xTrimoPGLM: Unified 100B-Scale Pre-trained Transformer for Deciphering the Language of Protein

Bo Chen[*†]
Tsinghua University

Xingyi Cheng[*‡]
BioMap Research

Yangli-ao Geng[†]
Tsinghua University

Shen Li
BioMap Research

Xin Zeng
BioMap Research

Boyan Wang[†]
Tsinghua University

Jing Gong
BioMap Research

Chiming Liu
BioMap Research

Aohan Zeng
Tsinghua University

Yuxiao Dong
Tsinghua University

Jie Tang[§]
Tsinghua University

Le Song[§]
BioMap Research & MBZUAI

Abstract

Protein language models have shown remarkable success in learning biological information from protein sequences. However, most existing models are limited by either autoencoding or autoregressive pre-training objectives, which makes them struggle to handle protein understanding and generation tasks concurrently. This paper proposes a unified protein language model, xTrimoPGLM, to address these two types of tasks simultaneously through an innovative pre-training framework. Our key technical contribution is an exploration of the compatibility and the potential for joint optimization of the two types of objectives, which has led to a strategy for training xTrimoPGLM at an unprecedented scale of 100 billion parameters and consuming 1 trillion training tokens. Our extensive experiments reveal that xTrimoPGLM significantly outperforms other advanced baselines in diverse protein understanding tasks (13 out of 15 tasks across four categories) and generates novel protein sequences which are structurally similar to natural ones. Furthermore, using the same xTrimoPGLM framework, we train an antibody-specific model (xTrimoPGLM-Ab) using 1 billion parameters. This model set a new record in predicting antibody naturalness and structures, both essential to the field of antibody-based drug design, and demonstrated a significantly faster inference speed than AlphaFold2. These results highlight the substantial capability and versatility of xTrimoPGLM in understanding and generating protein sequences.

1 Introduction

Proteins play vital roles in the sustenance, growth, and defense mechanisms of living organisms. They provide structural support for many essential biological processes such as synthesizing enzymes, facilitating transportation, regulating gene expression, and contributing to immune function.

[*]Equal Contribution. Emails: cb21@mails.tsinghua.edu.cn, xingyi@biomap.com
[†]Work done while interning at BioMap Research, California, USA.
[‡]The project leader at BioMap Research, California, USA.
[§]The corresponding authors. Emails: songle@biomap.com, jietang@tsinghua.edu.cn

Preprint. Work in progress.

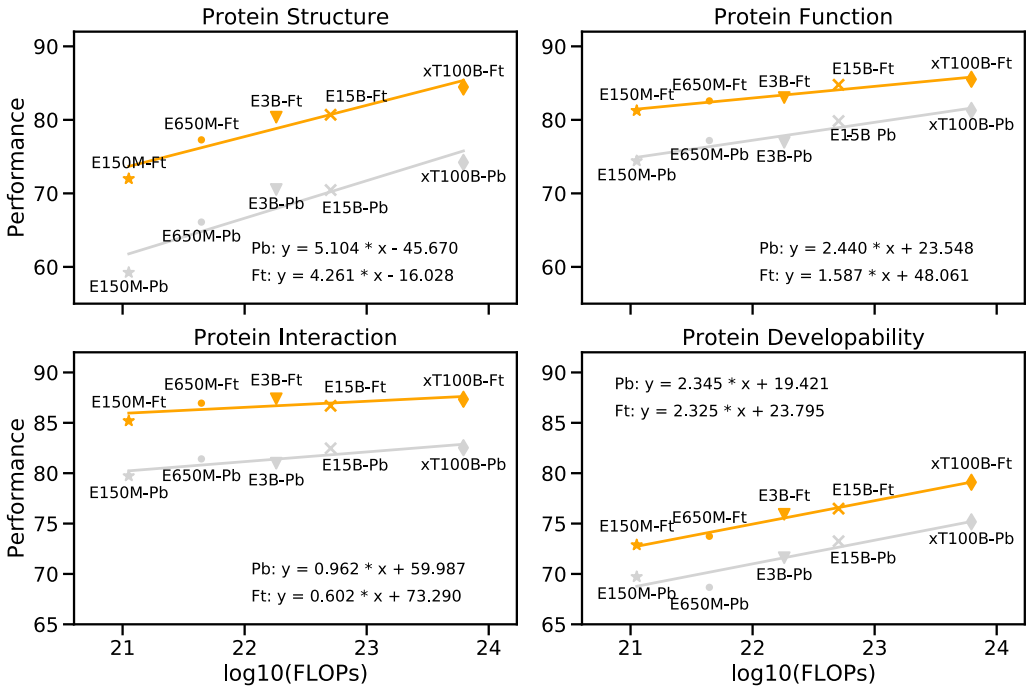


Figure 1: The relationship between the computational complexity of model training, quantified in Floating Point Operations (FLOPs), and the model performance across four distinctive task categories, each encompassing 3-4 subtasks. Each data point symbolizes the mean performance metric for a specific task category. We conduct two types of training methods on downstream tasks with varying complexities, utilizing both Probing (Pb) and Fine-tuning (Ft) techniques. Notably, an exponential increase in pre-training computations corresponds to linear growth in performance, thereby highlighting scaling effects [1, 2]. The labels **E150M**, **E650M**, **E3B**, **E15B**, and **xT100B** represent ESM2-150M, ESM2-650M, ESM2-3B, ESM2-15B, and xTrimoPGLM-100B, respectively. For a comprehensive analysis of the results, refer to Section 4.4. For a comparative study on the FLOPs across different models, refer to Appendix C.

Therefore, understanding the biological information encoded within proteins is crucial for unraveling the intricate workings of life and advancing fields such as medicine and biotechnology [3, 4]. As protein sequences serve as the blueprint for protein structure and function [5], pre-trained techniques on sequences, known as **P**rotein **L**anguage **M**odels (PLMs), e.g., the family of ESM models [6, 7], ProtTrans [8], ProGen [9], etc., offer a powerful tool for characterizing the properties and distributions of general protein sequences. These models are trained on large-scale protein datasets [3, 10, 11, 12] that encompass billions of sequences, allowing them to capture evolutionary patterns and sequence features that are inherent in protein structures. As a result, these models achieve state-of-the-art results in predicting protein functions and structures [3, 7] or generating novel sequences with faithful three-dimensional structures [9, 13].

It is worth noting that different categories of protein-related tasks necessitate divergent outputs from PLMs, such as protein understanding tasks call for PLMs to yield accurate residue-level or protein-level representations, while protein design tasks depend heavily on the potent generation capabilities of PLMs. Despite these varying outputs, all tasks reveal a consistent underlying dependency among protein sequences [5, 14, 15], which suggests the possibility of characterizing these tasks within one unified framework, potentially mitigating the disparity between task types and further augmenting the modeling power of PLMs. Unfortunately, prevailing PLMs are designed to address specific tasks depending on their pre-training framework. This presents a significant challenge to selecting appropriate PLMs for specific task types. A summary of the pre-training architectures adopted by existing PLMs and their comparative analysis is presented in Table 1. In light of this, we discuss the potential of unifying both the understanding tasks and the generation tasks, as dictated by their respective autoencoding and autoregressive pre-training objectives, in one unified framework, which

Table 1: Comparisons between different architectures of PLMs. "—" denotes that the approach has not been explored yet.

Downstream Task	**Autoenc.**	**Autoreg.**	**Enc.-Dec.**	**GLM**	Example
Protein Understanding	✓	×	✓	✓	Contact Prediction
Protein Generation	×	✓	—	✓	Antibody Re-design
Representatives	ESM-1b[6], ESM2[7], Pro.BERT[25]	ProGen[9], ProGen2[13], ProtGPT2[26]	ProtTrans[8], Ankh[27]	xTrimoPGLM	/

is capable of capturing the comprehensive underlying dependency embedded in the protein sequences and leads to more versatile and powerful PLMs.

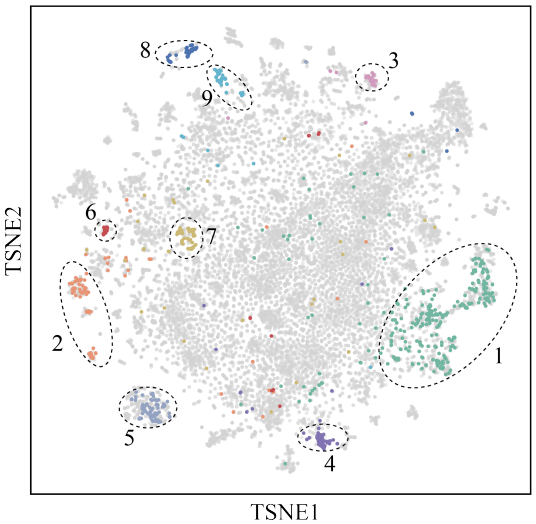
Although substantial efforts have been invested in the exploration of a unified pre-training paradigm within the natural language processing (NLP) domain [16, 17, 18], these studies typically adopt analogous training patterns. For instance, all pre-training objectives are commonly optimized using either the in-place token prediction format [17] or next-token prediction regime [16]. To fulfill the requirements of unified PLMs, it is essential to incorporate both the in-place token prediction formulation, exemplified by BERT-style objectives [19], to reinforce the model's representation ability, and the next-token prediction formulation, typified by GPT-style objectives [20], to ensure the model's generative capacity. Nevertheless, previous explorations in NLP have not discussed extrapolating conclusions observed in similar training formats to more general settings. The question remains open as to whether the in-place token prediction objectives can benefit from the next-token prediction objectives and vice versa, and whether these two types of objectives are compatible and can be optimized concurrently. In addition, the existing landscape of NLP is dominated by generative models, which afford various types of tasks via mapping task labels into a unified text space for zero/few-shot learning [20] or instruction-tuning [21, 22]. However, this capability is currently beyond the reach of PLMs. In practice, applications of protein modeling still rely on the bridging of representations with downstream task-specific labels, such as 3D coordinates for structures prediction [7, 23]. which heavily rely on BERT-style training to tackle protein understanding tasks. Consequently, this highlights the need for a unified model that incorporates both training objectives.

In this paper, we introduce the xTrimo **P**rotein **G**eneral **L**anguage **M**odel (xTrimoPGLM), which is a unified pre-training framework designed to concurrently address diverse protein-related tasks, including but not limited to, protein understanding and generation (or design). Concretely, xTrimoPGLM adopts the backbone of the General Language Model (GLM) [24] to leverage its bidirectional attention advantage and auto-regressive blank filling objective compared with vanilla encoder-only and causal decode-only language models. To enhance the representation capacity of xTrimoPGLM, we introduce the Masked Language Model (MLM) objective to the bidirectional prefix region, building upon the generation ability already encapsulated within the GLM objective. We provide empirical validation of the compatibility between the two categories of pre-training objectives. Our results further confirm that they can expedite convergence when transitioning from one to the other. Following extensive empirical verification, we pre-train xTrimoPGLM at 100B-scale utilizing a comprehensive dataset encompassing Uniref90 and ColdFoldDB - marking the largest scale of PLMs evaluated to date. The training stage involved over 1 trillion tokens processed on a cluster of 96 NVIDIA DGX-A100 (8×40G) GPU nodes between January 18 and June 30, 2023. As of the current date, we continue to pre-train xTrimoPGLM-100B in a unified manner to further enhance its modeling power.

We conduct extensive experiments to evaluate the effectiveness of our proposed xTrimoPGLM framework. In protein understanding tasks, we benchmark xTrimoPGLM-100B against the current state-of-the-art (SOTA) methods in 15 tasks spanning protein structure, function, interaction, and developability. Remarkably, xTrimoPGLM-100B significantly outperforms other SOTA techniques on 13 out of these 15 tasks. The correlation between performance improvement and increased pre-training computations, illustrated in Figure1, substantiates the scaling laws for large language models [1, 2]. This suggests that as the computational demands of training (measured in FLOPs), model parameters, and data size for PLMs increase exponentially, downstream performance on protein-related tasks continues to grow linearly. In protein generation tasks, xTrimoPGLM-100B demonstrates its capacity to generate novel protein sequences with varied lengths and sequence identities by adjusting generation hyper-parameters. Significantly, xTrimoPGLM-100B exhibits

the capability to generate structurally similar but low sequence identity novel protein sequences when compared to natural ones (Cf. Section 6.2). We visualize the embeddings produced by xTrimoPGLM-100B, as depicted in Figure 2, and find that the learned embeddings clearly capture functional information for the protein universe. These compelling results highlight the vast potential and versatility of xTrimoPGLM and its framework in understanding and generating protein sequences, which signal a promising future for the application of unified pre-training paradigms in the protein modeling domain.

We also developed an antibody-specific PLM, xTrimoPGLM-Ab-1B, containing 1 billion parameters. This model is pre-trained on the Observed Antibody Space (OAS) database [28], processing over 1 trillion tokens. It achieves SOTA performance on antibody naturalness and structure prediction tasks. Notably, xTrimoPGLM-Ab-1B rivals AlphaFold2 [3] on structural prediction of antibodies with 6,000× inference speedup. xTrimoPGLM-Ab-1B also has the capability to redesign the CDR region of Covid-19 antibodies (Cf. Section 6.3).



1: DNA-binding transcription repressor activity, RNA polymerase II-specific (GO:0001227, n=324)

2: transmembrane transporter activity (GO:0022857, n=107)

3: neuropeptide hormone activity (GO:0005184, n=30)

4: cytoplasmic translation (GO:0002181, n=87)

5: odorant binding (GO:0005549, n=118)

6: delayed rectifier potassium channel activity (GO:0005251, n=32)

7: protein glycosylation (GO:0006486, n=89)

8: immunoglobulin complex (GO:0019814, n=90)

9: collagen trimer (GO:0005581, n=60)

Figure 2: t-SNE visualization of xTrimoPGLM-100B context embedding for human protein sequences. The figure portrays xTrimoPGLM's capability to encapsulate biologically relevant latent embeddings across diverse functional protein sequences. Specifically, human protein sequences (n=20,255) from UniProt are fed to xTrimoPGLM to obtain the context embedding. We then use the t-SNE [29] method to reduce the embedding dimensions for visualization. Each dot signifies a distinct protein. Nine clusters related to known Gene Ontology annotation terms are highlighted with different colors.

2 Background

In this section, we discuss the background surrounding protein language models and other unified pre-training regimes.

2.1 Pre-training on Proteins

Pre-training on protein sequences has recently gained significant interest due to its potential applications in protein structure prediction, drug discovery, etc [3, 7, 30, 31]. A family of models including ProtTrans [8], ESM-1b [6] and ESM2 [7], employ individual protein sequences as input and are pre-trained via optimizing the masked language model objective based on encoder-only architectures. To further incorporate evolutionary information from homology protein sequences, MSA-Transformer [32] utilizes multiple sequence alignment (MSA) instead of the single query sequence as input. This 110-million parameter model performs comparably to ESM2-15B in protein structure prediction tasks, demonstrating that introducing co-evolutional information into the pre-training process is beneficial for protein structural-related tasks. To enable de-novo protein design, some PLMs like ProGen2 [13] and ProtGPT2 [26] are pre-trained in an autoregressive manner, which allows them to generate diverse, realistic proteins without conditions. ProGen [9] extends these

approaches by incorporating the functional tag and concatenating it with the protein sequences as input. Consequently, ProGen is capable of generating specialized protein sequences that exhibit desired properties. In summary, current PLMs have demonstrated a promising ability to improve the accuracy and efficiency of protein understanding and design tasks compared to traditional machine learning models. However, none of the aforementioned methods have explored the potential of a unified pre-training strategy that could thoroughly characterize the distribution of protein sequences.

2.2 Unifying Language Pre-Training

Several pioneering endeavors have explored the concept of a unified language pre-training framework in the domain of NLP. Among these, UniLM [17] is proposed to train on multiple language modeling objectives using the encoder-only architecture. It unifies multiple pre-training objectives, including unidirectional LM, bidirectional LM, and seq2seq LM into a single cloze-type formulation, i.e. the masked language model. This unification is achieved by introducing various masking patterns to the input to distinguish between the different objectives. UniLMv2 [18] utilizes partially autoregressive modeling for generation tasks, and the autoencoding objective for NLU tasks. This approach introduces a complex masking strategy to prevent information leakage. Aside from pre-training unification, there has been a recent trend of thematic unification, i.e., unifying common tasks into one model. UL2 [16] introduces the Mixture-of-Denoisers (MoD) pre-training objective, which combines diverse pre-training paradigms for more effective unified pre-training. This has become the primary pre-training strategy for PLAN-UL2. Although existing unified models have demonstrated the advantages of combining different pre-training objectives, they all adopt the same formulation for model optimization. This includes the in-place token prediction formulation used by UniLM and UniLM-v2, and the next-token prediction paradigm employed by UL2. In contrast, xTrimoPGLM seeks to combine these two training formats into a unified pre-training framework, endowing PLMs with the representational capacity for protein understanding tasks and the generative capacity for protein design tasks.

3 The Design Choices of xTrimoPGLM

Protein-related tasks can be broadly classified into two categories: 1) Understanding tasks, such as contact prediction [33, 34], fluorescence landscape prediction [35], etc. These tasks necessitate PLMs to provide accurate residue-level or sequence-level representations. 2) Generation or design tasks, such as antibody Complementarity-Determining Region (CDR) redesign, which rely on the generative capacity of PLMs. The conventional encoder-only PLMs, e.g., ESM [6, 36], or causal decoder-only PLMs, e.g., ProtGPT2 [26], struggle to concurrently address these two types of tasks, due to their inductive bias of the pre-trained framework. For instance, while ESM2 [36] presents superior performance on benchmarking most of understanding tasks compared to other PLMs, it cannot be utilized directly to generate novel protein sequences in an end-to-end regime.

