nature methods

Article

https://doi.org/10.1038/s41592-025-02636-z

xTrimoPGLM: unified 100-billion-parameter pretrained transformer for deciphering the language of proteins

Received: 7 February 2024

Accepted: 24 February 2025

Published online: 03 April 2025

Check for updates

Bo Chen[1,2,4] ✉, Xingyi Cheng[1,4] ✉, Pan Li[1], Yangli-ao Geng[1,2], Jing Gong[1], Shen Li[1], Zhilei Bei[1,2], Xu Tan[1], Boyan Wang[1,2], Xin Zeng[1], Chiming Liu[1], Aohan Zeng[2], Yuxiao Dong[2], Jie Tang[2] ✉ & Le Song[1,3] ✉

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 pretraining 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 pretraining 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 substantially outperforms other advanced baselines in 18 protein understanding benchmarks across four categories. The model also facilitates an atomic-resolution view of protein structures, leading to an advanced three-dimensional 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 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. Trained weight for the xTrimoPGLM model, and downstream datasets are available at https://huggingface.co/biomap-research.

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[1–3]. As protein sequences serve as the blueprint for protein structure and function[4], pretrained techniques on sequences, known as protein language models (PLMs), for example, the family of ESM models[5,6], ProtTrans[7] and PROGEN[8], offer a powerful tool for characterizing the properties and distributions of general protein sequences. These models are trained on large-scale protein datasets[9–11] 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

[1]BioMap Research, Palo Alto, CA, USA. [2]Tsinghua University, Beijing, China. [3]MBZUAI, Abu Dhabi, United Arab Emirates. [4]These authors contributed equally: Bo Chen, Xingyi Cheng. ✉e-mail: cb21@mails.tsinghua.edu.cn; derrickzy@gmail.com; jietang@tsinghua.edu.cn; dasongle@gmail.com

functions and structures[1,2,6] or generating novel sequences with faithful three-dimensional (3D) structures[8,12].

It is worth noting that different categories of protein-related tasks necessitate divergent outputs from PLMs, for example, 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[4,13], 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 pretraining framework. This presents a fundamental 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 pretraining 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 pretraining paradigms. However, these studies typically adopt analogous training patterns. For instance, all pretraining objectives are commonly optimized using either the BERT-style[14] or the GPT-style regime[15]. A balanced approach incorporating both bidirectional autoencoding and unidirectional autoregressive 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-shot/few-shot learning[16] or instruction tuning[17,18]. 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[6,19]. These tasks heavily rely on bidirectional autoencoding training to tackle protein understanding tasks. Consequently, this highlights the need for a unified model that incorporates both training objectives.

In this work, we develop the first, to our knowledge, xTrimo protein general language model (xTrimoPGLM), a unified pretraining framework and foundation model that scales up to 100 billion parameters, designed for various protein-related tasks, including understanding and generation (or design). The model differs from previous encoder-only (for example, ESM) or causal decode-only (for example, PROGEN) PLMs by leveraging the general language model (GLM)[20] as the backbone for its bidirectional attention and autoregressive 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 pretraining 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 × A100 GPU cards.

xTrimoPGLM-100B demonstrates the notable enhancement in the realm of protein understanding. By conducting extensive empirical experiments with linear probing and advanced fine-tuning techniques, we elevated the performance benchmarks in this domain. xTrimoPGLM-100B has substantially surpassed previous state-of-the-art methods in 15 of 18 tasks, covering a comprehensive range of areas including protein structure, interactions, functionality and developability (Fig. 2a). We also illustrate that xTrimoPGLM achieves lower perplexity (PPL) on two out-of-distribution (OOD) protein sets over other models (Fig. 1b). These results empirically validate the scaling behavior, demonstrating that larger models commonly tend to yield better performance (Figs. 1c and 2b).

xTrimoPGLM can serve as the base for developing a high-performance 3D structural prediction tool. Inspired by methodologies similar to those in ESMFold, merge folding modules with a PLM, 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 five-point TM-score increase over ESMFold in the CASP15 dataset, coupled with a faster inference speed across various scenarios (Fig. 3).

xTrimoPGLM also 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% (Fig. 4), but also can be tailored toward specific structural and biophysical properties through supervised fine-tuning (SFT; Figs. 5 and Fig. 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 PLM in practical protein applications. Although our study confirms the potential of PLMs, 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.

Results

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 bidirectionally. This dual approach is enhanced by the integration of an MLM objective[21] to enhance its understanding capacity. The core of our model is the simultaneous optimization of two distinct pretraining objectives, each characterized by unique indicator tokens, ensuring the proficiency in both understanding and generation capacities:

- 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[21] and ESM[6], focusing on bidirectional contextual understanding.
- 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 Fig. 1a. Motivated by the philosophy of curriculum learning, the pretraining stage is conducted in two distinct phases: (1) initial pretraining with the MLM objective, focusing on rapid loss minimization across approximately 400 billion tokens. This phase is geared toward enhancing the model's understanding capabilities; (2) subsequent training uses 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.

