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

Liuzhenghao Lv, Zongying Lin, Hao Li, Yuyang Liu, Jiaxi Cui, Calvin Yu-Chian Chen, Li Yuan, Yonghong Tian, *Fellow, IEEE*

***Abstract*—Recent advances in Protein Language Models (PLMs) have transformed protein engineering, yet unlike their counterparts in Natural Language Processing (NLP), current PLMs exhibit a fundamental limitation: they excel in either Protein Language Understanding (PLU) or Protein Language Generation (PLG), but rarely both. This fragmentation hinders progress in protein engineering. To bridge this gap, we introduce ProLLaMA, a multitask protein language model enhanced by the Evolutionary Protein Generation Framework (EPGF). We construct a comprehensive instruction dataset containing approximately 13 million samples with over 11,000 superfamily annotations to facilitate better modeling of sequence-function landscapes. We leverage a two-stage training approach to develop ProLLaMA, a multitask LLM with protein domain expertise. Our EPGF addresses the mismatch between statistic language modeling and biological constraints through three innovations: a multi-dimensional interpretable scorer, hierarchical efficient decoding, and a probabilistic-biophysical joint selection mechanism. Extensive experiments demonstrate that ProLLaMA excels in both unconditional and controllable protein generation tasks, achieving superior structural quality metrics compared to existing PLMs. Additionally, ProLLaMA demonstrates strong understanding capabilities with a 67.1% exact match rate in superfamily prediction. EPGF significantly enhances the biological viability of generated sequences, as evidenced by improved biophysical scores (+4.3%) and structural metrics (+14.5%).**

***Impact Statement*—Protein engineering is a rapidly evolving field that involves the design and modification of proteins to achieve desired functions, playing a critical role in diverse applications from medicine to materials science. Advances in this area have traditionally relied on labor-intensive experimental methods. In this work, we demonstrate the potential of LLMs to advance protein engineering. Our model is capable of not only predicting protein properties, but also designing structurally plausible proteins with desired functions from scratch. Moreover, EPGF enhances protein generation by integrating biophysical constraints, improving the biological viability of designed proteins and expanding the practical applications of AI-driven protein engineering.**

***Index Terms*—Biotechnology, Large language models, Protein engineering**

This work was supported in part by the National Natural Science Foundation of China (No. 62202014, No. 62332002, No. 62425101, No. 62088102).

Liuzhenghao Lv, Zongying Lin, Hao Li, Yuyang Liu, Calvin Yu-Chian Chen, Li Yuan and Yonghong Tian are with the Shenzhen Graduate School, Peking University, Shenzhen, China. Corresponding author: Li Yuan (yuanli-ece@pku.edu.cn), Yonghong Tian (yhtian@pku.edu.cn).

Liuzhenghao Lv and Yonghong Tian are also with the School of Computer Science, Peking University, Beijing, China.

Hao Li, Li Yuan and Yonghong Tian are also with the Peng Cheng Laboratory, Shenzhen, China.

Jiaxi Cui is with the Pandalla.AI, Beijing, China.

The project is available at https://github.com/PKU-YuanGroup/ProLLaMA.

I. Introduction

Large Language Models (LLMs), such as GPT-4 and LLaMA-2 [1, 2], have achieved outstanding performance in handling a wide range of Natural Language Processing (NLP) tasks [3, 4, 5, 6, 7, 8], including both Natural Language Generation (NLG) and Natural Language Understanding (NLU) tasks, in a generative manner. This surge in LLMs has extended their applications beyond traditional contexts, including their adoption in the challenging field of protein engineering [9, 10, 11, 12, 13].

Recent advances in Protein Language Models (PLMs) have opened new possibilities for protein engineering, leveraging large-scale protein sequence corpora to learn meaningful representations. However, existing PLMs exhibit a fundamental limitation: they lack multitasking capabilities, with most models excelling in either Protein Language Understanding (PLU)[14, 15, 16, 17] or Protein Language Generation (PLG)[18, 19, 20], but rarely both. This stands in contrast to LLM in NLP. It creates a fragmented approach to protein engineering, where different tasks require different models.

Among these two tasks, PLG is generally more challenging than PLU. While understanding mainly requires extracting meaningful features from existing sequences, generation demands the production of novel, functional proteins that adhere to strict biophysical and evolutionary constraints [11, 16]. Current PLMs, despite their ability to generate statistically plausible sequences, often struggle to ensure biological viability, limiting their practical applications in protein engineering [18].

We hypothesize that this limitation arises from the inherent mismatch between NLP decoding strategies such as nuclear sampling and beam search [21], and the nature of biological sequences. Unlike natural language, where fluency and coherence are primary concerns, protein sequences must fold into stable three-dimensional structures and maintain biochemical functionality. NLP decoding strategy, widely adopted in PLMs, optimizes for sequence likelihood but lacks explicit biophysical constraints, leading to sequences that are statistically coherent yet functionally deficient [22].

To address these challenges, we introduce ProLLaMA, a multitask protein language model that bridges the gap between understanding and generation, enhanced by our Evolutionary Protein Generation Framework (EPGF). EPGF improves PLMs by incorporating explicit biophysical guidance during inference, ensuring that generated sequences adhere to essential biological constraints. Specifically:

We construct a large-scale instruction dataset that contains

approximately 13 million samples and encompasses both generation and understanding. We introduce a two-stage training framework to obtain ProLLaMA. In the first training stage, we leverage a pre-trained general LLM (LLaMA-2-7B) to continually learn the protein language while maintaining the natural language knowledge. In the second stage, the model is further trained on the aforementioned instruction dataset. During inference, we use EPGF, a test-time computing framework, to enhance ProLLaMA's generation capabilities through three key innovations: (1) a Multi-dimensional Biophysical Scorer that evaluates sequences based on compositional biophysics, physicochemical properties, and functional characteristics; (2) a Hierarchical Efficient Decoding strategy that processes sequences at the segment level, aligning with natural protein folding patterns; and (3) a Probabilistic-Biophysical Joint Selection mechanism that dynamically balances statistical likelihood with biological viability.

Through extensive experiments, we demonstrate the multitask capabilities of ProLLaMA and the effectiveness of EPGF. In unconditional protein generation, ProLLaMA outperforms current PLMs on common metrics such as pLDDT and TM-score. In controllable protein generation, ProLLaMA generates novel proteins from scratch with desired functionalities, such as the SAM-MT superfamily, based on user-provided textual descriptions. For protein superfamily prediction, ProLLaMA achieves a 67.1% exact match rate on the test dataset and obtains an F1-score above 0.9 in many specific categories. Furthermore, EPGF significantly enhances the biological viability of generated sequences, as evidenced by improved biophysical scores (+4.3%) and structural metrics (+14.5%).

In summary, the contributions of our research are as follows:

- We construct an instruction dataset that contains 13 million samples and more than 11,000 kinds of superfamily annotations, which facilitates better modeling of sequence-function landscapes and enables multitask learning in the protein domain.
- We propose ProLLaMA, a multitask protein language model that bridges the gap between protein generation and understanding.
- We propose the Evolutionary Protein Generation Framework (EPGF), which ensures that generated protein sequences are not only statistically coherent but also biologically viable, addressing a critical limitation in current PLMs.
- Through extensive experiments, we demonstrate that ProLLaMA, enhanced by EPGF, achieves state-of-the-art results in protein generation tasks while excelling in protein understanding tasks.

II. Methods

In Section. II-A and Fig. 2A, we show how to construct the protein language dataset and the instruction dataset. In Section. II-B and Fig. 2B., we show how to develop ProLLaMA using our training framework. In Section. II-C, Fig. 4 and Algorithm. 1, we demonstrate Evolutionary Protein Generation Framework (EPGF).

