu et al., 2021; Liu et al., 2022; Hu et al., 2023), like Low-rank Adaptation (LoRA). We propose the use of LoRA in our training framework, which prevents catastrophic forgetting, enhances scalability, and reduces training costs. To our knowledge, this is the first application of parameter-efficient training techniques for ProLLMs.

6. Conclusion

Existing ProLLMs excel in the single task of protein generation but fall short in multiple tasks. In this work, we introduce an efficient training framework to transform any general LLM into a multi-task ProLLM. We also develop ProLLaMA, a versatile ProLLM for multiple tasks like con-

trollable protein generation and protein property prediction. Experiments indicate that ProLLaMA performs exceptionally well. We are confident that our framework and model will have a significant impact on the AI4Science community.

Impact Statements

Our work holds the potential to revolutionize the field of computational biology and biotechnology. With multi-task capabilities in the protein field, our ProLLaMA could significantly accelerate the pace of research and innovation in areas such as drug discovery, synthetic biology, and the development of novel biomaterials.

Moreover, our work can enable a wider range of researchers and institutions, particularly those with limited resources, to participate in cutting-edge research, fostering a more inclusive AI4Science community.

However, our work may raise concerns regarding model reliability and the potential for misuse. The uncertainty in model outputs could lead to misinformed decisions in critical research areas like drug discovery, necessitating cautious reliance and thorough validation by human experts to mitigate risks. Moreover, the powerful capabilities of ProLLaMA could be exploited for harmful purposes, posing bio-security challenges and ethical challenges.

A collaborative effort among researchers, policymakers, and the broader community is crucial to harness the benefits of our work while addressing the potential risks and ethical considerations.

References

Aghajanyan, A., Gupta, S., and Zettlemoyer, L. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 7319–7328, 2021.

Alamdari, S., Thakkar, N., van den Berg, R., Lu, A. X., Fusi, N., Amini, A. P., and Yang, K. K. Protein generation with evolutionary diffusion: sequence is all you need. *bioRxiv*, pp. 2023–09, 2023.

Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M., and Church, G. M. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, 16(12):1315–1322, 2019.

Bang, Y., Cahyawijaya, S., Lee, N., Dai, W., Su, D., Wilie, B., Lovenia, H., Ji, Z., Yu, T., Chung, W., et al. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*, 2023.

Bengio, Y., Ducharme, R., and Vincent, P. A neural probabilistic language model. *Advances in neural information processing systems*, 13, 2000.

Bepler, T. and Berger, B. Learning the protein language: Evolution, structure, and function. *Cell systems*, 12(6): 654–669, 2021.

Berman, H. M., Battistuz, T., Bhat, T. N., Bluhm, W. F., Bourne, P. E., Burkhardt, K., Feng, Z., Gilliland, G. L., Iype, L., Jain, S., et al. The protein data bank. *Acta Crystallographica Section D: Biological Crystallography*, 58(6):899–907, 2002.

Brandes, N., Ofer, D., Peleg, Y., Rappoport, N., and Linial, M. Proteinbert: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110, 2022.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Cui, Y., Yang, Z., and Yao, X. Efficient and effective text encoding for chinese llama and alpaca. *arXiv preprint arXiv:2304.08177*, 2023.

Davey, N. E. The functional importance of structure in unstructured protein regions. *Current opinion in structural biology*, 56:155–163, 2019.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., Hu, S., Chen, Y., Chan, C.-M., Chen, W., et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, 2023.

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2021.

Ferruz, N. and Höcker, B. Controllable protein design with language models. *Nature Machine Intelligence*, 4(6): 521–532, 2022.

Ferruz, N., Schmidt, S., and Höcker, B. Protgpt2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348, 2022.

He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., and Neubig, G. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*, 2021.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Hu, Z., Lan, Y., Wang, L., Xu, W., Lim, E.-P., Lee, R. K.-W., Bing, L., and Poria, S. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2304.01933*, 2023.

Huang, F., Kwak, H., and An, J. Is chatgpt better than human annotators? potential and limitations of chatgpt in explaining implicit hate speech. *arXiv preprint arXiv:2302.07736*, 2023.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Kocoń, J., Cichecki, I., Kaszyca, O., Kochanek, M., Szydło, D., Baran, J., Bielaniewicz, J., Gruza, M., Janz, A., Kanclerz, K., et al. Chatgpt: Jack of all trades, master of none. *Information Fusion*, pp. 101861, 2023.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637): 1123–1130, 2023.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023a.

Liu, X., Ji, K., Fu, Y., Tam, W., Du, Z., Yang, Z., and Tang, J. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 61–68, 2022.