https://doi.org/10.1038/s41592-025-02636-z

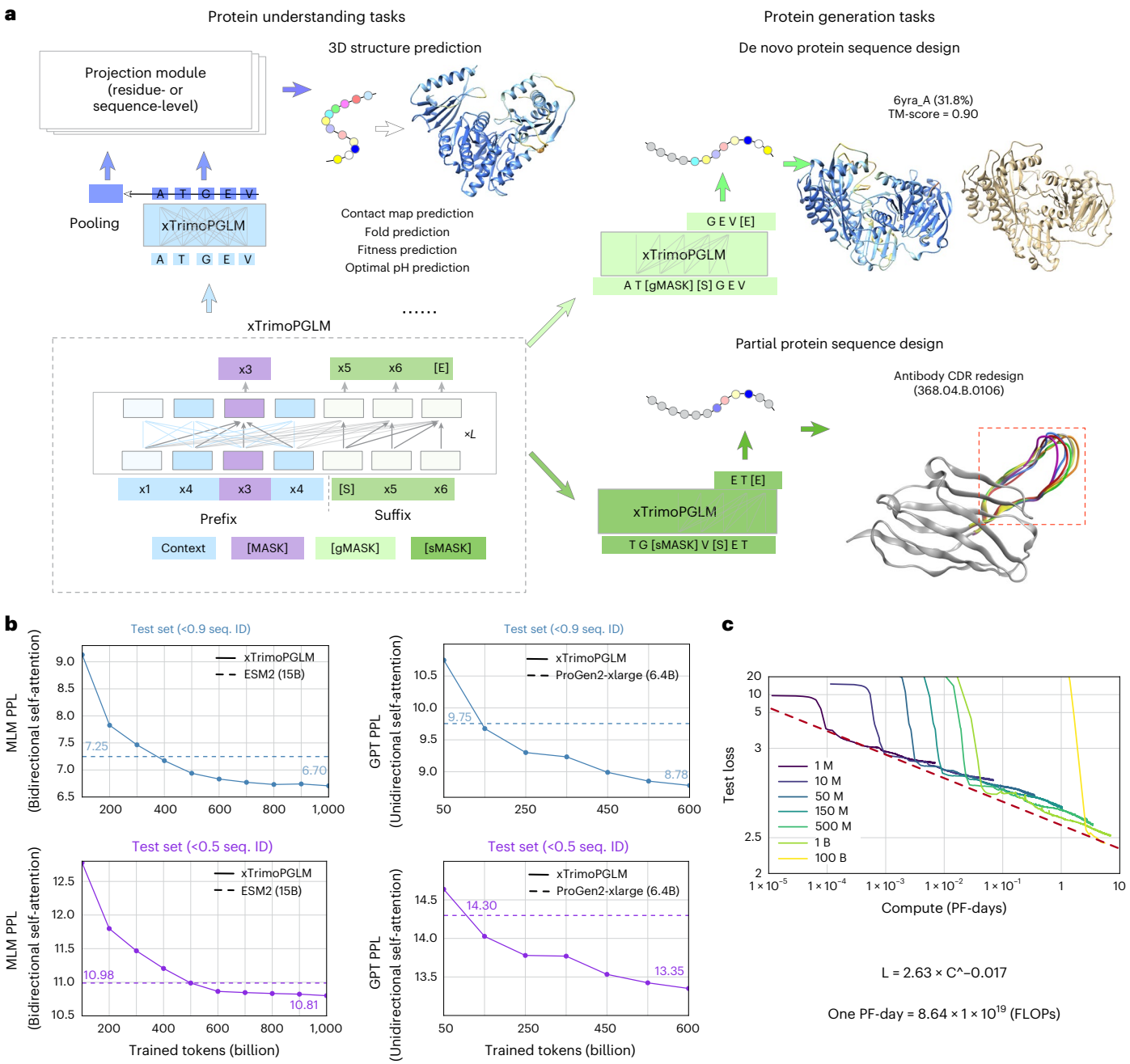


Fig. 1 | Comprehensive insights into xTrimoPGLM. a, The pretraining and fine-tuning stages of xTrimoPGLM, combining BERT-style (blue and purple, for masking and predicting tokens) and GPT-style (green, from [S] to [E], for autoregressive generation) objectives. The prefix's bidirectional attention facilitates protein understanding tasks like structure prediction, while the suffix supports both de novo and conditional protein design through sequence generation. **b**, xTrimoPGLM shows lower perplexity than other leading PLMs like ESM2 and ProGen2-xlarge in evaluations on two distinct OOD datasets, indicating its advanced performance. **c**, The scaling behavior of xTrimoPGLM-series 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. CDR, complementarity-determining region.

Quantification scaling law of xTrimoPGLM family models. The scaling law is a crucial concept for understanding the performance of LLMs during pretraining. 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 use the formula $L = a \times C^b$, where C is approximated by $6ND$, N is the model size and D is the pretrained dataset size (set to 100 billion tokens in this case for all the xTrimoPGLM family models)[22,23]. Thus the scaling behavior between $(L(C))$ and total compute C can also be viewed as the rule between $(L(C))$ and model scale (N). We quantify C in terms of training floating-point operations (FLOPs), using PetaFLOP-days (PF-days) as the unit of measure, where one PF-day = 8.64×10^{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 (Fig. 1c). Remarkably, this curve demonstrates 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

https://doi.org/10.1038/s41592-025-02636-z

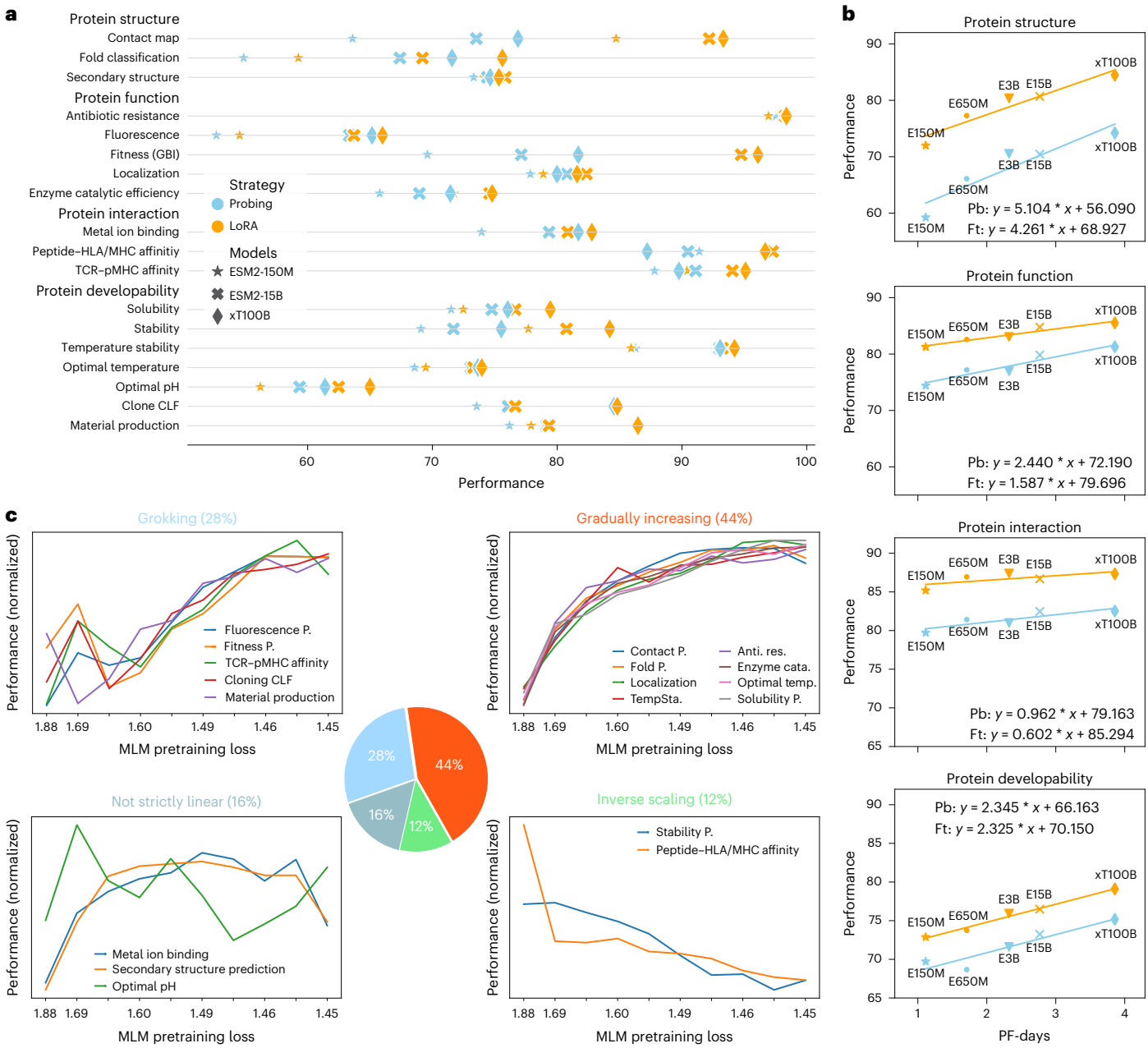


Fig. 2 | The performance of protein understanding benchmark. a, For the classification task, four metrics are used, including TopL/5 accuracy (contact map), accuracy (fold classification, secondary structure, antibiotic resistance, solubility, localization, metal ion binding), area under the curve (peptide–human leukocyte antigen (HLA)/major histocompatibility complex (MHC) affinity, T cell antigen receptor (TCR)–pMHC affinity, clone CLF, material production) and Matthews correlation coefficient (temperature stability). For the regression task, two metrics are used including the Spearman correlation coefficient (fluorescence, fitness, stability, optimal temperature, optimal pH) and the Pearson correlation coefficient (enzyme catalytic efficiency). GB1 is the data domain of the fitness prediction task. CLF is a binary label of the Clong CLF task indicating whether a protein failed during the cloning stage of experimental structure determination. **b**, The scaling trend between the computational cost of model training, quantified by PF-days, where one PF-day = 8.64×10^{19} FLOPs, and the model performance. Each data point symbolizes the mean performance metric for a specific task category (Pb, probing; Ft, fine-tuning with LoRA). E150M/650M/3B/15B and xT100B represent ESM2-150M/650M/3B/15B and xTrimoPGLM-100B, respectively. **c**, Correlations between the pretraining validation loss measured by MLM objective and the performance of the downstream tasks. For the fair comparison, we normalized the value by subtracting the mean value and dividing it by the standard deviation.

