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

Bo Chen[2*†], Xingyi Cheng[1*‡]
Pan Li[1], Yangli-ao Geng[2†], Jing Gong[1], Shen Li[1], Zhilei Bei[2†], Xu Tan[1],
Boyan Wang[2†], Xin Zeng[1],Chiming Liu[1], Aohan Zeng[2], Yuxiao Dong[2]
Jie Tang[2§], Le Song[1,3§]
[1]BioMap Research [2]Tsinghua University [3]MBZUAI

January 11, 2024

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. We propose 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 1 trillion training tokens. Our extensive experiments reveal that 1) xTrimoPGLM significantly outperforms other advanced

*Equal Contribution. Emails: cb21@mails.tsinghua.edu.cn, xingyi@biomap.com
†Work done during 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

baselines in 18 protein understanding benchmarks across four categories. The model also facilitates an atomic-resolution view of protein structures, leading to an advanced 3D structural prediction model that surpasses existing language model-based tools. 2) xTrimoPGLM not only can generate de novo protein sequences following the principles of natural ones, but also can perform programmable generation after supervised fine-tuning (SFT) on curated sequences. These results highlight the substantial capability and versatility of xTrimoPGLM in understanding and generating protein sequences, contributing to the evolving landscape of foundation models in protein science.

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. 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 (*1, 2*). As protein sequences serve as the blueprint for protein structure and function (*3*), pre-trained techniques on sequences, known as **P**rotein **L**anguage **M**odels (PLMs), e.g., the family of ESM models (*4, 5*), ProtTrans (*6*), ProGen (*7*), etc., offer a powerful tool for characterizing the properties and distributions of general protein sequences. These models are trained on large-scale protein datasets (*1, 8–10*) 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 (*1, 5*) or generating novel sequences with faithful three-dimensional structures (*7, 11*).

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 (*3, 12, 13*), 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, existing 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. Consequently, we explore the feasibility of integrating tasks of understanding and generation, dictated by autoencoding and autoregressive pre-training objectives, respectively, into one unified framework. This unified approach aims to encapsulate the intricate dependencies inherent in protein sequences, potentially resulting in more versatile and robust protein foundation models.

Large Language Models (LLMs) have explored the revenue of developing unified pre-training paradigms. However, these studies typically adopt analogous training patterns. For instance, all pre-training objectives are commonly optimized using either the BERT-style (*14, 15*) or GPT-style regime (*16*). A balanced approach incorporating both bi-directional auto-encoding and uni-directional auto-regressive objectives could fulfill the requirements of unified PLMs, yet the feasibility of such integration remains an open question. Practically, the current landscape of natural language processing 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 (*17*) or instruction-tuning (*18, 19*). 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 discrete values of categories or continuous values of 3D coordinates (*5, 20*). These tasks heavily rely on bi-directional auto-encoding training to tackle protein understanding tasks. Consequently, this highlights the need for a unified model that

incorporates both training objectives.

Key Results

Our first key result is the xTrimo Protein General Language Model (xTrimoPGLM), a unified pre-training framework and foundation model designed for various protein-related tasks, including understanding and generation (or design). The model differs from previous encoder-only (e.g., ESM) or causal decode-only (e.g., ProGen) protein language models by leveraging the General Language Model (GLM) (*21*) as the backbone for its bidirectional attention and auto-regressive objective. To enhance the representation capacity of xTrimoPGLM, we further introduce the Masked Language Model (MLM) objective to the bidirectional prefix region, building upon the generation ability encapsulated within the GLM objective. Additionally, we compiled a large pre-training dataset, comprising approximately 940 million unique protein sequences with roughly 200 billion residues, and trained a model with 100 billion parameters over 1 trillion tokens over a cluster of 96 NVIDIA DGX machines each with 8$\times$A100 GPU cards. To further justify the superiority of the xTrimoPGLM framework, we train an antibody-specific model (xTrimoPGLM-Ab) using the same framework with 1 billion parameters. This model sets a new record in predicting antibody naturalness and structures, both essential to the field of antibody-based drug design, and demonstrates a significantly faster inference speed than AlphaFold2, see supplementary material (SM) section 2.

The second key result is a significant enhancement in the realm of protein understanding with the xTrimoPGLM-100B as a protein foundation model. By conducting extensive empirical experiments with linear probing and advanced fine-tuning techniques, we elevated the performance benchmarks in this domain. xTrimoPGLM-100B has significantly surpassed previous state-of-the-art (SOTA) methods in 15 out of 18 tasks, covering a comprehensive range of areas including protein structure, interactions, functionality, and developability (Figure 2A). We also

illustrate that xTrimoPGLM achieves lower Perplexity (PPL) on two Out-Of-Distribution (OOD) protein sets over other models (Figure 1B). These results empirically validate the scaling behavior, demonstrating that larger models commonly tend to yield better performance (Figure 1C & Figure 2B).

The third key result is the development of a high-performance 3D structural prediction tool based on xTrimoPGLM. Inspired by methodologies similar to those in ESMFold, merge folding modules with a protein language model, thereby refining protein structure training. Our version named xTrimoPGLM-Fold (xT-Fold for short) has shown promising results with impressive TM-scores in both CAMEO (n=194) and CASP15 (n=56) protein benchmarks. Additionally, we optimized xT-Fold through 4-bit quantization, enhancing its performance and efficiency, which makes xT-Fold a leading option in PLM-based structure prediction tools. As a result, xT-Fold achieves a 5-point TM-score increase over ESMFold in the CASP15 dataset, coupled with a faster inference speed across various scenarios (Figure 3).

The final result demonstrates that xTrimoPGLM showcases an extraordinary ability to generate de novo protein sequences. These sequences not only exhibit diverse structures closely akin to natural counterparts, as evidenced by a median sequence identity of just 11.7% (Figure 4), but can also be tailored towards specific structural and biophysical properties through supervised fine-tuning (Figure 5 & Figure 6). This "super alignment" capability of xTrimoPGLM underscores its potential as a programmable model for exploring and synthesizing the vast protein space.

Lastly, we discuss the key limitations of our protein language model in practical protein applications. Although our study confirms the potential of protein language models, it also highlights that critical enhancements are necessary for their effective deployment in real-world drug design. These enhancements include adapting models to diverse protein tasks, improving prediction accuracy for protein structures, and reducing generative protein hallucinations. Overcoming

these challenges is essential to bridge the gap between theoretical capabilities and their practical application in drug discovery and development.

The xTrimoPGLM Framework

We adopt the GLM as our foundational framework to exploit its strengths in autoregressive blank infilling for training, while simultaneously processing input text bi-directionally. This dual approach is enhanced by the integration of a Masked Language Model (MLM) objective (*22*) to enhance its understanding capacity. The core of our model is the simultaneous optimization of two distinct pre-training objectives, each characterized by unique indicator tokens, ensuring the proficiency in both understanding and generation capacities:

- **Masked Language Model (MLM) Objective**: This task involves the prediction of tokens that have been randomly masked within sequences. These tokens are indicated by the special marker `[MASK]`. This task aligns with the functionality of BERT (*22*) and ESM (*5*), focusing on bidirectional contextual understanding.

- **General Language Model (GLM) Objective**: This task entails predicting subsequent tokens in a sequence, including short masked spans (indicated by `[sMASK]`) and longer spans at sequence ends (marked by `[gMASK]`). While the GLM objective takes into account the unidirectional context for predicting subsequent words, the prefix encoding portion remains bidirectional.

The framework of xTrimoPGLM, along with its application in downstream tasks, is depicted in the lower panel of Figure 1A. Motivated by the philosophy of curriculum learning, the pre-training stage is conducted in two distinct phases: 1). Initial pre-training with the MLM objective, focusing on rapid loss minimization across approximately 400 billion tokens. This phase is geared towards enhancing the model's understanding capabilities, 2). Subsequent

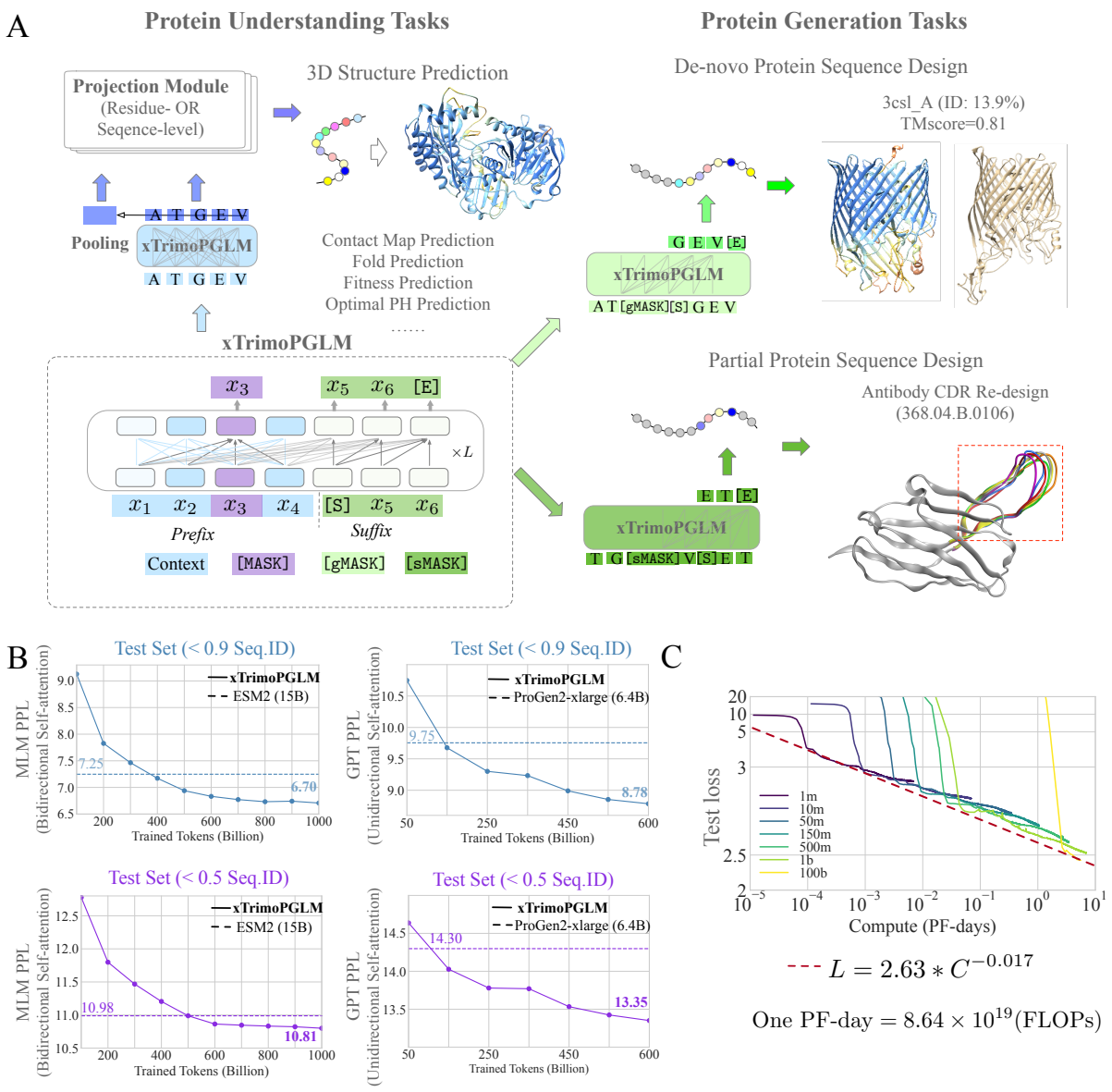


Figure 1: **Comprehensive Insights into xTrimoPGLM. A.** xTrimoPGLM Development Phases: illustrates the pre-training and fine-tuning stages of xTrimoPGLM, combining GPT-style and BERT-like objectives, and applying learned embeddings in protein understanding and design tasks. **B.** Perplexity Comparison: xTrimoPGLM shows lower perplexity than leading models like ESM2 and ProGen2-xlarge in evaluations on two distinct out-of-distribution datasets, indicating its advanced performance. **C.** Scaling Law Analysis: examines the scalability of models from 1 million to 1 billion parameters, trained with 100 billion tokens, demonstrating xTrimoPGLM-100B's efficiency through a power law fit of training losses against computational resources.

training employs a unified approach, merging MLM and GLM objectives at a specific ratio (20% MLM, 80% GLM). This stage, utilizing an additional 600 billion tokens, is dedicated to refining both the model's representational and generative abilities.

Quantification Scaling Law of xTrimoPGLM. The scaling law is a crucial concept for understanding the performance of large language models (LLMs) during pre-training. It suggests a power-law relationship between a model's performance, typically measured by cross-entropy test loss $L(C)$, adheres to a power-law relationship with the computer resources C used in training. To investigate the scaling law for xTrimoPGLM in the context of protein data, we employ the formula $L = a \times C^b$ where C as outlined by (*23, 24*). We quantify C in terms of training floating point operations (FLOPs), using PF-days (PetaFLOP-days) as the unit of measure, where one PF-day$= 8.64 \times 10^{19}$ FLOPs. We extract a range of $(C, L(c))$ data points from the loss trajectories of models varying from 1 million to 1 billion parameters, each trained with 100 billion tokens. The resultant scaling curve, fitted to these data points, is depicted in Figure 1C. Remarkably, this curve demonstrates a close alignment with the actual training loss observed for the 100 billion parameter model of xTrimoPGLM, trained with the same volume of tokens. This observation substantiates the model's adherence to the anticipated scaling law, thereby reinforcing the credibility of this principle as a pivotal guideline in the development of large-scale protein language models.

xTrimoPGLM Achieves Low Perplexity on Out-Of-Distribution Protein Sequences. Perplexity is a metric in language modeling that quantifies the uncertainty of a probability model in predicting a text sample, where lower values signify greater predictive accuracy. For a comprehensive evaluation, we construct two OOD datasets using a specific process: (1) Sampling an extensive collection of protein sequences from UniProt that were uploaded after January 2023, which is also the cutoff date for the training data. And (2) Filtering these sequences based on sequence identity thresholds of 0.5 and 0.9 with the pre-training datasets. Each OOD dataset

comprises approximately 10,000 protein sequences. The results are demonstrated in Figure 1B, xTrimoPGLM-100B achieves perplexity scores of 10.81 and 6.70 on the 0.5 and 0.9 sequence identity datasets, respectively, outperforming 15 billion parameters ESM2-15B (10.98 and 7.25). Similarly, against ProGen2-xlarge with 6.4 billion parameters, xTrimoPGLM recordes scores of 13.35 and 8.78, compared to ProGen2-xlarge's 14.30 and 9.75. We observed that large-scale models are more sample efficient, reaching these perplexity levels with substantially less training data: less than 500 million tokens to match ESM2 trained on 1 trillion tokens, and less than 150 billion tokens to parallel ProGen2-xlarge trained with 350 billion tokens, even if the learning rate schedule has not yet ended, resulting in an overestimated loss value at this stage.

Extensive Benchmarks of 18 Protein Understanding Tasks

To comprehensively assess the understanding capabilities of xTrimoPGLM-100B, we benchmark it against 18 downstream protein-related tasks. These tasks span four primary categories: protein structure, developability, interactions, and functions, as detailed in Figure 2A. A thorough description of these datasets can be found in SM Section 1.8. We compare xTrimoPGLM-100B with two other protein language models in the field, ESM2-150M and ESM2-15B (*5*), to provide a well-rounded evaluation. Our assessment methodology encompasses two distinct approaches to evaluate the effectiveness of the models' representations:

- **Probing with MLP.** We utilize a trainable multi-layer perceptron (MLP) as a probe to examine the evolutionary information encoded in the pre-trained representations. This method simply and efficiently identifies the types of protein information captured by the models, where pre-trained PLMs' parameters remain fixed, and only the MLP is trained. For comparisons, the embeddings from pre-trained models are projected into 128 dimensions followed by ReLU activation before passing to the next layer of MLP.

- **Fine-tuning with LoRA.** Considering the constraints of GPU memory, full-scale fine-tuning is not feasible for models with parameters on the scale of 100 billion. Hence, we employ Low-Rank Adaptation (LoRA) (*25*) as a parameter-efficient alternative. This technique involves freezing the pre-trained model's weights and adding trainable low-rank matrices to each transformer layer. LoRA significantly reduces the number of trainable parameters for downstream tasks while preserving the adaptability of the learned representations. The fine-tuning architecture and settings are similar to those in MLP probing, with only the transformer's W_q, W_k, W_v, W_o parameters being fine-tuned.