Conceptually, the two types of tasks reflect the consistent underlying distributions within the general protein sequences [14, 15], which ideally should be captured by a unified model to further enhance the modeling power of PLMs. Therefore, encoder-decoder models such as T5 [37], and non-causal decoder-only models, often referred as prefix language models, e.g., General Language Model (GLM) [24], are potential architectures that can concurrently handle understanding and generation tasks. These models accomplish this by optimizing an autoregressive blank-filling objective while consuming input bidirectionally.

Compared with GLM, T5 proves inefficient as it necessitates an order of magnitude larger parameter to achieve similar modeling power [24]. Consequently, we adopt the GLM framework as the backbone to exploit its bidirectional attention advantage and autoregressive blank infilling objective [38].

3.1 Backbone Framework: General Language Model (GLM)

GLM is a transformer-based language model that leverages autoregressive blank infilling as its training objective while consuming the input text in a bidirectional manner. It randomly blanks out continuous spans of tokens from the input text, following the idea of autoencoding, and trains the model to sequentially reconstruct these spans, following the idea of autoregressive pre-training.

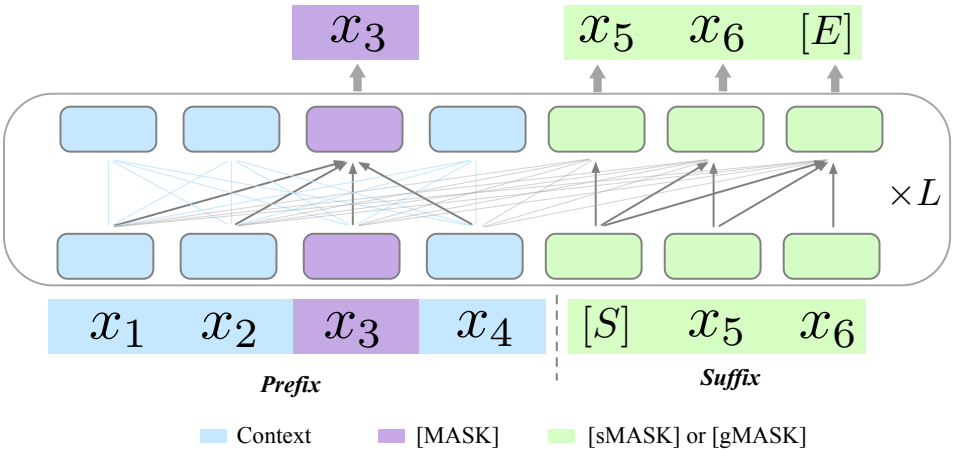


Figure 3: Illustration of the xTrimoPGLM framework. L is the number of transformer layers. (`[S]`, `[E]`) denotes the start or end of the input span, respectively. MLM, indicated by the `[MASK]` token, represents the in-place token predictions designed to improve the model's representation ability. GLM, signed by the `[gMASK]` and `[sMASK]` tokens, denotes the next-token predictions designed to enhance the model's generation ability. Noted that `[sMASK]` masks consecutive spans in the middle of sequences, while `[gMASK]` masks the rest of sequences except for the context.

GLM's bidirectional attention over unmasked (i.e., uncorrupted) contexts distinguishes it from causal decoder-only LMs in which only the unidirectional attention is used, as illustrated in Figure. 3.

3.2 Pre-Training Objectives

GLM incorporates two distinct pre-training objectives to ensure its generative capabilities: 1) *Span prediction*, to recover short blanks in sentences whose lengths add up to a certain portion of the input, and 2) *Long-text generation*, to generate long blanks with random-length at the end of sentences with prefix contexts provided. To further equip xTrimoPGLM with the understanding capacity, we additionally incorporate the masked language model [19] (MLM) as the understanding objective. This inclusion ensures that the xTrimoPGLM is capable of generating both accurate residue-level and sequence-level representations.

In summary, xTrimoPGLM simultaneously employs two types of pre-training objectives, each with its specific indicator tokens, to ensure both understanding and generative capacities:

- **MLM**: The in-place token prediction task that predicts the tokens randomly masked with a special indicator `[MASK]` within sequences.
- **GLM**: The next-token prediction task recovering short spans masked within sequences using `[sMASK]` or longer spans at the end of sequences using `[gMASK]`.

Conceptually, when the `[MASK]` indicator is utilized, xTrimoPGLM mirrors the functionality of BERT [19]. Conversely, upon utilizing `[sMASK]` or `[gMASK]`, the operation of xTrimoPGLM resembles PrefixLM [37, 39] or GPT [20], as depicted in Figure 3. More specifically,

Masked Language Models (MLM) for Understanding. The MLM objective aims at in-place masked token predictions. Formally, for an input protein sequence $\mathbf{x} = [x_1, \cdots, x_n]$ and the positions of masks $M = \{m_1, \cdots, m_{|M|}\}$, then the MLM pre-training loss is defined as

$$\mathcal{L}_{\text{MLM}} = \mathbb{E}_M \left[\sum_{m \in M} -\log p(x_m | \mathbf{x}_{/M}) \right], \tag{1}$$

where $x_{/M}$ denotes all input tokens except the ones that are in M.

General Language Models (GLM) for Generation. The GLM objective aims at recovering the masked consecutive tokens, i.e., spans, in an autoregressive manner. Concretely, for an input sequence $\mathbf{x}$, sequence spans $\{\mathbf{s}_1, \cdots, \mathbf{s}_m\}$ are sampled from it. Each span $\mathbf{s}_i$, consisting of a consecutive section of tokens $[s_{i,1}, \cdots, s_{i,l_i}]$ in $\mathbf{x}$, is replaced with a single mask token `[sMASK]` or `[gMASK]` to form $\mathbf{x}_{\text{corrupt}}$. To make sure the interactions among corrupted spans, xTrimoPGLM randomly permutes the order of spans like GLM, and defines the pre-training objective as

$$\mathcal{L}_{\text{GLM}} = \mathbb{E}_{\mathbf{z} \sim Z_m} \left[\sum_{i=1}^{m} \sum_{j=1}^{l_i} -\log p\left(s_{z_i,j} | \mathbf{x}_{\text{corrupt}}, \mathbf{s}_{z_{<i}}, s_{z_i,<j}\right) \right], \tag{2}$$

where Z_m denotes the set of the sequence's all permutations and $\mathbf{s}_{z_{<i}}$ represents $\{\mathbf{s}_{z_1}, \cdots, \mathbf{s}_{z_{i-1}}\}$.

Unified Pre-Training. The two types of pre-training objectives are jointly optimized to pre-train the xTrimoPGLM model. The unified pre-training objective, which aims to maximize the likelihood of the oracle tokens, is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{MLM}} + \alpha \cdot \mathcal{L}_{\text{GLM}}, \tag{3}$$

where α is a weighting factor used to balance the different pre-training objectives. As a result, the proposed unified framework effectively takes advantage of the GLM architecture to characterize both the understanding ability via $\mathcal{L}_{\text{MLM}}$ and the generation capacity via $\mathcal{L}_{\text{GLM}}$.

3.3 The Pre-Training Setup of xTrimoPGLM -100B

The pre-training phase of xTrimoPGLM-100B can be divided into two stages. Initially, xTrimoPGLM-100B is pre-trained utilizing the MLM objective, enhancing its representation ability. The primary aim of this stage is to quickly reduce the loss to a low level. In the second stage, xTrimoPGLM-100B is trained with the unified objectives that combines MLM and GLM loss at a certain ratio to improve both the representation and generation abilities.

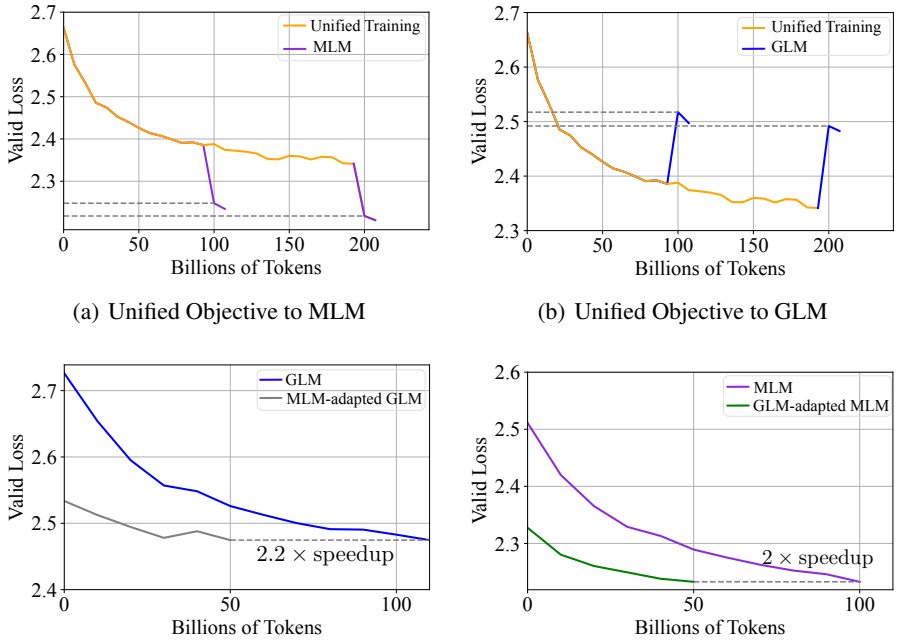
Masked Language Model (10% pre-trained tokens). `[MASK]` is employed to mask random tokens within the sequence, with the total length of the masked tokens amounting to 15% of the input.

General Language Model (90% pre-trained tokens). GLM [24] uses both `[sMASK]` and `[gMASK]` for this task. `[sMASK]` is used to mask consecutive spans serving the purpose of blank infilling. The lengths of these spans follow a Poisson distribution (λ = 6), with the total masked span amounting to 15% of the input. `[gMASK]` is used to mask the rest of the sequence given the prefix of the input sequences preserved as context. The length of the masked segment is drawn from a uniform distribution, with at least 40% of the tokens masked.

3.4 The Empirical Analysis of Unified Training

In this section, we delve deeper into the feasibility of simultaneously optimizing the two distinct objectives. Unlike existing unified pre-training frameworks [16, 17, 18, 40, 41] which employ analogous formulations to pre-train various objectives, we explore how to extend the conclusions drawn from a similar training format to a broader setting. Specifically, we investigate if a model optimized via in-place token predictions benefits the one trained via the next-token prediction regime, and vice versa. To achieve this, we must address two principal questions: 1). *Is the in-place token prediction objective compatible with the next-token prediction one?* 2). *Does the in-place token prediction objective contribute to the capability of the next-token prediction one, and vice versa?*

Pre-training settings. Empirical experiments are conducted based on xTrimoPGLM-150m encompassing 30 layers, 20 attention heads, 640 embedding dimensions, and FP16 precision. The other hyper-parameter settings are consistent with those of xTrimoPGLM-100B. Each model is pre-trained on Uniref50 [42]. Training is conducted on batches of 2,048 sequences, each of length 1,024 tokens. To operate within a fixed compute budget, we focus on the number of tokens observed during pre-training (corresponding to the total computational cost), rather than those actually trained (i.e., those on which a loss is calculated). These differences are considered intrinsic efficiency trade-offs between training objectives. Further specification for each objective is as follows:



(a) Unified Objective to MLM

(b) Unified Objective to GLM

(c) MLM-adapted GLM vs. GLM (Scratch)

(d) GLM-adapted MLM vs. MLM (Scratch)

Figure 4: The empirical analysis of unified training. (a)(b) The MLM and GLM objectives are optimized simultaneously. (c)(d) Adapting the model from the pre-trained one significantly accelerates convergence compared to that trained from scratch.

- **MLM.** Approximately 15% of input tokens are masked, leading to around 1,024 input and 154 target tokens, with the loss being computed exclusively on the targets.
- **GLM (`[gMASK]`).** Only the long-text generation objectives (signified by `[gMASK]`) are utilized, given the compatibility of the span corruption objective (`[sMASK]`) with the `[gMASK]` objectives has been verified. The loss computation pertains to the masked regions, encompassing a minimum of 40% of tokens.

Results. We answer the first question regarding the compatibility of these two types of objectives. Concretely, we alternate between using the MLM, representing the in-place token prediction, and employing the GLM (`[gMASK]`) objective, representing the next-token prediction, each occupying 50% of the time in the training batch. We switch the unified pre-training objective to each individual objective upon reaching the timesteps corresponding to the consumption of 100B and 200B tokens. Such transitions are feasible since the parameters and overall architecture remain constant, requiring only a switch in the attention mask. The corresponding validation loss is illustrated in Figure 4(a)(b). Interestingly, despite the seemingly conflicting nature of the two objectives incorporated within the unified pre-training objective, both the MLM loss and GLM loss are optimized simultaneously.

Then we investigate whether the in-place token prediction objective influences the convergence speed of the next-token prediction one, and vice versa. We conduct comparisons when adapting models to one objective after pre-training on another objective. The adapting model only trains over 50B tokens. First, we compare the xTrimoPGLM (GLM) adapted from the pre-trained xTrimoPGLM (MLM), denoted as MLM-adapted GLM, with the xTrimoPGLM (GLM) trained from scratch. Similarly, we compare the xTrimoPGLM (MLM) adapted from the pre-trained xTrimoPGLM (GLM), denoted as GLM-adapted MLM with the xTrimoPGLM (MLM) trained from scratch. The validation loss is depicted in Figure 4(c)(d). In general, we observe that adapting the model from the pre-trained one significantly accelerates convergence compared to training from scratch. Specifically, to match the loss of the GLM-adapted MLM model, GLM from scratch requires consuming 110B tokens (2.2$\times$ speedup). Analogically, to match the loss of the MLM-adapted GLM model, MLM from scratch requires consuming 100B tokens (2$\times$ speedup).

These empirical experiments confirm that modeling the inherent protein data distributions is not limited to specific training patterns. This finding narrows the gap between autoencoding PLMs, such as ESM [7], and autoregressive PLMs like ProGen2 [13], providing empirical evidence that supports the efficacy of the xTrimoPGLM training pipeline.

3.5 The Train Stability of Unified Training

Training stability is the crucial factor accounting for the successful training of 100B-scale large language models [20, 38, 43, 44]. Given a fixed computing budget, it is essential to balance efficiency and stability with respect to floating-point (FP) formats. Lower-precision FP formats such as 16-bit precision (FP16) can enhance computational efficiency but are susceptible to overflow and underflow errors, potentially leading to catastrophic collapses during training. xTrimoPGLM borrows ideas from the implementation of GLM-130B [38] which has already addressed many unstable training issues. However, xTrimoPGLM-100B still encounters catastrophic training collapses during the transition from the first to the second stage of training - an issue that is not observed in smaller-scale models (at the 10B-scale). That is, directly incorporating a fixed ratio of GLM loss into the pre-training can trigger these collapses, even with the addition of a mere 1% ratio of GLM loss at the beginning, as illustrated in Figure 5. To alleviate this issue, we propose the implementation of a smooth transition strategy.

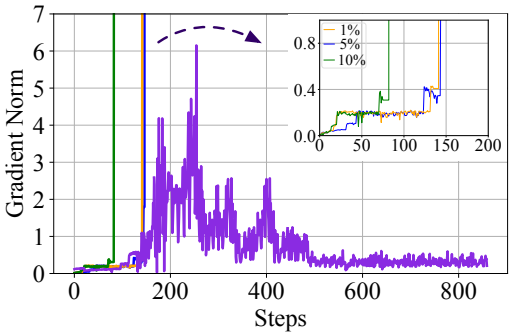


Figure 5: Trials on different strategies for transition from Stage-1 to Stage-2. Directly incorporating a fixed ratio of GLM loss into the pre-training triggers training collapses represented by the abnormal "spike" of gradient norm [38]. In contrast, the Smooth Transition strategy (Purple) that gradually improves the ratio of GLM loss to reach the expected amount makes the successful transition.

Smooth Transition (ST). Instead of directly introducing a fixed ratio of GLM loss into the training process, our empirical investigations suggest a smooth transition strategy divided into two phases. In the first phase, our primary goal is to gradually improve the ratio of GLM loss to reach the expected amount. Specifically, starting from 0, we incrementally increase the GLM loss ratio to reach the target value R in K steps using linear growth. Consequently, the GLM loss ratio R_k at the current step k is determined by the formula $R_k = \frac{k \times R}{K}$. It is worth noting that the learning rate should be kept exceptionally low during this phase. In our practical application, we set K = 1000 and learning_rate = 1e-7. Upon completion of the transition, the learning rate can be gradually increased back to its original level following the pre-defined pre-training script.

In fact, the final xTrimoPGLM-100B training run only experiences the loss divergence case at the transition stage, though it fails numerous times due to hardware failures.

4 xTrimoPGLM-100B: Mapping the Protein Universe

The goal of xTrimoPGLM-100B is rooted in a compelling vision: harnessing the capabilities of the largest-ever pre-trained protein language model to illuminate the vast *Protein Universe*. To achieve this, we enrich the existing pre-trained data with an abundance of metagenomic data, followed by a rigorous data integration process. The details of pre-training data and setup are demonstrated in 4.1 and 4.2 respectively. Through an extensive review of the current literature, we gather a variety of datasets. Subsequently, we conduct a comprehensive evaluation of our model using these datasets. We demonstrate the complexities of downstream task data 4.3, along with our approaches to downstream supervised training 4.4. We conduct a comprehensive analysis that offers fair performance comparisons between xTrimoPGLM-100B and ESM2, the current SOTA and the largest protein pre-training model [6], on downstream tasks. This comparison highlights the potential of xTrimoPGLM-100B in advancing future protein research and applications.