law, thereby reinforcing the credibility of this principle as a pivotal guideline in the development of large-scale PLMs.

xTrimoPGLM achieves low perplexity on OOD 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 pretraining datasets. Each OOD dataset comprises approximately 10,000 protein sequences (Fig. 1b). xTrimoPGLM-100B achieves perplexity

https://doi.org/10.1038/s41592-025-02636-z

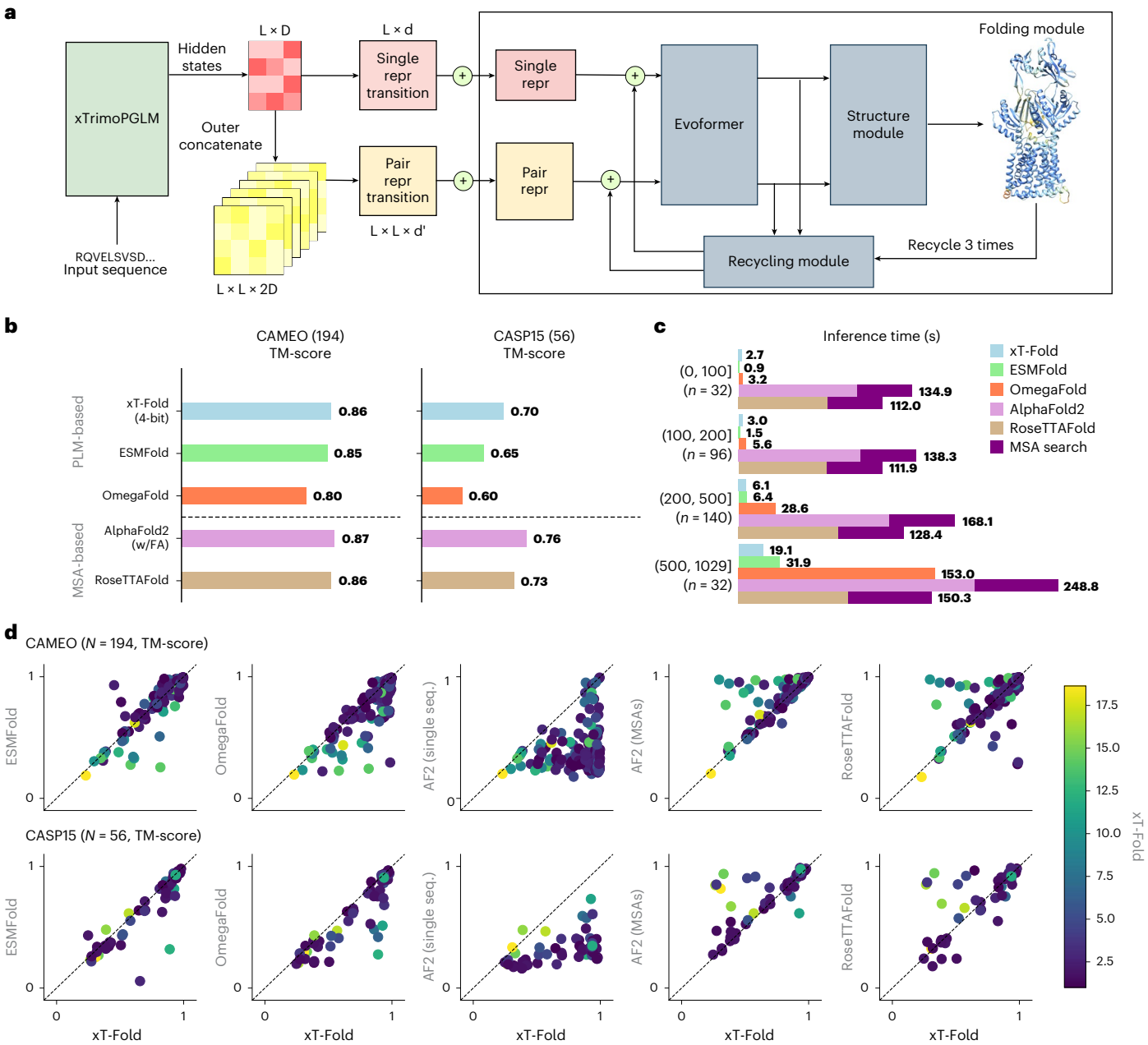


Fig. 3 | Structure prediction with xT-Fold. a, xT-Fold architecture leverages a MLP to convert PLM representations into inputs for the folding modules, which generate 3D coordinates and pLDDT confidence scores. **b**, TM-score benchmarks for structure prediction models. The bar chart shows the performance 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**, Scatterplots compare xT-Fold predictions (*x* axis) to other models (*y* axis), color coded by perplexity (green for high, purple for low).

scores of 10.81 and 6.70 on the 0.5 and 0.9 sequence identity datasets, respectively, outperforming the ESM2 15-billion parameter model (ESM2-15B; 10.98 and 7.25). Similarly, against ProGen2-xlarge with 6.4 billion parameters, xTrimoPGLM recorded 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 with respect to the corresponding pretraining objective: less than the actual 480 billion trained tokens for MLM pretraining (400 billion tokens at phase 1 and 80 of 100 billion at phase 2) to match ESM2 trained on 1 trillion tokens, and less than the actual 120 billion tokens for the GLM pretraining (120 of 150 billion tokens at phase 2) to parallel the ProGen2-xlarge model trained with 350 billion tokens, even if the learning rate schedule had not yet ended, resulting in an overestimated loss value at this stage. Note that, the empirical studies in the Methods confirm that the GLM model, when continued from MLM pretraining, converges faster than GLM model pretrained from scratch. Thus, the initial 400 billion MLM tokens serve as a warm-up stage, enhancing the model's ability to understand sequence distributions. This foundational understanding accelerates the convergence during the subsequent GLM pretraining phase to enhance the model's generation capacity.

Evaluation of protein understanding benchmarks

To comprehensively assess the understanding capabilities of xTrimoPGLM-100B, we benchmarked it against 18 downstream protein-related tasks. These tasks span four primary categories:

https://doi.org/10.1038/s41592-025-02636-z

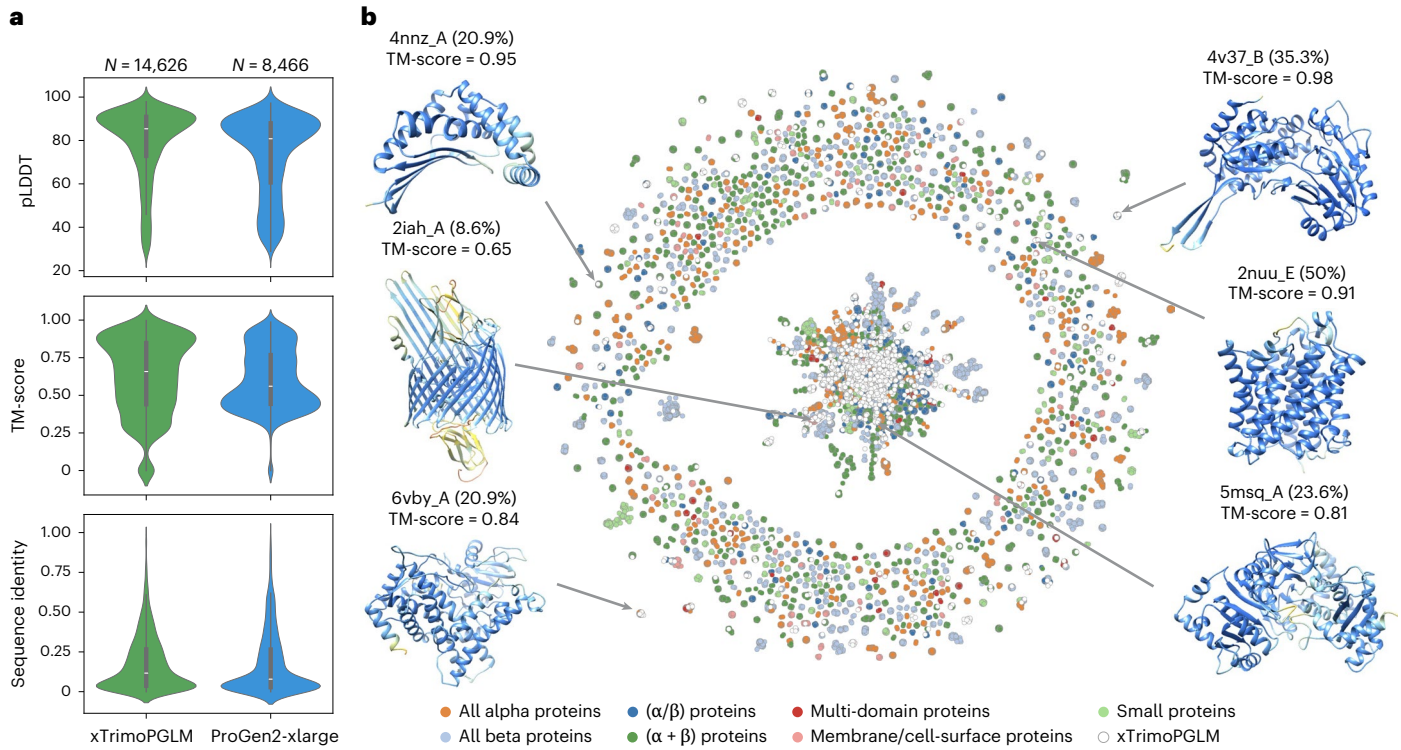


Fig. 4 | Diversification of generated proteins by xTrimoPGLM. a, Violin plots comparing ESMFold-predicted confidence (pLDDT scores) and similarity to PDB entries—measured by TM-score and sequence identity—for sequences generated by xTrimoPGLM (green, $N = 14{,}626$) and ProGen2-xlarge (blue, $N = 8{,}466$). The plots display the median, and upper and lower quartiles, and whiskers represent 1.5 times the interquartile range. **b**, The 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 illustrations (Extended Data Fig. 5), we showcase more generated examples. 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 the structure matches the xT-Fold pLDDT values. The blue color represents high confidence (pLDDT>90).

protein structure, developability, interactions and functions (Fig. 2a and Supplementary Section 1). The protein structure category involves modeling the protein structure from the primary sequence, including secondary and tertiary structures. The protein developability category focuses on the engineering characteristics of proteins, such as their stability and manufacturability. The protein interaction category includes scenarios where proteins interact with other molecules, such as short peptides or other proteins. The protein function category encompasses tasks that predict the intrinsic cellular features and activities of proteins, including enzyme catalytic efficiency. We compared xTrimoPGLM-100B with two other PLMs in the field, ESM2-150M and ESM2-15B[6], to provide a well-rounded evaluation. Our assessment methodology encompasses two distinct approaches to evaluate the effectiveness of the models' representations:

- Probing with multilayer perceptron (MLP). We utilized a trainable MLP as a probe to examine the evolutionary information encoded in the pretrained representations. This method simply and efficiently identifies the types of protein information captured by the models, where pretrained PLMs' parameters remain fixed, and only the MLP is trained. For comparisons, the embeddings from pretrained models are projected into 128 dimensions followed by ReLU activation before passing to the next layer of MLP.
- Fine-tuning with low-rank adaptation (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 used LoRA[24] as a parameter-efficient alternative. This technique involves freezing the pretrained model's weights and adding trainable low-rank matrices to each transformer layer. LoRA drastically 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 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) substantially outperformed the smaller ESM2-150M model. 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 was 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,