Figure 2A provides a comprehensive visualization of performance across all benchmarked protein-related tasks. Distinct colors and shapes in the figure represent different evaluation strategies and models, respectively. The ESM2-150M model, being relatively smaller, serves as an indicator to assess the difficulty of these tasks. The performance distribution highlights the inherent relationships between the complexity of tasks and the advantages brought by the scale of the model. The distribution of performance across tasks underscores the relationship between task complexity and the benefits derived from the scale of the model. In more complex tasks, such as Contact Map prediction (Protein Structure category), Fluorescence (Protein Function), Metal Binding (Protein Interaction), and Stability (Protein Development), the larger models (xTrimoPGLM-100B and ESM2-15B) significantly outperform the smaller ESM2-150M. This disparity in performance highlights the necessity for more advanced models to effectively address complex tasks. Conversely, for simpler tasks, like Antibiotic Resistance (Protein Function category), the performance gap between the large and small models is notably smaller. This observed pattern suggests that larger models are more adept at capturing the intrinsic evolutionary information of protein sequences. Consequently, as models scale up, they exhibit marked improvements in performance, particularly for complex tasks. The application of LoRA consistently enhances overall task performance compared to the static Probing method,

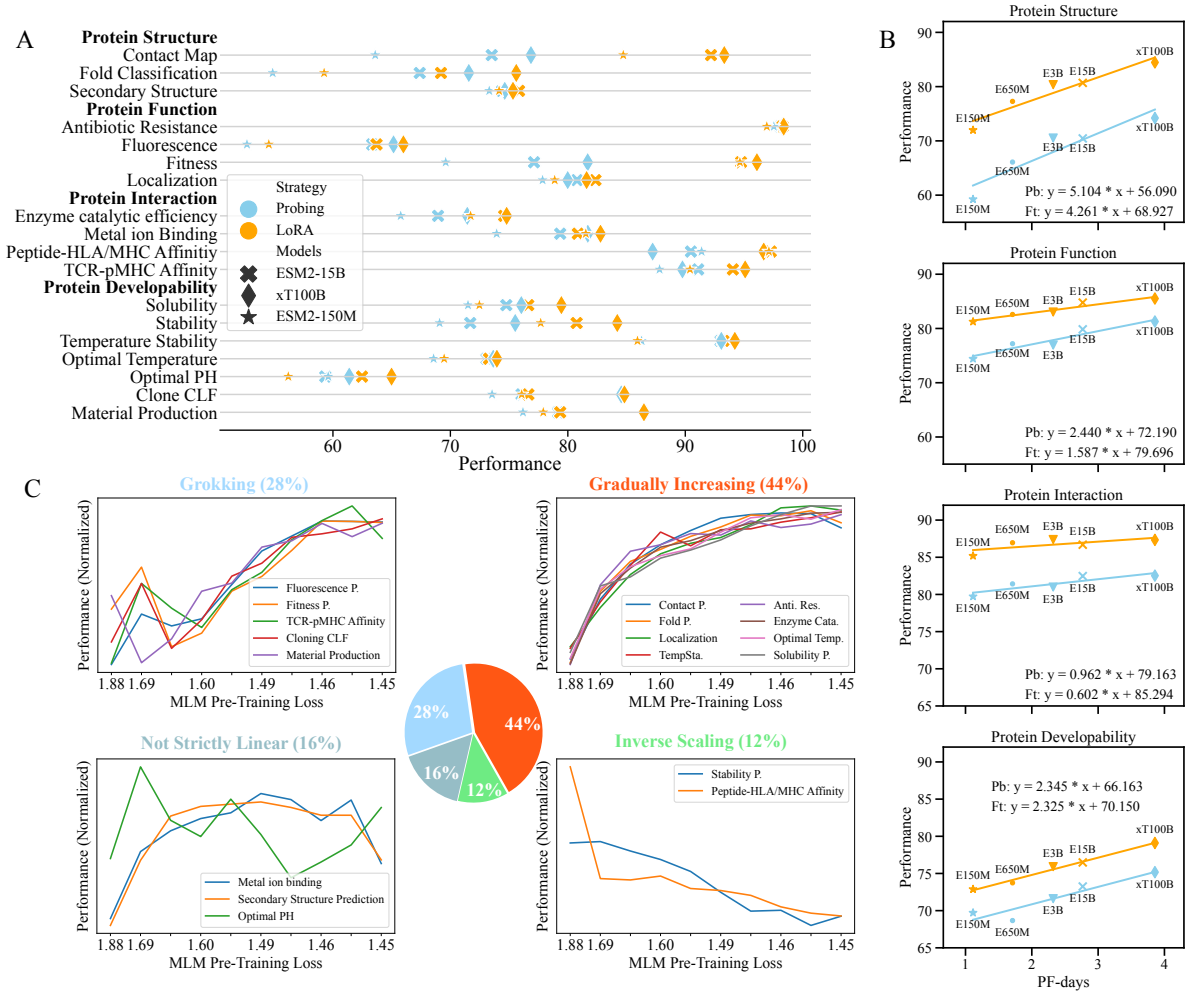


Figure 2: **The Performance of Protein Understanding Benchmark. A.** Visualization of Model Performance across all Tasks. **B.** The scaling trend between the computational complexity of model training, quantified by PF-days, and the model performance across four distinctive task categories, each encompassing 3-4 subtasks. Each data point symbolizes the mean performance metric for a specific task category, where Pb stands for Probing and Ft for Fine-tuning with LoRA techniques. The labels E150M, E650M, E3B, E15B, and xT100B represent ESM2-150M, ESM2-650M, ESM2-3B, ESM2-15B, and xTrimoPGLM-100B, respectively. **C.** Correlations between the Pre-Training validation loss measured by MLM objective and the downstream task performance. To facilitate comparison, we normalize this performance by subtracting the mean value and dividing by the standard deviation. Most tasks (44% Gradually Increasing, 28% Grokking, and 16% Not Strictly Linear) obey the trends that the task performance demonstrates the positive correlations with the ongoing pre-training process.

which limits the capacity of pre-trained models (as illustrated in Figure2B). LoRA's efficiency lies in its ability to refine and utilize key features without significantly increasing the number of trainable parameters (less than 1% increase with LoRA rank set to 8). The empirical findings also highlight a scaling trend in task performance with supervised fine-tuning, suggesting a strong correlation between model scale and performance. The following sections will delve deeper into the varying scaling behaviors observed across different types of tasks.

Scaling Behaviors of Downstream Tasks. We extend the investigation of scaling laws to downstream tasks. We examine the correlation between computational complexity (measured by PF-days) during model training and performance across four distinct categories of protein-related tasks. Each category encompasses 3-4 subtasks, analyzed using both Linear Probing with MLP (Pb) and Fine-tuning with LoRA (Ft) techniques (Figure 2B). While the intensity of the scaling effect differs among task types, a common trend observed from pre-training extends to these downstream tasks: an exponential increase in computational resources during pre-training correlates with linear improvements in task performance. This extension of the scaling law to downstream tasks in protein modeling is a novel observation.

A prevalent assumption posits that large language models, due to their higher computational complexity, universally outperform smaller models across all tasks. To validate this, we take a deep insight into the correlations between the downstream task performance with the ongoing pre-training process (measured by the MLM validation loss), as shown in Figure 2C. We observe that most tasks demonstrate positive correlations. Specifically, 44% show a gradual increase in performance, 28% exhibit a "Grokking" phenomenon, and 16% do not follow a strictly linear pattern. The 'Grokking' phenomenon mirrors emergent abilities seen in large NLP models (*26*). It occurs when models initially overfit the training data, especially when task datasets have limited overlap with pre-training data. As the model's knowledge base expands, it begins to apply its understanding to out-of-distribution (OOD) scenarios, resulting in a sudden increase

in test data performance. Thus the performance of the test data suddenly increased, denoted as the Grokking phenomenon. However, 16% of tasks indicate a potential negative impact from increased computational resources, suggesting an inverse scaling effect.

Our findings highlight the potential of scaling up models to enhance performance across a broad spectrum of protein-related tasks. This approach contrasts with other methods that seek data-efficient, cost-reduced, and knowledge-guided PLMs without relying on large-scale language models (*27*). These empirical observations provide valuable insights for future research in model development and advancement in the field of protein language modeling.

Evaluation on Protein Structure Prediction

Protein sequences encode rich information about their 3D structures and function through evolutionary history (*3*). Advanced methods using multiple sequence alignments (MSAs) like AlphaFold2 (*1*) and RoseTTAFold (*28*) are highly accurate in predicting protein structures and are key tools in computational biology. Meanwhile, PLM-based models such as ESMFold (*5*) and OmegaFold (*29*), while not as precise as MSA-based models, provide faster predictions. This rapid prediction is crucial for high-throughput applications, accelerating our understanding of biological mechanisms and hastening drug discovery efforts.

In this section, we propose xT-Fold, where building on the xTrimoPLGM-100B framework, marks a significant advancement by achieving SOTA results for the PLM-based structure prediction model on benchmarks such as CAMEO and the latest CASP15. Notably, the model offers both high accuracy and computational efficiency, utilizing 4-bit quantization and FlashAttention (*30*) techniques to run effectively on a single A100 GPU card (SM Section 1.9) This balance of speed and precision makes xT-Fold a tool choice for fast-paced research and drug discovery.

The overall architecture of xT-Fold closely resembles that of ESMFold (Figure 3A). It involves training Multi-Layer Perceptron (MLP) layers to map the respective single representation

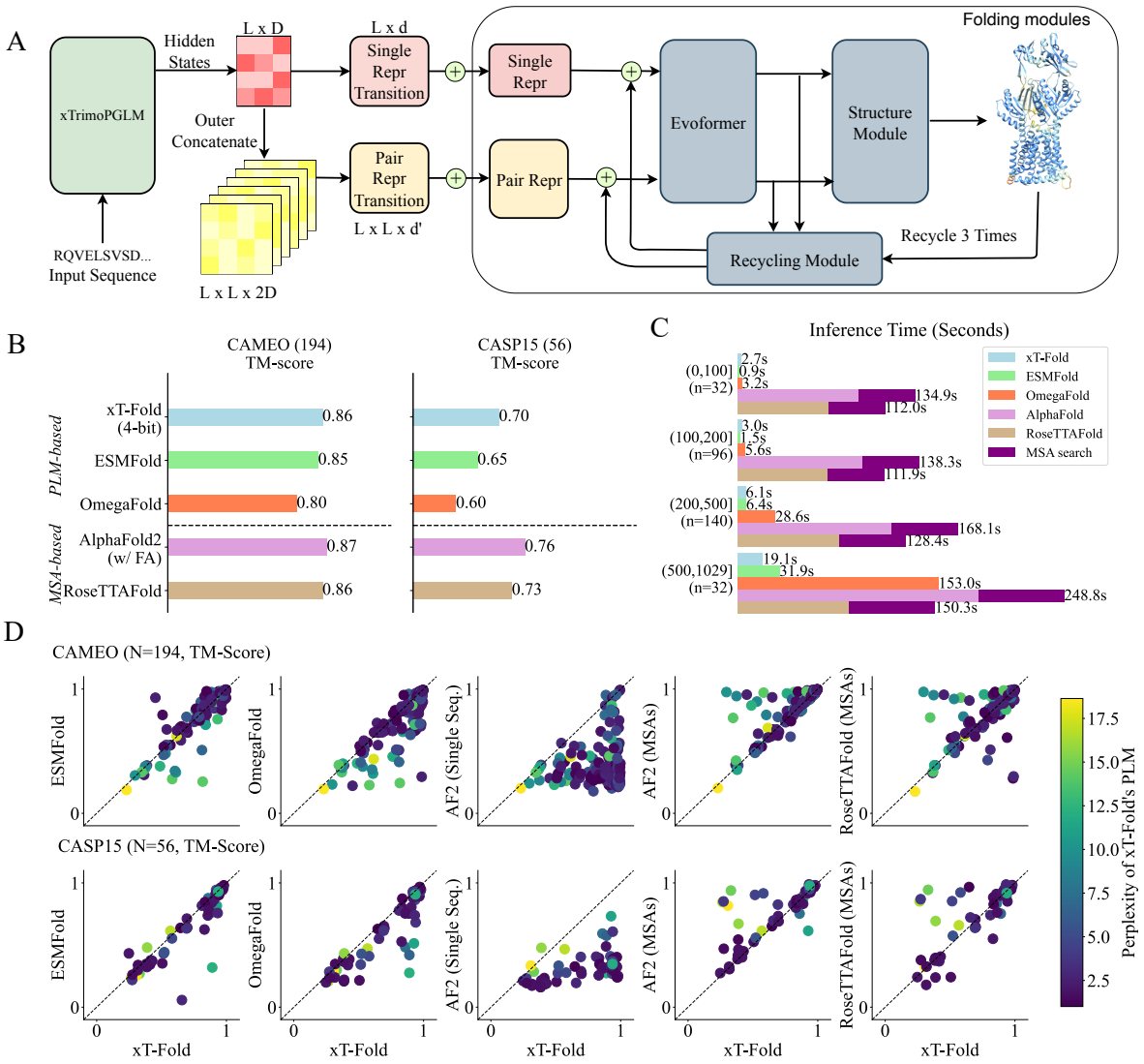


Figure 3: **Structure Prediction with xT-Fold. A.** xT-Fold's architecture utilizes a Multi-Layer Perceptron (MLP) connecting language model representations to folding modules, which then generate 3D coordinates and confidence scores (pLDDT). **B.** TM-score benchmarks for structure prediction models. The bar chart shows the performance of single-sequence PLM-based models and MSA-based models on CAMEO and CASP15 datasets. **C.** Inference time comparison across models for varying sequence length intervals, showing xT-Fold, ESMFold, OmegaFold, AlphaFold, RoseTTAFold, and MSA search times in seconds.**D.** Scatter plots compare xT-Fold predictions (x-axis) to other models (y-axis), color-coded by perplexity (green for high, purple for low).

and pair representation to d and d' from D dimension, which are fed into the structure module for 3D coordinates prediction. xT-Fold performs at most three recycling steps before outputting the final atomic-level structure.

We evaluated two individual test sets, CAMEO and CASP15, both of which are out-of-distribution in terms of their timelines, ensuring differences from our training set (SM Section 1.10). CAMEO includes 194 samples (released date between April and June 2022), while CASP15 consists of 56 publicly available proteins (released in May 2022). Initially, we observed the perplexity (the lower are better) of these two datasets on PLMs, which served as the backbones of the folding models (Figure S13). On CAMEO and CASP15, xTrimoPGLM achieved a perplexity of 4.01 and 4.45, respectively, in contrast to ESMFold's PLM (ESM2-3B), which scored 5.21 and 6.18. The perplexity (PPL) demonstrates that language models generally have a better understanding of the CAMEO dataset compared to CASP15. This suggests that CAMEO might be less challenging than CASP15, indicating that CASP15, relative to CAMEO, is closer to OOD data for language models.

We conducted a comparative analysis against other MSA-free PLM-based models (Figure 3B), including ESMFold, OmegaFold, and 4-bit xT-Fold. In performance evaluation, xT-Fold achieved a TM-score of 0.86 on the CAMEO dataset and 0.70 on CASP15. The scores for ESMFold were 0.85 and 0.65, respectively, while OmegaFold scored 0.80 and 0.60 on these datasets. When compared to methods that integrate MSAs and template retrievals, such as AlphaFold2 and RosettaFold, PLM-based models like xT-Fold and ESMFold showed performance closely matching that of RosettaFold on the CAMEO dataset. However, on the more OOD dataset, i.e. CASP15, the overall efficacy of the PLM-based methods still lagged behind MSA-augmented approaches.

On the other hand, when benchmarked on the CASP15 and CAMEO test sets across various sequence length intervals, the inference speed of MSAs/template-based models is notably

slower than that of PLM-based models (as shown in Figure 3C), lagging by approximately 10x to 50x. This disparity remains even for accelerated AlphaFold2, which is optimized with FlashAttention (FA), achieving an acceleration of 2x to 8x compared to the open-source variant (available at `https://github.com/google-deepmind/alphafold`) across sequence lengths ranging from 200 to 1000. For the PLM-based methods, the open-source versions of ESMFold (`https://github.com/facebookresearch/esm`) and OmegaFold (`https://github.com/HeliXonProtein/OmegaFold`) were utilized. As result, xT-Fold overall exhibits a marginal speed advantage over ESMFold and OmegaFold, due to the optimization with FA. Particularly on longer sequence intervals, xT-Fold demonstrates its superiority. The relative slowness of OmegaFold is attributed to its default setting of using 10 recycling passes to achieve better results.