4.1 Pre-training Datasets

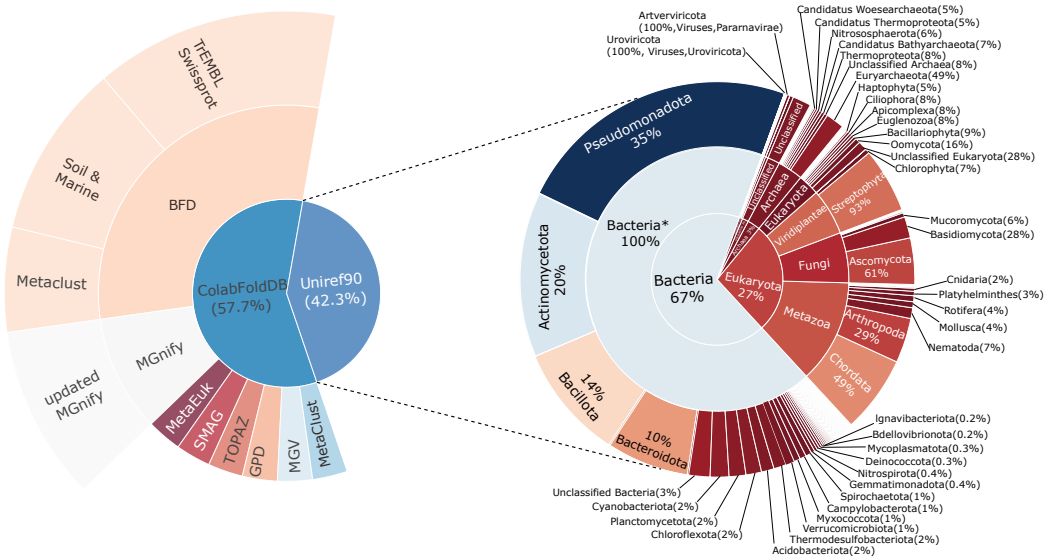


Figure 6: The pre-training dataset of xTrimoPGLM-100B: The left panel illustrates the composition of the dataset used for pre-training the model. The right panel depicts the distribution of taxonomic categories of Uniref90, visualized as concentric circles representing the levels of superkingdom, kingdom, and phylum from innermost to outermost. The innermost circle represents four superkingdoms: Bacteria (67%), Archaea (3%), Eukarya (27%), and Viruses (1%), with 2% of sequences labeled as unclassified. The middle circle encompasses 17 classified kingdoms, including an unclassified bacteria category, denoted as "bacteria*". The outermost circle denotes the phylum level, marking only those labels with counts over 200,000. In total, Uniref90 includes 273 known phyla.

The pre-training dataset of xTrimoPGLM-100B is curated from two extensive data repositories: Uniref90 [5] and ColAbFoldDB [45] [6]. The initial contributions from Uniref90 and ColAbFoldDB encompasse approximately 153M and 950M (210M representatives plus 740M members) entries, respectively.

Uniref, a cluster from UniProt, is broadly acknowledged as a high-quality protein dataset often utilized in pre-training PLMs such as ESM [7] and ProtTrans [8]. UniRef90 clusters are generated from the UniRef100 seed sequences with a 90% sequence identity threshold using the MMseqs2 [7] algorithm, as depicted in Figure 6. Except for unclassified entries, the dataset encapsulates all 4 NCBI taxonomic classifications of biological superkingdoms including Archaea, Bacteria, Eukaryotes, and Viruses. Additionally, it exhibits full coverage of the kingdom and phylum levels classifications, with spanning 17 kingdoms, and 273 phyla. This comprehensive representation across multiple taxonomic levels demonstrates the rich biodiversity encapsulated within the Uniref90 dataset and affirms its value for wide-ranging biological investigations. Protein sequences that are published prior to January 1, 2023, are incorporated into the training set. Given its robustness and reliability, our training process also substantially prioritizes this dataset.

ColAbFoldDB is established through an amalgamation of various metagenomic databases including BFD [8], MGnify [46], SMAG (eukaryotes) [47], MetaEuk (eukaryotes) [48], TOPAZ (eukaryotes) [49], MGV (DNA viruses) [50], GPD (bacteriophages) [51], and an updated version of the MetaClust [52] dataset. Built upon the foundation of UniProtKB, ColAbFoldDB is substantially augmented with a large corpus of metagenomic sequences derived from diverse environmental niches. Metagenomic data introduces a new level of diversity to the database, encompassing numerous environmental niches ranging from the human gut to marine ecosystems. This offers unparalleled opportunities

[5] https://www.uniprot.org/help/downloads, the Uniref90 version preceding December 2022 is downloaded
[6] https://colAbFold.mmseqs.com
[7] https://github.com/soedinglab/MMseqs2
[8] https://bfd.mmseqs.com

for the discovery of novel proteins. To comprehensively map the entirety of protein sources in the biological world, the pre-training dataset has been expanded by incorporating protein sequences sourced from ColAbFoldDB in addition to those from the Uniref90 dataset.

Training Set. The entirety of sequences in the ColabFoldDB comprises about 950M sequences. Due to an approximate count of 125M duplicate sequences and the need for diverse training data, only representative sequences are employed, reducing the dataset to an estimated 210M sequences. Subsequently, we cross-reference the ColAbFoldDB with the Uniref90 database, eliminating 1.1M sequences that show a 100% match, thereby avoiding redundancy. Then, we conduct a composition analysis of each remaining sequence, excluding any that exhibit an individual amino acid composition exceeding 80% as this may indicate an anomaly or bias in the data. These steps leave us a more representative subset of around 200M sequences. Then, we combine the two refined datasets (i.e., Uniref90 and ColAbFoldDB), yielding a collection of 360M unique sequences, equivalent to roughly 100B tokens, as depicted in Figure 6 (left). During training, to capitalize on the high-quality data, we assign a greater weight to the Uniref90 data, resulting in a sampling ratio of approximately 60%. This strategy effectively doubles the Uniref90 data contribution, enhancing our model's capacity to fine-tune based on superior-quality data. For the training data distribution, see Appendix A.

Validation Set. Sequences from UniProt released between January 1 and March 30, 2023, are utilized as the validation datasets. The 18M sequence increment is applied as a query to scrutinize the target database (i.e., Uniref50 and the training dataset), and sequences over 90% similarity are eliminated from the query set (`mmseqs easy-search -db-load-mode 2 -min-seq-id 0.9 -alignment-mode 3 -max-seqs 300 -s 4 -c 0.8 -cov-mode`). The remaining after filtering is used as the validation set.

4.2 xTrimoPGLM-100B Configurations

Here we introduce the implementation details of pre-training the xTrimoPGLM-100B model. Since the xTrimoPGLM-100B borrows the idea from the GLM-130B [38] framework, we only emphasize the specific hyper-parameter of xTrimoPGLM-100B. For more discussion and design choices please refer to GLM-130B [38].

xTrimoPGLM-100B is trained on a cluster of 96 DGX-A100 GPU (8×40G) servers in FP16 precision from January 18 to June 30, 2023. During this time, xTrimoPGLM-100B has consumed 1 trillion tokens from the dataset consisting of Uniref90 and ColAbFoldDB. As of the current date, xTrimoPGLM-100B continues its pre-training process to pass through as many tokens as possible, as a recent study [53] suggests that most existing LLMs are largely under-trained. We adopt 3D parallel strategy with the 4-way tensor parallelism [54], 8-way pipeline parallelism [55], and 24-way data parallelism [56] based on DeepSpeed [57]. The model owns 72 transformer layers, 80 attention heads, and 10,240 embedding dims with 31,744 feed-forward embedding dims using GeGLU [58]. We adopt the Post-LN initialized with the DeepNorm [59]. We follow the mixed-precision [60] strategy (Apex O2), i.e., FP16 for forwards and backwards and FP32 for optimizer states and master weights, to reduce the GPU memory usage and improve training efficiency. We also adopt the Embedding Layer Gradient Shrink (EGS) strategy [38, 61] with $\alpha = 0.1$ to stabilize the xTrimoPGLM-100B training. We warm-up the batch size from 240 to 4224 over the first 2.5% samples. We use AdamW [62] as our optimizer with β_1 and β_2 set to 0.9 and 0.95, and a weight decay value of 0.1. We warm up the learning rate from 10^{-7} to 4×10^{-5} over the first 3.0% samples, then decay it by a 10 $\times$ cosine schedule to the minimum learning 4×10^{-6}. We use a dropout rate of 0.1 and clip gradients using a clipping value of 1.0. Each sample contains a fixed sequence length of 2,048 (We concatenate all protein sequences with a separator into a single document, and sample protein sequences from this document in such a way that there is virtually no padding during pre-training.). To adapt the model to the different lengths of proteins in the downstream tasks, we adopt the mix-length pre-training strategy with four different context windows of 256, 512, 1,024, and 2,048. Taking, 512, for example, we concatenate four samples together to cater for the 2,048-sequence-length. The ratio of different context lengths is $[\#256 : \#512 : \#1,024 : \#2,048 = 0.1 : 0.4 : 0.4 : 0.1]$. We implement the two-dimensional RoPE from its author blog [9] as our position embedding. For the tokenization of the protein data, we use the residue-level tokenizer which is adopted in several PLMs [7, 13, 25, 49]. Except for the basic amino acid types, we add special tokens `[MASK]`, `[sMASK]`, and `[gMASK]` for

[9] https://kexue.fm/archives/8397

model prediction. We also add special tokens <sop>, <eop>, <eos> for sequence separation (Cf. Table 10 for the full configurations).

4.3 Downstream Tasks Datasets & Evaluation Metrics

Table 2: Summary information for 15 benchmarked downstream tasks, including task category, evaluation metric, data size, and performance. In this table, 'Struc.' represents protein structure, 'Dev.' represents protein developability, 'Inter.' represents protein interactions, 'Func.' represents protein functions, and 'Perf.' denotes performance (%). The ♣ denotes the results that we produce or reproduce, while the ♦ represents direct citations from original papers with the same split train/valid/test sets. For any dataset without established benchmarks, we employ the results of our own ESM2-15B with LoRA fine-tuning.

Type	Task	Metric	Train	Valid	Test	Prev.Method	Perf.	xT-100B Perf.
Struc.	Cont. P.	Top L/5 ACC	12K	1.5K	1.5K	ESM2-15B [7]	92.19♣	**93.32**
	Fold. P.	12K-cls ACC	12.3K	736	3.2K	Ankh_L [27]	61.10♦	**75.61**
	Sec. Struc. P.	3-cls ACC	11K	-	39	Ankh_L [27]	80.70♦	75.33
Dev.	Sol. P.	2-cls ACC	62.4K	-	6.9K	ESM2-15B [7]	76.49♣	**79.45**
	Stab. P.	SRCC	53.6K	2.5K	12.8K	ESM2-15B [7]	80.75♣	**84.21**
	Temp. Stab.	MCC	283K	63K	73.2K	TemStaPro [63]	83.80♦	**94.22**
	Opt. Temp.	SRCC	1.7K	-	190	DeepET [64]	62.40 ♦	**73.96**
Inter.	Metal B.	2-cls ACC	6K	-	1.3K	ESM2-15B [7]	79.35♣	**82.78**
	Enzyme Eff.	PCC	13.5K	1.7K	1.7K	DLKcat [65]	71.00♦	**74.79**
	Pept.-HLA Aff.	AUC	575K	144K	171K	CcBHLA [66]	95.00♦	**96.68**
	TCR-pMHC Aff.	AUC	19.5K	-	4.5K	epiTCR [67]	92.50♦	**95.10**
Func.	Antib. Res.	19-cls ACC	2K	-	1.3K	ESM2-15B [7]	98.28♣	**98.38**
	Fluor. P.	SRCC	21.4K	5.4K	27.2K	Ankh_L [27]	62.00 ♦	**66.00**
	Fitness P.	SRCC	6.3K	699	1.7K	Ankh_L [27]	84.00 ♦	**96.10**
	Loc. P.	10-cls ACC	6.6K	-	1.8K	Ankh_L [27]	83.20♦	81.60

To systematically evaluate xTrimoPGLM-100B, we have benchmarked 15 downstream protein-related tasks across multiple domains. Table 2 shows a comprehensive overview of our benchmark performance on all evaluated tasks, divided into four main categories: protein structure, protein developability, protein interactions, and protein functions. The table elucidates these tasks along with the latest SOTA methodologies employed, their respective performances, and the achievements attained by our proposed xTrimoPGLM-100B model. We emphasize that this comparison is primarily from a task-based perspective, where xTrimoPGLM is combined with fine-tuning techniques to achieve the results. The results reveal that xTrimoPGLM-100B significantly outperforms current SOTA approaches in most protein-related tasks, hence catalyzing advancements in this field. For a thorough comparison involving both probing and fine-tuning techniques with the ESM2, we direct readers to Section 4.4. Next, we individually delve into these subtasks, elaborating on the corresponding task definitions, dataset processing, evaluation metrics, and other relevant details.

Contact Map. Contact map prediction (Cont. P.) aims to determine whether two residues, *i* and *j*, are in contact or not, based on their distance with a certain threshold (<8Å). This task is an important part of the early Alphafold version [68] for structural prediction. The trRosetta dataset [69] is employed and split into 12,041 training samples, 1,505 validation samples, and 1,505 test samples for this task. The evaluation metric used is the Top L/5 accuracy, considering residue pairs with a separation length greater than 6 and a sequence length cutoff of 512.

Fold Classification. Fold class prediction (Fold. P.) is a scientific classification task that assigns protein sequences to one of 1,195 known folds. The dataset employed for this task is based on SCOP 1.75 [70], a release from 2009, and has been widely adopted by DeepSF [71] and Ankh [27]. The primary application of this task lies in the identification of novel remote homologs among proteins of interest, such as emerging antibiotic-resistant genes and industrial enzymes [72]. The study of protein fold holds great significance in fields like proteomics and structural biology, as it facilitates the analysis of folding patterns, leading to the discovery of remote homologies and advancements in disease research [73].

Secondary Structure. The study of a protein's secondary structure (Sec. Struc. P.) forms a fundamental cornerstone in understanding its biological function. This secondary structure, comprising helices, strands, and various turns, bestows the protein with a specific three-dimensional configuration, which is critical for the formation of its tertiary structure. In the context of this work, a given protein sequence is classified into three distinct categories, each representing a different structural element: Helix (H), Strand (E), and Coil (C). The datasets applied in this study are originally published by NetSurfP-2.0 [74] and have also been utilized by Ankh [27]. The datasets employed for testing in our investigation are specifically assembled from the Critical Assessment of Protein Structure Prediction (CASP) editions 12 and 14, which contain 18 and 21 samples. The result we reported is an average of these two datasets.

Solubility. This task (Sol. P.) involves a binary classification of a heterogenous set of proteins, assessing them as either soluble or insoluble. The solubility metric is a crucial design parameter in ensuring protein efficacy, with particular relevance in the pharmaceutical domain. We've adhered to the same dataset division as is employed in the development of DeepSol [75]. Within this framework, any protein exhibiting a sequence identity of 30% or greater to any protein within the test subset is eliminated from both the training and evaluation subsets, ensuring robust and unbiased evaluation.

Stability. The task (Stab. P.) is to predict the concentration of protease at which a protein can retain its folded state. Protease, being integral to numerous biological processes, bears significant relevance and a profound comprehension of protein stability during protease interaction can offer immense value, especially in the creation of novel therapeutics. The dataset applied in this task is initially sourced from Rocklin et al [76] and subsequently collected within the Task Assessing Protein Embeddings (TAPE) [77]. In this regression-based task, we employ the SpeaRman Correlation Coefficient (SRCC) as the evaluation metric to measure the prediction consistency.

Temperpature Stability. The accurate prediction of protein thermal stability (Temp. Stab.) has far-reaching implications in both academic and industrial spheres. This task primarily aims to predict a protein's capacity to preserve its structural stability under a temperature condition of 65 degrees Celsius. We employed the same database and dataset division strategy used in the development of TemStaPro [63]. The performance of our prediction is evaluated and reported using the Matthews Correlation Coefficient (MCC) score.

Optimal Temperature. Grasping the catalytic activity of enzymes is pivotal for industrial enzyme design, particularly in predicting the optimal temperature (Opt. Temp.) for a given enzyme's catalytic effect. The dataset utilized for this task is primarily procured by DeepET [64], a recent advancement in the field that uses deep learning techniques to understand enzyme thermal adaptation. To quantify the relationship between these variables, we use the SRCC.

Metal Ion Binding. Metal ion binding (Metal B.) sites within proteins play a crucial role across a spectrum of processes, spanning from physiological to pathological, toxicological, pharmaceutical, and diagnostic. Consequently, the development of precise and efficient methods to identify and characterize these metal ion binding sites in proteins has become an imperative and intricate task for bioinformatics and structural biology. This task involves a binary classification challenge aimed at predicting the existence of metal-ion binding site(s) on a given protein sequence. We employ data [78] curated from the Protein Data Bank (PDB).

Enzyme Catalytic Efficiency. This task (Enzyme Eff.) is focused on predicting kcat values, which are enzymatic turnover numbers denoting the maximum chemical conversion rate of a reaction, for metabolic enzymes originating from any organism. These predictions are based on substrate structures and protein sequences. The underlying importance of this task lies in its potential to yield high-throughput and accurate kcat predictions applicable to any organism or enzyme. Such capabilities are crucial for advancing our understanding of cellular metabolism and physiology. The data, sourced from a variety of repositories including BRENDA, SABIO-RK, KEGG, UniProt, and MetaCyc, are curated by Li et al [65].