Liu, Y., Tao, S., Zhao, X., Zhu, M., Ma, W., Zhu, J., Su, C., Hou, Y., Zhang, M., Zhang, M., et al. Automatic instruction optimization for open-source llm instruction tuning. *arXiv preprint arXiv:2311.13246*, 2023b.

Madani, A., Krause, B., Greene, E. R., Subramanian, S., Mohr, B. P., Holton, J. M., Olmos Jr, J. L., Xiong, C., Sun, Z. Z., Socher, R., et al. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, pp. 1–8, 2023.

Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., and Rives, A. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in Neural Information Processing Systems*, 34:29287–29303, 2021.

Min, B., Ross, H., Sulem, E., Veyseh, A. P. B., Nguyen, T. H., Sainz, O., Agirre, E., Heintz, I., and Roth, D. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, 56(2):1–40, 2023.

Moffat, L., Kandathil, S. M., and Jones, D. T. Design in the dark: learning deep generative models for de novo protein design. *bioRxiv*, pp. 2022–01, 2022.

Nijkamp, E., Ruffolo, J. A., Weinstein, E. N., Naik, N., and Madani, A. Progen2: exploring the boundaries of protein language models. *Cell Systems*, 14(11):968–978, 2023.

Notin, P., Dias, M., Frazer, J., Hurtado, J. M., Gomez, A. N., Marks, D., and Gal, Y. Tranception: Protein fitness prediction with autoregressive transformers and inference-time retrieval. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16990–17017. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/notin22a.html.

Ofer, D., Brandes, N., and Linial, M. The language of proteins: Nlp, machine learning & protein sequences. *Computational and Structural Biotechnology Journal*, 19: 1750–1758, 2021.

Pan, X. and Kortemme, T. Recent advances in de novo protein design: Principles, methods, and applications. *Journal of Biological Chemistry*, 296, 2021.

Paysan-Lafosse, T., Blum, M., Chuguransky, S., Grego, T., Pinto, B. L., Salazar, G. A., Bileschi, M. L., Bork, P., Bridge, A., Colwell, L., et al. Interpro in 2022. *Nucleic acids research*, 51(D1):D418–D427, 2023.

Qin, C., Zhang, A., Zhang, Z., Chen, J., Yasunaga, M., and Yang, D. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*, 2023.

Rao, R. M., Liu, J., Verkuil, R., Meier, J., Canny, J., Abbeel, P., Sercu, T., and Rives, A. Msa transformer. In *International Conference on Machine Learning*, pp. 8844–8856. PMLR, 2021.

Ren, Z., Li, J., Ding, F., Zhou, Y., Ma, J., and Peng, J. Proximal exploration for model-guided protein sequence design. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 18520–18536. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/ren22a.html.

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

Song, Z. and Li, L. Importance weighted expectation-maximization for protein sequence design. *arXiv preprint arXiv:2305.00386*, 2023.

Strokach, A. and Kim, P. M. Deep generative modeling for protein design. *Current opinion in structural biology*, 72: 226–236, 2022.

Suzek, B. E., Wang, Y., Huang, H., McGarvey, P. B., Wu, C. H., and Consortium, U. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015.

Tamkin, A., Brundage, M., Clark, J., and Ganguli, D. Understanding the capabilities, limitations, and societal impact of large language models. *arXiv preprint arXiv:2102.02503*, 2021.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Varadi, M., Anyango, S., Deshpande, M., Nair, S., Natassia, C., Yordanova, G., Yuan, D., Stroe, O., Wood, G., Laydon, A., et al. Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research*, 50(D1):D439–D444, 2022.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wang, Z., Zhang, Q., Ding, K., Qin, M., Zhuang, X., Li, X., and Chen, H. Instructprotein: Aligning human and protein language via knowledge instruction. *arXiv preprint arXiv:2310.03269*, 2023.

Wright, S. et al. The roles of mutation, inbreeding, crossbreeding, and selection in evolution. 1932.

Xu, M., Yuan, X., Miret, S., and Tang, J. Protst: Multi-modality learning of protein sequences and biomedical texts. *arXiv preprint arXiv:2301.12040*, 2023.

Yang, K. K., Wu, Z., Bedbrook, C. N., and Arnold, F. H. Learned protein embeddings for machine learning. *Bioinformatics*, 34(15):2642–2648, 2018.

Zeng, Z., Yu, J., Gao, T., Meng, Y., Goyal, T., and Chen, D. Evaluating large language models at evaluating instruction following. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*, 2023.

Zhang, S., Dong, L., Li, X., Zhang, S., Sun, X., Wang, S., Li, J., Hu, R., Zhang, T., Wu, F., et al. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*, 2023.

Zhang, Y. and Skolnick, J. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710, 2004.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

Zheng, Z., Deng, Y., Xue, D., Zhou, Y., Ye, F., and Gu, Q. Structure-informed language models are protein designers. *bioRxiv*, pp. 2023–02, 2023.