A. Dataset Construction

The protein language dataset is utilized in the first training stage to enable LLaMA-2-7B to grasp the language of proteins. Specifically, the dataset is sourced from UniRef50_2023_03 [23] on the UniProt website. We eliminate the descriptive parts of UniRef50, retaining only the pure protein sequences. Furthermore, We filter UniRef50 to ensure that the protein sequences consisted only of the 20 standard amino acids. We also retain sequences with a length of less than 512, aligning with ProGen [19]. Given that the lengths of protein sequences follow a long-tail distribution, the sequences that are deleted constitute only a small portion of the total dataset. To preprocess the protein sequences, we employ a specific prefix and a suffix. This standardized format aids LLaMA-2 in distinguishing the new protein language from its existing natural language knowledge, thus reducing confusion. The original Uniref50 contains 60,952,894 sequences, while after the above series of processing, our dataset comprises 52,807,283 protein sequences, with 90% for training and 10% reserved for testing.

The instruction dataset is utilized in the second training stage to enable ProLLaMA to perform various tasks. We first obtain the protein2ipr database from InterPro [24], which includes all proteins from UniProtKB along with their corresponding InterPro annotation information. Subsequently, we iterate through each protein's rep_id in UniRef50 to retrieve the corresponding annotation information from protein2ipr. This retrieval process is implemented using a distributed Redis database, and only proteins with lengths less than 256 participate in the retrieval to enhance efficiency. We utilize regular expressions to filter out superfamily annotation from the whole annotation. In the end, we obtain 6,350,106 data instances, each of which contains one protein sequence and its superfamily annotation. And the number of unique superfamily annotations is 11,268.

Then, we process the obtained data into a multi-task instruction dataset following the Alpaca format [25], where each instance comprises three parts: instruction, input, and output. The instruction specifies the task type. We design two tasks: generating proteins based on superfamily and determining the superfamily of the given protein. For the former task, the input is the superfamily annotation, and the output is the expected protein. The latter task is the opposite. In the end, the instruction dataset comprises 12,700,212 (6,350,106 * 2) instances, with 90% for training and the rest reserved for testing.

B. Training Framework for ProLLaMA

We propose a parameter-efficient training framework to transform general LLMs into PLMs capable of handling multiple tasks. Our framework leverages Low-Rank Adaptation (LoRA) [26], which injects protein-related knowledge into LLaMA-2 while preserving its natural language capabilities.

LoRA is a parameter-efficient technique that freezes the original LLM parameters and introduces trainable low-rank adapters. Theoretically, fine-tuning can be conceptualized as finding the parameter change $\Delta W = W - W_0$, where

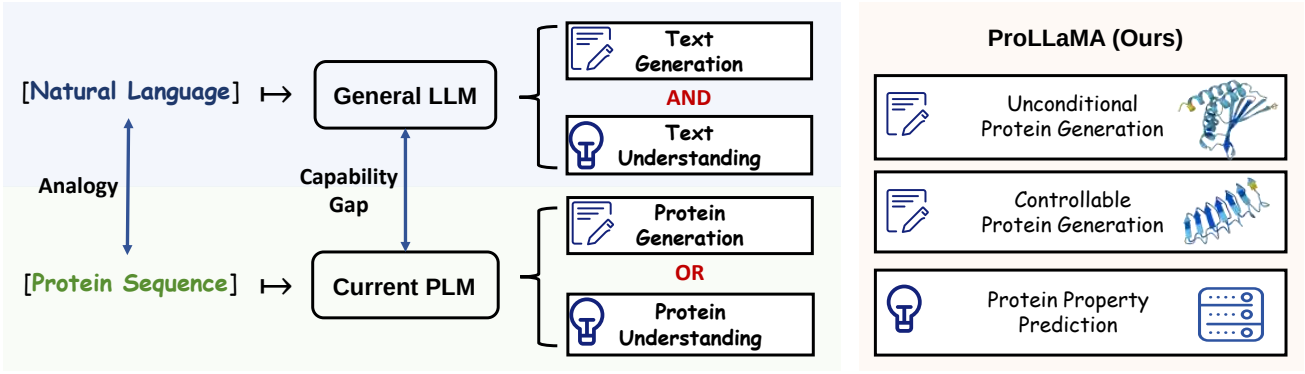

Fig. 1. **Left**: LLMs can handle both generation and understanding tasks, whereas PLMs cannot. This highlights the disparity in capabilities between the two. **Right**: Our ProLLaMA can handle generation tasks (unconditional protein generation, controllable protein generation) and understanding tasks (protein superfamily prediction), surpassing current PLMs.

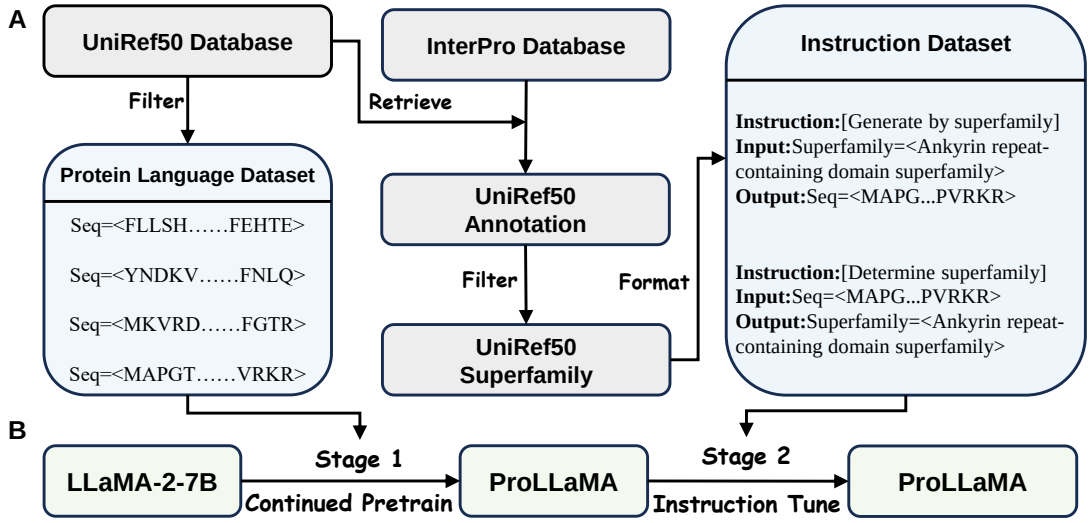

Fig. 2. (A) **Overview of the dataset construction.** The protein language dataset contains 53 million samples, which is used for training in Stage 1. The instruction dataset contains 13 million instances with 11,268 unique superfamily annotations, which is used for training in Stage 2. (B) **Overview of the training framework.** Stage 1: The pre-trained LLaMA-2 learns the protein language, resulting in ProLLaMA. Stage 2: ProLLaMA learns to perform multiple tasks by instruction tuning.

W_0 and W represent the original and fine-tuned parameters, respectively. Assuming ΔW has a low rank r [27], it can be decomposed as $\Delta W = AB$, yielding:

$$W = W_0 + AB \tag{1}$$

where $W, W_0 \in \mathbb{R}^{d\times h}$, $A \in \mathbb{R}^{d\times r}$, and $B \in \mathbb{R}^{r\times h}$. This reduces trainable parameters from dh to $r(d+h)$, with $r \ll d$ and $r \ll h$.

Our training framework consists of two key stages:

Stage 1: Learning Protein Language. We apply LoRA adapters to specific weights in each LLaMA-2 decoder block, including W_q, W_k, W_v, W_o, W_{up}, W_{gate}, and W_{down}. Due to significant differences between protein and natural languages, we use a relatively high rank for LoRA to prevent underfitting. We also train the embedding and output layers since tokens may have different meanings in protein sequences versus natural language. Through causal language modeling on protein sequences, we train only about 8% of the parameters, which substantially reduces computational costs while preserving natural language abilities.