Upon further examination of the scatter plots (Figure 3D), a perceptible correlation between the perplexity from xTrimoPGLM and the structural metric TM-score is observed. Notably, for samples with intermediate PPL values, xT-Fold demonstrates enhanced performance compared to ESMFold and OmegaFold. In the third column of Figure 3D, xT-Fold is juxtaposed with AlphaFold2 with single sequence input. Given that AlphaFold2 is not trained on single sequences, a significant drop in overall effectiveness is evident. The last two columns compare xT-Fold against AlphaFold2 and RosettaFold. It is apparent that where PPL is moderately high, corresponding to less accurate predictions, xT-Fold's performance is less effective than the two methods, For samples with lower PPL predictions, xT-Fold occasionally surpasses RosettaFold. In summary, while scaling single-sequence models enhances the performance, but still struggles with OOD data, which is effectively addressed by MSAs augmentation.

Evaluation on Protein Sequence Generation

Autoregressive models have emerged as powerful tools for representing the diverse array of evolutionary sequences found in nature. This capability facilitates the generation of novel protein sequences, exhibiting diverse folding patterns that significantly diverge from naturally occurring proteins (*11, 31, 32*). To validate the generative ability of xTrimoPGLM-100B, we conduct an extensive analysis of the properties of protein sequences synthesized by xTrimoPGLM-100B under various generative scenarios. Our investigation spans several generative contexts: universal protein synthesis utilizing pre-training data and task-specific sequence generation via Supervised Fine-Tuning (SFT), sequence creation following Reinforcement Self-Training (ReST). This multifaceted approach provides deep insights into the potential and versatility of xTrimoPGLM-100B in advancing protein sequence generation.

xTrimoPGLM Generate Sequences with Diverse Structures. To evaluate the generative capacity of xTrimoPGLM , we generate 2,861 sequences with the xTrimoPGLM-100B model utilizing `[gMASK]` indicators as the inserting prompt. This process generates new sequences by continuously predicting the next token in an auto-regressive manner until the `<eos>` token is predicted or the pre-set maximum length is reached. We use nucleus sampling (temperature=0.8, top p=0.8) to increase the generation diversity. For comparison, we also generated 2,759 sequences with the ProGen2-xlarge model with the same hyper-parameters (temperature=0.8, top p=0.8).

We employed xT-Fold to predict the structures of all generated sequences. Our model generated proteins with higher confidence scores than ProGen2-xlarge (median pLDDT scores 85.4 vs 80.8, Figure 4A upper). We then used Foldseek to search for the most structurally similar natural proteins from the PDB database. We measured the structural and sequence similarity using TM-score and sequence identity, respectively. Our model exhibited much higher structural

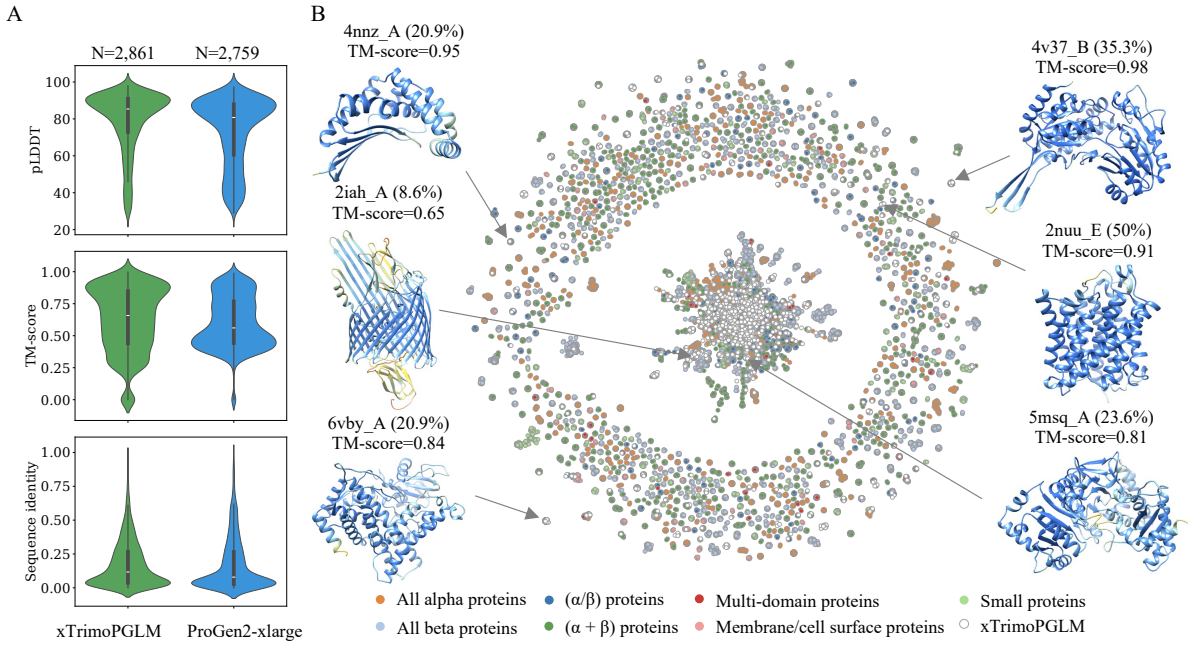


Figure 4: **Diversification of Generated Proteins by xTrimoPGLM. A.** Structural Prediction Confidence and Natural Protein Analogues: We assess the xT-Fold predicted confidence (pLDDT scores), the resemblance to proteins cataloged in the Protein Data Bank (TM-score and sequence identity) of generated sequences by xTrimoPGLM (green, N=2,861) and ProGen2-xlarge (blue, N=2,759). The violin plots mark the median, upper and lower quartiles, and 1.5× interquartile range (whiskers). **B.** Cartography of Protein Structure Space: We present a comprehensive mapping of protein structural space as informed by sequences generated by xTrimoPGLM. Each node represents a sequence generated by xTrimoPGLM or a sequence from SCOPe70_2.08. Two nodes are linked when one of them can be searched from SCOPe70_2.08 with an alignment of at least 20 amino acids and 70% hhsearch probability. The color coding corresponds to distinct SCOP structural classes, with xTrimoPGLM-generated sequences highlighted in white. For illustrative purposes, we extract and show 6 examples from generated sequences. The the PDB chain ID with the highest structural similarity to the generated sequence, their sequence identity and TM-score are displayed above each example. The color of structure matches to the xT-Fold pLDDT values. The blue color represent high confidence (pLDDT>90). 4nnz_A: Probable zinc protease. 2iah_A: Ferripyoverdine receptor. 6vby_A: Cinnamic acid 4-hydroxylase. 4v37_B: Betaine aldehyde dehydrogenase. 2nuu_E: Ammonia channel. 5msq_A: Carboxylic acid reductase. More visualized cases generated by xTrimoPGLM-100B refer to Figure S10.

resemblance to PDB entries than ProGen2-xlarge (median TM-score of 0.658 vs 0.560) with very low sequence identity (median sequence identity of 0.117 vs 0.079).

This potentially allows us to access a larger sequence in the protein manifold while we try to design certain protein structures. Yet, the critical question remains: Does xTrimoPGLM have the capability to not only replicate but also expand the known protein universe, introducing novel topologies and functions? Figure 4B showcases a comprehensive network of the protein structural space, informed by sequences synthesized by xTrimoPGLM. Each node corresponds to a xTrimoPGLM-generated sequence or a sequence from SCOPe70_2.08. Sequences originating from xTrimoPGLM are distinctly marked in white. This network vividly illustrates that xTrimoPGLM successfully generates novel protein sequences encompassing a broad spectrum of protein folds, while maintaining low sequence identity, indicating the model's potential in pioneering new functional and structural realms. A further illustration of additional generative examples and hyper-parameter analysis refers to SM Section 1.12.

Enhanced Protein Sequence Generation through Supervised Fine-Tuning and Reinforcement Self-Training. The xTrimoPGLM-100B excels in generating diverse protein sequences but faces challenges in aligning to produce sequences with specific properties or families, such as lysozymes or immunoglobulins. This limitation is a critical bottleneck for applications in various industries, including pharmaceuticals and agriculture. Adopting strategies from OpenAI's GPT models (*17*), xTrimoPGLM-100B serves as a protein foundational model, equipped with vast knowledge from trillions of residue tokens. We apply established alignment methods like Supervised Fine-Tuning (SFT) (*33*) on select protein families and enhance this with Reinforcement Self-Training (ReST) (*34*), based on the SFT model. We fine-tune xTrimoPGLM-100B on datasets representing common protein structures or chemical properties. We choose five tasks from 18 benchmark protein understanding tasks (Fold Prediction, Temperature Stability, Localization Prediction, Fluorescence Prediction, and Fitness Prediction) for fine-tuning, detailed in SM Section 1.11. Comparative analysis is conducted against ProtGPT2 and ProGen2 using identical SFT protocols. To circumvent trivial results, we employ the non-fine-tuned

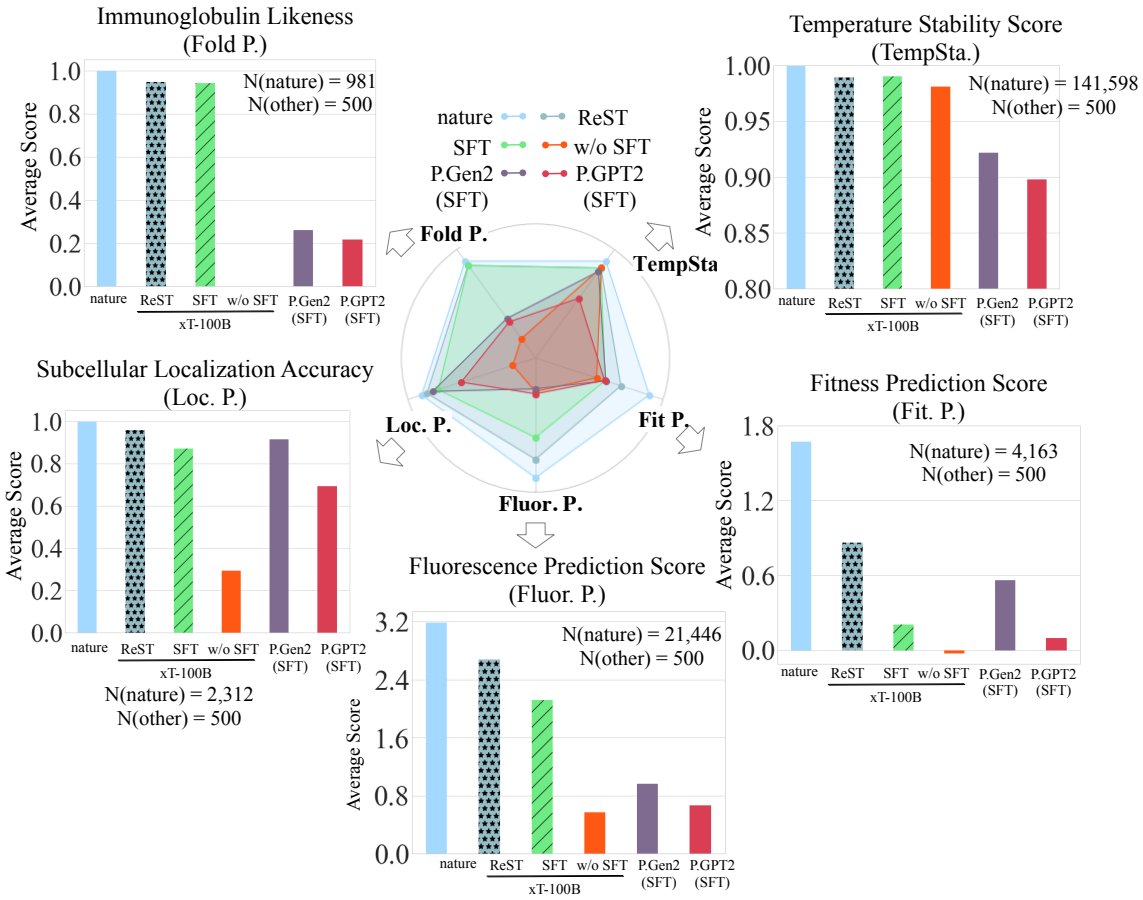


Figure 5: **Robust Alignment Capabilities of xTrimoPGLM in Protein Sequence Generation towards Desired Properties.** Quantitative result in five selective tasks: We present a detailed quantitative analysis of xTrimoPGLM, enhanced through Supervised Fine-Tuning (SFT) and Reinforcement Self-Training (ReST), across five specifically chosen tasks. The analysis highlights xTrimoPGLM's proficiency in achieving alignment with specific task objectives, as evidenced by the average scores (where higher scores indicate superior alignment performance).

xTrimoPGLM-100B as a control. Task-specific predictors evaluate the quality of the generated sequences, acting as biased evaluators. Due to the impracticality of in vitro validation, we used in silico simulators, a common approach in prior research (*11, 31*). Our findings, illustrated in Figure 5, reveal that: 1). Sequences from xTrimoPGLM with SFT consistently score higher on targeted properties than the non-fine-tuned baseline, 2). xTrimoPGLM surpasses ProGen2 and ProtGPT2 under the same SFT conditions for most tasks, underscoring its efficacy as a

foundational model capable of superior alignment with minimal data or fewer tuning steps, in line with observed scaling behavior. Furthermore, we implement a one-step Reinforcement Self-Training process, which utilizes task predictors as reward models, guiding the self-training of the SFT-enhanced xTrimoPGLM-100B as follows: 1). Task predictors evaluate the quality of sequences from the SFT model, 2). Sequences of higher quality are then selected to form a new dataset, which is used for further fine-tuning. This iterative process results in the development of the Reinforcement Self-Training (ReST) model. Remarkably, the ReST model effectively synthesizes sequences closely resembling natural datasets, highlighting xTrimoPGLM's potential as a robust protein synthesizer for industrial applications.

To further showcase that xTrimoPGLM can generate proteins that mimic the natural sequences with the SFT alignment pipeline. We fine-tune the model on four SCOP fold-type sequences. Quantitative analyses exploring the relationship between structural prediction confidence, as indicated by xT-Fold's pLDDT scores, sequence identities, and their structural counterparts in the Protein Data Bank (PDB), as measured by TM-score, are presented in Figure 6. These findings underscore xTrimoPGLM's exceptional capability in generating protein sequences that not only embody specific structural characteristics but also align closely with established PDB entries, thereby reinforcing its potential as a tool for synthesizing proteins with targeted structural attributes.