Peptide-HLA/MHC Affinity. The human leukocyte antigen (HLA) gene encodes major histocompatibility complex (MHC) proteins, which can bind to peptide fragments and be presented to the cell surface for subsequent T cell receptors (TCRs) recognition. Accurately predicting the interaction between peptide sequence and HLA molecule will boost the understanding of immune responses,

antigen presentation, and designing therapeutic interventions such as peptide-based vaccines or immunotherapies. The classification task aims to predict whether a given paired peptide and HLA sequence can bind or not. The modeling data is from Wu et al [66].

TCR-pMHC Affinity. The interaction between T cell receptors (TCRs) and peptide-major histocompatibility complexes (pMHCs) plays a crucial role in the recognition and activation of T cells in the immune system. TCRs are cell surface receptors found on T cells, and pMHCs are complexes formed by peptides derived from antigens bound to major histocompatibility complexes (MHCs) on the surface of antigen-presenting cells. The classification task is to predict whether a given paired TCR sequence and peptide can bind or not. The evaluated data is major from VDJdb, processed and curated from Pham et al [67].

Antibiotic Resistance. Antibiotic resistance (Antib. Res.) refers to the ability of bacteria and other microorganisms to resist the effects of an antibiotic to which they are once sensitive. In this task (Antib. Res.), an input protein sequence is categorized according to which of 19 antibiotics it is resistant to. Thus, the scope of antibiotic drug development and research could be explored as an understanding in this topic is accumulated. The Dataset used in this task is curated by CARD [79].

Fluorescence. The Fluorescence Prediction (Fluor. P.) task [35] focuses on predicting the fluorescence intensity of green fluorescent protein mutants, a crucial function in biology that allows researchers to infer the presence of proteins within cell lines and living organisms. This regression task utilizes training and evaluation datasets that feature mutants with three or fewer mutations, contrasting the testing dataset, which comprises mutants with four or more mutations. The partitioning of the datasets mirrors the splitting method implemented in the TAPE [77]. The quality of these predictions is assessed using the Spearman score as the primary evaluation metric.

Fitness. The task of Fitness Prediction (Fitness P.) is dedicated to anticipating the fitness landscape of the GB1 domain, a process that plays a pivotal role in understanding and enhancing the binding affinity and stability of this domain. As a prevalent protein interaction domain, GB1 finds wide usage in diverse applications such as protein purification, biosensors, and drug delivery [80, 81]. This task is configured as a regression problem, where the goal is to predict the fitness of GB1 binding following mutations at four specific positions. The data for this task is sourced from the FLIP database [82]. Predictive efficacy is assessed using the Spearman score as the principal evaluation metric.

Localization. The task of Protein Subcellular Localization Prediction (Loc. P.) bears substantial relevance in bioinformatics, owing to its contributions to proteomics research and its potential to augment our comprehension of protein function and disease mechanisms [83]. In this task, the input to the model is an amino acid sequence of a protein, which is transformed into an output comprising a probability distribution over 10 unique subcellular localization categories. The dataset applied for this task is derived from Uniprot, meticulously curated by Armenteros et al [84].

4.4 Downstream Performance

We compare three distinct large language models for proteins: ESM2-150M, ESM2-15B [7], and xTrimoPGLM-100B. To evaluate the effectiveness of these models' representations, we perform both feature-based probing and fine-tuning evaluations. Such comparisons in both model architectures and adaptation techniques offer a comprehensive understanding of the model's performance across a range of scenarios. We systematically document the variations in performance metrics and draw insights into the respective strengths and weaknesses of each model under consideration.

- **MLP Probing.** We utilize a trainable multilayer perceptron (MLP) model as a probe to examine the information encoded in the pre-trained representations. This method offers a straightforward and efficient way to identify what kind of protein information, is captured by the underlying models. It is crucial to note that during the probing process, the parameters of pre-trained PLMs are kept frozen, and only the MLP is trained. For the pair comparison, the embeddings from all models are projected into 128 dimensions followed by ReLU activation before passing to the next layer of MLP (except for the Fold Prediction task which is directly projected to the target classes without activation functions), weights of the MLP are initialized with Kaiming initialization, and the used optimizer is Adam.

- **Fine-tuning with LoRA.** Given the limitations imposed by GPU memory, full-scale fine-tuning becomes impractical for models with 100 billion parameters. Consequently, we resort to parameter-efficient adaptation techniques, such as the Low-Rank Adaptation (LoRA) [85]. LoRA, a widely employed method, freezes the weights of the pre-trained model and incorporates trainable low-rank matrices into each layer of the transformer architecture. This approach substantially reduces the number of trainable parameters for downstream tasks while preserving the flexibility of learned representations. The architecture and other settings for the fine-tuning model remain analogous to those utilized in the MLP probing, with only the parameters W_q, W_k, W_v, W_o in the transformer being fine-tuned. Detailed information about the hyperparameter settings can be found in Table 10.

Results. Figure 7 illustrates the performance across all benchmark tasks, with distinct colors signifying various evaluation strategies and different shapes denoting different models. We use the relatively smaller ESM2-150M model as an indicator to understand the degree of difficulty associated with respective downstream protein-related tasks. The performance distribution highlights the inherent relationships between the complexity of tasks and the advantages brought by the scale of the model. For the complex tasks, the large models (xTrimoPGLM-100B and ESM2-15B) perform significantly better than the small model (ESM2-150M), illustrating the requirement for a more powerful and complex model to address these tasks effectively. For instance, the larger models consistently surpass the smaller ones by a substantial margin in most of the tasks, such as Contact Map prediction (under Protein Structure), Fluorescence (under Protein Function), Metal Binding (under Protein Interaction), and Stability (under Protein Development). In contrast, for simpler tasks (e.g., Antibiotic Resistance under Protein Function), the difference in performance between the large and small models is marginal. This pattern demonstrates that the larger models are better equipped to capture intrinsic latent features of protein sequences. Hence, as the model sizes scale up, they contribute to significant enhancements in performance especially for the complex tasks, which corresponds to the emergent abilities of large language models [1].

In the quest to enhance parameter efficiency during fine-tuning, the incorporation of Low-Rank Adaption (LoRA) has resulted in consistent improvements in overall task performance compared to the MLP probing method. MLP probing, employed as a static embedding method, circumscribes the capacity of pre-trained models. In contrast, LoRA enables the pre-trained model to extract and exploit pertinent features. Furthermore, there is no significant increase in the number of trainable parameters when LoRA is applied. These advantages have already been proven across a wide range of NLP tasks. In the following section, we provide a detailed analysis of the four types of downstream tasks.

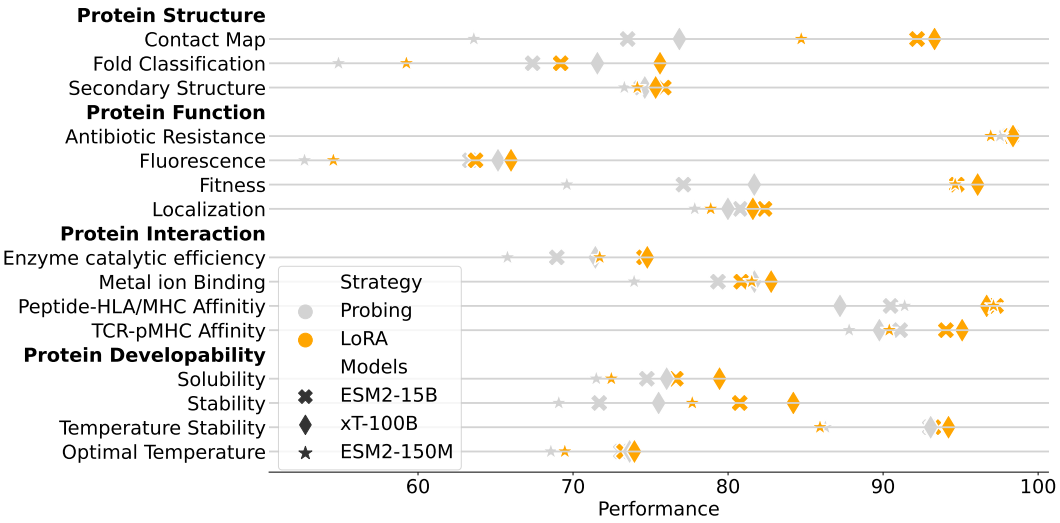


Figure 7: Visualization of model performance across all benchmark tasks. Details please refer to Appendix B.

Protein Structure. For protein structure-related tasks, we collect three datasets, including residual (Secondary Structure), pairing (Contact Map) and ensemble structure level (Fold Classification). It is evident that the large pre-trained models (xTrimoPGLM-100B and ESM2-15B) bring substantial improvements, as does the application of LoRA. Concretely, the accuracy of the xTrimoPGLM-100B is improved from 76.86 to 93.32 when LoRA is applied. This implies the potential of incorporating LoRA into protein structure prediction models. More importantly, the contact map prediction task is intricately interconnected with the task of predicting the three-dimensional structure of proteins, as precise residue contact map prediction can significantly expedite the process. Existing structure prediction models may not exhaustively harness the non-linear transfer capabilities intrinsic to the pre-trained model. For instance, a popular model, ESMFold [7], freezes ESM2 and appends a folding trunk (a transformer encoder with 48 layers) as a representation learner. Conversely, the LoRA technique, by enabling fine-tuning, pioneers a promising trajectory for exploiting pre-training of large language models to augment the precision of 3D protein structure prediction.

Protein Function. Several tasks have been established to experimentally assess the consequences of a synthesized protein sequence, with each observation tied to a specific biological function. Accordingly, we evaluate four such tasks within this category. For instance, the antibiotic resistance task predicts whether a protein sequence is sensitive to a specific bacteria. The results (Figure 7) manifest the consistently superior performance of larger models in comparison to smaller counterparts, such as xTrimoPGLM-100B and ESM2-15B vs ESM2-150M. The tendency is evidenced by a notably higher Spearman correlation margin on the fitness task and 10-class classification accuracy on localization prediction. Therefore, we believe that larger PLMs could be deployed in the frontier of biomedical research and pharmaceutical applications.

Protein Interaction. Proteins tend to interact with different types of molecules to trigger subsequent bioactivity, including the endogenous catalyzing substrate, metal ions on the cell surface, and exogenous antigen peptides. Here we focus on four tasks related to protein interactions. Specifically, for enzyme catalytic efficiency and metal ion binding prediction, only the protein sequence is utilized. For immunity-based peptide-MHC and TCR-pMHC interaction prediction, we simply concatenate two sequences with special token `<eos>` as model input. The results show that LoRA fine-tuning consistently outperforms the probing method, extending its advantage to sequence pair cases where the task pattern has not been seen during the pre-training stage. However, we find that the margin between xTrimoPGLM-100B and ESM models tends to be small in peptide-MHC and TCR-pMHC interaction tasks. This may be due to the relative simplicity of the task, as the baseline model already achieves high performance (AUC > 0.9).

Protein Developability. The biophysical condition surrounding protein molecules determines whether they can work normally. Here, we select three related tasks—solubility, stability, and sensitivity—as representatives for evaluation. The results indicate that xTrimoPGLM-100B significantly outperforms ESM models on solubility and stability tasks, even though the two tasks are relatively difficult (ESM-150M performance is around 70). However, the improvement in temperature-related tasks remains marginal. We find a similar performance trend for the two datasets: xTrimoPGLM-100B is slightly better than ESM. Since both ESM and xTrimoPGLM-100B achieve high performance (with MCC > 0.93) in the Temperature Stability task, we could hypothesize that this task may present some challenges for prediction. On the other hand, the Optimal Temperature task has the smallest training sample size (approximately 1.7k) among all benchmark tasks. Therefore, it could potentially constrain the achievable performance of models.

Overall, xTrimoPGLM-100B outperforms ESM2-15B on 12 of 15 tasks. The results also reveal the scaling law in the performance of downstream tasks with supervised fine-tuning (in Figure 1), i.e., the performance seems to have a strong correlation with the model scale. This suggests that scaling models could potentially be a simple yet effective way to enhance the model performance on a wide range of protein-related tasks, although other methods [27] attempt to find a path towards building data-efficient, cost-reduced, and knowledge-guided PLMs without resorting to large language models. These empirical observations offer clear guidance for future research endeavors focused on model advancement.

5 OAS Fine-tuning for Antibody

In this section, we adopt the xTrimoPGLM framework to explore a special family of proteins: antibodies. Antibodies are ubiquitous yet vital proteins that are generated by the adaptive immune system to bind and neutralize foreign pathogens such as SARS-CoV-2 [86, 87]. It functions via binding to antigens on the pathogen, recognizing it, and finally inhibiting it. Generally, antibodies, composed of two identical heavy chains and two identical light chains, form a large Y-shaped structure. The specificity of antibody binding is determined by CDRs at the tips of the Y-shaped proteins (VH and VL). The estimated number of potential human antibodies ranges from 10^{13} to 10^{16}, signifying the incredible diversity of the immune response. Their specificity, combined with this abundance, renders antibodies invaluable for the development of targeted therapies.

We do not directly fine-tune on xTrimoPGLM-100B, mainly due to limitations in computational budgets and considering the inherent lack of diversity in OAS antibody data, most of which are of similar length and have similar framework areas. Hence, we first pre-train xTrimoPGLM-1B model on the general protein dataset 4.1 This process undertakes 500B tokens. Since antibodies represent a distinct subset of general proteins, then we finetune the model using the OAS dataset[10], comprising 1 billion antibody sequences. Considering that the CDRs are the most important parts of an antibody, we randomly mask one or two whole CDRs for 40% of samples with `[sMASK]`. A further 40% of the samples undergo a random span masking process, while the remaining 20% are subjected to the MLM objective. We exclude the `[gMASK]` loss from consideration, as it is not required for downstream antibody-related tasks involving long-text generation. When fine-tuning the xTrimoPGLM-Ab-1B model on OAS data, we decrease the maximum learning rate to 2e-5 and make the model consume 500B tokens with 2,048 samples per batch and the 1,024 input length per sample. It takes about 168 hours to use 128 Nvidia A100 80G GPU cards with mixed precision training. We carry out evaluations on two critical undertakings within the realm of drug discovery including assessing the zero-shot naturalness of antibodies and predicting the structural conformation of antibodies.

5.1 Zero-shot Naturalness

Table 3: Performance of different models in zero-shot naturalness datasets. Since xTrimoPGLM-Ab-1B can not only be considered as an auto-regressive mode but also an auto-encoder model, we calculate both PPL and PPPL of them. xTrimoPGLM-Ab-1B-GLM or -MLM means that the model is finetuned with the supervision of GLM or MLM from the base xTrimoPGLM-Ab-1B model.

Model	DATASET 1			DATASET 2		
	H Chain	L Chain	Pair	H Chain	L Chain	Pair
Iglm [89]	0.698	0.651	0.683	0.703	0.594	0.665
AbLang [90]	0.655	0.497	0.613	0.713	0.671	0.679
ESM2-15B [36]	0.682	0.552	0.686	0.716	0.510	0.626
AntiBERTy [91]	0.763	0.549	0.699	0.723	0.678	0.679
Progen2-oas [13]	0.703	**0.734**	0.748	0.701	0.565	0.644
xTrimoPGLM-Ab-1B PPL	0.745	0.696	**0.756**	0.702	0.688	0.704
xTrimoPGLM-Ab-1B PPPL	0.754	0.683	0.750	0.741	0.668	0.700
xTrimoPGLM-Ab-1B-GLM PPL	**0.763**	0.676	0.742	0.703	0.685	**0.724**
xTrimoPGLM-Ab-1B-MLM PPPL	0.733	0.682	0.746	**0.766**	**0.704**	0.722
Ablation Study						
xTrimoPGLM-Ab-1B-GLM-CDR PPL	0.652	0.700	0.689	0.699	0.647	0.671
xTrimoPGLM-Ab-1B-GLM-Random PPL	0.736	0.666	0.725	0.715	0.640	0.708

In protein design and expression, a crucial step involves filtering out proteins with low expression while retaining those with high naturalness. Perplexity (PPL) given by a protein language model can be used to predict the naturalness of proteins [13, 92, 93]. For the GLM objective, PPL is calculated by:

$$\mathrm{PPL}(\mathbf{x}) = \exp\left(-\sum_{i=1}^{l} \log P_{\text{model}}(x_i | x_{\hat{i}}, x_i = \texttt{[sMASK]})\right), \tag{4}$$

[10]Observed Antibody Space (OAS) [28] data. Following the paper, we filter OAS data with IMGT schema [88] and therefore get 678m sequences without disorder and incompletion.

where $P_{\text{model}}(x_i|x_{\hat{i}}, x_i = \texttt{[sMASK]})$ is the probability of the i-th amino acid, denoted by x_i, as predicted by the model. Here, the context $x_{\hat{i}}$ is given, with a [sMASK] token in the i-th position. Note that $x_{\hat{i}}$ represents all tokens excluding the i-th token. For the MLM objective, pseudo-perplexity [94] is usually utilized as a substitute for perplexity since perplexity is only computed via the auto-regressive model. Pseudo-perplexity (PPPL) of a protein is defined as

$$\text{PPPL}(\mathbf{x}) = \exp\left(-\sum_{i=1}^{l} \log P_{\text{model}}(x_i|x_{\hat{i}}, x_i = \texttt{[MASK]})\right), \tag{5}$$

where $P_{\text{model}}(x_i|x_{\hat{i}}, x_i = \texttt{[MASK]})$ represents the probability of the i-th amino acid x_i predicted by the model given the context $x_{\hat{i}}$ with a [MASK] in i-th position.