Stage 2: Instruction Tuning. To enable multi-task capabilities, we perform instruction tuning on the model from Stage 1. The training objective is:

$$\mathcal{L}(\Theta) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{u}\sim\mathcal{D}}\left[-\sum_i \log p(x_i|\boldsymbol{u}, x_0, x_1, \ldots, x_{i-1}; \Theta)\right] \tag{2}$$

where $\boldsymbol{u}$ denotes the instruction (including input) and $\boldsymbol{x} = \{x_0, x_1, \ldots, x_{n-1}\}$ represents the expected output. During this stage, we exclusively train LoRA at a lower rank than in Stage 1.

Our framework is flexible and extensible, allowing ProLLaMA to be easily adapted to additional tasks. Researchers can customize instruction datasets and perform further instruction tuning on ProLLaMA with minimal training resources. In our experiments, we demonstrate this extensibility by successfully adapting ProLLaMA to protein solubility prediction through additional instruction tuning.

The framework enables ProLLaMA to understand both protein language and natural language, follow instructions, and

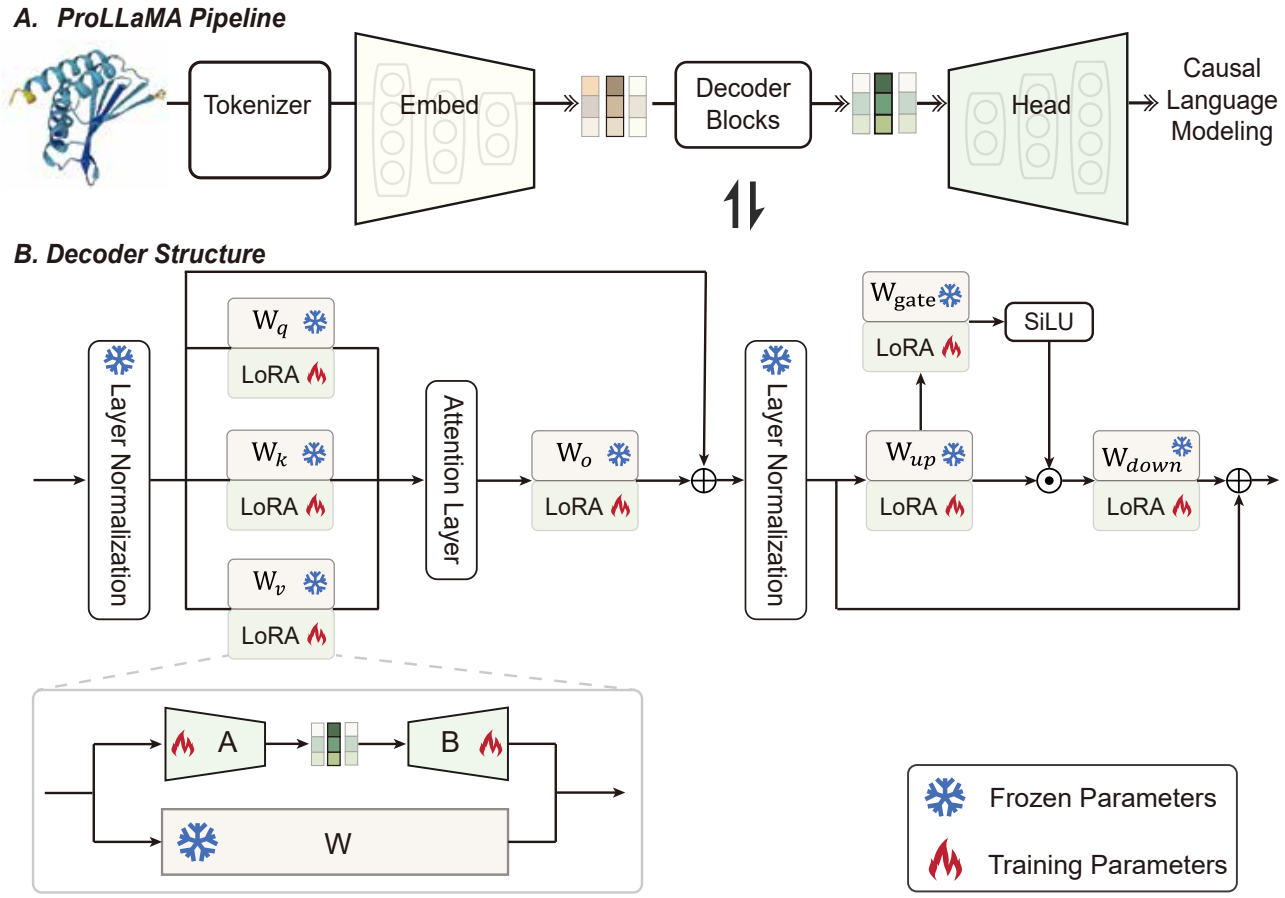

Fig. 3. **The overview of the ProLLaMA model.** We add LoRA adapters to certain weights. We freeze original parameters, focusing solely on training LoRA adapters (*Embed* and *Head* are also involved in the first training stage).

perform multiple protein-related tasks efficiently.

C. Evolutionary Protein Generation Framework

We propose the **Evolutionary Protein Generation Framework (EPGF)**, a novel inference framework designed to enhance the sequence generation capabilities of PLMs by explicitly incorporating biological constraints. EPGF bridges the gap between statistical language modeling and functional protein design through three key innovations:

- **Multi-dimensional Biophysical Scorer**: A biologically interpretable scoring system that evaluates protein sequences based on compositional biophysics, physicochemical properties, sequence complexity, and functional characteristics.
- **Hierarchical Efficient Decoding**: A structure-aware generation paradigm that processes sequences at the segment level (approximately 30 amino acids), aligning with natural protein folding patterns and significantly improving evaluation efficiency.
- **Probabilistic-Biophysical Joint Selection with Adaptive Diversity Control**: A unified mechanism that combines statistical likelihood with biological viability assessment, dynamically balancing exploration and exploitation during sequence generation through a simulated annealing-inspired approach.

We demonstrate the effectiveness of EPGF and its individual components through comprehensive experiments, including ablation studies, as presented in the experimental section and Supplementary Material.

1) Multi-dimensional Biophysical Scorer: Our proposed scorer aims to comprehensively evaluate protein sequences from the following four aspects and provides an overall score by weighting and averaging key metrics:

$$\mathcal{B}(S) = \frac{1}{n}\sum_{i=1}^{n} \text{Metric}_i(S), \tag{3}$$

where S represents a protein sequence, and Metric_i represents the i-th metric drawn from four major categories:

- **Compositional Biophysics**: Evaluates amino acid distribution, diversity, and prevalence of rare amino acids to ensure natural sequence composition. Research [28] shows proteins with abnormal amino acid distributions typically exhibit reduced stability and functionality. Rare amino acids like cysteine and tryptophan serve critical roles but their improper distribution can compromise structure.
- **Physicochemical Properties**: Assesses hydrophobicity, charge balance, and sequence stability, which are crucial for proper protein folding [29]. Hydrophobicity patterns drive tertiary structure formation with hydrophobic