Limitations

Challenges of Model Scalability

A significant limitation of xTrimoPGLM-100B is the high computational cost associated with the model, which presents a considerable barrier to their practical application. Fine-tuning without quantization requires at least four A100 80G GPUs, which may not be feasible for all users. To mitigate this, more advanced efficient compression technologies in terms of parameters or

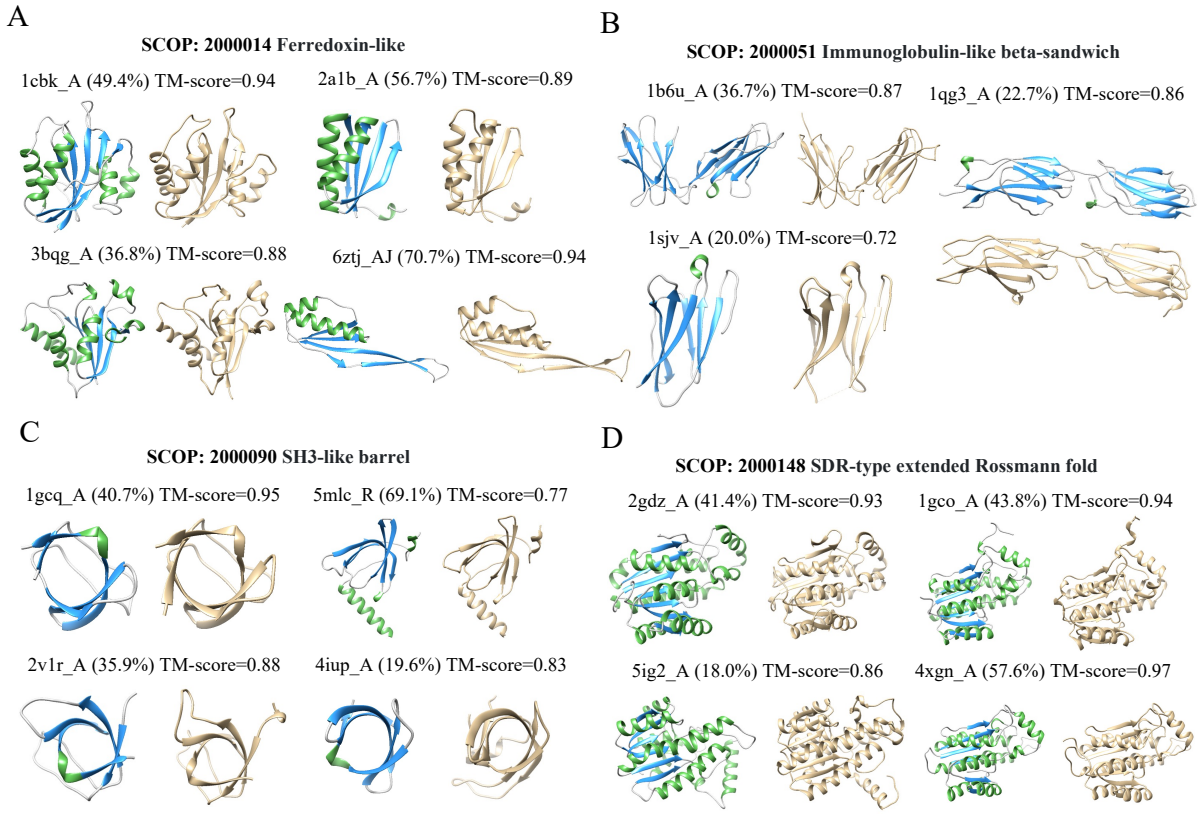


Figure 6: **Cases Study of xTrimoPGLM in Controllable Generation.** Generation of four SCOP fold types by xTrimoPGLM (SFT): Ferredoxin-like (**A**), Immunoglobulin-like beta-sandwich (**B**), SH3-like barrel (**C**) and SDR-type extended Rossmann fold (**D**). The structures with interleaved green, blue and gray color represents generated proteins. And the structures with gold color are the most structurally similar proteins from PDB database. Percentages in parentheses indicate sequence identity.

memory, such as quantization (*35*), kernel fusion (*30*) and other accelerate technologies (*36–39*) could be applied. These methods might enable the training and deployment of larger models with reduced computational resources. In this area, there is a wealth of research exploring various methods. We have only validated a few of these methods so far. However, more research is needed to assess their effectiveness in real-world scenarios.

OOD Structure Prediction

While xT-Fold has shown impressive both TM-score and inference speed in PLM-based models, a notable challenge persists in its OOD test performance (such as CASP proteins). This performance gap remains consistent, even as the pre-training model becomes more powerful. Though Figure1D illustrates that the model is gradually improving in compressing OOD sequences, it already diminishes the return. We speculate that protein sequences, unlike natural language, are less smooth in semantic space and more akin to a factual nature. Currently, it remains a formidable task for PLM-based methods to outperform those MSA-based (or retrieval-augmented), especially focusing on significant OOD test data. To bridge this gap, more abundant data resources, along with compute-optimal pre-training could be key. Another option is that integrating MSA modules with neural network retrievers (*40, 41*), might lead to better end-to-end optimization and faster inference speeds than traditional MSA methods. Such developments could enhance the model's ability to generalize on OOD data while accelerating its processing capabilities.

Generative Protein Hallucinations

In the preceding sections, we demonstrated that the xTrimoPGLM is capable of generating a diverse protein sequence. However, akin to most LLMs, our model also suffers from the issue of generating hallucinations. In the generation process, when the set sampling temperature is low, the model tends to generate fragments with a high repetition of amino acids. Although certain types of repetitive fragments (such as repeated Alanine) might be predicted with high-confidence structures, these fragments do not exist in nature. On the other hand, out of the 2,861 sequences we generated, approximately 21.7% of the sequences could not be confidently predicted by xT-Fold (pLDDT < 70), leaving us uncertain about whether these sequences can exist and fold stably. Even after SFT, only 17.8% to 88% of the generated sequences could find similar

structures in the Protein Data Bank (PDB). To avoid or reduce the hallucination problem of LLMs, augmenting constraints (*42, 43*) during the training and sampling process can enhance the efficiency of the model's generation.

Conclusion

We explored unified understanding and generation pre-training with an extremely large-scale protein language model. Our experiments suggest that such scaling can extend to downstream tasks, including the key 3D structure prediction. We have further unlocked new possibilities in protein sequence design through supervised fine-tuning, paving the way for groundbreaking advancements in this field. Our work serves as a stepping stone for future research in the protein foundation model, and we hope it can facilitate further progress in a broader spectrum of protein-related applications.

References

1. J. Jumper, *et al.*, Highly accurate protein structure prediction with alphafold, Nature (2021).

2. M. Baek, *et al.*, Accurate prediction of protein structures and interactions using a three-track neural network, Science (2021).

3. C. B. Anfinsen, *et al.*, The molecular basis of evolution., The molecular basis of evolution. (1959).

4. A. Rives, *et al.*, Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences, Proceedings of the National Academy of Sciences (2021).

5. Z. Lin, *et al.*, Evolutionary-scale prediction of atomic-level protein structure with a language model, Science (2023).

6. A. Elnaggar, *et al.*, & 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.

7. A. Madani, *et al.*, Large language models generate functional protein sequences across diverse families, Nature Biotechnology (2023).

8. R. Apweiler, *et al.*, Uniprot: the universal protein knowledgebase, Nucleic acids research (2004).

9. R. D. Finn, *et al.*, Pfam: the protein families database, Nucleic acids research (2014).

10. M. Steinegger, M. Mirdita, J. Söding, Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold, Nature methods (2019).

11. E. Nijkamp, J. Ruffolo, E. N. Weinstein, N. Naik, A. Madani, Progen2: exploring the boundaries of protein language models, arXiv preprint arXiv:2206.13517 (2022).

12. B. Hie, *et al.*, A high-level programming language for generative protein design, bioRxiv (2022).

13. R. Verkuil, *et al.*, Language models generalize beyond natural proteins, bioRxiv (2022).

14. L. Dong, *et al.*, Unified language model pre-training for natural language understanding and generation, Advances in neural information processing systems (2019).

15. H. Bao, *et al.*, Unilmv2: Pseudo-masked language models for unified language model pre-training, International conference on machine learning (2020).

16. Y. Tay, *et al.*, Ul2: Unifying language learning paradigms, The Eleventh International Conference on Learning Representations (2023).

17. T. Brown, *et al.*, Language models are few-shot learners, Advances in neural information processing systems (2020).

18. J. Wei, *et al.*, Finetuned language models are zero-shot learners, arXiv preprint arXiv:2109.01652 (2021).

19. H. W. Chung, *et al.*, Scaling instruction-finetuned language models, arXiv preprint arXiv:2210.11416 (2022).

20. R. M. Wu, *et al.*, High-resolution de novo structure prediction from primary sequence, bioRxiv (2022).

21. Z. Du, *et al.*, Glm: General language model pretraining with autoregressive blank infilling, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (2022).

22. J. D. M.-W. C. Kenton, L. K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, Proceedings of NAACL-HLT (2019).

23. J. Kaplan, *et al.*, Scaling laws for neural language models, arXiv preprint arXiv:2001.08361 (2020).

24. J. Hoffmann, *et al.*, Training compute-optimal large language models, arXiv preprint arXiv:2203.15556 (2022).

25. E. J. Hu, *et al.*, Lora: Low-rank adaptation of large language models, arXiv preprint arXiv:2106.09685 (2021).

26. J. Wei, *et al.*, Emergent abilities of large language models, Transactions on Machine Learning Research (2022).

27. A. Elnaggar, *et al.*, Ankh: Optimized protein language model unlocks general-purpose modelling, bioRxiv (2023).

28. M. Baek, *et al.*, Efficient and accurate prediction of protein structure using rosettafold2, bioRxiv (2023).

29. R. Wu, *et al.*, High-resolution de novo structure prediction from primary sequence, BioRxiv (2022).

30. T. Dao, D. Fu, S. Ermon, A. Rudra, C. Ré, Flashattention: Fast and memory-efficient exact attention with io-awareness, Advances in Neural Information Processing Systems (2022).

31. N. Ferruz, S. Schmidt, B. Höcker, Protgpt2 is a deep unsupervised language model for protein design, Nature communications (2022).

32. N. Ferruz, S. Schmidt, B. Höcker, A deep unsupervised language model for protein design, bioRxiv (2022).

33. L. Ouyang, *et al.*, Training language models to follow instructions with human feedback, Advances in Neural Information Processing Systems (2022).

34. C. Gulcehre, *et al.*, Reinforced self-training (rest) for language modeling, arXiv preprint arXiv:2308.08998 (2023).

35. T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, Qlora: Efficient finetuning of quantized llms, arXiv preprint arXiv:2305.14314 (2023).

36. W. Kwon, *et al.*, Efficient memory management for large language model serving with pagedattention, Proceedings of the 29th Symposium on Operating Systems Principles (2023).

37. J. Ainslie, *et al.*, Gqa: Training generalized multi-query transformer models from multi-head checkpoints, arXiv preprint arXiv:2305.13245 (2023).

38. C. Chen, *et al.*, Accelerating large language model decoding with speculative sampling, arXiv preprint arXiv:2302.01318 (2023).

39. Y. Leviathan, M. Kalman, Y. Matias, Fast inference from transformers via speculative decoding, International Conference on Machine Learning (2023).

40. K. Guu, K. Lee, Z. Tung, P. Pasupat, M. Chang, Retrieval augmented language model pre-training, International conference on machine learning (2020).

41. S. Borgeaud, *et al.*, Improving language models by retrieving from trillions of tokens, International conference on machine learning (2022).

42. P. Lewis, *et al.*, Retrieval-augmented generation for knowledge-intensive nlp tasks, Advances in Neural Information Processing Systems (2020).

43. M. Post, D. Vilar, *arXiv preprint arXiv:1804.06609* (2018).

44. Z. Lin, *et al.*, Language models of protein sequences at the scale of evolution enable accurate structure prediction, bioRxiv (2022).

45. C. Raffel, *et al.*, Exploring the limits of transfer learning with a unified text-to-text transformer, The Journal of Machine Learning Research (2020).

46. A. Zeng, *et al.*, Glm-130b: An open bilingual pre-trained model, arXiv preprint arXiv:2210.02414 (2022).

47. T. Wang, *et al.*, What language model architecture and pretraining objective works best for zero-shot generalization?, International Conference on Machine Learning (2022).

48. Y. Tay, *et al.*, Transcending scaling laws with 0.1% extra compute, arXiv preprint arXiv:2210.11399 (2022).

49. B. E. Suzek, *et al.*, Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches, Bioinformatics (2015).

50. A. Chowdhery, *et al.*, Palm: Scaling language modeling with pathways, arXiv preprint arXiv:2204.02311 (2022).

51. T. L. Scao, *et al.*, What language model to train if you have one million gpu hours?, arXiv preprint arXiv:2210.15424 (2022).

52. M. Shoeybi, *et al.*, Megatron-lm: Training multi-billion parameter language models using model parallelism, arXiv preprint arXiv:1909.08053 (2019).

53. D. Narayanan, A. Phanishayee, K. Shi, X. Chen, M. Zaharia, Memory-efficient pipeline-parallel dnn training, International Conference on Machine Learning (2021).

54. L. G. Valiant, A bridging model for parallel computation, Communications of the ACM (1990).

55. J. Rasley, S. Rajbhandari, O. Ruwase, Y. He, Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters, Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2020).

56. N. Shazeer, Glu variants improve transformer, arXiv preprint arXiv:2002.05202 (2020).

57. H. Wang, *et al.*, Deepnet: Scaling transformers to 1,000 layers, arXiv preprint arXiv:2203.00555 (2022).

58. P. Micikevicius, *et al.*, Mixed precision training, International Conference on Learning Representations (2018).

59. M. Ding, *et al.*, Cogview: Mastering text-to-image generation via transformers, Advances in Neural Information Processing Systems (2021).

60. I. Loshchilov, F. Hutter, Decoupled weight decay regularization, International Conference on Learning Representations (2018).

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

62. H. Alexander, *et al.*, Eukaryotic genomes from a global metagenomic dataset illuminate trophic modes and biogeography of ocean plankton, bioRxiv (2021).

63. M. Mirdita, *et al.*, Colabfold: making protein folding accessible to all, Nature methods (2022).

64. A. L. Mitchell, *et al.*, Mgnify: the microbiome analysis resource in 2020, Nucleic acids research (2020).

65. T. O. Delmont, *et al.*, Functional repertoire convergence of distantly related eukaryotic plankton lineages abundant in the sunlit ocean, Cell Genomics (2022).

66. E. Levy Karin, M. Mirdita, J. Söding, Metaeuk—sensitive, high-throughput gene discovery, and annotation for large-scale eukaryotic metagenomics, Microbiome (2020).

67. S. Nayfach, *et al.*, Metagenomic compendium of 189,680 dna viruses from the human gut microbiome, Nature microbiology (2021).

68. L. F. Camarillo-Guerrero, A. Almeida, G. Rangel-Pineros, R. D. Finn, T. D. Lawley, Massive expansion of human gut bacteriophage diversity, Cell (2021).

69. M. Steinegger, J. Söding, Clustering huge protein sequence sets in linear time, Nature communications (2018).

70. A. W. Senior, *et al.*, Improved protein structure prediction using potentials from deep learning, Nature (2020).

71. J. Yang, *et al.*, Improved protein structure prediction using predicted interresidue orientations, Proceedings of the National Academy of Sciences (2020).

72. L. Lo Conte, *et al.*, Scop: a structural classification of proteins database, Nucleic acids research (2000).

73. J. Hou, B. Adhikari, J. Cheng, Deepsf: deep convolutional neural network for mapping protein sequences to folds, Bioinformatics (2018).

74. D. Chen, X. Tian, B. Zhou, J. Gao, Profold: Protein fold classification with additional structural features and a novel ensemble classifier, BioMed research international (2016).

75. J. Chen, M. Guo, X. Wang, B. Liu, A comprehensive review and comparison of different computational methods for protein remote homology detection, Briefings in bioinformatics (2018).

76. M. S. Klausen, *et al.*, Netsurfp-2.0: Improved prediction of protein structural features by integrated deep learning, Proteins: Structure, Function, and Bioinformatics (2019).

77. S. Khurana, *et al.*, Deepsol: a deep learning framework for sequence-based protein solubility prediction, Bioinformatics (2018).

78. G. J. Rocklin, *et al.*, Global analysis of protein folding using massively parallel design, synthesis, and testing, Science (2017).

79. R. Rao, *et al.*, Evaluating protein transfer learning with tape, Advances in Neural Information Processing Systems (2019).

80. I. Pudžiuvelyte, *et al.*, Temstapro: protein thermostability prediction using sequence representations from protein language models, bioRxiv (2023).

81. G. Li, *et al.*, Learning deep representations of enzyme thermal adaptation, Protein Science (2022).

82. J. E. Gado, *et al.*, Deep learning prediction of enzyme optimum ph, bioRxiv (2023).

83. H. Wang, *et al.*, Predppcrys: Accurate prediction of sequence cloning, protein production, purification and crystallization propensity from protein sequences using multi-step heterogeneous feature fusion and selection, PLOS ONE (2014).

84. Y. Cheng, *et al.*, Co-evolution-based prediction of metal-binding sites in proteomes by machine learning, Nature Chemical Biology (2023).

85. F. Li, *et al.*, Deep learning-based kcat prediction enables improved enzyme-constrained model reconstruction, Nature Catalysis (2022).

86. Y. Wu, *et al.*, Ccbhla: pan-specific peptide–hla class i binding prediction via convolutional and bilstm features, bioRxiv (2023).

87. M.-D. N. Pham, *et al.*, epiTCR: a highly sensitive predictor for TCR–peptide binding, Bioinformatics (2023). Btad284.

88. P. Chhibbar, A. Joshi, Generating protein sequences from antibiotic resistance genes data using generative adversarial networks, arXiv preprint arXiv:1904.13240 (2019).

89. K. S. Sarkisyan, *et al.*, Local fitness landscape of the green fluorescent protein, Nature (2016).

90. Y. Luo, *et al.*, Ecnet is an evolutionary context-integrated deep learning framework for protein engineering, Nature communications (2021).

91. Y. Qiu, J. Hu, G.-W. Wei, Cluster learning-assisted directed evolution, Nature computational science (2021).

92. C. Dallago, *et al.*, Flip: Benchmark tasks in fitness landscape inference for proteins, bioRxiv (2021).

93. R. David, *et al.*, Identifying protein subcellular localisation in scientific literature using bidirectional deep recurrent neural network, Scientific Reports (2021).