Datasets. To assess the performance of various models, we construct two datasets derived from protein expression experiments conducted in a wet laboratory. Any samples that yield less than 10 mg/L of the purified proteins in the supernatant liquid are categorized as unexpressed, whereas those yielding more than 10 mg/L are deemed as successfully synthesized. The first dataset (Dataset 1) comprises 601 antibody sequences, derived from experiments conducted on CHO and HEK293 cells. These sequences include 114 proteins from humans, 90 from mice, 1 from rhesus, and 396 from undefined species (not directly sourced from specific species). Of these, 516 are successfully expressed. The second dataset (Dataset 2) – sourced from HEK293 cells – contains 98 human antibody sequences targeting a specific antigen, of which 90 are successfully expressed.

Metrics. Each sample comprises both a heavy chain and a light chain. For models that do not incorporate chain types, we calculate the perplexity of each chain individually, then multiply these values to obtain the overall perplexity for the pair. For models incorporating chain types, we concatenate both chains in the following format: [human] [heavy chain] sequence1<eos> [human] [light chain] sequence2<eos>, where [human] is a special token to indicate the species of sequences, [heavy chain] and [light chain] are two tokens to represent the types of sequences, <eos> means the end of sequences. We use the area under the receiver operating characteristic (ROC) curve (AUC) as a measure to evaluate the models' ability to predict protein naturalness. Notably, Iglm [89] and ProGen2 [13] are auto-regressive models, while AbLang [90], ESM2 [36], and AntiBERTy [91] are auto-encoder models. Thus we evaluate Iglm and ProGen2 using PPL, while the remaining models are tested using PPPL. As xTrimoPGLM-Ab-1B can function as either an auto-regressive or an auto-encoder model, we employ both PPL and PPPL to calculate its AUC score.

Results. The results are shown in Table 3. Among these, xTrimoPGLM-Ab-1B surpasses other baselines in two datasets. Moreover, we further fine-tune xTrimoPGLM-Ab-1B with the GLM objective with 30 billion tokens to gain xTrimoPGLM-Ab-1B-GLM. Analogously, we fine-tune it with the MLM objective with 100 billion tokens to get xTrimoPGLM-Ab-1B-MLM. Since the consumed tokens (80% tokens) of the GLM objective is 4 times more than that (20% tokens) of the MLM objective in the pre-training stage, xTrimoPGLM-Ab-1B-MLM is fine-tuned with more tokens than xTrimoPGLM-Ab-1B-GLM for a relatively fair comparison. Consequently, xTrimoPGLM-Ab-1B-GLM and xTrimoPGLM-Ab-1B-MLM keep similar results on Dataset 1 with little difference of AUC on pair test, while they benefit from additional training on Dataset 2, as the AUC scores are improved by 0.02 consistently.

Ablation Study. To justify the contribution of different components, i.e, [sMASK] within random spans or [sMASK] with CDR regions, of the GLM objective, we train xTrimoPGLM-Ab-1B-GLM-CDR only with the CDR span task and xTrimoPGLM-Ab-1B-GLM-Random with the random span task, based on the pre-trained xTrimoPGLM-Ab-1B. xTrimoPGLM-Ab-1B-GLM (50% CDR span task and 50% random span task) outperforms these two models on Dataset 1 and Dataset 2. These distinctions highlight the importance of the combination of CDR span task and random span task.

5.2 Antibody structure prediction

In this section, our aim is to predict antibody structures based on their sequences. The study of protein structure assists in the design and modification of proteins, as well as in target identification and structural analysis for protein-based drug design. A popular method to predict protein structures is leveraging Multiple Sequence Alignment (MSA) and structure templates to encode sequences and

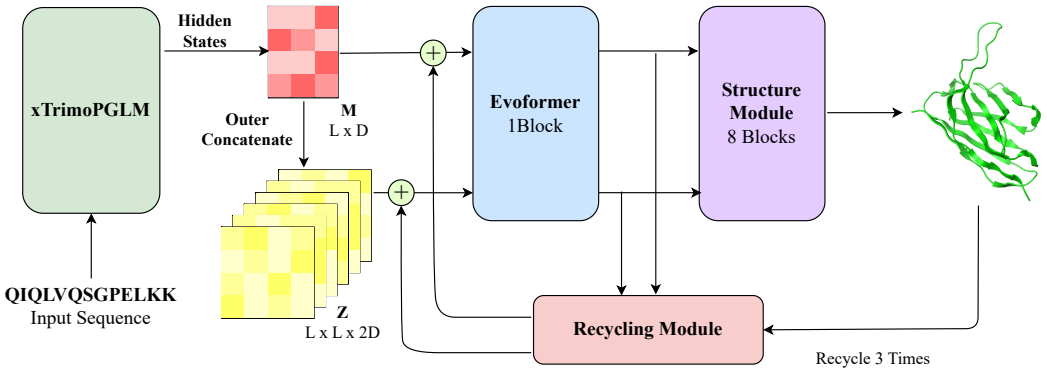


Figure 8: Architecture of xTrimoPGLM-AbFold for structure prediction. xTrimoPGLM-AbFold only leverages a single Evoformer block and does not need MSA and template search.

then using encoded matrices to generate structures. However, MSA requires significant computational resources and time. Given that xTrimoPGLM is trained using the MLM task, it is naturally suited to serve as an encoder for understanding tasks Therefore, we develop xTrimoPGLM-AbFold, which is based on xTrimoPGLM-Ab-1B, with the aim of predicting three-dimensional antibody structures directly from amino acid sequences. Our experiments encompass both single-chain structure prediction and complex structure prediction, i.e., the VH-VL complex.

Datasets & Metrics. The structure prediction dataset for single chains is derived from the RCSB Protein Data Bank (PDB) [95] prior to April 13, 2022, which consists of both amino acid sequences and structures. We collect all antibody data in PDB, which contains 19k antibody chains (VL or VH). Structural data with missing resolution values or resolutions greater than 9 Å are excluded to maintain quality. Additionally, sequences with an amino acid repetition rate exceeding 90% are filtered out. Finally, we obtain about 7.5k unique sequences (VL or VH chains). The training set consists of 7,234 sequences, and 350 sequences are left as the test set. The dataset for VH-VL complexes includes approximately 4.7k antibodies from PDB, which are released before January 2022. We select 68 samples as the test set, which are released between January 2022 and November 2022.

Root mean square deviation (RMSD) and TM-score [96] are used as evaluation metrics for both tasks. Another important metric DockQ [97] is involved in the structure prediction of complexes.

Model Architecture. Our principal hypothesis is that with an adequately proficient encoder, structure prediction models can accommodate complex structures using shallow Evoformer layers and structure modules. Therefore, compared with the current prevailing folding structures, such as ESMFold, AlphaFold2, we introduce the following modifications to xTrimoPGLM-AbFold: 1) We eliminate MSA and template search modules, as they offer minimal benefit for antibody folding in our pre-training and fine-tuning paradigm; 2) Unlike Alphafold2, which employs 48 blocks of Evoformer, and ESMfold, which utilizes 48 layers of folding trunk, we significantly **reduce the number of downstream folding blocks from 48 to 1**. The architecture of xTrimoPGLM-AbFold is depicted in Figure 8.

Training Settings. For single-chain structure prediction, we convert protein sequences of length L into the format of `[human]` `[chain type]` sequence`<eos>`, and feed it into the xTrimoPGLM-Ab-1B model to obtain the hidden representation $\mathbf{M}$ of the last layer. The information corresponding to `[human]`, `[chain type]` and `<eos>` are removed from $\mathbf{M}$, where $\mathbf{M} \in \mathbb{R}^{L \times D}$ and D is the size of the hidden dimension of the xTrimoPGLM-Ab-1B model. Then, we extend $\mathbf{M}$ along its L dimension in a pairwise manner to obtain a tensor $\mathbf{Z} \in \mathbb{R}^{L \times L \times 2D}$ (Figure 8). After that, $\mathbf{M}$ and $\mathbf{Z}$ are fed into a single-block Evoformer module for information fusion and then into the structure module for prediction of the angle and position of each residue. For the VH-VL complex, it should be noted that the input is converted into the format of vh_sequence`[linker]`vl_sequence, where the `[linker]` is composed of four groups of residue sequences, each of which is composed of four G residues and one S residue, just like GGGGSGGGGSGGGGSGGGGS.

Table 4: Structure prediction of VH and VL in antibodies. RMSD H1-3 means RMSD on CDR1-3 of heavy chains and RMSD L1-3 means RMSD on CDR1-3 of light chains.

Model	RMSD↓	TM-SCORE↑	HEAVY CHAIN RMSD↓			LIGHT CHAIN RMSD↓		
			H1	H2	H3	L1	L2	L3
AlphaFold2	1.225	0.951	1.254	1.091	2.826	0.89	0.723	1.329
OmegaFold	1.337	0.946	1.418	1.183	3.246	0.860	0.598	1.360
ESMFold	1.421	0.943	1.464	1.320	3.409	1.048	0.679	1.520
IgFold	1.261	0.945	1.324	1.126	2.998	0.948	0.589	1.318
xTrimoAbFold	1.089	0.958	1.176	0.912	2.472	0.811	**0.566**	1.038
xTrimoPGLM-AbFold	**0.9823** ±0.007	**0.961** ±0.001	**1.089** ±0.012	**0.866** ±0.011	**2.230** ±0.04	**0.779** ±0.017	0.573 ±0.008	**0.937** ±0.014

For structure prediction of single chains, the loss function of structure prediction mainly follows the work of AlphaFold2 [98] , which contains Frame Aligned Point Error (FAPE) and a number of auxiliary losses but excludes MSA loss. The loss can be formalized as follows:

$$\mathcal{L} = 0.5\mathcal{L}_{\text{FAPE}} + 0.5\mathcal{L}_{\text{aux}} + 0.3\mathcal{L}_{\text{dist}} + 0.01\mathcal{L}_{\text{conf}} + \mathcal{L}_{\text{angle_norm}} + 0.5\mathcal{L}_{\text{rmsd_ca}} \tag{6}$$

where $\mathcal{L}_{\text{aux}}$ is the auxiliary loss from the structure module, $\mathcal{L}_{\text{dist}}$ is an averaged cross-entropy loss for distogram prediction, $\mathcal{L}_{\text{conf}}$ is the model confidence loss, $\mathcal{L}_{\text{angle_norm}}$ is the side chain and backbone torsion angle loss [98] and $\mathcal{L}_{\text{rmsd_ca}}$ is the rmsd for carbo alpha. In addition to the loss described by the formula above, the VH-VL complex replaces the rmsd-ca loss with a chain center-of-mass loss [30] and a structural violation loss [98], with weights of 1.0 and 0.03, respectively. The concrete loss is shown as follows:

$$\mathcal{L}_{\text{vh-vl}} = 0.5\mathcal{L}_{\text{FAPE}} + 0.5\mathcal{L}_{\text{aux}} + 0.3\mathcal{L}_{\text{dist}} + 0.01\mathcal{L}_{\text{conf}} + \mathcal{L}_{\text{angle_norm}} + \mathcal{L}_{\text{centre_mass}} + 0.03\mathcal{L}_{\text{violation}}. \tag{7}$$

Baselines. For single-chain structure prediction tasks, we conduct a comparison of existing influential folding models, including Alphafold2 and four PLM-based models: OmegaFold [23], ESMFold [99], IgFold [100], and xTrimoAbFold [101]. We use public checkpoints [11] [12] [13] [14] to infer the test set.

For the prediction of VH-VL complex structures, we compared ZDock [102] , a rigid protein docking algorithm based on fast Fourier transform correlation techniques, ClusPro [103], a docking method using bioinformatics and computational chemistry techniques, EquiDock [104], a genetic evolution-based algorithm, HDOCK [105], an algorithm that combines global and local search, and AlphaFold-Multimer [30], which predicts complex structures based on protein sequence, MSA, and template information.

Results. Each experiment is conducted 5 times with different random seeds and reports the averaged results. As demonstrated in Table 4, xTrimoPGLM-AbFold significantly outperforms all other models, notably xTrimoAbFold—an existing state-of-the-art model—in every metric related to antibody structure prediction. The impressive performance of xTrimoPGLM-AbFold implies that a pre-trained antibody model, when fine-tuned with merely a single additional Evoformer [98] block, can emerge as a leading model for antibody structure prediction, even without the presence of MSA and templates.

Table 5 shows the performance of VH-VL complex in different models. AlphaFold-Multimer, which relies heavily on MSA and template information, outperforms most of protein docking algorithms. However, xTrimoPGLM-AbFold, which does not use any MSA or template information, performs comparable with AlphaFold-Multimer, indicating that xTrimoPGLM-Ab-1B has learned sufficient and rich information on antibodies. Crucially, xTrimoPGLM-AbFold achieves a speedup of

[11] AlphaFold2:https://github.com/deepmind/alphafold/blob/main/scripts/download_alphafold_params.sh
[12] ESMFold:https://github.com/facebookresearch/esm
[13] OmegaFold:https://github.com/HeliXonProtein/OmegaFold
[14] IgFold:https://github.com/Graylab/IgFold

Table 5: Structure prediction of VH-VL complexes. The inference time is calculated on the whole test set with a single A100 GPU. xTrimoPGLM-AbFold (evo 1) and xTrimoPGLM-AbFold (evo 16) are xTrimoPGLM-AbFold with 1 Evoformer block and 16 Evoformer blocks respectively.

	RMSD↓	TM-SCORE↑	DOCKQ ↑	INFERENCE TIME ↓
ZDock	10.982	0.596	0.108	34h
ClusPro	5.899	0.792	0.404	1.3h
EquiDock	18.293	0.559	0.032	2m
HDOCK	2.032	0.926	0.705	3.2h
AlphaFold-Multimer	1.325	0.962	0.765	56.6h (original) 55m (faster MSA)
xTrimoPGLM-AbFold (evo 1)	1.304	0.962	0.762	**32s**
xTrimoPGLM-AbFold (evo 16)	**1.234**	**0.966**	**0.770**	82s

6,300× over the original AlphaFold-Multimer and **103**× over the faster MSA-searching AlphaFold-Multimer [106], owing to the original AlphaFold-Multimer consumes a long time to search MSA (0.8 hour per sample). When we increase the number of Evoformer blocks to 16, xTrimoPGLM-AbFold attains the best performance on all metrics while still maintaining a **2,400**× speedup than the original AlphaFold-Multimer and **40**× speedup than the accelerated AlphaFold-Multimer. It is noteworthy that only a marginal improvement is attained when the number of Evoformer blocks is increased from 1 to 16, which indicates that xTrimoPGLM -Ab has already captured sufficient information for downstream tasks to predict atom positions with precision.

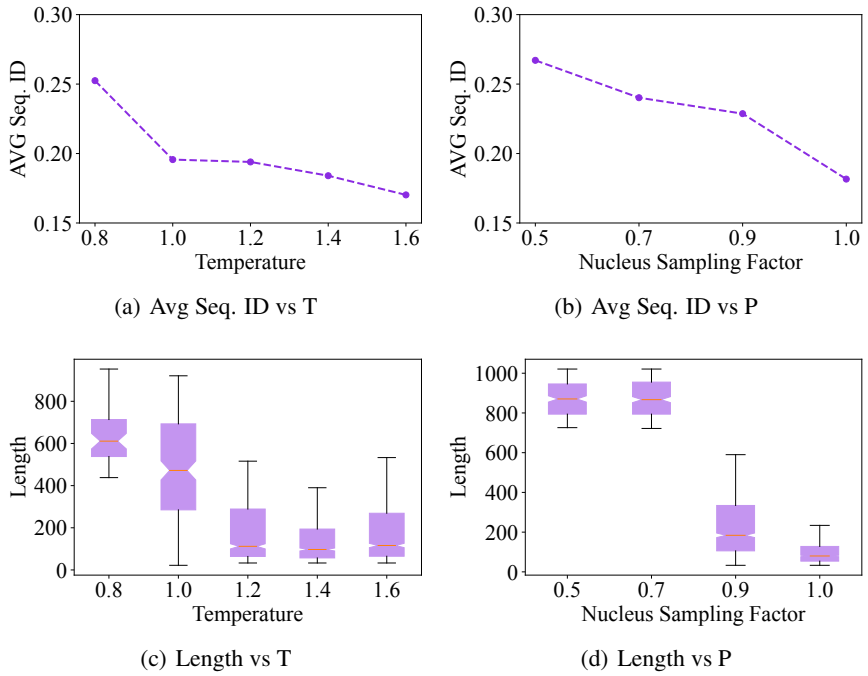
6 Generation

The autoregressive PLMs can characterize the distribution of observed evolutionary sequences, thereby enabling the generation of novel sequences with diverse folds, markedly distinct from observed natural proteins [13, 26, 107]. To assess the generation ability of xTrimoPGLM-100B, we analyze the properties of protein sequences generated via xTrimoPGLM-100B under different generation settings. Specifically, a diverse set of sequences is sampled using a cross product of temperature ($T \in$ 0.8, 1.0, 1.2, 1.4, 1.6) and nucleus sampling probability ($P \in$ 0.5, 0.7, 0.9, 1.0) parameters. For each combination of T and P, we sample 600 sequences for the comprehensive sequence analysis. The structures of all generated sequences are predicted with AlphaFold2 [3] for 3 recycles without model ensemble. The similarity of predicted structures to the natural ones in the PDB is measured by calculating the TM-score using Foldseek [108]. We also use xTrimoPGLM-Ab-1B to generate the CDR region of Covid-19 antibodies. All the generated sequences are predicted with xTrimoPGLM-AbFold.