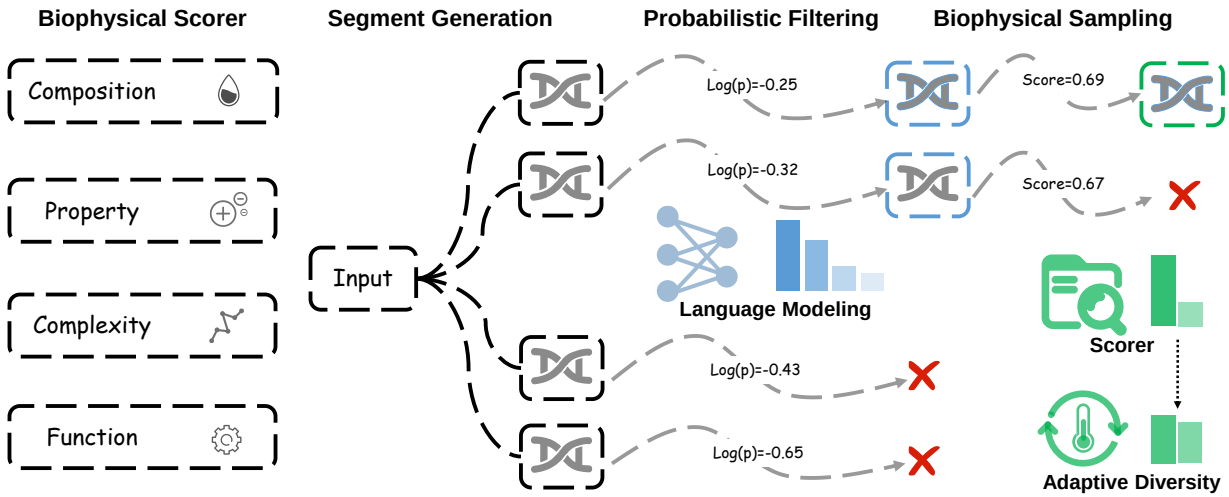

Fig. 4. **The overview of EPGF.** EPGF has three key components: (1) a multi-dimensional biophysical scorer; (2) a hierarchical efficient decoding strategy which generates protein candidates at segment-level; (3) probabilistic-biophysical joint selection with adaptive diversity control, which selects the superior candidate for the next round of generation.

Algorithm 1 Evolutionary Protein Generation Framework (EPGF)

Require: Language modeling distribution p, segment length L, diversity controller τ, biophysical scorer $\mathcal{B}$, number of candidates N, minimum acceptable score e

Ensure: Generated protein sequence S.

Initialize sequence S_0 with start token.
Initialize temperature τ_0 and decay rate γ.
while S not finished **do**
  Sample N candidate segments: $\{C_1, C_2, \ldots, C_N\} \sim p, |C_i| \leq N$
  for each candidate segment C_j **do**
    Log-probability $P(C_j) = \sum_{i=1}^{L} \log p(c_i \mid c_{<i})$.
  end for
  Retain top $K = \lceil N/2 \rceil$ candidates based on $P(C_j)$.
  for each retained candidate C_j **do**
    Compute biophysical score $\mathcal{B}(C_j)$.
    Ensure $\mathcal{B}(C_j) \geq e$
  end for
  Compute selection probability:

$$P_{\text{select}}(C_j) = \frac{\exp(\mathcal{B}(C_j)/\tau_t)}{\sum_{k=1}^{K} \exp(\mathcal{B}(C_k)/\tau_t)}.$$

  Sample candidate: $C^* \sim P_{\text{select}}$.
  Update Sequence: $S = S + C^*$.
  Update Diversity Controller: $\tau = \max(\tau_{\text{final}}, \tau \cdot \gamma)$.
end while
return Final sequence S.

residues clustering in the protein core. Charge distribution affects solubility and stability, with imbalanced charges linked to aggregation.

- **Sequence Complexity**: Measures sequence entropy, repetitive patterns, and local complexity to avoid unnatural homopolymers or overly repetitive sequences. Low-complexity regions rarely occur in functional globular proteins and often indicate intrinsically disordered regions or pathological aggregation [30]. Sequences with abnormally low complexity frequently fail to form stable structures.
- **Functional Characteristics**: Examines secondary structure propensities and functional motifs associated with specific protein superfamilies. Well-balanced secondary structure elements demonstrate better stability and functionality. Conserved functional motifs represent evolutionarily optimized patterns essential for specific biochemical functions and activity [31].

The specific calculation formula is shown in Supplementary Material. The proposed scorer enables quantitative evaluation of both complete and partial protein sequences, providing explicit insights into their biophysical properties during the design process. By assessing key metrics across multiple dimensions, it identifies potential issues such as unnatural amino acid distributions, improper physicochemical profiles, or structural anomalies, guiding the generation process of PLMs.

2) Hierarchical Efficient Decoding: Traditional methods generate sequences token-by-token, which is inherently inefficient during evaluation, as the computational cost scales with residue numbers. It also poses challenges for meaningful biological evaluation, as assessing individual tokens in isolation lacks biological context and significance. To address these limitations, we propose a segment-level decoding strategy within EPGF. By evaluating sequences at the segment level, we not only significantly improve computational efficiency but also ensure the evaluation aligns with biologically relevant units, such as structural motifs and functional domains.

Although sequence generation within each segment proceeds in a token-by-token manner, the principal innovation of our method resides in the evaluation paradigm. Instead of assessing individual tokens, we evaluate contiguous sequence fragments of length $L = 20$ tokens (approximately 30 amino acids), which aligns with the characteristic size of typical protein structural motifs. The candi-

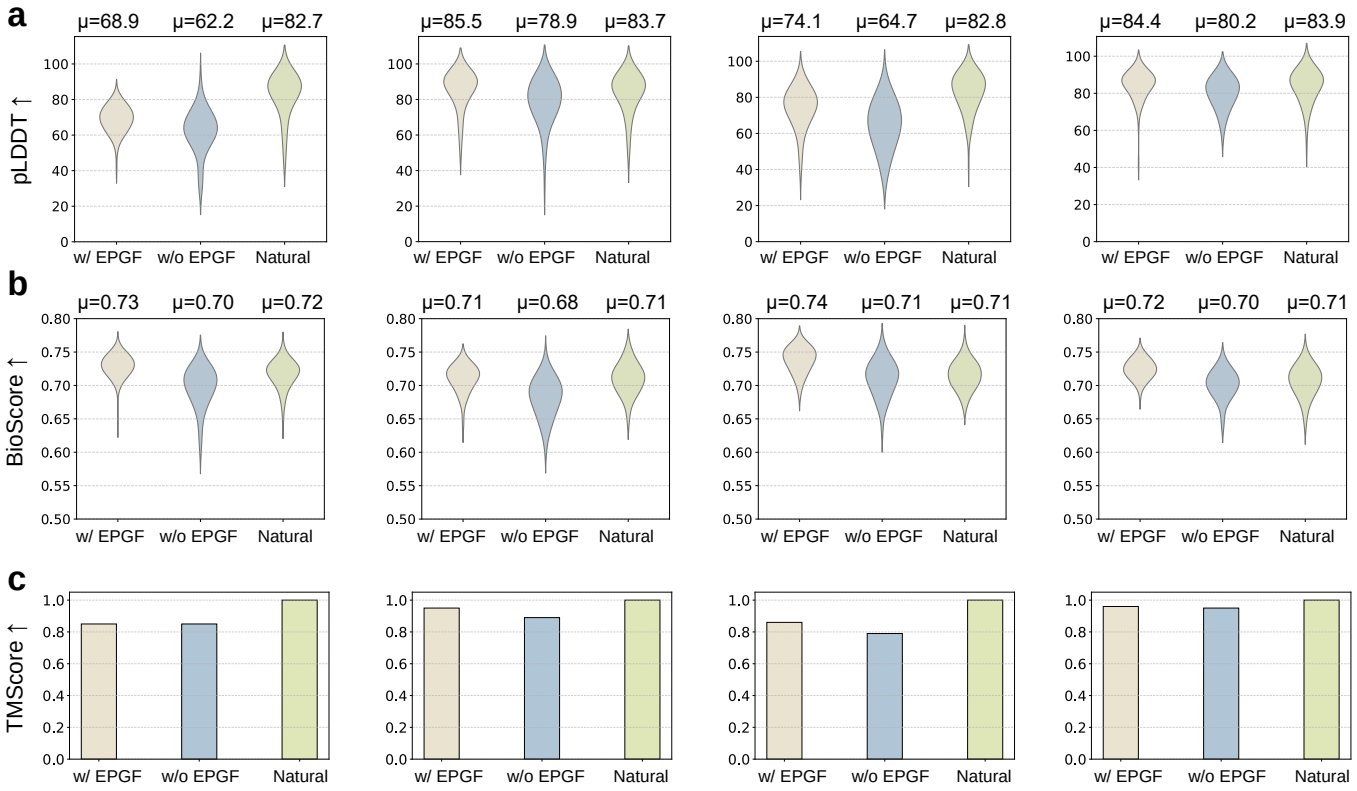

Fig. 5. **ProLLaMA generates better protein sequences with EPGF.** We visualize the (a) pLDDT (b) BioScore (c) TM-Score values of proteins belonging to four superfamiles (in order: SAM-MT, TPHD, Trx, CheY). $w/$: ProLLaMA with EPGF; w/o: ProLLaMA alone; $Natural$: Natural proteins as reference; μ: the average value; $BioScore$: the biophysical score calculated by our scorer. EPGF improves the performance of ProLLaMA and even makes the generated proteins approach or even surpass the natural proteins on pLDDT.