https://doi.org/10.1038/s41592-025-02636-z

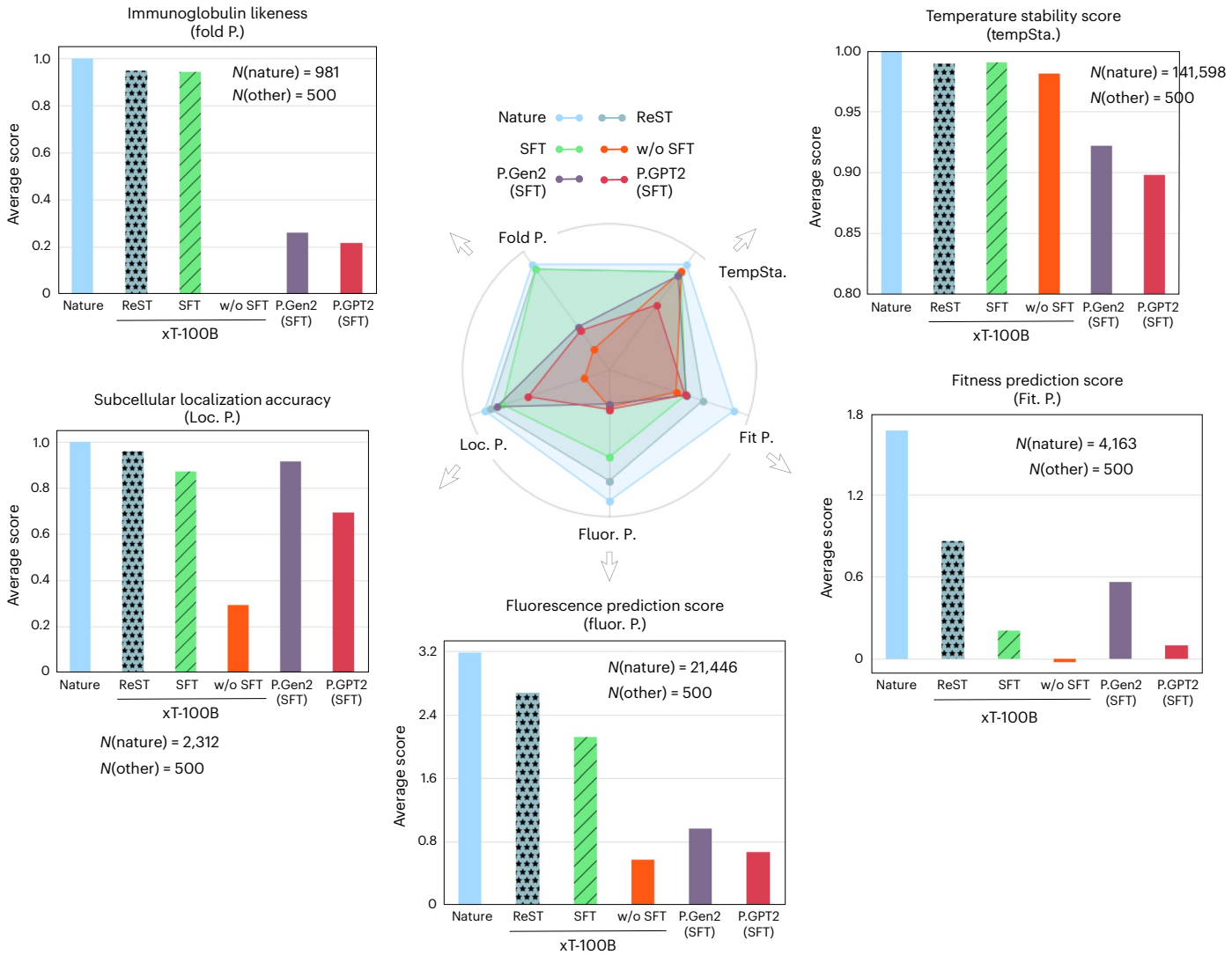


Fig. 5 | Robust alignment capabilities of xTrimoPGLM in protein sequence generation toward desired properties. Quantitative analysis of xTrimoPGLM, enhanced with SFT and ReST, across five selected tasks. The number of sampled generated sequences, *N*(other), and natural sequences, *N*(nature), used in the analysis are illustrated. The results demonstrate xTrimoPGLM's effectiveness in aligning with specific task objectives, as shown by the average scores (higher scores indicate better alignment). P.Gen2 refers to the ProGen2-xlarge model[12] with 6.4 billion parameters, and P.GPT2 denotes the ProtGPT2 model[29] with 740 million parameters.

particularly for complex tasks. The application of LoRA consistently enhances overall task performance compared to the static probing method, which limits the capacity of pretrained models (Fig. 2b). LoRA's efficiency lies in its ability to refine and utilize key features without appreciably 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 SFT, 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 extended the investigation of scaling laws to downstream tasks in four distinct categories of protein-related tasks from three perspectives.

Comparison between ESM and xTrimoPGLM family models. We observed the scaling behavior of downstream tasks among the ESM and xTrimoPGLM family models using both linear probing with MLP (Pb) and fine-tuning with LoRA (Ft) techniques (Fig. 2b). The *x* axis measures the total computational cost in PF-days, while the *y* axis represents the performance of various task categories. The results indicate that, although the intensity of the scaling effect varies among task types, a common trend persists: an exponential increase in computational resources during pretraining correlates with linear improvements in downstream task performance. This extension of the scaling law to downstream tasks in protein modeling is a new observation.

Comparison among different training states of xTrimoPGLM. To eliminate the differences in backbone architectures, pretrain datasets and so on, we obtained a deeper insight into the correlations between the downstream task performance and the ongoing pretraining process (measured by the MLM validation loss; Fig. 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[25]. It occurs when models initially overfit the training data, especially when task datasets have limited overlap with pretraining data. As the model's knowledge base expands, it begins to apply its understanding to OOD scenarios, resulting in a sudden increase in

https://doi.org/10.1038/s41592-025-02636-z

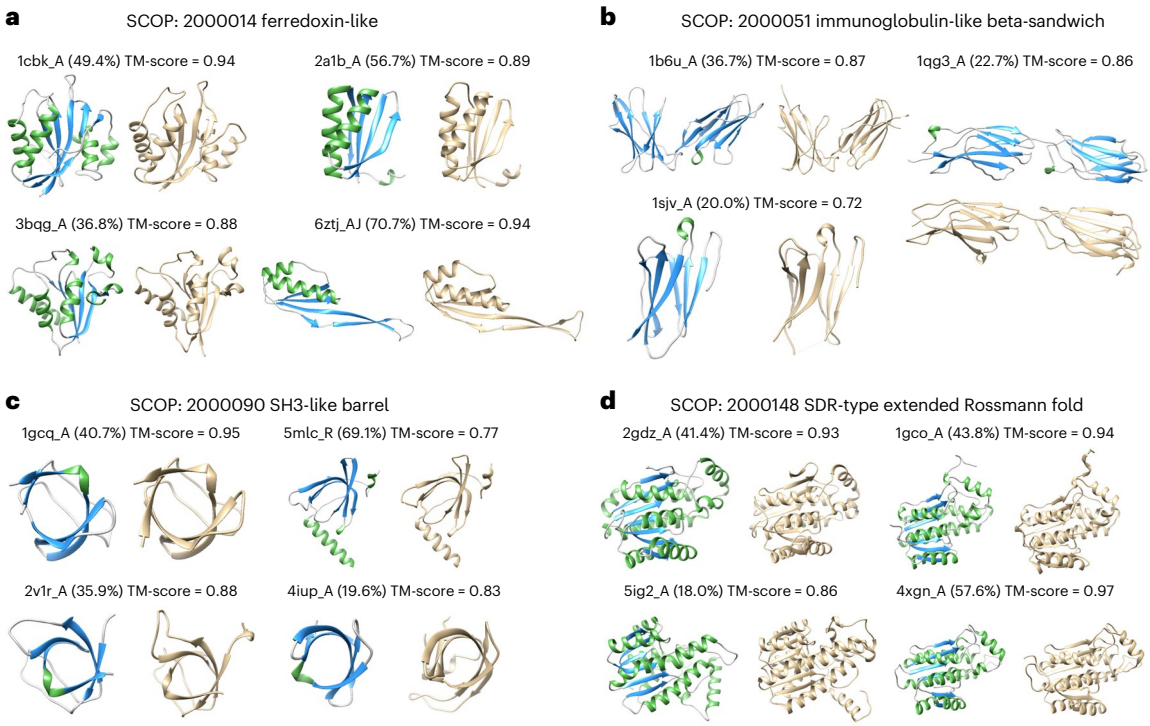


Fig. 6 | Cases study of xTrimoPGLM in controllable generation. **a**–**d**, Generation of four SCOP fold types by xTrimoPGLM (SFT): ferredoxin-like (**a**), immunoglobulin-like beta-sandwich (**b**), SH3-like barrel (**c**) and SDR-type extended Rossmann fold (**d**). Generated protein structures are depicted in interleaved green, blue and gray, whereas gold-colored structures represent the most structurally similar proteins from the PDB. Percentages in parentheses indicate sequence identity.

test data performance. Thus, the performance of the test data suddenly increased, denoted as the Grokking phenomenon. However, 12% of tasks indicated a potential negative impact from increased computational resources, suggesting an inverse scaling effect.

Comparison among different scales of xTrimoPGLM family models. We further analyzed the scaling behavior by comparing xTrimoPGLM family models (Extended Data Fig. 1), which showed a similar trend that most tasks exhibit a positive relationship between task performance and the scaling of training FLOPs and model size.

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[26]. These empirical observations provide valuable insights for future research in model development and advancement in the field of protein language modeling.

Evaluation of protein structure prediction

Protein sequences encode rich information about their 3D structures and function through evolutionary history[4]. Advanced methods using multiple sequence alignments (MSAs) like AlphaFold2 (ref. 1) and RoseTTAFold[27] are highly accurate in predicting protein structures and are key tools in computational biology. Meanwhile, PLM-based models such as ESMFold[6] and OmegaFold[19], 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.

Building on the xTrimoPGLM-100B framework, we propose xT-Fold, a model that marks a remarkable advancement by achieving state-of-the-art results for the PLM-based structure prediction model on benchmarks such as CAMEO and the latest CASP15. Notably, xT-Fold offers both high accuracy and computational efficiency, utilizing 4-bit quantization and FlashAttention (FA)[28] techniques to run effectively on a single A100 GPU card (Supplementary Section 2) This balance of speed and precision makes xT-Fold a tool of choice for fast-paced research and drug discovery.