94. J. J. Almagro Armenteros, C. K. Sønderby, S. K. Sønderby, H. Nielsen, O. Winther, Deeploc: prediction of protein subcellular localization using deep learning, Bioinformatics (2017).

95. Z. Dosztanyi, V. Csizmok, P. Tompa, I. Simon, The pairwise energy content estimated from amino acid composition discriminates between folded and intrinsically unstructured proteins, Journal of molecular biology (2005).

96. G. Erdős, M. Pajkos, Z. Dosztányi, Iupred3: prediction of protein disorder enhanced with unambiguous experimental annotation and visualization of evolutionary conservation, Nucleic acids research (2021).

97. Z. Dosztányi, Prediction of protein disorder based on iupred, Protein Science (2018).

98. F. Zhu, *et al.*, Who international standard for sars-cov-2 antibodies to determine markers of protection, The Lancet Microbe (2022).

99. Q. Li, *et al.*, Immune response in covid-19: what is next?, Cell Death & Differentiation (2022).

100. A. Kovaltsuk, *et al.*, Observed antibody space: a resource for data mining next-generation sequencing of antibody repertoires, The Journal of Immunology (2018).

101. M.-P. Lefranc, *et al.*, Imgt®, the international immunogenetics information system®, Nucleic acids research (2009).

102. S. Bachas, *et al.*, Antibody optimization enabled by artificial intelligence predictions of binding affinity and naturalness, bioRxiv (2022).

103. D. Hesslow, N. Zanichelli, P. Notin, I. Poli, D. Marks, Rita: a study on scaling up generative protein sequence models, arXiv preprint arXiv:2205.05789 (2022).

104. J. Salazar, D. Liang, T. Q. Nguyen, K. Kirchhoff, Masked language model scoring, arXiv preprint arXiv:1910.14659 (2019).

105. R. W. Shuai, J. A. Ruffolo, J. J. Gray, Generative language modeling for antibody design, bioRxiv (2021).

106. T. H. Olsen, I. H. Moal, C. M. Deane, Ablang: an antibody language model for completing antibody sequences, Bioinformatics Advances (2022).

107. J. A. Ruffolo, J. J. Gray, J. Sulam, Deciphering antibody affinity maturation with language models and weakly supervised learning, arXiv preprint arXiv:2112.07782 (2021).

108. H. M. Berman, The protein data bank: a historical perspective., Acta crystallographica. Section A, Foundations of crystallography (2008).

109. Y. Zhang, J. Skolnick, Scoring function for automated assessment of protein structure template quality, Proteins: Structure, Function, and Bioinformatics (2004).

110. S. Basu, B. Wallner, Dockq: a quality measure for protein-protein docking models, PloS one (2016).

111. K. Tunyasuvunakool, *et al.*, Highly accurate protein structure prediction for the human proteome, Nature (2021).

112. R. Evans, *et al.*, Protein complex prediction with alphafold-multimer, bioRxiv (2021).

113. Z. Lin, *et al.*, Evolutionary-scale prediction of atomic level protein structure with a language model, bioRxiv (2022).

114. J. A. Ruffolo, L.-S. Chu, S. P. Mahajan, J. J. Gray, Fast, accurate antibody structure prediction from deep learning on massive set of natural antibodies, Nature Communications (2022).

115. Y. Wang, *et al.*, xtrimoabfold: De novo antibody structure prediction without msa, ArXiv (2022).

116. R. Chen, L. Li, Z. Weng, Zdock: An initial-stage protein-docking algorithm, Proteins: Structure (2003).

117. D. Kozakov, *et al.*, The cluspro web server for protein–protein docking, Nature Protocols (2017).

118. O.-E. Ganea, *et al.*, Independent se(3)-equivariant models for end-to-end rigid protein docking, ArXiv (2021).

119. Y. Yan, H. Tao, J. He, S. Huang, The hdock server for integrated protein–protein docking, Nature Protocols (2020).

120. Y. Luo, *et al.*, xtrimodock: Rigid protein docking via cross-modal representation learning and spectral algorithm, bioRxiv (2023).

Supplementary materials

Contents

1 Methods and Material **39**

1.1 Backbone Framework: General Language Model (GLM) 39

1.2 Pre-Training Objectives 40

1.3 Pre-Training Strategy for xTrimoPGLM-100B 41

1.4 Empirical Analysis of Unified Training 42

1.5 The Training Stability of Unified Training 44

1.6 xTrimoPGLM-100B Pre-Training Configurations 45

1.7 xTrimoPGLM-100B Pre-Training Datasets 46

1.8 Downstream Tasks Datasets & Evaluation Metrics 49

1.9 xT-Fold Acceleration Methods 57

1.10 xT-Fold Training Settings 57

1.11 The SFT & ReST Pipelines 58

1.11.1 Dataset 58

1.11.2 Supervised Fine-tuning. 59

1.11.3 Reinforcement Self-Training 59

1.12 Properties Analysis of Generated Sequences using xTrimoPGLM-100B 60

1.13 Model FLOPs Comparisons 62

2 xTrimoPGLM-Ab **64**

2.1 xTrimoPGLM-Ab: OAS Fine-tuning for Antibody 64

2.1.1 Zero-shot Naturalness 65

2.1.2 xTrimoPGLM-AbFold: Antibody structure prediction 67

2.1.3 Antibody-specific Generation using xTrimoPGLM-Ab. 71

Additional Figures

Figure S1 Comprehensive Insights into xTrimoPGLM 7
Figure S2 The Performance of Protein Understanding Benchmark 11
Figure S3 Structure Prediction with xT-Fold 14
Figure S4 Diversification of Protein Structures by xTrimoPGLM 18
Figure S5 Alignment Capabilities of xTrimoPGLM in Protein Sequence Generation . . 20
Figure S6 Cases Study of xTrimoPGLM in Controllable Generation 22
Figure S7 Human Protein Sequence Mapping. 73
Figure S8 The empirical analysis of unified training. 74
Figure S9 Trials on different strategies for transition from Stage-1 to Stage-2. 75
Figure S10 Generated Protein Sequences Cases 76
Figure S11 The Pre-training Dataset of xTrimoPGLM-100B 77
Figure S12 Training Data Distribution 77
Figure S13 Perplexity Delta Comparison on CAMEO and CASP15 78
Figure S14 xTrimoPGLM-Abfold Architecture 78
Figure S15 Generated Antibodies Sequences 79
Figure S16 Conformation of Antibody Generation 79
Figure S17 Structure Examples of Generated Protein Sequences 80
Figure S18 Distributions of generated sequences 80

Additional Tables

Table S1 Comparisons between different architectures of PLMs. 74

Table S2 Overview of 18 Downstream Benchmark Tasks 81
Table S3 Parameter Efficient Fine-tuning on Downstream Tasks 82
Table S4 Compute Comparisons of Pre-training Language Models 83
Table S5 Zero-shot performance of xTrimoPGLM-AB 83
Table S6 Structure Prediction of VH and VL Individuals 84
Table S7 Structure Prediction of VH-VL Complexes 84
Table S8 Full Configurations for xTrimoPGLM-100B Training. 85
Table S9 Edit Distance of Antibody Generation 86
Table S10Distorder Proteins and Domains 86

1 Methods and Material

1.1 Backbone Framework: General Language Model (GLM)

Current PLM frameworks are commonly categorized as either encoder-only, like ESM (*4, 44*), or decoder-only, such as ProGen (*7, 11*), exhibit limitations in addressing both task categories effectively due to their inherent inductive biases. Conceptually, the two types of tasks mirror the broader spectrum of protein sequence distributions (*12, 13*), suggesting the need for a unified model capable of encapsulating this diversity. To achieve this, encoder-decoder architectures like T5 (*45*) and non-causal decoder-only models like the General Language Model (GLM) (*21, 46*) are optimized through an auto-regressive generating and bidirectional input-processing objective, which emerge as promising candidates for this dual capability. However, the GLM, with its parameter efficiency, stands out as a more viable option compared to the T5, which requires significantly 2x larger parameters for similar efficacy. GLM is a transformer-based language model characterized by its unique training methodology. It employs autoregressive blank infilling while processing input text bi-directionally. This approach involves randomly blanking out continuous spans of tokens from the input and training the model to sequentially reconstruct

these spans. This dual focus on autoencoding and autoregressive pre-training differentiates GLM from causal decoder-only language models, which rely solely on unidirectional attention.

1.2 Pre-Training Objectives

GLM incorporates two distinct pre-training objectives to ensure its generative capabilities: 1) *Span prediction*. This objective focuses on recovering short blanks within sentences, with the blank lengths cumulatively forming a significant portion of the input. and 2) *Long-text generation*. Aimed at generating extended blanks at sentence ends, this objective works with variable-length blanks, utilizing prefix contexts for guidance. Additionally, to enhance xTrimoPGLM's comprehension abilities, we have integrated the Masked Language Model (MLM) strategy (*22*). This inclusion ensures that xTrimoPGLM not only excels in accurate residue-level representation but also effectively captures sequence-level representations, providing a comprehensive understanding of protein sequences.

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 $\mathcal{L}_{\text{GLM}}$.

1.3 Pre-Training Strategy for xTrimoPGLM-100B

Motivated by the philosophy of curriculum learning, we begin the pre-training of xTrimoPGLM -100B with a rather simpler MLM objective, then followed by the GLM objective, which unfolds in two methodically structured stages:

Masked Language Model Stage (400 Billion Pre-Trained Tokens). In this initial stage, the `[MASK]` token is employed for masking random tokens in the sequence, with these masked tokens comprising 15% of the total input. This stage, consuming about 400 billion tokens, is dedicated to advancing the model's representation abilities.

Unified Training Stage (600 Billion Pre-Trained Tokens). Subsequently, the model undergoes training with a combined approach of MLM and GLM objectives, in a ratio of 20% MLM to

80% GLM. In this stage, the model processes an additional 600 billion tokens, aiming to further refine both its representational and generative capabilities.

- **Masked Language Model Component.** Leveraging the `[MASK]` token, this component focuses on enhancing the model's understanding of protein sequences.
- **General Language Model Component.** This component employs two types of masking: `[sMASK]` for consecutive span masking, following a Poisson distribution ($\lambda = 6$), and `[gMASK]` for masking larger sequence segments based on a uniform distribution (minimally 40% of the tokens masked). The `[sMASK]` token aids in blank infilling tasks, while `[gMASK]` facilitates the model in generating extended masked segments using the unmasked prefix.

This dual-stage training strategy meticulously integrates the MLM and GLM objectives, thereby optimizing xTrimoPGLM-100B for a comprehensive understanding and generation of protein sequences.

1.4 Empirical Analysis of Unified Training

This section presents an in-depth analysis of the feasibility of concurrently optimizing two distinct pre-training objectives in xTrimoPGLM. Unlike prior unified pre-training frameworks (*14–16, 47, 48*), which typically adopt similar formulations for diverse objectives, our approach extends these methodologies to a broader context. We critically examine whether models benefit from joint optimization of in-place token predictions (Masked Language Model) and next-token predictions (General Language Model). Central to our investigation are two pivotal questions: *Objective Compatibility.* Does the in-place token prediction objective be optimized with the next-token prediction approach simultaneously? This inquiry is essential to understand whether these objectives can be effectively integrated within a single training framework; *Mutual Contribution.*

Can the in-place token prediction strategy enhance the performance of next-token prediction tasks, and does the reverse also hold true? This question addresses the potential synergistic effects of combining these two objectives in the training regime of xTrimoPGLM. Our exploration into these questions aims to shed light on the intricate dynamics of unified training models, particularly in the context of large-scale language models specialized for protein sequence analysis.

Pre-training Settings. Our experiments utilize xTrimoPGLM-150m, featuring 30 layers, 20 attention heads, 640 embedding dimensions, and FP16 precision. This configuration aligns with xTrimoPGLM-100B's hyperparameters. Pre-training is conducted on the Uniref50 database (*49*). We employ batches of 2,048 sequences, each 1,024 tokens in length. 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.

- **MLM.** Roughly 15% of input tokens were masked, resulting in approximately 1,024 input and 154 target tokens. Loss calculations are confined to target tokens.
- **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.

We evaluate the compatibility of MLM (in-place token prediction) and GLM ([gMASK], next-token prediction) objectives. Each occupies 50% of the training batch time, alternating between them. Shifts in objectives occur at 100B and 200B token consumption milestones, facilitated by constant model parameters and architecture, requiring only adjustments in the

attention mask. Validation losses (refer to Figure S8(a)(b)) indicate that despite their differing natures, both MLM and GLM objectives optimize simultaneously.

Furthermore, we investigate the impact of pre-training objectives on convergence speed. Models pre-trained on one objective adapt to another, training over an additional 50B tokens. Comparisons include: *MLM-adapted GLM versus GLM trained from scratch* and *GLM-adapted MLM versus MLM trained from scratch*. Our results (see Figure S8(c)(d)) show significantly faster convergence in adapted models compared to those trained from scratch. The MLM-adapted GLM matches the loss of the GLM from-scratch model with a 2.2× speedup (110B tokens). Similarly, the GLM-adapted MLM achieves a 2× speedup (100B tokens).

These findings suggest that modeling protein data distribution is not limited to specific training patterns. This bridges the gap between autoencoding PLMs (e.g., ESM (*5*)) and autoregressive PLMs (e.g., ProGen2 (*11*)), underscoring the effectiveness of the xTrimoPGLM training pipeline.

1.5 The Training Stability of Unified Training

Training stability is a critical factor for the successful training of large language models (LLMs) at the 100B-scale (*17, 46, 50, 51*). Given a fixed computing budget, it is essential to balance efficiency and stability, particularly in relation to floating-point (FP) formats. Lower-precision FP formats, such as 16-bit precision (FP16), enhance computational efficiency but are vulnerable to overflow and underflow errors. These vulnerabilities can potentially lead to catastrophic collapses during training. xTrimoPGLM, drawing on the implementation strategies of GLM-130B (*46*), addresses many unstable training issues. Nonetheless, xTrimoPGLM-100B still experiences catastrophic training collapses during the transition from the first to the second stage, a challenge not present in smaller-scale models (10B-scale). Incorporating a fixed ratio of GLM loss into pre-training can trigger these collapses, even with a minimal 1% GLM loss ratio, as illustrated in

Figure S9. To mitigate this issue, we propose the implementation of a smooth transition strategy.

Smooth Transition Strategy. Our empirical investigations suggest a two-phase smooth transition strategy to integrate GLM loss into training:

Gradual Increase in GLM Loss Ratio. We start by incrementally increasing the GLM loss ratio from 0, aiming to reach the target value R in K steps through linear growth. The GLM loss ratio R_k at each step k is calculated as $R_k = \frac{k \times R}{K}$. Notably, the learning rate remains exceptionally low during this phase. In practice, we set K = 1000 and the learning rate to 1e-7.

Normalization of the Learning Rate. After completing the transition, the learning rate gradually returns to its standard pre-training level as defined in the pre-training script. The final xTrimoPGLM-100B training run demonstrates that loss divergence occurs only at the transition stage, though it initially faces numerous failures due to hardware issues.

1.6 xTrimoPGLM-100B Pre-Training 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 (*46*) framework, we only emphasize the specific hyper-parameter of xTrimoPGLM-100B. For more discussion and design choices please refer to GLM-130B (*46*).

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. We adopt 3D parallel strategy with the 4-way tensor parallelism (*52*), 8-way pipeline parallelism (*53*), and 24-way data parallelism (*54*) based on DeepSpeed (*55*). The model owns 72 transformer layers, 80 attention heads, and 10,240 embedding dims with 31,744 feed-forward embedding dims using GeGLU (*56*). We adopt the Post-LN initialized with the DeepNorm (*57*). We follow the mixed-precision (*58*) strategy (Apex O2), i.e., FP16 for forwards and backward 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 (*46, 59*) 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 (*60*) 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 https://kexue.fm/archives/8397 as our position embedding. For the tokenization of the protein data, we use the residue-level tokenizer which is adopted in several PLMs (*5, 11, 61, 62*). Except for the basic amino acid types, we add special tokens `[MASK]`, `[sMASK]`, and `[gMASK]` for model prediction. We also add special tokens `<sop>`, `<eop>`, `<eos>` for sequence separation (Cf. Table S8 for the full configurations).