date segments $\{C_1, C_2, \ldots, C_N\}$, where each C_j consists of L tokens $\{c_1, c_2, \ldots, c_L\}$, are subsequently processed by our *Probabilistic-Biophysical Joint Selection* mechanism (described below). This hierarchical efficient decoding strategy maintains the model's ability to generate high-quality sequences while reducing computational overhead.

3) Probabilistic-Biophysical Joint Selection with Adaptive Diversity Control: We propose a joint selection strategy that establishes an equilibrium between the statistical likelihood derived from language models and the biochemical validity determined by our specialized scorer. Models that optimize solely for probabilities often generate sequences that are statistically plausible but biologically aberrant. Conversely, methods prioritizing biochemical constraints may produce sequences lacking the statistical signatures characteristic of natural proteins. To address this limitation, the candidate segments are evaluated using a two-stage *Probabilistic-Biophysical Joint Selection* mechanism, enhanced by an *Adaptive Diversity Control* strategy to maintain sequence diversity.

Probabilistic Filtering. In the first stage, we evaluate the statistical coherence of the candidate segments. For each candidate segment C_j, we compute its log-probability under the PLM:

$$P(C_j) = \sum_{i=1}^{L} \log p(c_i \mid c_{<i}), \tag{4}$$

where $p(c_i \mid c_{<i})$ is the conditional probability of token c_i given the preceding tokens $c_{<i}$. The top K candidates with the highest $P(C_j)$ are retained for further evaluation, where K is defined as $K = \lceil \frac{N}{2} \rceil$, and N is the total number of candidate segments. This step ensures that the generated sequences are statistically fluent within the PLM's learned distribution.

Biophysical Sampling with Adaptive Diversity Control. The retained candidates are then evaluated using the proposed *Multi-dimensional Biophysical Scorer*, with the selection process dynamically regulated by our *Adaptive Diversity Control* strategy. This integration addresses the potential limitation of excessive filtering, which might lead to reduced sequence diversity.

For each candidate segment C_j, we compute a biophysical score $\mathcal{B}(C_j)$ and ensure that it exceeds 0.55, which is the minimum acceptable score determined by the lowest biophysical score observed in natural proteins. The final selection probability is determined by:

$$P_{\text{select}}(C_j) = \frac{\exp(\mathcal{B}(C_j)/\tau_t)}{\sum_{k=1}^{K} \exp(\mathcal{B}(C_k)/\tau_t)}, \tag{5}$$

where τ is the adaptive diversity control parameter at the current step, controlled by the decay schedule:

$$\tau \leftarrow \max(\tau_{\text{final}}, \tau \cdot \gamma). \tag{6}$$

This adaptive diversity control mechanism enables a dynamic exploration-exploitation trade-off throughout the generation process. During early stages (high τ), the model maintains broad exploration capability by accepting a wider range of candidates, including those with suboptimal biophysical

TABLE I
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. AE: AUTO-ENCODER. AR: AUTO-REGRESSIVE.

Type	Method	pLDDT↑	SC-Perp↓	AFDB		PDB	
				TM-score↑	RMSD↓	TM-score↑	RMSD↓
CNN	CARP [32]	34.40±14.43	4.05±0.52	0.28	19.38	0.38	8.95
	LRAR [32]	49.13±15.50	3.59±0.54	0.40	14.47	0.43	9.47
PLM (AE)	ESM-1b [16]	59.57±15.36	3.47±0.68	0.34	20.88	0.44	8.59
	ESM-2 [33]	51.16±15.52	3.58±0.69	0.20	35.70	0.41	9.57
Diffusion	EvoDiff [32]	44.29±14.51	3.71±0.52	0.32	21.02	0.41	10.11
PLM (AR)	ProtGPT2 [18]	56.32±16.05	3.27±0.59	0.44	12.60	0.43	9.19
	ProGen2 [20]	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**

scores. As generation progresses and τ decreases, the selection becomes increasingly focused on candidates with superior biophysical properties.

This joint selection mechanism, enhanced by adaptive diversity control, ensures that the generated sequences are not only statistically coherent and biologically viable, but also maintain evolutionary diversity. The dynamic adjustment of τ prevents premature convergence to local optima while gradually guiding the selection toward functionally promising regions of the sequence space.

III. EXPERIMENTS

We introduce the experiment setup in Section III-A. And we evaluate the unconditional protein generation task in Section III-B, the controllable protein generation task in Section III-C, the protein property prediction task in Section III-D.

A. Experiment Setup

Training Settings: For continued pre-training, the LoRA rank is set to 128, employing the AdamW optimizer alongside a cosine annealing scheduler with a warm-up. The peak learning rate stands at 5e-5, with a total of one training epoch. It takes six days on eight A6000 GPUs using FlashAttention-2 [34]. For instruction tuning, the LoRA rank is set to 64 with two training epochs, and all other settings remain consistent with the continued pre-training setup. It takes 5 days on eight A6000 GPUs. More training details can be found in Supplementary Material.

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 superfamily. Property prediction involves predicting protein superfamily and solubility based on instructions, which include the protein sequences to be predicted. All evaluations are conducted on one GPU with 24GB of VRAM. For EPGF, the number of candidates is 8, the initial τ is 1.0, the final τ is 1.0, and the decay rate is 0.1.

Evaluation Metrics: We use the following metrics to evaluate the generated protein sequences. The pLDDT [35] is used to measure whether the sequences are structurally plausible. Self-Consistency Perplexity (SC-Perp) [32] serves as an additional metric of plausible structures since pLDDT falls short in dealing with intrinsically disordered regions (IDRs) [36]. The TM-score [37] reflects the structural similarity between the generated sequences and the known ones in AFDB [38] and PDB [39]. 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. More details are shown in Supplementary Material.

Baselines: The baselines cover various types of models. As shown in Table I, CARP and LRAR belong to Convolutional Neural Networks (CNN). ESM-1b and ESM-2 are language models based on Auto-Encoder (AE) architectures, and we use Gibbs sampling to make them generate proteins. EvoDiff is a diffusion model. ProtGPT2, Mol-Instructions, and our ProLLaMA are Auto-Regressive (AR) language models. We demonstrate the comparison of the parameters in Supplementary Material.

B. Unconditional Protein Generation

Table I shows the results. Our ProLLaMA is optimal on pLDDT, TM-score, and RMSD and suboptimal on SC-Perp. This indicates that ProLLaMA, through its training on protein sequence data, can generate structurally plausible proteins. In particular, 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 in [32], which are 68.25±17.85 and 3.09±0.63, respectively.