6.1 Properties Analysis of Generated Sequences using xTrimoPGLM-100B

In this section, we examine both the sequence and structural attributes of generated sequences, shedding light on their statistical properties.

Statistical properties of the sampled artificial sequences We present the pairwise sequence identity analysis of generated sequences obtained through various combinations of temperature and nucleus sampling factors, as illustrated in Figure 9(a)(b). We observe that higher nucleus sampling factors and temperatures, which flatten the token probability distribution during the generation process, lead to a broader range of sequence diversity. However, it should be noted that the likelihood of selecting the `<eos>` token also increases under these conditions. Consequently, higher factors may result in shorter sequences, as illustrated in Figure 9(c)(d). Furthermore, our empirical study suggests that the pre-trained model tends to generate repetitive tokens when the temperature drops below 1.0 and the nucleus sampling factor falls under 0.7, which results in abnormal long sequences. Conversely, higher values of these hyperparameters improve generation quality. Therefore, we recommend a careful calibration of the hyperparameters, specifically the balance between temperature and nucleus sampling factors, to generate protein sequences that conform to the desired specifications.



(a) Avg Seq. ID vs T (b) Avg Seq. ID vs P

(c) Length vs T (d) Length vs P

Figure 9: The sequence identity and length distributions of generated sequences. Seq. ID represents the pairwise sequence identity. T is short for the temperature. P is short for the nucleus sampling probability. Higher nucleus sampling factors and temperatures lead to a broader range of sequence diversity and shorter sequences.

Intrinsically unstructured/disordered proteins/domains Intrinsically unstructured or disordered proteins/domains (IUPs) [109] exist in a largely disordered structural state and carry out basic protein functions. Hence, it is essential to identify IUPs by the commonly used disorder prediction methods, IUPred3 [110], to reflect the biological evolutionary behaviors. Without extra functional annotations, we generate a dataset of protein sequences to evaluate our proposed method in the protein disorder task. For comparison, we also simulate a natural dataset by uniformly sampling from the original training dataset. Our generated dataset and the natural dataset consist of 6,523 and 10,000 sequences, respectively.

In order to compare the two datasets comprehensively, all three prediction types are provided in Table 6, i.e., short disorder, long disorder, and globular structured domains [111]. Short disorder (SHORT) emphasizes predicting short-disordered regions, while long disorder (LONG) chiefly targets global structural disorder encompassing a minimum of 30 consecutive protein residues. The prediction corresponding to globular domains (GLOBULAR) is a structured prediction for structural studies and structural genomics projects. We also present the ordered content (the proportion of ordered regions over the entire protein, termed ORDERED) from globular disorder predictions, to analyze the structural and biochemical attributes of sequences generated by xTrimoPGLM . This approach diverges from the definition of ordered content (ratio of ordered to disordered regions) employed in ProtGPT2 [26].

Consequently, the two datasets show similar disorder prediction results as reported in Table 6. Our generated sequences have close prediction results to the natural dataset in all four metrics, with the largest gap of 3.89% in LONG between them. The experimental results affirm that sequences generated by xTrimoPGLM-100B exhibit comparable tendencies for minimal, maximal, and structured predicted disorder, akin to natural sequences.

Table 6: Disorder proteins/domains predictions (%).

	SHORT	LONG	GLOBULAR	ORDERED
Natural Data (10K)	63.38	68.16	68.57	34.20
Generated	59.84	64.27	64.96	34.56

6.2 General Protein Generation using xTrimoPGLM-100B

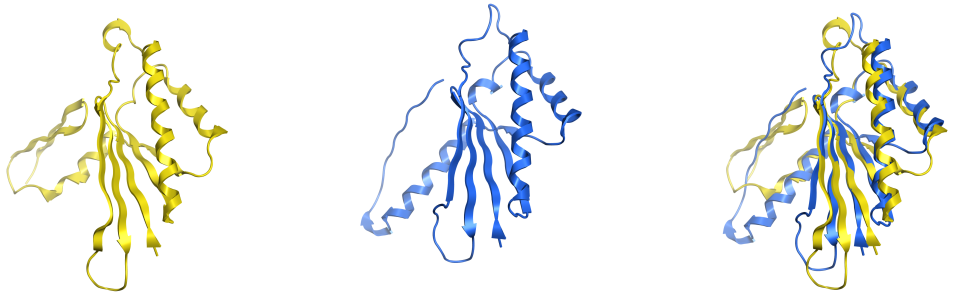
To further explore the potential of xTrimoPGLM in generating naturally functional sequences, we conduct an analysis of their corresponding structures. Our methodology encompasses generating thousands of sequences, guided by parameters (T=1.0, P=1.0) inferred from preceding statistical investigations. Each sequence is initiated with the `[gMASK]` token, subsequently inputted into xTrimoPGLM-100B. This process generates new sequences by continuously predicting the next token in an autoregression manner until the `<eos>` token is predicted or the pre-set maximum length is reached. Subsequently, AlphaFold2 [98] is employed to predict the three-dimensional folding structure corresponding to each sequence. In the following step, we utilized FoldSeek [108] to process each structure and identify any remote homology proteins.

Several interesting findings emerged when we examined the structures of the sequences generated First, we observed that the model exhibits the capacity to generate essential structural motifs, including alpha helices and beta sheets. These components form the basis of more sophisticated tertiary structures. As shown in Figure 10, Case-2 is the shortest and folds into a simple structure with two alpha helices and beta sheets. Moreover, for moderately longer sequences (Case-1 and Case-3), the generated structures are much more complex and multiple alpha helices are interconnected by loop regions. The results potentially imply an iterative process during sequence generation, aiming to attain global structural optimization. Second, sequence identity is correlated with structure similarity level. Compared with the generated sequence with extremely low identity (e.g., 11% in Case-3), a more similar sequence (e.g., 25.1% in Case-1) tends to achieve a better structural alignment (TM-score from 0.345 to 0.735). The tendency is expected as the folding algorithm largely depends on homology sequence information, e.g., the MSA alignment utilized in AlphaFold2. Last, it is noteworthy that although these sequences exhibit highly similar structures to known proteins, their sequence similarity is still very low. For example, the sequence identity of Case-3 is about 11%, but the contained six alpha helices are consistently aligned across a long stretch. The result demonstrates that the xTrimoPGLM model has the potential to search a much larger sequence space to generate functional structures. The advantage will greatly enhance the synthesis pathways for diverse protein structures, and potentially improve the design of antibodies targeting antigenic epitopes.

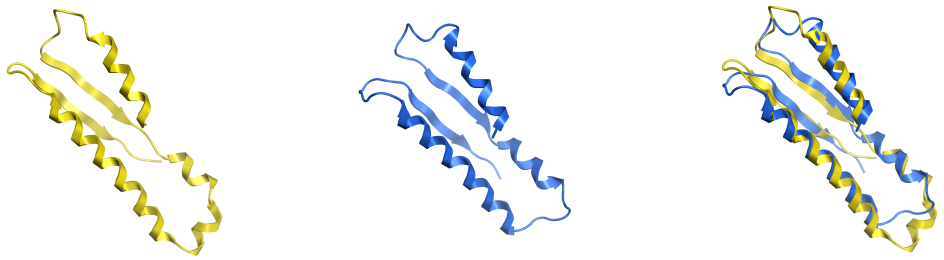
Limitations. There are still many challenges that exist in generating high-quality sequences. Primarily, the model has difficulty synthesizing proteins that resemble those found in nature when dealing with sequences longer than 200 amino acids. Instead, it tends to generate a large number of loops, as demonstrated in Figure 13. Although the model capably captures secondary structures and basic local combinations of these structures, it falls short in capturing the global or long-dependency characteristics intrinsic to protein modeling. However, these limitations can be mitigated with additional training of models, supported by increased computational resources and data volume, following the development trajectory of language models that have already shown preliminary comprehension of long-dependencies, such as emergent abilities [1].

Another limitation is the lack of specific guiding conditions during the generation of protein sequences by the model. Unlike natural language models, PLMs are unable to map content to text spaces like code, meta-knowledge, and data labels. Consequently, zero-shot or few-shot tasks are not feasible. However, if PLMs could be interfaced with natural language inputs, which allows them to generate corresponding protein sequences in response to explicit instructions and intents such as functional description, target information, and other modalities like structure. These would significantly enhance their practicality and utility.

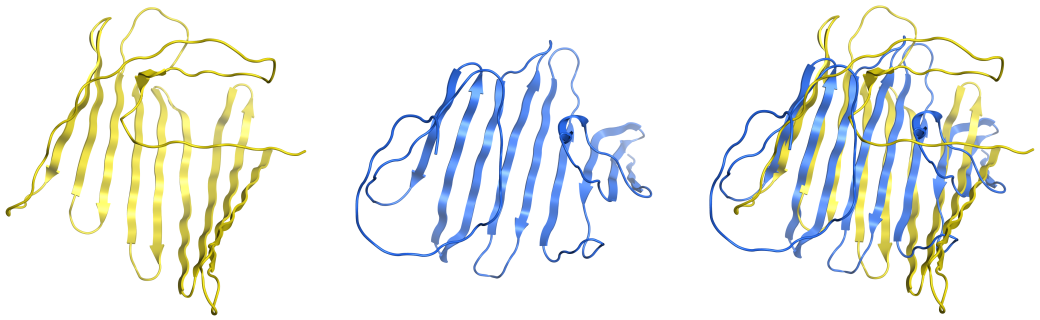
Repetition is a common issue in the generation process and arises from the tendency towards local optima and training oscillations. When generating new amino acids, models often choose options that locally maximize output probability, leading to repetitive sequences. Training oscillations can also cause overuse of certain patterns, exacerbating this issue. To mitigate such repetition, techniques



Case-1: TM-score=0.735, sequence identity=25.1%



Case-2: TM-score=0.594, sequence identity=13.4%



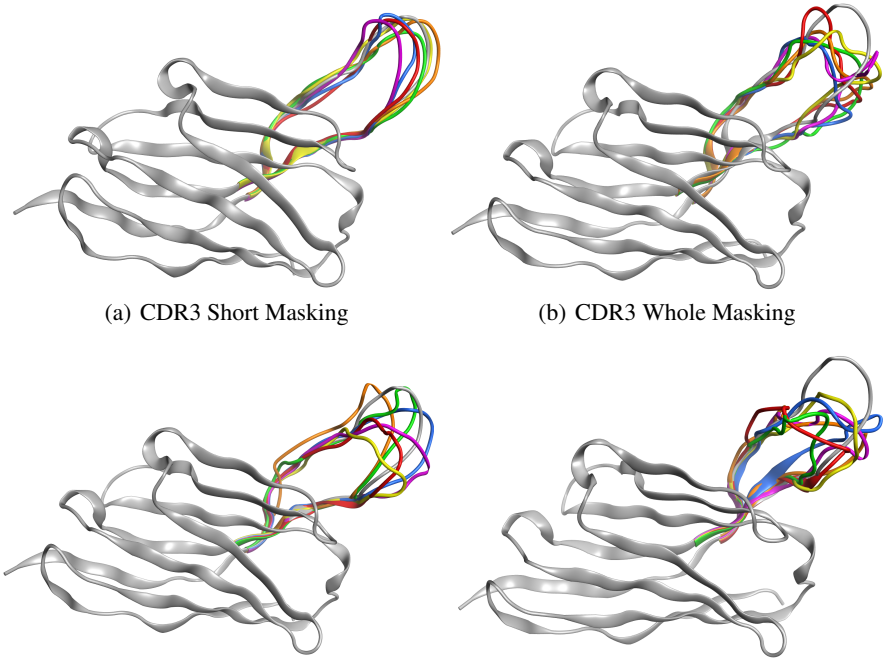
Case-3: TM-score=0.345, sequence identity=11%

Figure 10: Structural visualization of generated and searched natural template sequences. In each row, the left and middle panel depicts the predicted structure of xTrimoPGLM-100B generated sequence and the ground truth structure of searched remote protein sequence. The two structures are superposed at the right panel. We use TM-score and sequence identity as the metric to assess the similarity at both structural and sequence levels, respectively.

like N-gram penalty and temperature or nucleus factor tuning are used in Figure 13, thus enhancing the overall sequence quality.

6.3 Antibody Generation using xTrimoPGLM-Ab-1B

To demonstrate the generation capacity of xTrimoPGLM-Ab-1B, we select a heavy chain antibody sequence (specifically 368.04.B.0106) that interacts with SARS-CoV-2-WT. We implement four distinctive masking strategies to redesign the Complementarity Determining Region 3 (CDR3) of the selected sequence, as the CDR3 region is a critical element in the structure of an antibody or T cell receptor. This sequence is characterized by significant variability and plays an integral role in the specificity of antigen recognition. The four strategies are defined as follows,



(a) CDR3 Short Masking (b) CDR3 Whole Masking

(c) CDR3 Random Mutation (d) CDR3 Random Retrieval

Figure 11: The conformations of various methodologies implemented for xTrimoPGLM-AbFold. (a) CDR3 Short Masking: This setup represents a sequence modification scenario where a fragment of the CDR3 sequence is masked and redesigned. The generated antibodies are structurally similar to the original ones. (b) CDR3 Whole Masking: This strategy involves masking the entire CDR3 sequence, necessitating a de novo structural prediction, thus illustrating a more comprehensive redesign approach. This setup offers a broader framework for exploring the subtleties of antigen recognition and antibody functionality. (c) CDR3 Random Mutation: This strategy signifies the validation process using random mutagenesis of selected positions within the CDR3 domain. (d) CDR3 Random Retrieval: This demonstrates another validation method wherein the CDR3 region of the base sequence is replaced by a random CDR3 region from other antibodies in the SARS-CoV-2 wild-type library.

- **CDR3 Short Masking (CSM).** This strategy involves masking a partial segment of the CDR3 region. We select the length of the masked region based on a uniform distribution within the interval $[3, 6]$. Subsequently, a segment of the CDR3 region is randomly replaced with the `[sMASK]` token. Upon feeding this modified antibody sequence into xTrimoPGLM-Ab-1B, the masked segment of the CDR3 region undergoes a redesign. The comparison between the conformations of the CDR3-redesigned antibodies and the original sequence is depicted in Figure 11(a).
- **CDR3 Whole Masking (CWM).** This strategy involves masking the entirety of the CDR3 region with the `[sMASK]` token, thus necessitating a de novo design approach. Given the increased complexity of this setting, compared to the CSM, the CWM requires more sophisticated computational models. This method provides a comprehensive and integrative methodology to delve deeper into the complexities of antibody functionality, as shown in Figure 11(b).
- **CDR3 Random Mutation (CRM).** This strategy adopts a random mutagenesis approach focusing on specific sites within the CDR3 region. It involves randomly selecting 3-6 positions within the CDR3 domain and subsequently introducing random mutations at these sites. This method can be seen as a stochastic baseline that operates at a comparable edit distance. The result is shown in Figure 11(c).
- **CDR3 Random Retrieval (CRR).** This strategy comprises the random substitution of the CDR3 region using sequences from other antibodies present in the SARS-CoV-2 wild-type library. The predicted structures are illustrated in Figure 11(d).

Table 7: A collection of sequences produced via two distinct masking approaches: CDR3 Short Masking and CDR3 Whole Masking. In addition, it includes two parallel benchmark methods, namely CDR3 Random Mutations and CDR3 Random Retrieval. Each sequence's relative variation from the reference truth is also quantified, demonstrated through their respective edit distances.

Marker	CDR3 Short Masking	Edit Distance
Ground truth	AKDKDYGDLPTVDYYYHYGMDV	-
Red	AKDKDYGDLPTVLRYYYYGMDV	3
Green	AKDKDYGDLPQYYYYHYGMDV	3
Blue	AKDKDYGDLPSLSYYYHYGMDV	3
Yellow	AKDKDYGDLPTVDYFFLLGMDV	4
Purple	AKDKDYGDLSLSPPYYHYGMDV	5
Orange	AKDKDYGDLPTVDYYDYYGLDV	3
	CDR3 Whole Masking	
Red	AKDSYYGSGSYYNPDQGYYYYYGMDV	12
Green	AKDGPGGSGSYSADYYYYYGMDV	10
Blue	AKDKDCGGDCYLLDYHYYYGMDV	8
Yellow	AKDSTVTPLPAAIRTYYYYYYGMDV	12
Purple	AKDLNRRGISIFGVDNDYYFYGLDV	13
Orange	AKDSYYGSGSYSYVSYYYYYYGMDV	11
	CDR3 Random Mutations	
Red	AKDKDHVGFMTVDYYYHYGMDV	4
Green	AKDILFIDLPTVDYYYHYGMDV	5
Blue	AKDKDYGDLPTVDYYYLQLIPC	6
Yellow	AKDKDYGDLPTVDYDIGYGMDV	3
Purple	AKDKDYRHRETVDYYYHYGMDV	4
Orange	AKDKDYGDLPTVDYYYALRRRR	6
	CDR3 Random Retrieval	
Red	ARDRSGKDVLTGYPMFPAGMDV	14
Green	ARDLSAGHCTGGVCYTAGGIDY	16
Blue	ARGVITMVRGVIRDYYYYGMDV	13
Yellow	ARDLGGGYSNVYVNHYYGMDV	12
Purple	ARDEITVTAGAWGNYYYGMDY	14
Orange	AKGYCGGDCYSGLLDWYFDL	16

Results. Under the aforementioned settings, we generate a set of 6,000 antibodies via xTrimoPGLM-Ab-1B. Six antibodies are randomly selected as depicted in Figure 6.3. xTrimoPGLM-AbFold is utilized as the structure prediction model. In response to the observation that using CDR3 short masking tends to generate antibodies closely resembling the ground truth with a small edit distance, we implemented a filter to exclude any antibodies with an edit distance of 2 or less. A series of generated sequences and their corresponding edit distances from the ground truth is presented in Table 7. Importantly, it is noteworthy that both the CSM and CWM policies are capable of generating sequences of varying lengths without resorting to mutations or deletions. In contrast, the sequences generated by the two parallel baselines, CRM and CRR, display considerable disorder, regardless of whether there are few mutations or a complete replacement of the entire CDR3 fragment. Our analysis further identifies a relationship between the edit distance and the structure of the generated antibody's CDR3 region. Specifically, as the edit distance grows, the organization of the CDR3 region tends to degrade, suggesting that even large generative models currently face limitations.