The overall architecture of xT-Fold closely resembles that of ESMFold (Fig. 3a). It involves training MLP layers to map the respective single representation and pair representation to d and d' from the D dimension, which are fed into the structure module for prediction of 3D coordinates. We used a 48-layer evoformer with approximately 88 million parameters. The structure module accounts for about 2 million parameters, plus additional heads (81,000), bringing the total to around 90 million parameters. The maximum recycle count is 3. During training, the recycle count was randomly selected between 0 and 3, consistent with the original AF2 and ESMFold approaches.

We evaluated two individual test sets, CAMEO and CASP15, both of which are OOD datasets in terms of their timelines, ensuring differences from our training set (Supplementary Section 3). CAMEO includes 194 samples (release date between April 2022 and June 2022), while CASP15 consists of 56 publicly available proteins (released in May 2022). Initially, we observed the perplexity (the lower the better) of these two datasets in PLMs, which served as the backbone of the folding models (Extended Data Fig. 2). 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 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 including ESMFold, OmegaFold and 4-bit xT-Fold (Fig. 3b and Supplementary Table 1). 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, that is, 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 (Fig. 3c), lagging by approximately 10× to 50×. This disparity remains even for accelerated AlphaFold2, which is optimized with FlashAttention (FA), achieving an acceleration of 2× to 8× compared to the open-source variant (available at https://github.com/google-deepmind/alphafold/) across sequence lengths ranging from 200 to 1,000 and kept the output consistent with the canonical self-attention. For the PLM-based methods, the open-source versions of ESMFold (https://github.com/facebookresearch/esm/) and OmegaFold (https://github.com/HeliXonProtein/OmegaFold/) were used. As a 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 ten recycling passes to achieve better results.

Upon further examination of the scatterplots (Fig. 3d), a perceptible correlation between the perplexity from xTrimoPGLM and the structural metric TM-score was observed. Notably, for samples with intermediate PPL values, xT-Fold demonstrated enhanced performance compared to ESMFold and OmegaFold. xT-Fold is also juxtaposed with AlphaFold2 with single-sequence input. Given that AlphaFold2 is not trained on single sequences, a substantial drop in overall effectiveness is evident. The last two columns compare xT-Fold with 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. To provide a more comprehensive and quantified understanding of the relationship between sequence perplexities (PPL) and the corresponding TM-scores predicted by xT-Fold, we calculated the Pearson correlation coefficient and P value for the CAMEO, CASP14 and CASP15 datasets: $(-0.415, 1.7 \times 10^{-9})$, $(-0.579, 1 \times 10^{-5})$ and $(-0.239, 8 \times 10^{-2})$, respectively. These results clearly indicate negative correlations between PPLs and TM-scores on CAMEO and CASP14, but not strictly correlations in CASP15. Overall, these quantified metrics demonstrate that lower perplexities are associated with more accurate predicted structures. In summary, while scaling single-sequence models enhances the performance, it still struggles with OOD data, which is effectively addressed by MSA-based augmentation.

Evaluation of 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 markedly diverge from naturally occurring proteins[12,29]. To validate the generative ability of xTrimoPGLM-100B, we conducted 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 pretraining data and task-specific sequence generation via 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 generates sequences with diverse structures. To evaluate the generative capacity of xTrimoPGLM, we generated 14,626 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 autoregressive manner until the `<eos>` token is predicted or the preset maximum length is reached. We used nucleus sampling by combining different top P values (0.5, 0.7, 0.9, 1.0) and sampling temperature (0.2, 0.4, 0.6, 0.8, 1.0) parameters. For each parameter combination, we generated 2,000 protein sequences and limited the maximum length to 800 tokens. Then we performed a simple filtering on the generated sequences: (1) removing the generated sequences with perplexity > 10; (2) removing the sequences containing repeated fragments; (3) MMseqs2 clustering (--min-seq-id 0.9 -c 0.5) and only leaving the centroid sequence. We also used same strategy to generate 8,466 sequences for ProGen2.

We used ESMFold to predict the structures of all generated sequences. Our model generated proteins with higher confidence scores than ProGen2-xlarge (mean predicted local distance difference test (pLDDT) scores 84.0 versus 74.3; Fig. 4a). We then used Foldseek to search for the most structurally similar natural proteins from the Protein Data Bank (PDB) database. We measured the structural and sequence similarity using TM-score and sequence identity, respectively. Our model exhibited much higher structural resemblance to PDB entries than ProGen2-xlarge (mean TM-score 0.695 versus 0.522) with very low sequence identity (mean sequence identity 0.224 versus 0.165) and high diversity (Extended Data Fig. 3). We have stratified the generated sequences into four groups based on perplexity values (<2, 2–5, 5–8, >8). Across all perplexity ranges, our model consistently produced proteins with higher confidence scores and greater structural and sequence resemblance to PDB entries compared to ProGen2 (Extended Data Fig. 4).

To measure the generated protein distribution in the protein space, we used Foldseek to align the structures predicted by ESMFold to the AlphaFold/UniProt50-minimal structure dataset to obtain the maximum TM-score for each generated sequence. This dataset is constructed by MMseqs2 with 50% sequence similarity from the UniProt subset in the AlphaFold database. To obtain the maximum sequence similarity to natural protein sequences, we also retrieved each generated sequence from UniProt, UniClust30 and the BFD database with MMseqs2 and HHblits (Extended Data Fig. 3b). The UniProt and BFD databases contain more than 2.5 billion protein sequences, representing the space of currently known protein sequences in nature. For sequences with ESMFold pLDDT > 80 (N = 11,048), we show the maximum sequence similarity and maximum TM-score of all query sequences in the scatterplot (Extended Data Fig. 3c), with a mean sequence similarity value of 0.699 and a mean TM-score value of 0.864. The vast majority of generated structures have similar 3D structures to those in UniProt50, which confirms that our pretrained model comprehensively understands and represents the protein universe.

This potentially allows us to access a larger sequence in the protein manifold while we try to design certain protein structures. Figure 4b showcases a comprehensive network of the protein structural space, informed by sequences synthesized by xTrimoPGLM. Each node corresponds to an 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.

Enhanced protein sequence generation through SFT and ReST. 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[16], xTrimoPGLM-100B serves as a protein foundational model, equipped with vast knowledge from trillions of residue tokens. We applied established alignment methods like SFT[30] on select protein families and enhance this with ReST[31], based on the SFT model. We fine-tuned xTrimoPGLM-100B on datasets representing common protein structures or chemical properties. We chose five tasks from 18 benchmark protein understanding tasks (fold prediction, temperature stability, localization prediction, fluorescence prediction and fitness prediction) for fine-tuning. Specifically, we adopted two filtering strategies to obtain the SFT datasets: (1) regression tasks (fluorescence and fitness prediction), where we filtered samples whose label scores exceeded a certain threshold, then fine-tuned the models with these samples; and (2) classification tasks (fold prediction, temperature stability, localization), where we used samples from one category to fine-tune the model and generate new samples. Moreover, we further filtered the protein sequences generated by the SFT models as the ReST datasets. We applied the same filtering and fine-tuning settings during the ReST stages. We fine-tuned our model and the baseline models, such as ProGen2 and ProtGPT2, using the same causal language modeling regime (Supplementary Section 5). Comparative analysis was conducted against ProtGPT2 and ProGen2 using identical SFT protocols. To circumvent trivial results, we used the non-fine-tuned xTrimoPGLM-100B model 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[12,29]. More specifically, we used the corresponding task predictor to predict the scores of the desired class (for classification tasks) or the regression scores (for regression tasks) to validate whether the SFT models could generate sequences with the desired properties.