The de novo design of long and structurally plausible protein sequences is a huge challenge [18], yet our ProLLaMA performs well. As shown in Fig. 7(a), when the length is greater than 300, ProLLaMA performs the best in all three metrics. Although ProGen2's performance is better in short

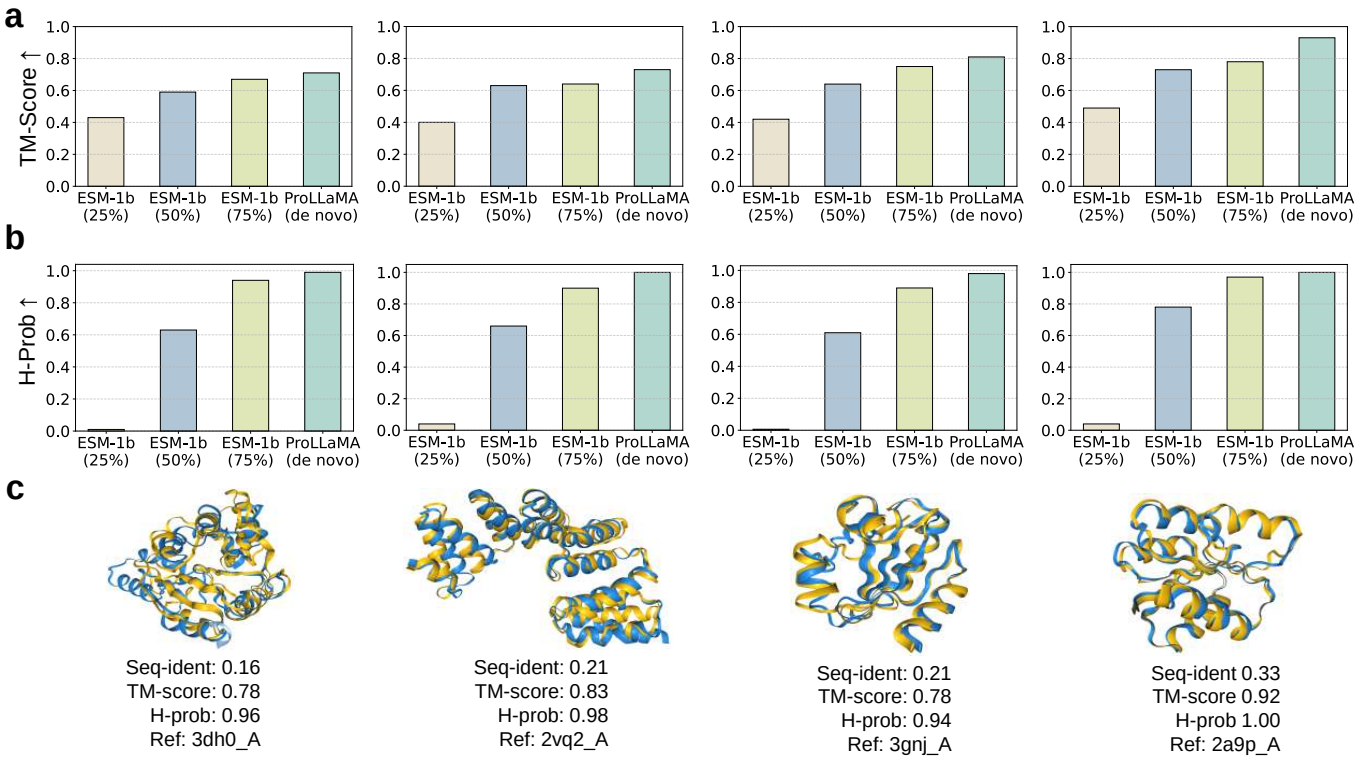

Fig. 6. **The performance of ProLLaMA in Conditional Protein Generation.** Proteins of four superfamilies generated by our ProLLaMA outperform ESM-1b in terms of (a) TM-score and (b) H-Prob, even partial residues are provided to ESM-1b. 25%: 25% residues are provided. (c) We visualize four proteins generated by ProLLaMA using SAM-MT, TPHD, Trx, and CheY as instructions. Blue is generated protein, and yellow is the natural protein as the reference, indicating that the generated protein is novel in sequence and reliable in structure.

sequences (length$\leq$200), it decreases as the length increases. This indicates that ProLLaMA is capable of capturing long-range dependencies between amino acids while other models struggle.

C. Controllable Protein Generation

We use 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. We randomly select 100 natural proteins from each of the four superfamilies as benchmarks for comparison. We employ Foldseek [40] to compare generated proteins with natural ones.

The protein generation of ProLLaMA is controllable. The TM-scores shown in Table II 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 and Trx, the TM-scores of our generated sequences are about 0.8; for TPHD and CheY, they are around 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. In contrast, other models exhibit significantly lower TM-score due to their lack of controllable generation.

Furthermore, ProLLaMA's de novo generation outperforms ESM-1b's non-de novo generation, even when ESM-1b is provided with 75% of the residues, as evidenced by higher TM-scores and H-Prob in Fig. 6(a)(b). As shown in Fig. 7(b), ProLLaMA demonstrates robust generation performance across different sequence lengths. Across all superfamilies, the proteins generated by ProLLaMA maintain a high pLDDT across different length ranges. The results in Supplementary Material also indicate that their TM-Score is high, around 0.9. These results highlight ProLLaMA's ability to effectively capture structural and evolutionary relationships through text and sequence learning, enabling precise control over protein generation.

EPGF benefits ProLLaMA to generate more biologically plausible proteins. We compared the performance of ProLLaMA with and without EPGF across four aforementioned protein superfamilies. As shown in Fig. 5, ProLLaMA+EPGF consistently outperformed ProLLaMA alone. Specifically, ProLLaMA+EPGF achieved higher pLDDT scores, indicating greater confidence in local structure prediction, with mean values closer to or even exceeding those of natural proteins in the TPHD and CheY superfamilies (85.5 vs. 83.7, 84.4 vs. 83.9). Additionally, ProLLaMA+EPGF generated sequences with significantly higher BioScores, demonstrating better adherence to evolutionary and biophysical constraints. Finally, the TM-scores of ProLLaMA+EPGF were consistently higher, particularly in the TPHD and Trx superfamilies, indicating greater structural similarity to natural proteins. These results highlight EPGF's ability to guide ProLLaMA in generating