7 Discussion & Conclusion

One must acknowledge that a substantial limitation of xTrimoPGLM-100B is the high computational cost linked to these models, posing a considerable barrier to their implementation. A possible approach to alleviating this might be the application of more advanced efficient technologies in terms of parameters or memory, like quantization, for instance QLoRA [112], kernel fusion, such as FlashAttention [113], and Multi-query [114]. Utilizing these methods could enable the training and

deployment of larger models with less computational resources. However, further investigation is needed to confirm its practical effectiveness.

Many task-specific methods are plausibly orthogonal to pre-training approaches, complementing each other to achieve robust performance and significant advancements, as is presented in protein structure prediction tasks. We underscore the importance of leveraging the substantial fitting capabilities of large pre-trained models for protein tasks. Instead of treating these models as simply feature extractors, we argue that it is critical to tap into their inherent learning capabilities [115]. For example, the contact map prediction task can significantly improve the performance of fine-tuning (including the use of LoRA) by 15-20 points. Adding inductive bias to these models, although it can improve performance in some cases, may also inadvertently constrain their learning capacity and limit the breadth and depth of their feature extraction capabilities. The introduced bias may oversimplify the problem at hand, hence reducing the robustness of the model. Instead, by exploiting the inherent strengths of these large pre-trained models, we can extract more diverse and complex features and build more robust and flexible predictive models. We envision that with continued advancement in these models and computing technology, the potential of large pre-trained models in protein-related tasks will be fully unlocked.

In conclusion, our key contribution is the exploration of unified understanding and generation pre-training with an extremely large-scale protein language model. This model is comparable to the scale of today's large natural language models, and our extensive experiments show that downstream tasks also comply with the scaling law. Additionally, we have opened up the generation of protein sequences with xTrimoPGLM-100B. By utilizing the xTrimoPGLM framework, we've made advancements in predicting antibody naturalness and structure prediction with our antibody-specific model, xTrimoPGLM-Ab. Our work serves as a stepping stone for future research in the protein foundation model, and we hope it can facilitate further progress in protein-related applications.

8 Acknowledgements

We would like to express our deepest gratitude to all those who provided us with the possibility to complete this work. We are grateful to Ming Ding (THU), and Xiao Liu (THU) for their consecutive suggestions about model training and inference, and Ke Wang (BioMap), Nachuan Shan (BioMap), Tengyun Liu (BioMap) and Beiqi Hongdu (BioMap) for their help in data and biological guidance. Our heartfelt thanks are also extended to Zhizhuo Zhang (BioMap), Hui Li (BioMap), and Taifeng Wang (BioMap) for advice on antibody evaluations. Additionally, we would like to thank Zezhi Wang (BioMap), Ming Yang (BioMap), and Xiaoming Zhang (BioMap) for their help in infrastructure and supercomputing facilities.

References

[1] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022.

[2] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[3] 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.

[4] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

[5] Christian Boehmer Anfinsen et al. The molecular basis of evolution. *The molecular basis of evolution.*, 1959.

[6] 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.

[7] 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.

[8] A Elnaggar, M Heinzinger, C Dallago, G Rehawi, Y Wang, and L Jones. & rost, b.(2021). prottrans: towards cracking the language of lifes code through self-supervised deep learning and high performance computing. *IEEE transactions on pattern analysis and machine intelligence*.

[9] 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.

[10] Rolf Apweiler, Amos Bairoch, Cathy H Wu, Winona C Barker, Brigitte Boeckmann, Serenella Ferro, Elisabeth Gasteiger, Hongzhan Huang, Rodrigo Lopez, Michele Magrane, et al. Uniprot: the universal protein knowledgebase. *Nucleic acids research*, 32(suppl_1):D115–D119, 2004.

[11] Robert D Finn, Alex Bateman, Jody Clements, Penelope Coggill, Ruth Y Eberhardt, Sean R Eddy, Andreas Heger, Kirstie Hetherington, Liisa Holm, Jaina Mistry, et al. Pfam: the protein families database. *Nucleic acids research*, 42(D1):D222–D230, 2014.

[12] Martin Steinegger, Milot Mirdita, and Johannes Söding. Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nature methods*, 16(7):603–606, 2019.

[13] Erik Nijkamp, Jeffrey Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. Progen2: exploring the boundaries of protein language models. *arXiv preprint arXiv:2206.13517*, 2022.

[14] Brian Hie, Salvatore Candido, Zeming Lin, Ori Kabeli, Roshan Rao, Nikita Smetanin, Tom Sercu, and Alexander Rives. A high-level programming language for generative protein design. *bioRxiv*, pages 2022–12, 2022.

[15] Robert Verkuil, Ori Kabeli, Yilun Du, Basile IM Wicky, Lukas F Milles, Justas Dauparas, David Baker, Sergey Ovchinnikov, Tom Sercu, and Alexander Rives. Language models generalize beyond natural proteins. *bioRxiv*, pages 2022–12, 2022.

[16] Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, et al. Ul2: Unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations*, 2023.

[17] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. *Advances in neural information processing systems*, 32, 2019.

[18] Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, et al. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *International conference on machine learning*, pages 642–652. PMLR, 2020.

[19] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[20] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[21] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[22] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

[23] Rui Min Wu, Fan Ding, Rui Wang, Rui Shen, Xiwen Zhang, Shitong Luo, Chenpeng Su, Zuofan Wu, Qi Xie, Bonnie Berger, Jianzhu Ma, and Jian Peng. High-resolution de novo structure prediction from primary sequence. *bioRxiv*, 2022.

[24] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, 2022.

[25] 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.

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

[27] Ahmed Elnaggar, Hazem Essam, Wafaa Salah-Eldin, Walid Moustafa, Mohamed Elkerdawy, Charlotte Rochereau, and Burkhard Rost. Ankh: Optimized protein language model unlocks general-purpose modelling. *bioRxiv*, pages 2023–01, 2023.

[28] Aleksandr Kovaltsuk, Jinwoo Leem, Sebastian Kelm, James Snowden, Charlotte M Deane, and Konrad Krawczyk. Observed antibody space: a resource for data mining next-generation sequencing of antibody repertoires. *The Journal of Immunology*, 201(8):2502–2509, 2018.

[29] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11):2579–2605, 2008.

[30] Richard Evans, Michael O'Neill, Alexander Pritzel, Natasha Antropova, Andrew Senior, Tim Green, Augustin Zídek, Russ Bates, Sam Blackwell, Jason Yim, Olaf Ronneberger, Sebastian Bodenstein, Michal Zielinski, Alex Bridgland, Anna Potapenko, Andrew Cowie, Kathryn Tunyasuvunakool, Rishub Jain, Ellen Clancy, Pushmeet Kohli, John M. Jumper, and Demis Hassabis. Protein complex prediction with alphafold-multimer. *bioRxiv*, 2021.

[31] Minkyung Baek and David Baker. Deep learning and protein structure modeling. *Nature methods*, 19(1):13–14, 2022.

[32] 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.

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

[34] Mohammed AlQuraishi. Proteinnet: a standardized data set for machine learning of protein structure. *BMC bioinformatics*, 20(1):1–10, 2019.

[35] Karen S Sarkisyan, Dmitry A Bolotin, Margarita V Meer, Dinara R Usmanova, Alexander S Mishin, George V Sharonov, Dmitry N Ivankov, Nina G Bozhanova, Mikhail S Baranov, Onuralp Soylemez, et al. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603):397–401, 2016.

[36] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022.

[37] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[38] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*, 2022.

[39] Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. In *International Conference on Learning Representations*.

[40] Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. What language model architecture and pretraining objective works best for zero-shot generalization? In *International Conference on Machine Learning*, pages 22964–22984. PMLR, 2022.

[41] Yi Tay, Jason Wei, Hyung Won Chung, Vinh Q Tran, David R So, Siamak Shakeri, Xavier Garcia, Huaixiu Steven Zheng, Jinfeng Rao, Aakanksha Chowdhery, et al. Transcending scaling laws with 0.1% extra compute. *arXiv preprint arXiv:2210.11399*, 2022.

[42] 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.

[43] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[44] Teven Le Scao, Thomas Wang, Daniel Hesslow, Lucile Saulnier, Stas Bekman, M Saiful Bari, Stella Bideman, Hady Elsahar, Niklas Muennighoff, Jason Phang, et al. What language model to train if you have one million gpu hours? *arXiv preprint arXiv:2210.15424*, 2022.

[45] Milot Mirdita, Konstantin Schütze, Yoshitaka Moriwaki, Lim Heo, Sergey Ovchinnikov, and Martin Steinegger. Colabfold: making protein folding accessible to all. *Nature methods*, 19(6):679–682, 2022.

[46] Alex L Mitchell, Alexandre Almeida, Martin Beracochea, Miguel Boland, Josephine Burgin, Guy Cochrane, Michael R Crusoe, Varsha Kale, Simon C Potter, Lorna J Richardson, et al. Mgnify: the microbiome analysis resource in 2020. *Nucleic acids research*, 48(D1):D570–D578, 2020.

[47] Tom O Delmont, Morgan Gaia, Damien D Hinsinger, Paul Frémont, Chiara Vanni, Antonio Fernandez-Guerra, A Murat Eren, Artem Kourlaiev, Leo d'Agata, Quentin Clayssen, et al. Functional repertoire convergence of distantly related eukaryotic plankton lineages abundant in the sunlit ocean. *Cell Genomics*, 2(5):100123, 2022.

[48] Eli Levy Karin, Milot Mirdita, and Johannes Söding. Metaeuk—sensitive, high-throughput gene discovery, and annotation for large-scale eukaryotic metagenomics. *Microbiome*, 8:1–15, 2020.

[49] Harriet Alexander, Sarah K Hu, Arianna I Krinos, Maria Pachiadaki, Benjamin J Tully, Christopher J Neely, and Taylor Reiter. Eukaryotic genomes from a global metagenomic dataset illuminate trophic modes and biogeography of ocean plankton. *bioRxiv*, pages 2021–07, 2021.

[50] Stephen Nayfach, David Páez-Espino, Lee Call, Soo Jen Low, Hila Sberro, Natalia N Ivanova, Amy D Proal, Michael A Fischbach, Ami S Bhatt, Philip Hugenholtz, et al. Metagenomic compendium of 189,680 dna viruses from the human gut microbiome. *Nature microbiology*, 6(7):960–970, 2021.

[51] Luis F Camarillo-Guerrero, Alexandre Almeida, Guillermo Rangel-Pineros, Robert D Finn, and Trevor D Lawley. Massive expansion of human gut bacteriophage diversity. *Cell*, 184(4):1098–1109, 2021.

[52] Martin Steinegger and Johannes Söding. Clustering huge protein sequence sets in linear time. *Nature communications*, 9(1):2542, 2018.

[53] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[54] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

[55] Deepak Narayanan, Amar Phanishayee, Kaiyu Shi, Xie Chen, and Matei Zaharia. Memory-efficient pipeline-parallel dnn training. In *International Conference on Machine Learning*, pages 7937–7947. PMLR, 2021.

[56] Leslie G Valiant. A bridging model for parallel computation. *Communications of the ACM*, 33(8):103–111, 1990.

[57] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506, 2020.

[58] Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

[59] Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. Deepnet: Scaling transformers to 1,000 layers. *arXiv preprint arXiv:2203.00555*, 2022.

[60] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. In *International Conference on Learning Representations*, 2018.

[61] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835, 2021.

[62] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

[63] Ieva Pudžiuvelyte, Kliment Olechnovič, Egle Godliauskaite, Kristupas Sermokas, Tomas Urbaitis, Giedrius Gasiunas, and Darius Kazlauskas. Temstapro: protein thermostability prediction using sequence representations from protein language models. *bioRxiv*, 2023.

[64] Gang Li, Filip Buric, Jan Zrimec, Sandra Viknander, Jens Nielsen, Aleksej Zelezniak, and Martin K. M. Engqvist. Learning deep representations of enzyme thermal adaptation. *Protein Science*, 31(12):e4480, 2022.

[65] Feiran Li, Le Yuan, Hongzhong Lu, Gang Li, Yu Chen, Martin K. M. Engqvist, Eduard J. Kerkhoven, and Jens Nielsen. Deep learning-based kcat prediction enables improved enzyme-constrained model reconstruction. *Nature Catalysis*, 5(8):662–672, Aug 2022.

[66] Yejian Wu, Lujing Cao, Zhipeng Wu, Xinyi Wu, Xinqiao Wang, and Hongliang Duan. Ccbhla: pan-specific peptide–hla class i binding prediction via convolutional and bilstm features. *bioRxiv*, 2023.

[67] My-Diem Nguyen Pham, Thanh-Nhan Nguyen, Le Son Tran, Que-Tran Bui Nguyen, Thien-Phuc Hoang Nguyen, Thi Mong Quynh Pham, Hoai-Nghia Nguyen, Hoa Giang, Minh-Duy Phan, and Vy Nguyen. epiTCR: a highly sensitive predictor for TCR–peptide binding. *Bioinformatics*, 39(5), 04 2023. btad284.

[68] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.

[69] Jianyi Yang, Ivan Anishchenko, Hahnbeom Park, Zhenling Peng, Sergey Ovchinnikov, and David Baker. Improved protein structure prediction using predicted interresidue orientations. *Proceedings of the National Academy of Sciences*, 117(3):1496–1503, 2020.

[70] Loredana Lo Conte, Bart Ailey, Tim JP Hubbard, Steven E Brenner, Alexey G Murzin, and Cyrus Chothia. Scop: a structural classification of proteins database. *Nucleic acids research*, 28(1):257–259, 2000.

[71] Jie Hou, Badri Adhikari, and Jianlin Cheng. Deepsf: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, 34(8):1295–1303, 2018.

[72] Daozheng Chen, Xiaoyu Tian, Bo Zhou, and Jun Gao. Profold: Protein fold classification with additional structural features and a novel ensemble classifier. *BioMed research international*, 2016, 2016.

[73] Junjie Chen, Mingyue Guo, Xiaolong Wang, and Bin Liu. A comprehensive review and comparison of different computational methods for protein remote homology detection. *Briefings in bioinformatics*, 19(2):231–244, 2018.

[74] Michael Schantz Klausen, Martin Closter Jespersen, Henrik Nielsen, Kamilla Kjaergaard Jensen, Vanessa Isabell Jurtz, Casper Kaae Soenderby, Morten Otto Alexander Sommer, Ole Winther, Morten Nielsen, Bent Petersen, et al. Netsurfp-2.0: Improved prediction of protein structural features by integrated deep learning. *Proteins: Structure, Function, and Bioinformatics*, 87(6):520–527, 2019.

[75] 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.

[76] Gabriel J Rocklin, Tamuka M Chidyausiku, Inna Goreshnik, Alex Ford, Scott Houliston, Alexander Lemak, Lauren Carter, Rashmi Ravichandran, Vikram K Mulligan, Aaron Chevalier, et al. Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science*, 357(6347):168–175, 2017.

[77] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S Song. Evaluating protein transfer learning with tape. In *Advances in Neural Information Processing Systems*, 2019.

[78] Yao Cheng, Haobo Wang, Hua Xu, Yuan Liu, Bin Ma, Xuemin Chen, Xin Zeng, Xianghe Wang, Bo Wang, Carina Shiau, et al. Co-evolution-based prediction of metal-binding sites in proteomes by machine learning. *Nature Chemical Biology*, pages 1–8, 2023.

[79] Prabal Chhibbar and Arpit Joshi. Generating protein sequences from antibiotic resistance genes data using generative adversarial networks. *arXiv preprint arXiv:1904.13240*, 2019.

[80] Yunan Luo, Guangde Jiang, Tianhao Yu, Yang Liu, Lam Vo, Hantian Ding, Yufeng Su, Wesley Wei Qian, Huimin Zhao, and Jian Peng. Ecnet is an evolutionary context-integrated deep learning framework for protein engineering. *Nature communications*, 12(1):5743, 2021.

[81] Yuchi Qiu, Jian Hu, and Guo-Wei Wei. Cluster learning-assisted directed evolution. *Nature computational science*, 1(12):809–818, 2021.

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

[83] Rakesh David, Rhys-Joshua D Menezes, Jan De Klerk, Ian R Castleden, Cornelia M Hooper, Gustavo Carneiro, and Matthew Gilliham. Identifying protein subcellular localisation in scientific literature using bidirectional deep recurrent neural network. *Scientific Reports*, 11(1):1696, 2021.

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

[85] 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.

[86] Feng Zhu, Thomas Althaus, Chee Wah Tan, Alizée Costantini, Wan Ni Chia, Nguyen Van Vinh Chau, Giada Mattiuzzo, Nicola J Rose, Eric Voiglio, Lin-Fa Wang, et al. Who international standard for sars-cov-2 antibodies to determine markers of protection. *The Lancet Microbe*, 3(2):e81–e82, 2022.