Our findings revealed 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 (Fig. 5). Furthermore, we implemented a one-step ReST 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 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-tuned 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, sequenced identities and their structural counterparts in the PDB, as measured by TM-score (Fig. 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.

Discussion

Although xTrimoPGLM demonstrated impressive performance, it has limitations. A major limitation of xTrimoPGLM-100B is the high computational cost associated with the model, which presents a considerable barrier to its 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 memory, such as quantization[32], kernel fusion[28] and other accelerate technologies[33–35], 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.

Moreover, we observed diminishing returns in performance with increasing model parameters. Specifically, while xT-Fold has shown impressive TM-scores and inference speeds 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 pretraining model becomes more powerful. Although the model is gradually improving in compressing OOD sequences, it already diminishes the return (Fig. 1b). We speculate that protein sequences, unlike natural language, are less smooth in semantic space and more akin to a factual nature. Currently, PLM-based methods still struggle to outperform MSA-based or retrieval-augmented approaches, especially when dealing with substantial OOD test data. To bridge this gap, leveraging more abundant data sources—such as sequences, structures and functional descriptions[36]—and expanding data modalities to include proteins, DNA and RNA[2], along with implementing compute-optimal pretraining strategies[37], could be crucial. Therefore, we advocate for proportional scaling of both data and models and emphasize the importance of exploring their efficient frontier further. Another option is that integrating MSA modules with neural network retrievers[38,39] could 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.

Similarly to most LLMs, xTrimoPGLM also experiences the issue of generating hallucinations. During the generation process, when the sampling temperature is set low, the model tends to produce 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, of the sequences we generated, approximately 20% of the sequences could not be confidently predicted by xT-Fold (pLDDT < 70), leaving us uncertain 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 PDB. To avoid or reduce the hallucination problem of LLMs, augmenting constraints during the training and sampling process can improve the efficiency of the model generation.

In summary, we explored unified understanding and generation pretraining with an extremely large-scale PLM. 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 SFT, 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.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41592-025-02636-z.

References

1. Jumper, J. et al. Highly accurate protein structure prediction with alphafold. *Nature* **596**, 583–589 (2021).
2. Abramson, J. et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature* **630**, 493–500 (2024).
3. Baek, M. et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **373**, 871–876 (2021).
4. Anfinsen, C. B. et al. *The Molecular Basis of Evolution* (John Wiley and Sons, Inc., 1959).
5. Rives, A. et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl Acad. Sci. USA* **118**, e2016239118 (2021).
6. Lin, Z. et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379**, 1123–1130 (2023).
7. Elnaggar, A. et al. ProtTrans: toward cracking the language of lifes code through self-supervised deep learning and high performance computing. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 7112–7127 (2021).
8. Madani, A. et al. Large language models generate functional protein sequences across diverse families. *Nat. Biotechnol.* **41**, 1099–1106 (2023).
9. Apweiler, R. et al. Uniprot: the universal protein knowledgebase. *Nucleic Acids Res.* **32**, 115–119 (2004).
10. Finn, R. D. et al. Pfam: the protein families database. *Nucleic Acids Res.* **42**, 222–230 (2014).
11. Steinegger, M., Mirdita, M. & Söding, J. Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nat. Methods* **16**, 603–606 (2019).
12. Nijkamp, E. et al. ProGen2: exploring the boundaries of protein language models. *Cell Syst*. https://doi.org/10.1016/j.cels.2023.10.002 (2023).
13. Verkuil, R. et al. Language models generalize beyond natural proteins. Preprint at *bioRxiv* https://doi.org/10.1101/2022.12.21.521521 (2022).
14. Bao, H. et al. Unilmv2: pseudo-masked language models for unified language model pre-training. In *Proc. 37th International Conference on Machine Learning* 642–652 (PMLR, 2020).
15. Tay, Y. et al. Ul2: unifying language learning paradigms. In *Eleventh International Conference on Learning Representations* https://openreview.net/pdf?id=6ruVLB727MC (ICLR, 2023).
16. Brown, T. et al. Language models are few-shot learners. In *34th Conference on Neural Information Processing Systems* https://papers.nips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf (NeurIPS, 2020).
17. Wei, J. et al. Finetuned language models are zero-shot learners. In *Tenth International Conference on Learning Representations* https://openreview.net/pdf?id=gEZrGCozdqR (ICLR, 2022).
18. Chung, H. W. et al. Scaling instruction-finetuned language models. *J. Mach. Learning Res.* **25**, 1–53 (2024).
19. Wu, R. et al. High-resolution de novo structure prediction from primary sequence. Preprint at *bioRxiv* https://doi.org/10.1101/2022.07.21.500999 (2022).
20. Du, Z. et al. GLM: general language model pretraining with autoregressive blank infilling. In *Proc. 60th Annual Meeting of the Association for Computational Linguistics* (Volume 1: Long Papers) 320–335 (Association for Computational Linguistics, 2022).
21. Kenton, J. D. M.-W. C. & Toutanova, L. K. Bert: pre-training of deep bidirectional transformers for language understanding. In *Proc. 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* Volume 1 (Long and Short Papers) https://aclanthology.org/N19-1423.pdf (Association for Computational Linguistics, 2019).
22. Kaplan, J. et al. Scaling laws for neural language models. Preprint at https://arxiv.org/abs/2001.08361 (2020).
23. Hoffmann, J. et al. Training compute-optimal large language models. In *Proc. 36th International Conference on Neural Information Processing Systems* https://proceedings.neurips.cc/paper_files/paper/2022/file/c1e2faff6f588870935f114ebe04a3e5-Paper-Conference.pdf (NeurIPS, 2022).
24. Hu, E. J. et al. Lora: Low-rank adaptation of large language models. In *Tenth International Conference on Learning Representations* https://openreview.net/pdf?id=nZeVKeeFYf9 (ICLR, 2022).
25. Wei, J. et al. Emergent abilities of large language models. *Trans. Mach. Learn. Res.* https://openreview.net/pdf?id=yzkSU5zdwD (2022).
26. Elnaggar, A. et al. Ankh: optimized protein language model unlocks general-purpose modelling. Preprint at https://arxiv.org/abs/2301.06568 (2023).
27. Baek, M. et al. Efficient and accurate prediction of protein structure using RoseTTAFold2. Preprint at *bioRxiv* https://doi.org/10.1101/2023.05.24.542179 (2023).
28. Dao, T. et al. FlashAttention: fast and memory-efficient exact attention with IO-awareness. In *36th Conference on Neural Information Processing Systems* https://openreview.net/pdf?id=H4DqfPSibmx (NeurIPS, 2022).
29. Ferruz, N., Schmidt, S. & Höcker, B. ProtGPT2 is a deep unsupervised language model for protein design. *Nat. Commun.* **13**, 4348 (2022).
30. Ouyang, L. et al. Training language models to follow instructions with human feedback. In *36th Conference on Neural Information Processing Systems* https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf (NeurIPS, 2022).
31. Gulcehre, C. et al. Reinforced self-training (rest) for language modeling. Preprint at https://arxiv.org/abs/2308.08998 (2023).
32. Dettmers, T., Pagnoni, A., Holtzman, A., Zettlemoyer & L. Qlora: efficient finetuning of quantized LLMs. In *37th Conference on Neural Information Processing Systems* https://openreview.net/pdf?id=OUIFPHEgJU (NeurIPS, 2024).
33. Kwon, W. et al. Efficient memory management for large language model serving with PagedAttention. In *Proc. 29th Symposium on Operating Systems Principles* 611–626 (Association for Computing Machinery, 2023).
34. Ainslie, J. et al. GQA: training generalized multi-query transformer models from multi-head checkpoints. In *Proc. 2023 Conference on Empirical Methods in Natural Language Processing* 4895–4901 (Association for Computational Linguistics, 2023).
35. Chen, C. et al. Accelerating large language model decoding with speculative sampling. Preprint at https://arxiv.org/abs/2302.01318 (2023).
36. Hayes, T. et al. Simulating 500 million years of evolution with a language model. *Science* **387**, 850–858 (2025).
37. Cheng, X. et al. Training compute-optimal protein language models. In *38th Conference on Neural Information Processing Systems* https://openreview.net/pdf?id=uCZI8gSfD4 (NeurIPS, 2025).
38. Rao, R. M. et al. MSA transformer. In *Proc. 38th International Conference on Machine Learning* 8844–8856 (PMLR, 2021).
39. Chen, B. et al. Msagpt: neural prompting protein structure prediction via MSA generative pre-training. In *38th Conference on Neural Information Processing Systems* https://proceedings.neurips.cc/paper_files/paper/2024/file/41f3347f8f47c17bbadaed584e68d8bd-Paper-Conference.pdf (2025).