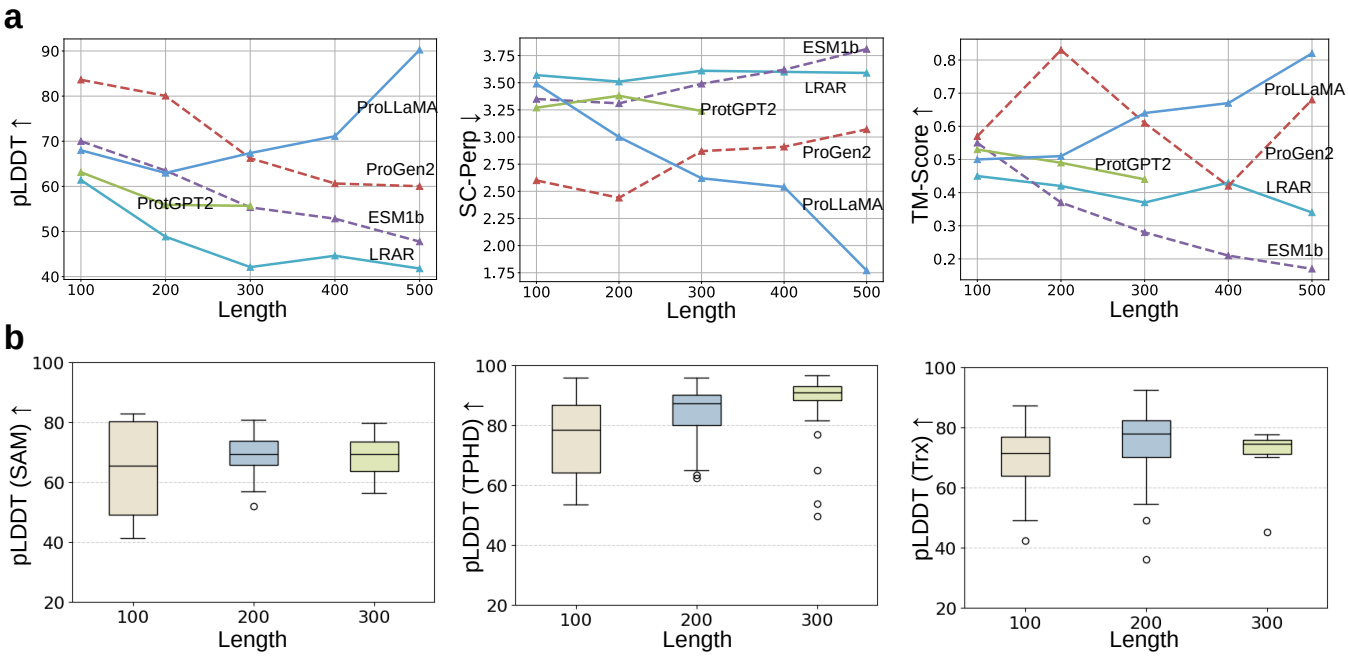

Fig. 7. **Comparison of proteins across different length intervals.** (a) Unconditional Generation: Compared to other methods, ProLLaMA maintains a high quality of generated proteins as their length increases. (b) Conditional Generation: Distribution of pLDDT values of proteins generated by ProLLaMA in different length intervals, validating the effectiveness of ProLLaMA.

TABLE II
CONTROLLABLE GENERATION OF PROLLAMA. SAM-MT, TPHD, TRX, AND CHEY ARE FOUR SUPEFAMILES.

Methods	SAM-MT	TPHD	Trx	CheY
ESM-1b	0.58	0.55	0.61	0.63
ESM-2	0.52	0.51	0.53	0.57
EvoDiff	0.46	0.42	0.42	0.46
ProtGPT2	0.45	0.43	0.44	0.45
ProGen2	0.44	0.45	0.43	0.44
Mol-Instructions	0.39	0.38	0.39	0.45
ProLLaMA	0.85	0.89	0.79	0.95
ProLLaMA+EPGF	**0.85**	**0.95**	**0.86**	**0.96**

TABLE III
PROTEIN SUPERFAMILY PREDICTION.

Metric	5-fold Validation	Test
Accuracy	0.671±0.005	0.671
Precision	0.702±0.004	0.701
Recall	0.700±0.005	0.697
Jaccard	0.691±0.004	0.690

biologically viable and structurally coherent protein sequences, bridging the gap between statistical language modeling and functional protein design. We provide more results and ablation experiments of EPGF in Supplementary Material.

Case study of four generated proteins. In Fig. 6(c), we visualize four examples of proteins generated by ProLLaMA (colored in blue) alongside the most structurally similar natural proteins from PDB as reference(colored in yellow). The significant overlap in 3D structures and the high TM-score confirm structural similarity. Low Seq-ident indicates sequence diversity. In summary, through controllable protein generation, ProLLaMA is capable of generating desired proteins with structures similar to natural proteins, yet with novel sequences.

TABLE IV
PROTEIN SOLUBILITY PREDICTION. *:VALUES ARE SOURCED FROM GRAPHSOL [41].

Method	Accuracy	Precision	Recall	F1-score
Protein-Sol* [42]	0.714	0.689	0.688	0.693
DeepSol* [43]	0.763	0.771	**0.738**	0.695
GraphSol* [41]	**0.779**	0.775	0.693	0.732
ProLLaMA (ours)	0.775	**0.788**	0.685	**0.733**

D. Property Prediction

Superfamily 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. Although ProLLaMA 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 the text. The advantage of the latter lies in its flexibility, such as the ability to easily handle situations where a sample belongs to multiple categories simultaneously. However, this increases task difficulty due to the much larger number of potential classification categories.

As shown in Table III, our model achieves an accuracy of 67.1% in predicting protein superfamilies on the test set, matching the performance observed in 5-fold validation (67.1% ± 0.5%). The model achieves a precision of 70.1%, recall of 69.7%, and a Jaccard score of 69.0% on the test set, all closely aligned with the validation results These results indicate the model's robustness and generalizability. The calculation formulas for these metrics can be found in Supplementary Material.

Solubility Prediction. We transform the eSol dataset [44,

45] into an additional instruction dataset, which includes two tasks: generating proteins based on solubility and determining the solubility of proteins. We binarize the solubility, with "Solubility is False" indicating insoluble and "Solubility is True" indicating soluble.

We train ProLLaMA (the one after the first training stage) on this additional instruction dataset. The LoRA rank is 64, the learning rate is 5e-5, and the number of training steps is 370. We compare our ProLLaMA with other methods in predicting protein solubility. The results are shown in Table. IV. Our ProLLaMA outperforms models specifically designed for solubility prediction in terms of precision and F1-score. And its accuracy is almost the same as that of GraphSol. In particular, ProLLaMA utilizes only protein sequences, whereas other models incorporate additional features such as absolute charge, secondary structure probabilities, etc (See Supplementary Material for detailed discussion).

IV. Related Work

Protein Language Models. Recognizing the similarity between natural language sequences and protein sequences, many methods of NLP have been applied to protein sequence data [46, 47, 48, 49]. This has led to the development of PLMs, which are broadly categorized into two types [11, 50]: Auto-Regressive (AR) PLMs and Auto-Encoder (AE) PLMs. AR PLMs adopt decoder-only architecture and Causal Language Modeling (CLM) [51, 52]. They mainly concentrate on PLG [53, 18, 19, 20], with a minority also focusing on fitness prediction [9]. AE PLMs adopt the encoder-only architecture and Masked Language Modeling (MLM) [14, 15, 16, 17, 33]. They excel in PLU, with the learned protein representations applied to downstream predictive tasks [54]. However, they face challenges in de novo protein generation. Our ProLLaMA is capable of multitasking, excelling in tasks in which both types specialize, and surpassing existing PLMs. This multitasking capability is achieved through instruction following, making it user-friendly. We have also noticed the recent emergence of scientific LLMs [12, 13, 55, 56, 57]. Although these models can also address certain protein-related problems, they lack a deep understanding of protein sequences due to the absence of large-scale pretraining on protein-specific corpora. As a result, they are better suited for handling general scientific question-answering tasks, but usually perform poorly when faced with complex protein tasks, especially protein generation.

V. Conclusion

Existing PLMs excel in either protein generation tasks or protein understanding tasks. In this work, we introduce an efficient training framework to transform any general LLM into a multi-task PLM. We construct an instruction dataset containing both generation tasks and understanding tasks. We developed ProLLaMA, a versatile PLM for multiple tasks such as controllable protein generation and prediction of protein properties. Evolutionary Protein Generation Framework (EPGF) plays a crucial role in bridging the gap between statistical language modeling and biological constraints. Experiments indicate that ProLLaMA and EPGF perform exceptionally well. We are confident that our work will have a significant impact on the AI4Science community and also open up exciting avenues for further exploration in biologically grounded generation strategies such as optimized EPGF.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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

[3] Alex Tamkin, Miles Brundage, Jack Clark, and Deep Ganguli. Understanding the capabilities, limitations, and societal impact of large language models. *arXiv preprint arXiv:2102.02503*, 2021.

[4] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

[5] Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szydło, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, et al. Chatgpt: Jack of all trades, master of none. *Information Fusion*, page 101861, 2023.

[6] Fan Huang, Haewoon Kwak, and Jisun An. Is chatgpt better than human annotators? potential and limitations of chatgpt in explaining implicit hate speech. *arXiv preprint arXiv:2302.07736*, 2023.