1.7 xTrimoPGLM-100B Pre-Training Datasets

The pre-training dataset of xTrimoPGLM-100B is curated from two extensive data repositories: Uniref90 (`https://www.uniprot.org/help/downloads`), the Uniref90 version preceding December 2022 is downloaded and ColabFoldDB (*63*)(`https://colabfold.mmseqs.com`). The initial contributions from Uniref90 and ColabFoldDB encompasses ap-

proximately 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 (*5*) and ProtTrans (*6*). UniRef90 clusters are generated from the UniRef100 seed sequences with a 90% sequence identity threshold using the MMseqs2 `https://github.com/soedinglab/MMseqs2` algorithm, as depicted in Figure S11. 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. 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 `https://bfd.mmseqs.com`, MGnify (*64*), SMAG (eukaryotes) (*65*), MetaEuk (eukaryotes) (*66*), TOPAZ (eukaryotes) (*62*), MGV (DNA viruses) (*67*), GPD (bacteriophages) (*68*), and an updated version of the MetaClust (*69*) 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 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 complete dataset in ColabFoldDB initially contained approximately 950M sequences. After initial deduplication and short-length filtering, which removed about 150M duplicate sequences, and further refinement by cross-referencing and deduplicating with Uniref90, we narrowed down the dataset to 780M unique sequences, ensuring diversity and representativeness for effective training. 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 940M unique sequences, equivalent to roughly 200 billion tokens, as depicted in Figure S11 (left). During training, to capitalize on the high-quality data, we assign a greater weight to the Uniref90 data, resulting in a ColabFoldDB sampling ratio of approximately 60%. This approach triples or quadruples the contribution of Uniref90 data, boosting our model's fine-tuning capability with high-quality data.

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% or 0.5% 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`). The remaining after filtering is used as the validation set.

Pre-training Data Distribution. As shown in Figure S12, 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.

1.8 Downstream Tasks Datasets & Evaluation Metrics

To systematically evaluate xTrimoPGLM-100B, we have benchmarked 18 downstream protein-related tasks across multiple domains. Table S2 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. 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. 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. 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 (*70*) for structural prediction. The trRosetta dataset (*71*) 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 (*72*), a release from 2009, and has been widely adopted by DeepSF (*73*) 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 (*74*). 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 (*75*).

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 (*76*) 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 (*77*). 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 (*78*) and subsequently collected within the Task Assessing Protein Embeddings (TAPE) (*79*). In this regression-based task, we employ the SpeaRman Correlation Coefficient (SRCC) as the evaluation metric to measure the prediction consistency.

Temperature 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 (*80*). 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 (*81*), 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.

Optimal PH. Enzyme functions normally under a specific range of pH of the surrounding

environment. However, an optimal pH for the reaction will largely boost the catalytic ability. We collected a dataset from EpHod (*82*), which established a deep learning method to predict the optimal enzyme catalytic pH based on protein sequence only. In the regression-based task, we use the SRCC as the evaluation metric.

Clone CLF. Protein structure determination includes a series of experimental stages to yield stable proteins for X-ray crystallography. Specifically, the proteins are first selected and expressed, then purified for crystal structure determination. Each step corresponds to a "stage tag" to denote whether the protein is stable under a certain stage. We collected a dataset from PredPPCrys (*83*), which manually annotated thousands of proteins with different experimental procedures. The binary classification task is to predict whether a protein sequence tends to be a cloning failure (Clone CLF.). We use the AUC as the evaluation metric.

Material Production. The task is to predict whether a protein sequence fails at the protein material stage (MF). The dataset is also collected from PredPPCrys (*83*) and the AUC metric is employed as the measurement.

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 (*84*) 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 (*85*).

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 (*86*).

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 (*87*).

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 (*88*).

Fluorescence. The Fluorescence Prediction (Fluor. P.) task (*89*) 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 (*79*). 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 (*90, 91*). 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 (*92*). 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 (*93*). 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 (*94*).

We evaluated all benchmarked downstream tasks with xTrimoPGLM-100B and ESM2 models. The performance results are followed as Table S3. 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.

Protein Structure. 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 (*5*), 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 the 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 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.

1.9 xT-Fold Acceleration Methods

We developed and trained an xT-Fold model based on xTrimoPGLM-100B. Initially, the model undergoes a 4-bit quantization process, effectively implementing the W8A16 (weights for 8 bits, Activation with 16 bits) strategy before further compressing into a 4-bit model. Due to reduced model communication time among the model weights and memory consumption, this approach resulted in at least a 4x increase in training speed with the same devices. However, during the inference phase, it encountered memory overflow issues with longer sequences (exceeding 700 in length). To address this, we incorporated the FlashAttention (*30*) technique into xT-Fold. This technique enhances self-attention through operator fusion and softmax reduction, allowing our model to perform inference on sequences up to 2000 in length in a ~200 seconds, at least ~5x GPU inference time reduction compared with a standalone AlphaFold2 that not include the time of MSAs feature process.

1.10 xT-Fold Training Settings

The dataset preprocessing follows similar settings with ESMFold and AlphaFold2. We train xT-Fold on ~380K experimentally determined structures of ~25k clusters from the PDB (release date is less than May 2020), further augmented with a distilled dataset of ~4M greater than 90% pLDDT structures predicted from AlphaFoldDB. Predicted structures are sampled 75% of the time, and real structures 25% of the time during training. The model is trained with the same loss as AlphaFold2 except for omitting masked MSA loss. We trained on 10M samples with crop size 256 on the first stage, and fine-tuned cropsize 384 on data parallel 64 cards A100 80G with effective batch size 128. It was trained for the first stage 78K steps on protein crops of size 256 and then fine-tuned at the second stage with the structural violation loss with 0.01 coefficient for 70K steps, on crop sizes of 384. In the second stage, we use a cosine decay to make the learning rate from 5e-5 to 1e-5 after a warm-up of 3000 steps.

1.11 The SFT & ReST Pipelines

1.11.1 Dataset

For the comprehensive assessment of the alignment ability of xTrimoPGLM-100B, we curate 5 datasets from the 18 benchmarks described in SM Section 1.8 for supervised fine-tuning and reinforcement self-training.

Strong Fluorescence Intensity Proteins. Targeting generating protein sequences characterized by elevated fluorescence intensity, we select sequences exhibiting fluorescence intensity scores surpassing 0.8 in the Fluorescence task dataset, yielding a final dataset comprising 21,446 sequences for training and 5,362 sequences for validation.

High Fitness Mutations. Targeting generating protein sequences with mutations of high fitness, we selected sequences with fitness scores exceeding 0.5 in the Fitness task dataset, yielding a final dataset comprising 4,163 sequences for training and 469 sequences for validation.

Low Thermal Stability Proteins. Targeting generating protein sequences with reduced stability at 65 degrees Celsius, we selected sequences labeled as '0' in the Temperature Stability task dataset, yielding a final dataset comprising 141,598 sequences for training and 32,790 sequences for validation.

Immunoglobulins. Targeting generating protein sequences that correspond to the specific fold class of immunoglobulins, we selected sequences labeled as '36' in the Fold Classification task dataset, yielding a final dataset comprising 981 sequences for training and 221 sequences for validation.

Nucleus Localization Proteins. Targeting generating protein sequences that localize in the nucleus, we select sequences with labeled as '7' in the Localization task dataset, yielding a final dataset comprising 2,312 sequences for training and 610 sequences for validation.

1.11.2 Supervised Fine-tuning.

Supervised Fine-tuning. We trained the xTrimoPGLM-100B model with the GPT objective using LoRA fine-tuning with $\text{LoRA}_r = 16$ and $\text{LoRA}_\alpha = 32$. In addition, We use Adam as our optimizer with β_1 and β_2 set to 0.9 and 0.95, and a weight decay value of 0.01. We warm up the learning rate from 5×10^{-7} to 10^{-5} over the first 10.0% samples, then decay it linearly to the minimum learning rate 10^{-6}.

We trained the Progen2-xlarge model with the GPT objective in full scale. We use Adam as our optimizer with β_1 and β_2 both set to 0.9, ϵ set to 10^{-7}, and a weight decay value of 0.01 (except for the Fold Classification task, which corresponds to a weight decay value of 10^{-4}). We warm up the learning rate from 10^{-7} to 10^{-5} over the first 500 samples, then decay it by a cosine schedule to the minimum learning rate 5×10^{-6}. The gradient accumulation step is 10 for the Fold task and 4 for others.

We trained the ProtGPT2 model with the official fine-tuning script on Huggingface. We use Adam as our optimizer with β_1 and β_2 set to 0.9 and 0.999, and a constant learning rate 10^{-5}.

1.11.3 Reinforcement Self-Training

To explore the capacity for iterative improvement of xTrimoPGLM-100B, we implement a 1-step reinforcement self-training regime based on the initial supervised fine-tuning, leveraging feedback from task predictors as the reward model to further enhance the model's performance. The task predictor on each task is fine-tuned by qLoRA (*35*). More specifically, the reward models assess the generated sequences by predicting the numeric features for regression tasks (Fluorescence and Fitness), or the likelihood of the desired label for classification tasks (Temperature Stability, Fold and Localization). Then we apply a filtering process with strategies detailed below, and using the selected high-quality sequences as training data, fine-tune the checkpoint saved after supervised fine-tuning until the model converges.

Filtering Generated Sequences. For the Fluorescence task, we select sequences with predicted fluorescence intensity scores greater than 2.5 (1,386 sequences in total). For the Fitness task, we select the top 300 sequences in descending order of their predicted fitness scores. For the Temperature Stability task, we select the top 200 sequences in descending order of their predicted likelihood of being classified as unstable at 65 degrees Celsius. For the Fold task, we select the top 500 sequences in descending order of their predicted likelihood of being classified as corresponding to the fold class of DNA-binding domains. For the Localization Task, we select sequences with a likelihood of being classified as localizing in the nucleus surpassing 0.5 (1,084 sequences in total).

1.12 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. Mor 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.

We present the pairwise sequence identity analysis of generated sequences obtained through various combinations of temperature and nucleus sampling factors, as illustrated in Figure S18(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 S18(c)(d). Furthermore, our empirical study suggests that the pre-trained model tends to

generate repetitive tokens when the temperature drops below 0.8 and the nucleus sampling factor falls under 0.7, which results in abnormal long sequences. 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.

Intrinsically unstructured/disordered proteins/domains. Intrinsically unstructured or disordered proteins/domains (IUPs) (*95*) 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 (*96*), 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 S10, i.e., short disorder, long disorder, and globular structured domains (*97*). 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 (*31*).

Consequently, the two datasets show similar disorder prediction results as reported in Table S10. 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.

Trial & Error of Generated Structures with N-gram penalty. We first produced batches of samples with an n-gram penalty (N-gram=3) to reduce the probability of generating repetitive sequences, as shown in Figure S17. 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. 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.

Network Construction. We download SCOPe70_2.08 HH-suite database from HH-suite database website. Then we extracted all sequence clusters in scope70_a3m.ffdata file. Each cluster is used as query to SCOPe70 with hhsearch tool. Each sequence in SCOPe70 is represented as a node. Two nodes are linked if one of them can be searched by the other with at least 20 amino acids and 70% hhsearch probability. For generated sequences, we first search UniClust30 v2021_03 database with hhblits. The output profiles are used as input to search SCOPe70 with hhsearch. Edges are also constructed in a similar way.

1.13 Model FLOPs Comparisons

We conduct a comparative analysis of computational resources utilized by different pre-trained protein language models (Table S4). 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 computed 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.

2 xTrimoPGLM-Ab

2.1 xTrimoPGLM-Ab: OAS Fine-tuning for Antibody

We further 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 (*98, 99*). 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. By utilizing the xTrimoPGLM framework, we've made advancements in predicting antibody naturalness and structure prediction with our antibody-specific model, xTrimoPGLM-Ab.

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 SM Section 1.7 This process undertakes 500B tokens. Since antibodies represent a distinct subset of general proteins, then we finetuned the model using the OAS dataset[1], 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

[1]Observed Antibody Space (OAS) (*100*) data. Following the paper, we filter OAS data with IMGT schema (*101*) and therefore get 678m sequences without disorder and incompletion.

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.

2.1.1 Zero-shot Naturalness

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 (*11, 102, 103*). For the GLM objective, PPL is calculated by:

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

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 (*104*) 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]`sequence1`<eos>``[human][light]`sequence2`<eos>`, where `[human]` is a special token to indicate the species of sequences, `[heavy]` and `[light]` 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 (*105*) and ProGen2 (*11*) are auto-regressive models, while AbLang (*106*), ESM2 (*44*), and AntiBERTy (*107*) 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 S5. 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.

2.1.2 xTrimoPGLM-AbFold: 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 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) (*108*) 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 selected 68 samples as the test set, which are released between January 2022 and November 2022.

Root mean square deviation (RMSD) and TM-score (*109*) are used as evaluation metrics for both tasks. Another important metric DockQ (*110*) 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 S14.

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 S14). 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.

For structure prediction of single chains, the loss function of structure prediction mainly follows the work of AlphaFold2 (*111*) , which contains Frame Aligned Point Error (FAPE) and a number of auxiliary losses but excludes MSA loss. The loss can be formalized as follows:

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

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 (*111*) 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 (*112*) and a structural violation loss (*111*), with weights of 1.0 and 0.03, respectively. The concrete loss is shown as follows:

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

Baselines. For single-chain structure prediction tasks, we conduct a comparison of existing influential folding models, including Alphafold2 and four PLM-based models: OmegaFold (*20*), ESM-

Fold (*113*), IgFold (*114*), and xTrimoAbFold (*115*). We use public checkpoints AlphaFold2 [2], ESMFold [3], IgFold [4], OmegaFold [5] to infer the test set.

For the prediction of VH-VL complex structures, we compared ZDock (*116*) , a rigid protein docking algorithm based on fast Fourier transform correlation techniques, ClusPro (*117*), a docking method using bioinformatics and computational chemistry techniques, EquiDock (*118*), a genetic evolution-based algorithm, HDOCK (*119*), an algorithm that combines global and local search, and AlphaFold-Multimer (*112*), 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 S6, 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 (*111*) block, can emerge as a leading model for antibody structure prediction, even without the presence of MSA and templates.

Table S7 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 **6,300**$\times$ over the original AlphaFold-Multimer and **103**$\times$ over the faster MSA-searching AlphaFold-Multimer (*120*), owing to the original AlphaFold-Multimer consumes

[2]https://github.com/deepmind/alphafold
[3]https://github.com/facebookresearch/esm
[4]https://github.com/Graylab/IgFold
[5]https://github.com/HeliXonProtein/OmegaFold

a long time to search MSA (0.8 hours 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**$\times$ speedup than the original AlphaFold-Multimer and **40**$\times$ 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.

2.1.3 Antibody-specific Generation using xTrimoPGLM-Ab.

To demonstrate the generation capacity of xTrimoPGLM-Ab, 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,

- **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 S16(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 S16(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 S16(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 S16(d).