Zhong, Q., Ding, L., Liu, J., Du, B., and Tao, D. Can chatgpt understand too? a comparative study on chatgpt and fine-tuned bert. *arXiv preprint arXiv:2302.10198*, 2023.

Zhou, J., Lu, T., Mishra, S., Brahma, S., Basu, S., Luan, Y., Zhou, D., and Hou, L. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

A. Appendix

A.1. Detailed Training Settings

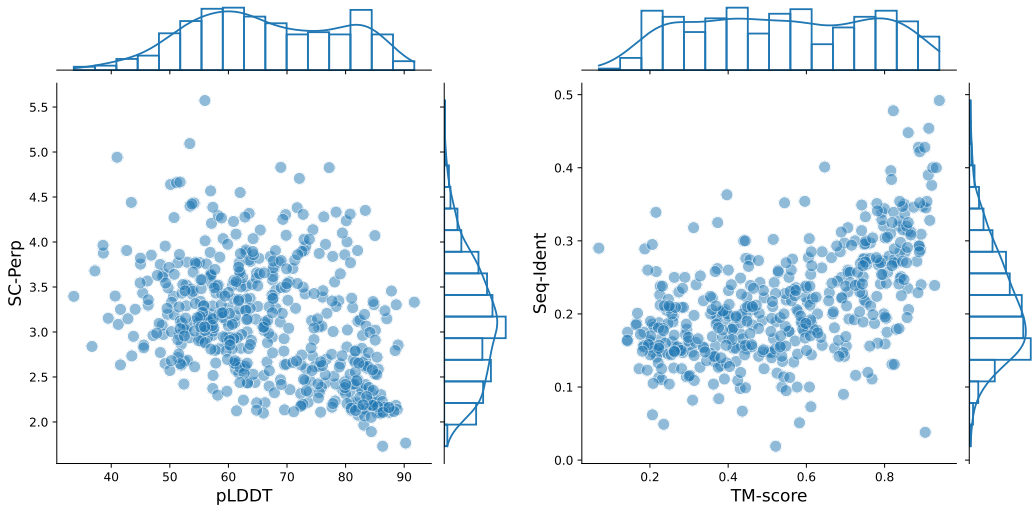
For continual learning, the block size is set to 2048, with a gradient accumulation step count of 8. The warm-up ratio is 0.05, and the weight decay is set to 0.01. The data type is bfloat16. The batch size per GPU is 4. DeepSpeed Zero Redundancy Optimizer stage 2 (ZeRO-2) is employed without using offload. For instruction tuning, the max sequence length is set to 256, with a gradient accumulation step count of 4. The warm-up ratio is 0.03. The batch size per GPU is 144. Other settings are the same as continual learning.

A.2. Detailed Evaluation Metrics

We use OmegaFold to calculate the values of the predicted Local Distance Difference Test (pLDDT) of protein sequences. Omegafold performs structure prediction without the need for homologous sequences or evolutionary information, relying solely on a single sequence for prediction. To calculate Self-Consistency Perplexity (SC-Perp), following the process mentioned in EvoDiff, we fold the sequence into a 3D structure using OmegaFold, then unfold it back into the sequence using ProteinMPNN. The self-consistency perplexity between the resulting sequence and the original sequence is referred to as SC-Perp.

We calculate TM-score, RMSD, H-Prob, and Seq-Ident using Foldseek. Foldseek facilitates the pairing of the queried protein p^{query} with structurally similar proteins from an existing protein database (AFDB or PDB), yielding pairs represented as $(p^{query},\ p^{target})$. Here, p^{target} denotes the protein in the database with a significant structural similarity to p^{query}. The magnitude of the average Template Modeling score (TM-score) value and Root-Mean-Square Deviation (RMSD) reflects the degree of structural similarity. TM-score takes into account the overall topological structure of proteins, focusing more on the overall structure. RMSD calculates the square root of the average position deviation of corresponding atoms between two protein structures, being highly sensitive to the size of the protein structure and local variations. Additionally, Foldseek also calculates the Sequence Identity (Seq-Ident) between p^{query} and p^{target}, reflecting their sequence-level similarity. Homologous probability (H-Prob) reflects the probability that p^{query} and p^{target} is homologous.

A.3. More Results



(a) (b)

Figure 6. (a) Distribution of pLDDT and SC-Perp. (b) Distribution of TM-score and Seq-Ident.

For the sequences generated without instructions by ProLLaMA, Figure 6(a) shows the distribution of pLDDT and SC-Perp, and Figure 6(b) shows the distribution of pLDDT and SC-Perp, where one spot denotes one sequence.