[7] Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. Can chatgpt understand too? a comparative study on chatgpt and fine-tuned bert. *arXiv preprint arXiv:2302.10198*, 2023.

[8] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*, 2023.

[9] Pascal Notin, Mafalda Dias, Jonathan Frazer, Javier Marchena Hurtado, Aidan N Gomez, Debora Marks, and Yarin Gal. Tranception: Protein fitness prediction with autoregressive transformers and inference-time retrieval. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 16990–17017. PMLR, 17–23 Jul 2022.

[10] Alexey Strokach and Philip M Kim. Deep generative modeling for protein design. *Current opinion in structural biology*, 72:226–236, 2022.

[11] Noelia Ferruz and Birte Höcker. Controllable protein design with language models. *Nature Machine Intelligence*, 4(6):521–532, 2022.

[12] Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. Mol-instructions-a large-scale biomolecular instruction dataset for large language models. In *The Twelfth International Conference on Learning Representations*, 2023.

[13] Qizhi Pei, Lijun Wu, Kaiyuan Gao, Xiaozhuan Liang, Yin Fang, Jinhua Zhu, Shufang Xie, Tao Qin, and Rui Yan. Biot5+: Towards generalized biological understanding with iupac integration and multi-task tuning. *arXiv preprint arXiv:2402.17810*, 2024.

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

[15] Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alex Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in Neural Information Processing Systems*, 34:29287–29303, 2021.

[16] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, 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.

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

[18] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. Protgpt2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348, 2022.

[19] 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. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, pages 1–8, 2023.

[20] Erik Nijkamp, Jeffrey A Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. Progen2: exploring the boundaries of protein language models. *Cell Systems*, 14(11):968–978, 2023.

[21] Markus Freitag and Yaser Al-Onaizan. Beam search strategies for neural machine translation. *arXiv preprint arXiv:1702.01806*, 2017.

[22] Giorgio Valentini, Dario Malchiodi, Jessica Gliozzo, Marco Mesiti, Mauricio Soto-Gomez, Alberto Cabri, Justin Reese, Elena Casiraghi, and Peter N Robinson. The promises of large language models for protein design and modeling. *Frontiers in Bioinformatics*, 3:1304099, 2023.

[23] Baris E Suzek, Yuqi Wang, Hongzhan Huang, Peter B McGarvey, Cathy H Wu, and UniProt Consortium. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015.

[24] Typhaine Paysan-Lafosse, Matthias Blum, Sara Chuguransky, Tiago Grego, Beatriz Lázaro Pinto, Gustavo A Salazar, Maxwell L Bileschi, Peer Bork, Alan Bridge, Lucy Colwell, et al. Interpro in 2022. *Nucleic acids research*, 51(D1):D418–D427, 2023.

[25] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

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

[27] Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 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)*, pages 7319–7328, 2021.

[28] M Michael Gromiha, Motohisa Oobatake, Hidetoshi Kono, Hatsuho Uedaira, and Akinori Sarai. Relationship between amino acid properties and protein stability: buried mutations. *Journal of Protein Chemistry*, 18:565–578, 1999.

[29] James T Kellis Jr, Kerstin Nyberg, Dasa S ail, and Alan R Fersht. Contribution of hydrophobic interactions to protein stability. *Nature*, 333(6175):784–786, 1988.

[30] Swagata Das, Uttam Pal, Supriya Das, Khyati Bagga, Anupam Roy, Arpita Mrigwani, and Nakul C Maiti. Sequence complexity of amyloidogenic regions in intrinsically disordered human proteins. *PLoS One*, 9(3):e89781, 2014.

[31] Patrice Koehl and Michael Levitt. Structure-based conformational preferences of amino acids. *Proceedings of the National Academy of Sciences*, 96(22):12524–12529, 1999.

[32] Sarah Alamdari, Nitya Thakkar, Rianne van den Berg, Alex Xijie Lu, Nicolo Fusi, Ava Pardis Amini, and Kevin K Yang. Protein generation with evolutionary diffusion: sequence is all you need. *bioRxiv*, pages 2023–09, 2023.

[33] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[34] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

[35] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure

prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

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

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

[38] Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, 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.

[39] Helen M Berman, Tammy Battistuz, Talapady N Bhat, Wolfgang F Bluhm, Philip E Bourne, Kyle Burkhardt, Zukang Feng, Gary L Gilliland, Lisa Iype, Shri Jain, et al. The protein data bank. *Acta Crystallographica Section D: Biological Crystallography*, 58(6):899–907, 2002.

[40] Michel Van Kempen, Stephanie S Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron LM Gilchrist, Johannes Söding, and Martin Steinegger. Fast and accurate protein structure search with foldseek. *Nature Biotechnology*, 42(2):243–246, 2024.

[41] Jianwen Chen, Shuangjia Zheng, Huiying Zhao, and Yuedong Yang. Structure-aware protein solubility prediction from sequence through graph convolutional network and predicted contact map. *Journal of cheminformatics*, 13:1–10, 2021.

[42] Max Hebditch, M Alejandro Carballo-Amador, Spyros Charonis, Robin Curtis, and Jim Warwicker. Protein–sol: a web tool for predicting protein solubility from sequence. *Bioinformatics*, 33(19):3098–3100, 2017.

[43] Sameer Khurana, Reda Rawi, Khalid Kunji, Gwo-Yu Chuang, Halima Bensmail, and Raghvendra Mall. Deepsol: a deep learning framework for sequence-based protein solubility prediction. *Bioinformatics*, 34(15):2605–2613, 2018.

[44] Tatsuya Niwa, Bei-Wen Ying, Katsuyo Saito, WenZhen Jin, Shoji Takada, Takuya Ueda, and Hideki Taguchi. Bimodal protein solubility distribution revealed by an aggregation analysis of the entire ensemble of escherichia coli proteins. *Proceedings of the National Academy of Sciences*, 106(11):4201–4206, 2009.

[45] Jack Hanson, Kuldip Paliwal, Thomas Litfin, Yuedong Yang, and Yaoqi Zhou. Accurate prediction of protein contact maps by coupling residual two-dimensional bidirectional long short-term memory with convolutional neural networks. *Bioinformatics*, 34(23):4039–4045, 2018.

[46] Kevin K Yang, Zachary Wu, Claire N Bedbrook, and Frances H Arnold. Learned protein embeddings for machine learning. *Bioinformatics*, 34(15):2642–2648, 2018.

[47] Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, 16(12):1315–1322, 2019.

[48] Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. In *International Conference on Machine Learning*, pages 8844–8856. PMLR, 2021.

[49] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, 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.

[50] Zaixiang Zheng, Yifan Deng, Dongyu Xue, Yi Zhou, Fei Ye, and Quanquan Gu. Structure-informed language models are protein designers. *bioRxiv*, pages 2023–02, 2023.

[51] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. *Advances in neural information processing systems*, 13, 2000.

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

[53] Lewis Moffat, Shaun M Kandathil, and David T Jones. Design in the dark: learning deep generative models for de novo protein design. *bioRxiv*, pages 2022–01, 2022.

[54] Minghao Xu, Xinyu Yuan, Santiago Miret, and Jian Tang. Protst: Multi-modality learning of protein sequences and biomedical texts. *arXiv preprint arXiv:2301.12040*, 2023.

[55] Songhua Yang, Hanjie Zhao, Senbin Zhu, Guangyu Zhou, Hongfei Xu, Yuxiang Jia, and Hongying Zan. Zhongjing: Enhancing the chinese medical capabilities of large language model through expert feedback and real-world multi-turn dialogue. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19368–19376, 2024.

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

[57] Qizhi Pei, Lijun Wu, Kaiyuan Gao, Jinhua Zhu, Yue Wang, Zun Wang, Tao Qin, and Rui Yan. Leveraging biomolecule and natural language through multi-modal learning: A survey. *arXiv preprint arXiv:2403.01528*, 2024.