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 S16. 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 S9. 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

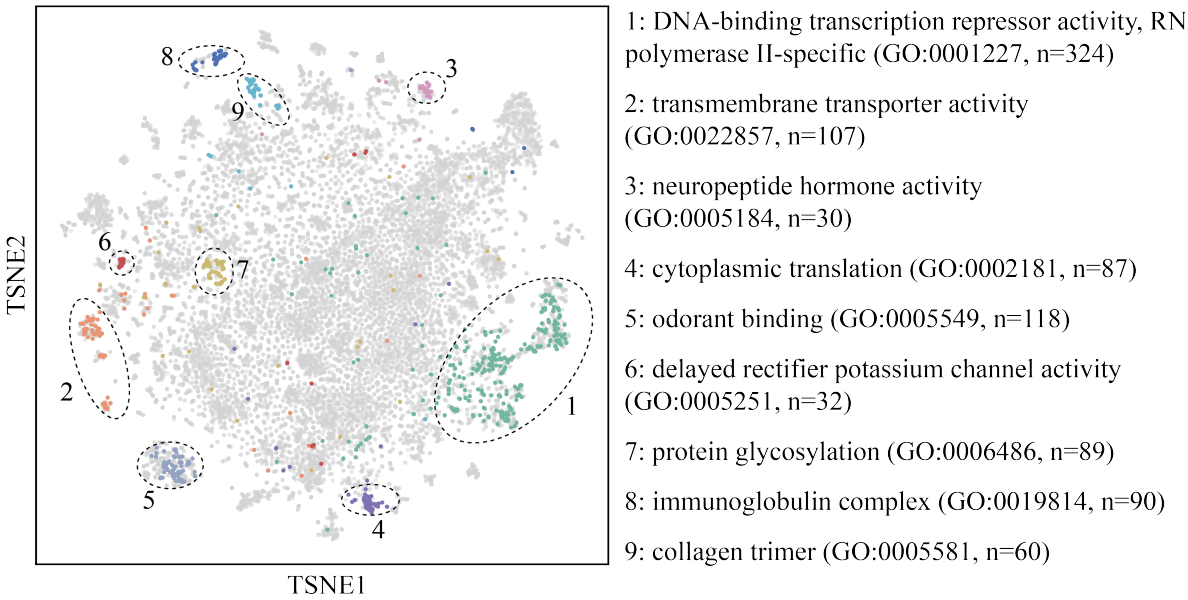
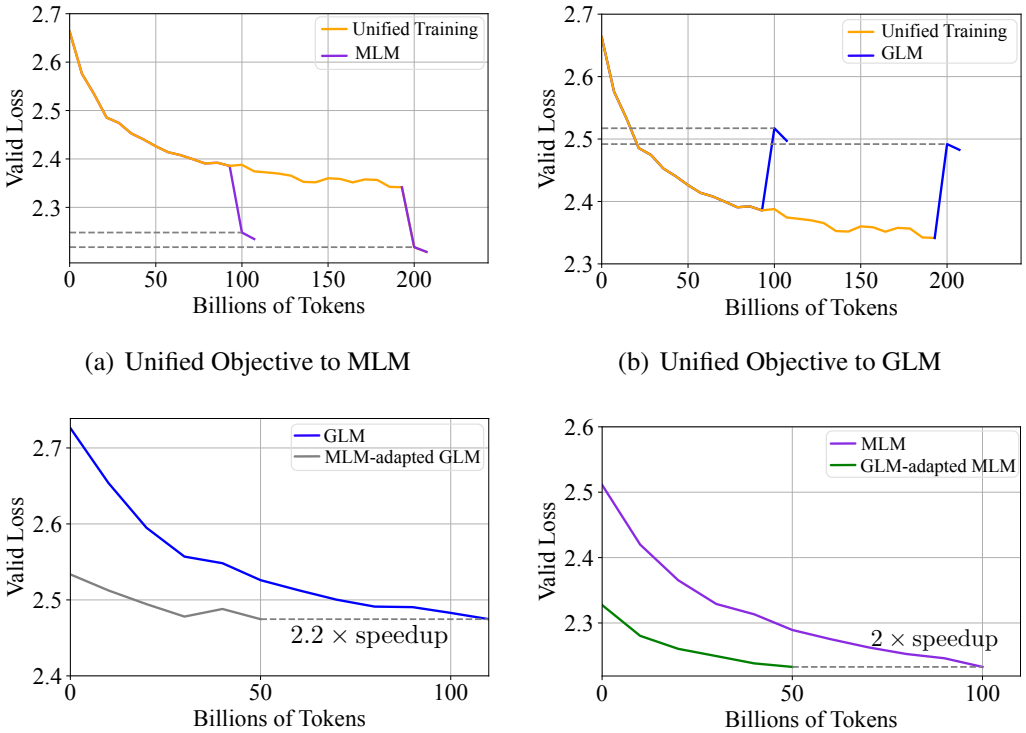


Figure 7: Human Protein Sequence Mapping. This panel illustrates the t-SNE visualization of xTrimoPGLM-100B embeddings for human protein sequences. The visualization demonstrates xTrimoPGLM's ability to capture biologically significant latent embeddings across a variety of functional protein sequences. Using a dataset of 20,255 human protein sequences from UniProt, each protein is represented as a distinct dot. The figure highlights nine clusters, each correlating with specific Gene Ontology annotation terms, differentiated by unique colors.

generative models currently face limitations.



(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 8: The empirical analysis of unified training.

Table 1: Comparisons between different architectures of PLMs.

Downstream Task	**Autoenc.**	**Autoreg.**	**Enc.-Dec.**	**GLM**	Example
Protein Understanding	✓	×	✓	✓	Contact Prediction
Protein Generation	×	✓	—	✓	Antibody Re-design
Representatives	ESM-1b (*4*), ESM2 (*5*), Pro.BERT (*61*)	ProGen (*7*), ProGen2 (*11*), ProtGPT2 (*31*)	ProtTrans (*6*), Ankh (*27*)	xTrimoPGLM	/

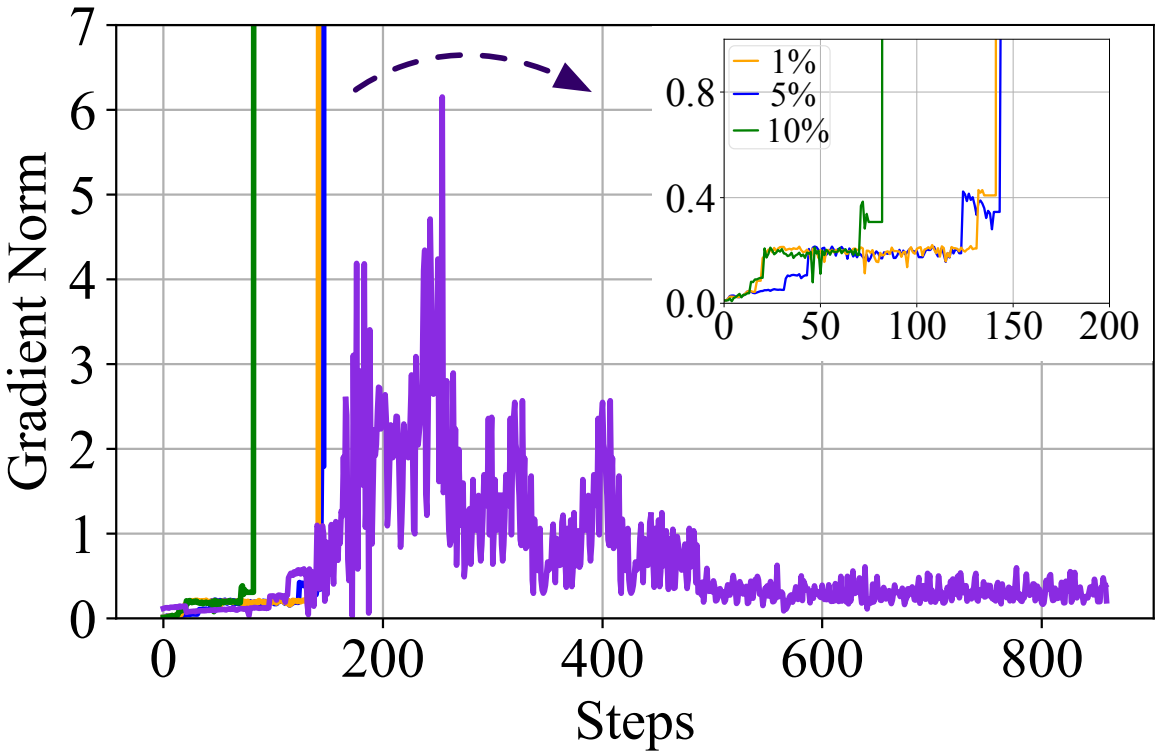


Figure 9: Trials on different strategies for transition from Stage-1 to Stage-2.

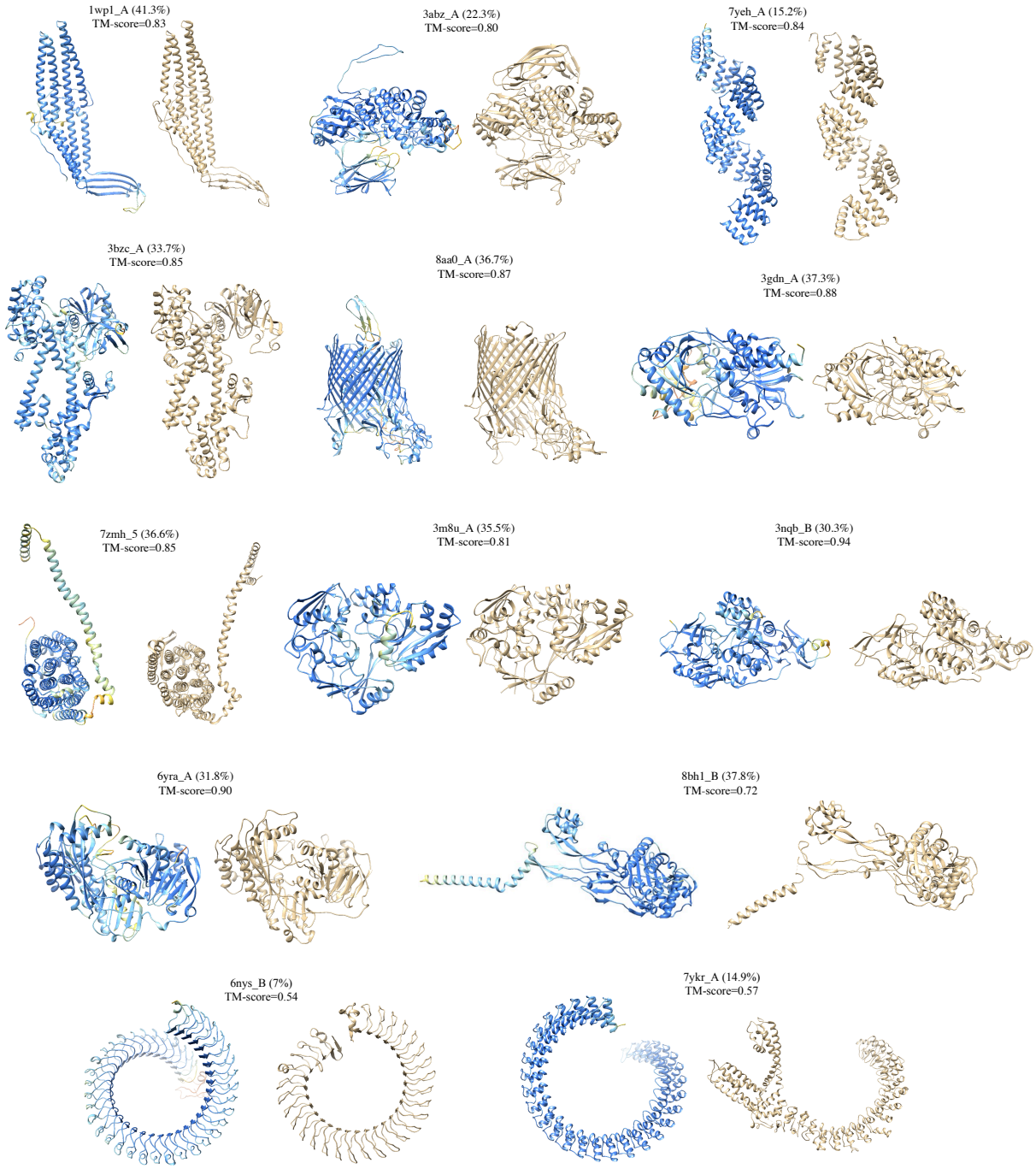


Figure 10: More protein sequences visualizations generated by xTrimoPGLM-100B. (Left: Generated by xTrimoPGLM-100B. Right: Natural Proteins.)

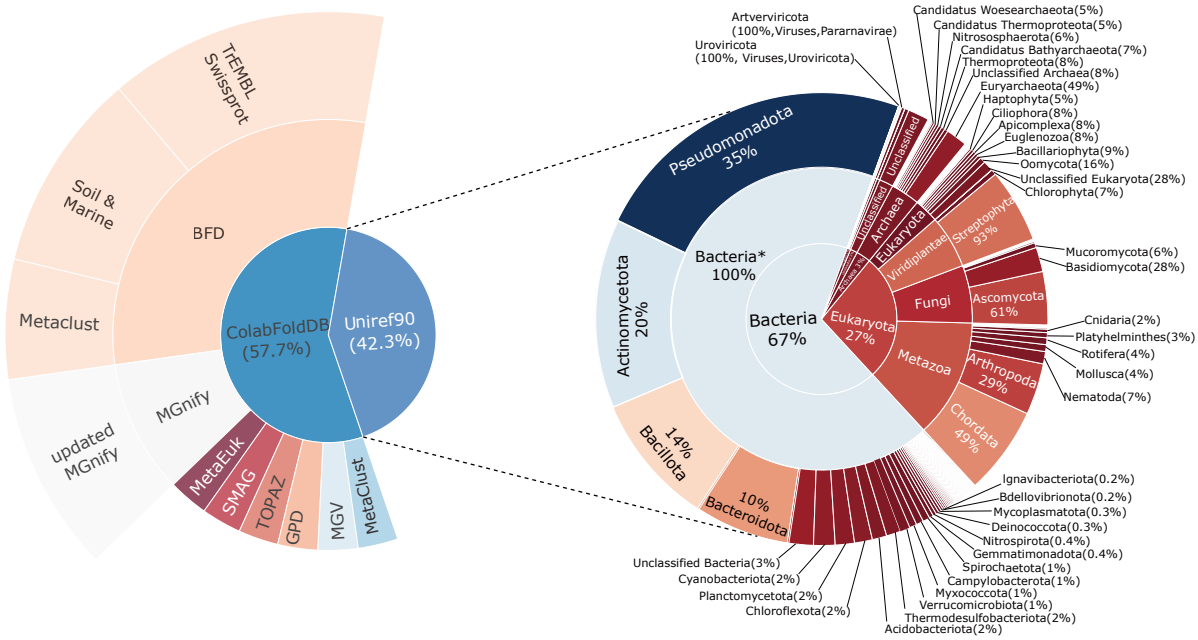


Figure 11: The left panel illustrates the dataset composition used for pre-training the model. The right panel depicts the distribution of taxonomic categories of Uniref90, visualized as concentric circles representing the levels of superkingdom, kingdom, and phylum from innermost to outermost. The innermost 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.

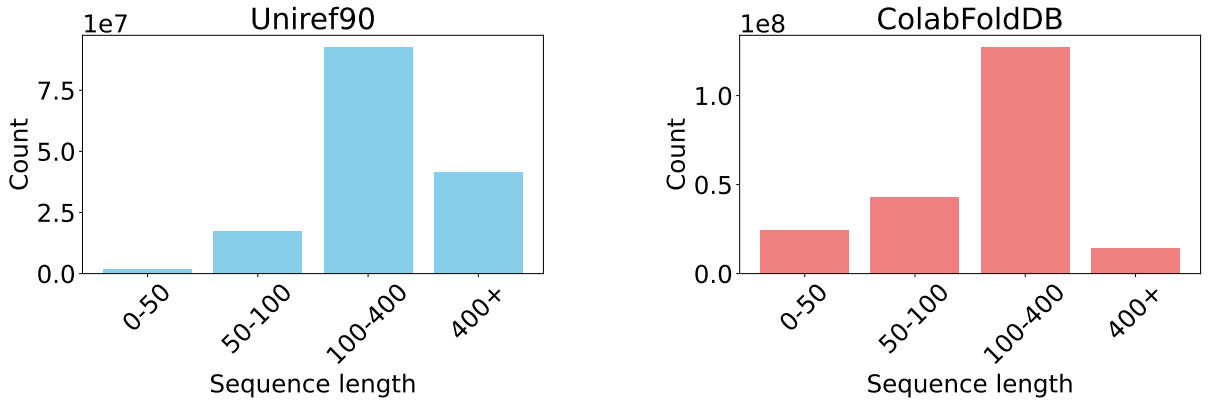


Figure 12: Training Data Distribution

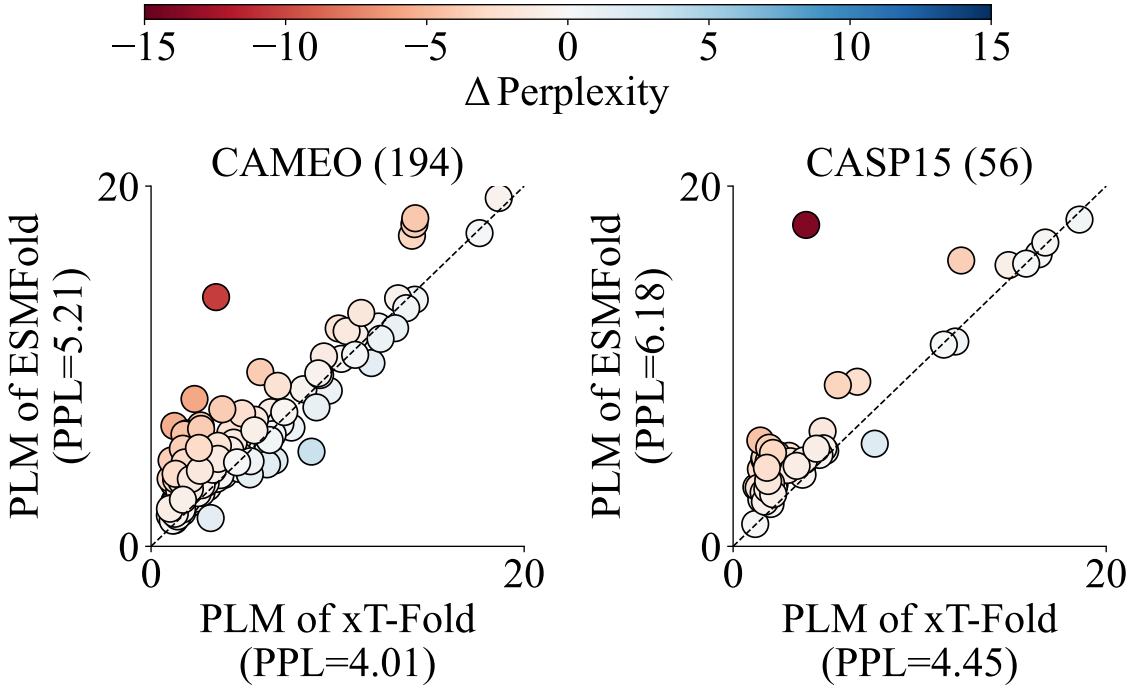


Figure 13: Scatter plots show delta perplexity for CAMEO and CASP15. The perplexities (PPL) are from the PLM modules of xT-Fold and ESMFold. Points represent proteins, with color gradients indicating perplexity delta by the x-axis PPL minus the y-axis PPL.