[87] Qing Li, Ying Wang, Qiang Sun, Jasmin Knopf, Martin Herrmann, Liangyu Lin, Jingting Jiang, Changshun Shao, Peishan Li, Xiaozhou He, et al. Immune response in covid-19: what is next? *Cell Death & Differentiation*, 29(6):1107–1122, 2022.

[88] Marie-Paule Lefranc, Veronique Giudicelli, Chantal Ginestoux, Joumana Jabado-Michaloud, Geraldine Folch, Fatena Bellahcene, Yan Wu, Elodie Gemrot, Xavier Brochet, Jero^me Lane, et al. Imgt®, the international immunogenetics information system®. *Nucleic acids research*, 37(suppl_1):D1006–D1012, 2009.

[89] Richard W Shuai, Jeffrey A Ruffolo, and Jeffrey J Gray. Generative language modeling for antibody design. *bioRxiv*, pages 2021–12, 2021.

[90] Tobias H Olsen, Iain H Moal, and Charlotte M Deane. Ablang: an antibody language model for completing antibody sequences. *Bioinformatics Advances*, 2(1):vbac046, 2022.

[91] Jeffrey A Ruffolo, Jeffrey J Gray, and Jeremias Sulam. Deciphering antibody affinity maturation with language models and weakly supervised learning. *arXiv preprint arXiv:2112.07782*, 2021.

[92] Sharrol Bachas, Goran Rakocevic, David Spencer, Anand V Sastry, Robel Haile, John M Sutton, George Kasun, Andrew Stachyra, Jahir M Gutierrez, Edriss Yassine, et al. Antibody optimization enabled by artificial intelligence predictions of binding affinity and naturalness. *bioRxiv*, pages 2022–08, 2022.

[93] Daniel Hesslow, Niccoló Zanichelli, Pascal Notin, Iacopo Poli, and Debora Marks. Rita: a study on scaling up generative protein sequence models. *arXiv preprint arXiv:2205.05789*, 2022.

[94] Julian Salazar, Davis Liang, Toan Q Nguyen, and Katrin Kirchhoff. Masked language model scoring. *arXiv preprint arXiv:1910.14659*, 2019.

[95] Helen M. Berman. The protein data bank: a historical perspective. *Acta crystallographica. Section A, Foundations of crystallography*, 64 Pt 1:88–95, 2008.

[96] 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.

[97] Sankar Basu and Björn Wallner. Dockq: a quality measure for protein-protein docking models. *PloS one*, 11(8):e0161879, 2016.

[98] Kathryn Tunyasuvunakool, Jonas Adler, Zachary Wu, Tim Green, Michal Zielinski, Augustin Zídek, Alex Bridgland, Andrew Cowie, Clemens Meyer, Agata Laydon, Sameer Velankar, Gerard J. Kleywegt, Alex Bateman, Richard Evans, Alexander Pritzel, Michael Figurnov, Olaf Ronneberger, Russ Bates, Simon A A Kohl, Anna Potapenko, Andrew J. Ballard, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Ellen Clancy, David A. Reiman, Stig Petersen, Andrew W. Senior, Koray Kavukcuoglu, Ewan Birney, Pushmeet Kohli, John M. Jumper, and Demis Hassabis. Highly accurate protein structure prediction for the human proteome. *Nature*, 596:590 – 596, 2021.

[99] Zeming Lin, Halil Akin, Roshan Rao, Brian L. Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic level protein structure with a language model. *bioRxiv*, 2022.

[100] Jeffrey A. Ruffolo, Lee-Shin Chu, Sai Pooja Mahajan, and Jeffrey J. Gray. Fast, accurate antibody structure prediction from deep learning on massive set of natural antibodies. *Nature Communications*, 14, 2022.

[101] Yining Wang, Xumeng Gong, Shaochuan Li, Bing Yang, Yiwu Sun, Chuan Shi, Yangang Wang, Cheng Yang, Hui Li, and Le Song. xtrimoabfold: De novo antibody structure prediction without msa. *ArXiv*, abs/2212.00735, 2022.

[102] Rong Chen, Li Li, and Zhiping Weng. Zdock: An initial-stage protein-docking algorithm. *Proteins: Structure*, 52, 2003.

[103] Dima Kozakov, David R. Hall, Bing Xia, Kathryn A. Porter, Dzmitry Padhorny, Christine Yueh, Dmitri Beglov, and Sandor Vajda. The cluspro web server for protein–protein docking. *Nature Protocols*, 12:255–278, 2017.

[104] Octavian-Eugen Ganea, Xinyuan Huang, Charlotte Bunne, Yatao Bian, Regina Barzilay, T. Jaakkola, and Andreas Krause. Independent se(3)-equivariant models for end-to-end rigid protein docking. *ArXiv*, abs/2111.07786, 2021.

[105] Yumeng Yan, Huanyu Tao, Jiahua He, and Shengua Huang. The hdock server for integrated protein–protein docking. *Nature Protocols*, 15:1829–1852, 2020.

[106] Yujie Luo, Shaochuan Li, Yiwu Sun, Ruijia Wang, Tingting Tang, Beiqi Hongdu, Xingyi Cheng, Chuan Shi, Hui Li, and Le Song. xtrimodock: Rigid protein docking via cross-modal representation learning and spectral algorithm. *bioRxiv*, pages 2023–02, 2023.

[107] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. A deep unsupervised language model for protein design. *bioRxiv*, pages 2022–03, 2022.

[108] 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*, pages 1–4, 2023.

[109] Zsuzsanna Dosztanyi, Veronika Csizmok, Peter Tompa, and Istvan Simon. The pairwise energy content estimated from amino acid composition discriminates between folded and intrinsically unstructured proteins. *Journal of molecular biology*, 347(4):827–839, 2005.

[110] Gábor Erdős, Mátyás Pajkos, and Zsuzsanna Dosztányi. Iupred3: prediction of protein disorder enhanced with unambiguous experimental annotation and visualization of evolutionary conservation. *Nucleic acids research*, 49(W1):W297–W303, 2021.

[111] Zsuzsanna Dosztányi. Prediction of protein disorder based on iupred. *Protein Science*, 27(1):331–340, 2018.

[112] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

[113] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

[114] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*, 2023.

[115] Kurtland Chua, Qi Lei, and Jason D Lee. How fine-tuning allows for effective meta-learning. *Advances in Neural Information Processing Systems*, 34:8871–8884, 2021.

A Training Data Distribution

As shown in Figure 12, the bar charts represent the distribution of sequence lengths within the Uniref90 and ColAbFoldDB datasets. In both datasets, sequences in the '100-400' length category predominate, followed by the '50-100' category. The '0-50' and '400+' categories contain significantly fewer sequences. Note the comparison between the distribution of Uniref90 and ColAbFoldDB, indicating the variety of sequence lengths used for model training.

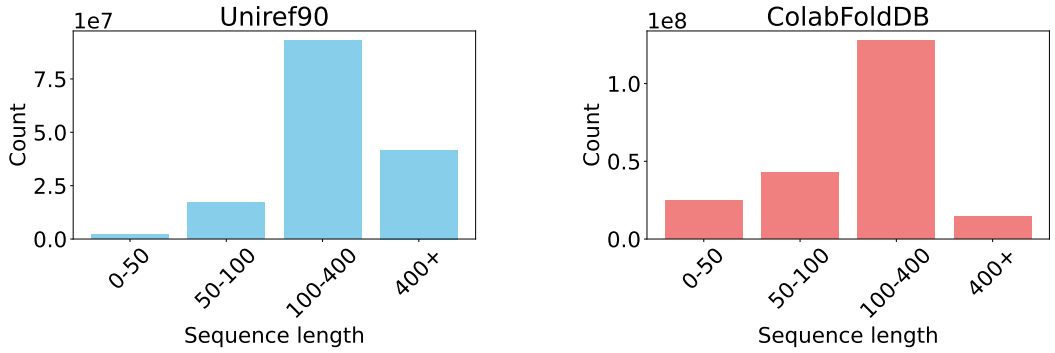


Figure 12: Training data distribution

B Tasks Comparison

We evaluated all benchmarked downstream tasks with xTrimoPGLM-100B and ESM2 models. The performance results are followed as Table 8. This table is the digitized result of the previous Figure 7.

Table 8: Performance of different models across all benchmarked downstream protein-related tasks. xT100B depicts xTrimoPGLM-100B model, E15B and E150M for ESM-15B and ESM-150M model respectively. Metric values are shown in both probing and LoRA (in parentheses) fine-tuning modes, where the underline denotes the best performance of probing and **bold** indicates the best performance of LoRA fine-tuning.

Type	Task	Metric	Model		
			xT100B (LoRA)	E15B (LoRA)	E150M (LoRA)
P. Struc.	Cont. Pred.	Top L/5 ACC	76.86 (**93.32**)	73.52 (92.19)	63.60 (84.72)
	Fold Pred.	12K-cls ACC	71.57 (**75.61**)	67.39 (69.20)	54.87 (59.25)
	Sec. Struc. Pred.	3-cls ACC	74.63 (75.33)	74.40 (**75.85**)	73.31 (74.15)
P. Func.	Antib. Res.	19-cls ACC	98.29 (**98.38**)	98.13 (98.28)	97.54 (96.94)
	Fluor.	SRCC	65.16 (**66.00**)	63.84 (63.71)	52.68 (54.54)
	Fitness	SRCC	81.69 (**96.10**)	77.12 (94.75)	69.60 (94.65)
	Localization	10-cls ACC	79.99 (81.60)	80.78 (**82.35**)	77.85 (78.88)
P. Inter.	Enzyme eff.	PCC	71.44 (**74.79**)	68.95 (74.58)	65.77 (71.72)
	Metal Bind.	2-cls ACC	81.70 (**82.78**)	79.35 (80.85)	73.94 (81.53)
	Pept.-HLA/MHC Aff.	AUC	87.22 (96.68)	90.48 (**97.28**)	91.39 (97.12)
	TCR-pMHC Aff.	AUC	89.76 (**95.10**)	91.10 (94.05)	87.81 (90.40)
P. Dev.	Solubility	2-cls ACC	76.04 (**79.45**)	74.76 (74.63)	71.50 (72.47)
	Stability	SRCC	75.52 (**84.21**)	71.69 (80.75)	69.08 (77.69)
	Temp. Stabit.	MCC	93.07 (**94.22**)	93.01 (93.24)	86.28 (85.93)
	Opt. Temp.	SRCC	73.64 (**73.96**)	73.08 (73.29)	68.57 (69.47)

C Model FLOPs Comparison

We conduct a comparative analysis of computational resources utilized by different pre-trained protein language models (Table 9). The parameters detailed in this table are meticulously calculated by implementing the models as per the configurations outlined in their respective source papers and accompanying resources, such as code and model checkpoints. When discrepancies arise between a paper's theoretical account and its practical application, we favor the metrics provided in the paper. From the right-hand side, the total training tokens are computed by multiplying the training steps, global batch size, and sequence length, given that all models listed are sequence language models. The model's parameters are estimated directly by following the authors' released implementations and hyperparameters, with the sum of the training parameters calculated while disregarding tied

weights and buffers. The total training compute is estimated by first approximating the FLOPs for one forward propagation (1F) of a single training sample. This is then multiplied by three to account for one forward and one backward propagation without activation recomputation (1F1B). The resulting number is then multiplied by the number of samples used during the entire pre-training process, which is equivalent to the total training tokens divided by the sequence length during pre-training. Only matrix multiplication (matmul) operations are considered in the compute statistics, with other operations such as embedding, element-wise addition, softmax, and layer normalization excluded from the FLOP count. The matmuls considered within the attention block include key, query, and value transformations, attention matrix computation, attention over values, and post-attention linear transformation. Hidden size transformations in the feed-forward block, a projection from hidden dimensions into vocabulary dimensions, and a linear transformation in the language model head (if one exists), are also included in the matmul FLOPs. As an example, if hidden states of size (B, L, D) are multiplied by a weight matrix of size (D, 4D), the resulting FLOPs is BLD4D2 (the factor of 2 accounts for multiplication and addition operations). The total training compute for ProtGPT2 is estimated assuming each A100 GPU performs 120 TFLOPs per second. Consequently, 128 A100 GPUs would achieve approximately 5.3e+21 FLOPs over four days of training.

Table 9: Comparison of training computes between different pre-trained protein language models.

Model	Total train compute (FLOPs)	Params	Training tokens
ESM150M	1.1E+21	150M	1,000B
ESM650M	4.4E+21	650M	1,000B
ESM3B	1.8E+22	2.8B	1,000B
ESM15B	5.1E+22	15B	864B
ProtBert	2.5e+12	2.8B	1,929B
ProtT5-xl	1.7E+22	2.8B	1,929B
ProtT5-xxl	3.7E+22	11B	1,039B
Ankh-base	2.6E+21	740M	952B
Ankh-large	6.5E+21	1.9B	952B
ProtGPT2	5.3E+21	740M	-
ProGen	7.6E+21	1.2B	1,049B
ProGen2-small	1.8E+20	150M	170B
ProGen2-medium	8.9E+20	760M	170B
ProGen2-base	1.1E+21	760M	200B
ProGen2-large	3.4E+21	2.8B	200B
ProGen2-xlarge	1.4E+22	6.4B	350B
xTrimoPGLM-Ab-1B	8.5E+21	1.2B	1,000B
xTrimoPGLM-100B	6.2E+23	100B	1,000B

D Pre-training Configurations

The detailed parameters for training the xTrimoPGLM-100B model are listed in Table 10. Hyperparameters for fine-tuning settings are also included.

E Generated Structures

We first produced batches of samples with an n-gram penalty (N-gram=3) to reduce the probability of generating repetitive sequences. However, we find many examples exhibiting low-complexity sequences (e.g., local repeats), where the predicted structures contain long loop disorder regions. We hypothesize that the n-gram penalty potentially impedes the model's capacity to generate grammatically correct sequences with ease. Once we remove the n-gram penalty, the generated structures tend to be more natural.

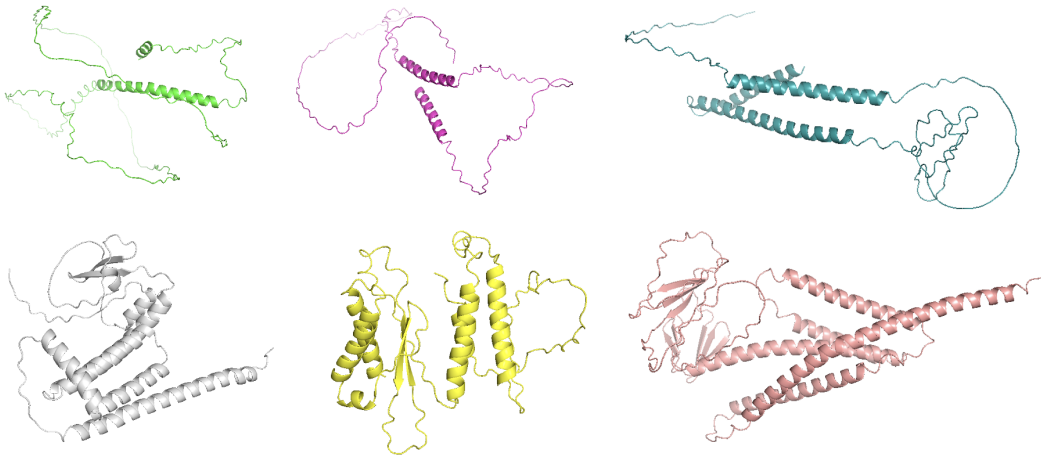


Figure 13: Structure examples of generated protein sequences with different parameter configurations. The first row depicts sequences with parameter (T=1.0, P=1.0, N-gram-penalty=3), while the second row removes the n-gram constraints to reduce long loop disorder regions.

Table 10: Full configurations for xTrimoPGLM-100B training

KEY	VALUE
glu_activation	GeGLU
hidden dim.	10,240
ffn size	31,744
# layers	72
# attention heads	80
sequence_length	2,048
global batch size	4,224
max learning rate	4e-05
min learning rate	4e-06
adam_beta1	0.9
adam_beta2	0.95
adam_eps	1e-08
aggregated_samples_per_sequence	1,2,4,8
attention_dropout	0.1
attention_softmax_in_fp32	True
average_block_length	6
bias_dropout_fusion	True
checkpoint_activations	True
checkpoint_in_cpu	False
checkpoint_num_layers	9
clip_grad	1.0
tensor_parallel_size	4
pipeline_parallel_size	8
data_parallel_size	24
deepnorm	True
distributed_backend	nccl
eval_interval	300
fp16	True
mlm_prob	0.1
span_prob	0.2
gpt_prob	0.7
hidden_dropout	0.1
init_method_std	0.0052
initial_loss_scale	65536
layernorm_epsilon	1e-05
rotary_embedding	2D
learnable_rotary_embedding	False
length_per_sample	2048
log_interval	1
lr_decay_iter	None
lr_decay_samples	439,453,125
lr_decay_style	cosine
lr_warmup_samples	14,648,437
make_vocab_size_divisible_by	128
masked_softmax_fusion	True
micro_batch_size	1
min_gmask_ratio	0.4
min_loss_scale	1.0
optimizer	adamw
partition_activations	True
rampup_batch_size	240,24,12207031
save_interval	300
seed	1234
short_seq_prob	0.02
shrink_embedding_gradient_alpha	0.1
single_span_prob	0.02
split	949,50,1
tokenizer_type	ProteinTokenizer
weight_decay	0.1
zero_stage	1
FINETUNE	
lora_(R, α)	(8,16),(16,32)