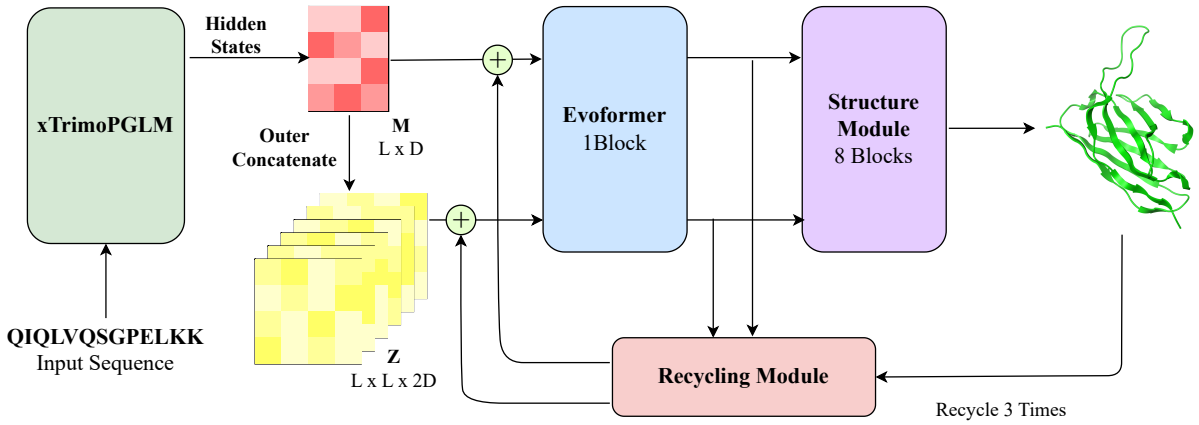
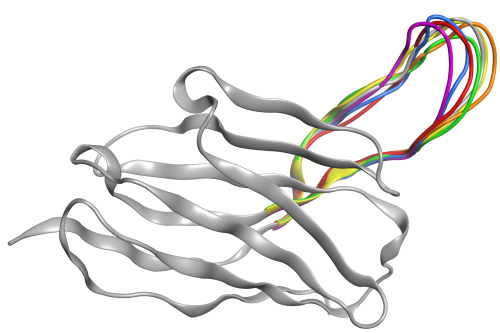


Figure 14: Architecture of xTrimoPGLM-AbFold for structure prediction.

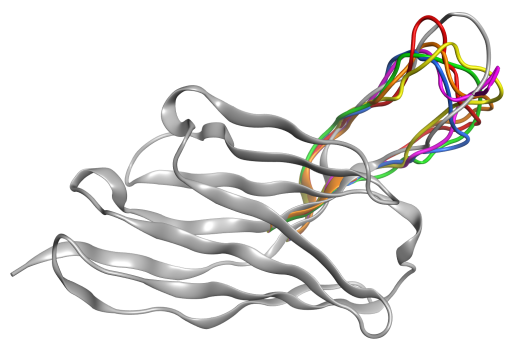
(“:” denotes the residue pairs of distance < 5)

```
Generated  -QLVESGGGVVQTGRSLRLSCAASGRTFSSHALGWFRQAPGKEREFVAAVGRGGGSAGYEDSVKGRFTVSRDNAKNTVYLQMKSLGPEDTAVYYCAYDYDYDGGGYHGARDYYDSWGRGTLVTVSSG
           :::::::::::::::::::::::::::: :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
6h1f_A     VQLQESGGGLVQAGGSLRLSCAASGRTFSSFVMGWFRQAPGKEREFVASISRSGSVTRYADSAKGRFTISKDNAKNTVSLQMDNLNPDDTAVYYCAADLHRPYGPGSQRTDDYDTWGQGTQVTVSSL
                                  CDR1                    CDR2                                                                CDR3

Generated  EVQLQQSGPELVKPGASVKLSCKASGYTFTSYGMNWVKQAPGKGLEWIGWIDTGTGTGKSFAHFKGKATLTVDTSSTTAYMQLRSLTSDDSAIYYCVAFG-YGGGYGGYGAGAMDYWGQGTLVTVSS
           ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
6ol6_H     QVQLVQSGAEVKKPGASVKVSCKASGYTFTSYAMHWVRQAPGQRLEWMGWINAGNGNTKYSQKFQGRVTITRDTSASTAYMELSSLRSEDTAVYYCARVGPYCGGDSPQDNYYMDVWGKGTTVTVSS
                                  CDR1                    CDR2                                                                CDR3

Generated  EVQLQESGGGLVQPGGSLRLSCAASGFTFSDYAMSWVRQAPGKGLEWVSGINGDGSTYYPDSVKGRFTISRDNAKNSLYLQMNSLRAEDTAVYYCARLGVSYYAGSY----YYYYGMDVWGQGTTVTVSS
           ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
8ded_U     EVQLVESGGGLAKPGGSLRLSCAASGFTFSSYAMHWVRQAPGKGLEWVSVISSHYSTYYADSVKGRFTISRDNSKNTVSLQVNSLRAEDTAVYYCAKVGRSGYCTASGCNALLFDWFDVWGPGVLVTVSS
                                  CDR1                    CDR2                                                                CDR3

Generated  EIVLTQSPGTLSLSPGERATLSCRASQSVSSSYLAWYQQKPGQAPRLLIYGASSRATGIPDRFSGSGSGTDFTLRISRLEPEDFAVYYCQHRGDSWTFGGGTKVEIKLIQSPGSL
           ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
6ol6_L     EIVLTQSPGTLSLSPGERATLSCRASQSVSSSYLAWYQQKPGQAPRLLIYGASSRATGIPDRFSGSGSGTDFTLTISRLEPEDFAVYYCQQYGSSSYTFGQGTKLEIKRLVPR--
                                  CDR1                    CDR2                                                                CDR3
```

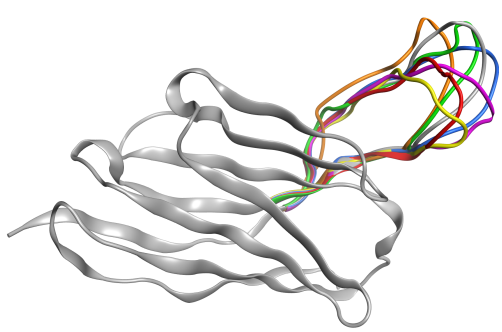
Figure 15: Sequences comparison about generated antibodies.



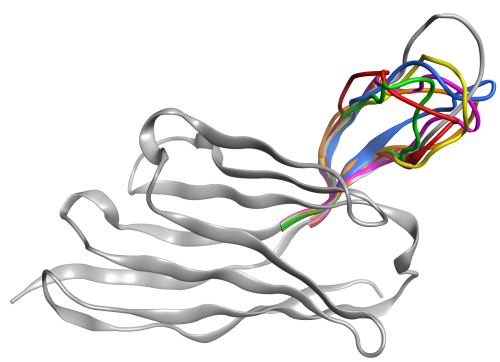
(a) CDR3 Short Masking



(b) CDR3 Whole Masking



(c) CDR3 Random Mutation



(d) CDR3 Random Retrieval

Figure 16: Conformations of antibody generation. The conformations of various methodologies implemented for xTrimoPGLM-AbFold.

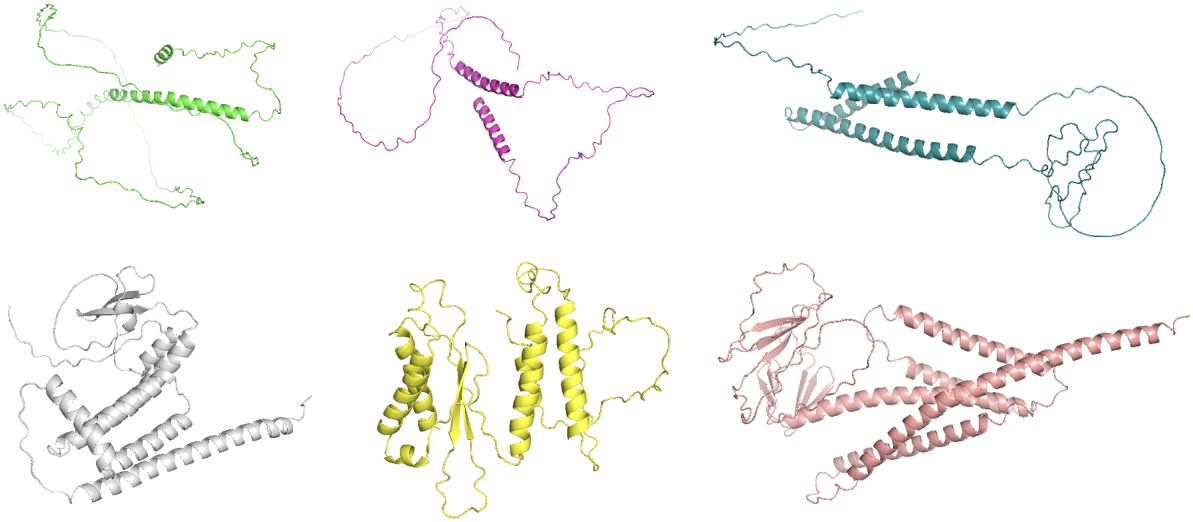


Figure 17: Structure examples of generated protein sequences with different parameter configurations.

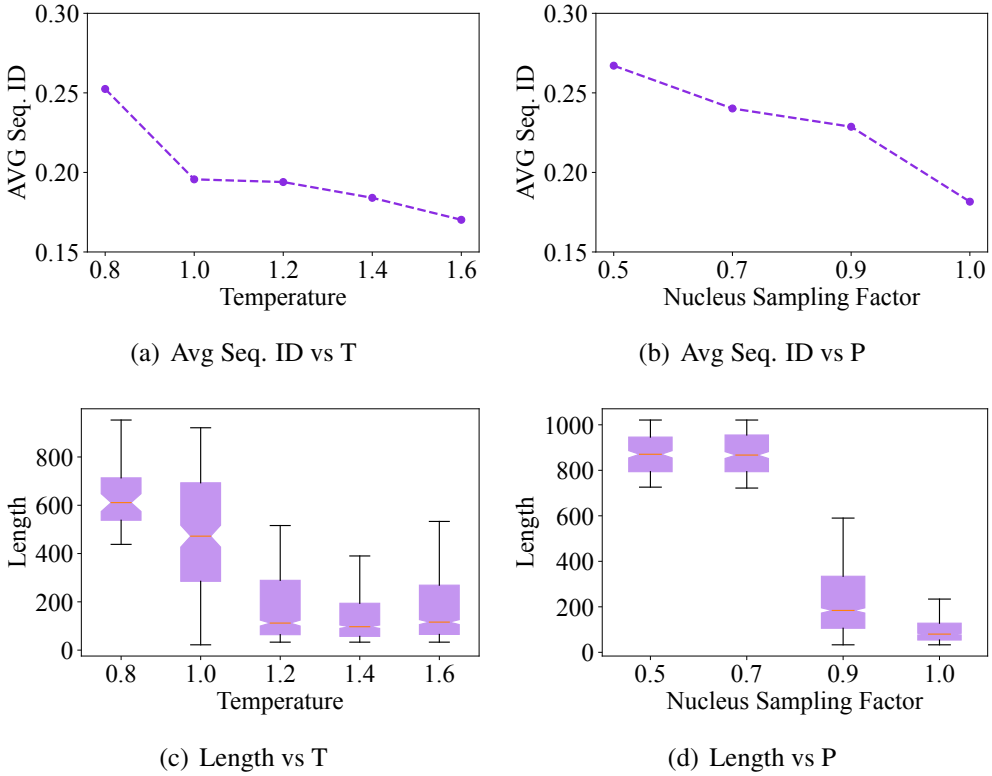


Figure 18: The sequence identity and length distributions of generated sequences with different temperatures (T) and nucleus sampling probability (P).

Table 2: Summary information for 18 benchmarked downstream tasks.

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 (*5*)	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 (*5*)	76.49♣	**79.45**
	Stab. P.	SRCC	53.6K	2.5K	12.8K	ESM2-15B (*5*)	80.75♣	**84.21**
	Temp. Stab.	MCC	283K	63K	73.2K	TemStaPro (*80*)	83.80♦	**94.22**
	Opt. Temp.	SRCC	1.7K	-	190	DeepET (*81*)	62.40 ♦	**73.96**
	Opt. PH	AUC	7.1K	760	1.9K	ESM2-15B (*5*)	62.48 ♦	**64.99**
	Clo. CLF	AUC	23.3K	-	4.7K	ESM2-3B (*5*)	77.09 ♦	**84.82**
	Mat. Pro.	AUC	23.3K	-	4.7K	ESM2-15B (*5*)	79.17 ♦	**86.48**
Inter.	Metal B.	2-cls ACC	6K	-	1.3K	ESM2-15B (*5*)	79.35♣	**82.78**
	Enzyme Eff.	PCC	13.5K	1.7K	1.7K	DLKcat (*85*)	71.00♦	**74.79**
	Pept.-HLA Aff.	AUC	575K	144K	171K	CcBHLA (*86*)	95.00♦	**96.68**
	TCR-pMHC Aff.	AUC	19.5K	-	4.5K	epiTCR (*87*)	92.50♦	**95.10**
Func.	Antib. Res.	19-cls ACC	2K	-	1.3K	ESM2-15B (*5*)	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

Table 3: Performance of different models across all benchmarked downstream protein-related tasks.

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. Sta.	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)
	Opt. PH	AUC	61.39 (**64.99**)	59.35 (62.48)	59.60 (56.20)
	Clone CLF.	AUC	84.55 (**84.82**)	76.06 (76.64)	73.55 (76.07)
	Mat. Pro.	AUC	86.52 (**86.48**)	79.35 (79.17)	76.18 (77.91)

Table 4: Comparison of training computes between different pre-trained protein language models.

Model	Total train compute (FLOPs)	Params	Training tokens
ESM2-150M	1.1E+21	150M	1,000B
ESM2-650M	4.4E+21	650M	1,000B
ESM2-3B	1.8E+22	2.8B	1,000B
ESM215B	5.1E+22	15B	864B
ProtBert	7.6E+21	420M	2516B
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	101B	1,000B

Table 5: Performance of different antibody pre-training models in zero-shot naturalness datasets.

Model	DATASET 1			DATASET 2		
	H Chain	L Chain	Pair	H Chain	L Chain	Pair
Iglm (*105*)	0.698	0.651	0.683	0.703	0.594	0.665
AbLang (*106*)	0.655	0.497	0.613	0.713	0.671	0.679
ESM2-15B (*44*)	0.682	0.552	0.686	0.716	0.510	0.626
AntiBERTy (*107*)	0.763	0.549	0.699	0.723	0.678	0.679
Progen2-oas (*11*)	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

Table 6: 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

Table 7: 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

Table 8: 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)

Table 9: A collection of sequences produced via four distinct masking approaches.

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

Table 10: Disorder proteins/domains predictions (%).

	SHORT	LONG	GLOBULAR	ORDERED
Natural Data (10K)	59.84	64.27	64.96	34.56
Generated	63.38	68.16	68.57	34.20
Random Generation	56.10	55.11	54.84	92.58