Methods

Backbone framework: GLM

Current PLM frameworks are commonly categorized as either encoder-only, like ESM[5,6], or decoder-only, such as PROGEN[8,12], 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[13], suggesting the need for a unified model capable of encapsulating this diversity. To achieve this, encoder–decoder architectures like T5 (ref. 40) and noncausal decoder-only models like the GLM[20,41] are optimized through an autoregressive 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 twice the number of parameters for similar efficacy. The GLM is a transformer-based language model characterized by its unique training methodology. It uses autoregressive blank infilling while processing input text bidirectionally. 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 pretraining differentiates the GLM from causal decoder-only language models, which rely solely on unidirectional attention.

Pretraining objectives

The GLM incorporates two distinct pretraining objectives to ensure its generative capabilities: (1) span prediction, which focuses on recovering short blanks within sentences, with the blank lengths cumulatively forming a substantial portion of the input; and (2) long-text generation, which is 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 MLM strategy[21]. 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.

MLMs 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 pretraining loss is defined as given by equation (1):

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

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

GLMs for generation. The GLM objective aims at recovering the masked consecutive tokens, that is, 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 $\mathbf{x}_{\mathrm{corrupt}}$. To make sure the interactions are among corrupted spans, xTrimoPGLM randomly permutes the order of spans like the GLM, and defines the pretraining objective according to equation (2):

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

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

Unified pretraining. The two types of pretraining objectives are jointly optimized to pretrain the xTrimoPGLM model. The unified pretraining objective, which aims to maximize the likelihood of the Oracle tokens, is defined according to equation (3):

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

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

The pretraining strategy

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

MLM stage (400 billion pretrained tokens). In this initial stage, the `[MASK]` token is used for masking random tokens in the sequence, with these masked tokens constituting 15% of the total input. This stage, consuming about 400 billion tokens, is dedicated to advancing the model's representation abilities. At this stage, the α for scaling the GLM loss is set to 0, focusing solely on minimizing the MLM loss to enhance the model's understanding capabilities.

Unified training stage (600 billion pretrained 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. Here, the α is set to 1, allowing both objectives to equally contribute to the overall loss function on each training instance, with the GLM objective appearing four times more frequently than the MLM objective.

- MLM component. Leveraging the `[MASK]` token, this component focuses on enhancing the model's understanding of protein sequences.
- GLM component. This component uses 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.

Empirical analysis of unified training

This section presents an in-depth analysis of the feasibility of concurrently optimizing two distinct pretraining objectives in xTrimoPGLM. Unlike prior unified pretraining frameworks[14,15], 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 (MLM) and next-token predictions (GLM). Central to our investigation are two pivotal questions: (1) Objective compatibility. Does the in-place token prediction objective need to 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; (2) 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.

Pretraining 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. Pretraining is conducted on the UniRef50 database[42]. We use 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 pretraining (corresponding to the total computational cost), rather than those actually trained (that is, 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]`) were 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 100-billion-token and 200-billion-token consumption milestones, facilitated by constant model parameters and architecture, requiring only adjustments in the attention mask. Validation losses indicate that despite their differing natures, both MLM and GLM objectives optimize simultaneously (Extended Data Fig. 7a,b).

Furthermore, we investigate the impact of pretraining objectives on convergence speed. Models pretrained on one objective adapt to another, training over an additional 50 billion tokens. Comparisons include MLM-adapted GLM versus GLM trained from scratch and GLM-adapted MLM versus MLM trained from scratch. Our results show markedly faster convergence in adapted models compared to those trained from scratch (Extended Data Fig. 7c,d). The MLM-adapted GLM matches the loss of the GLM from-scratch model with a 2.2× speedup (110 billion tokens). Similarly, the GLM-adapted MLM achieves a 2× speedup (100 billion tokens).

These findings suggest that modeling protein data distribution is not limited to specific training patterns. This bridges the gap between autoencoding PLMs (for example, ESM[6]) and autoregressive PLMs (for example, ProGen2)[12], underscoring the effectiveness of the xTrimoPGLM training pipeline.

The training stability of unified training

Training stability is a critical factor for the successful training of LLMs at the 100-billion scale[16,41,43]. 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[41], 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 (10-billion scale). Incorporating a fixed ratio of GLM loss into pretraining can trigger these collapses, even with a minimal 1% GLM loss ratio (Extended Data Fig. 6). 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 $K = 1{,}000$ and the learning rate to 1×10^{-7}.

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

Pretraining configurations

Here we introduce the implementation details of pretraining the xTrimoPGLM-100B model. Since the xTrimoPGLM-100B borrows the idea from the GLM-130B[41] framework, we only emphasize the specific hyperparameter of xTrimoPGLM-100B. For more discussion and design choices, please refer to GLM-130B[41].

xTrimoPGLM-100B is trained on a cluster of 96 DGX-A100 GPU (8 × 40G) servers in FP16 precision from 18 January 2023 to 30 June 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[44], 8-way pipeline parallelism[45] and 24-way data parallelism based on DeepSpeed[46]. The model owns 72 transformer layers, 80 attention heads and 10,240 embedding dims with 31,744 feed-forward embedding dims using GeGLU[47]. We adopt the Post-LN initialized with the DeepNorm[48]. We follow the mixed-precision strategy (Apex O2), that is, FP16 for forwards and backwards and FP32 for optimizer states and master weights, to reduce the GPU memory usage and improve training efficiency. We also adopt the embedding layer gradient shrink (EGS) strategy[41] with $\alpha = 0.1$ to stabilize the xTrimoPGLM-100B training. We warm up the batch size from 240 to 4,224 over the first 2.5% of samples. We use AdamW[49] 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% of samples, then decay it by a 10 × cosine schedule to the minimum learning rate of 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 pretraining.). To adapt the model to the different lengths of proteins in the downstream tasks, we adopt the mix-length pretraining 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's 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[6,7]. Except for the basic amino acid types, we add the special tokens `[MASK]`, `[sMASK]` and `[gMASK]` for model prediction. We also add the special tokens `<sop>`, `<eop>` and `<eos>` for sequence separation. Please refer to Supplementary Table 9 for the full configurations.

Pretraining datasets

The pretraining dataset of xTrimoPGLM-100B is curated from two extensive data repositories: UniRef90 (https://www.uniprot.org/help/downloads/; UniRef90 version preceding December 2022) and

ColabFoldDB[50] (https://colabfold.mmseqs.com/). The initial contributions from UniRef90 and ColabFoldDB encompass approximately 153 million and 950 million (210 million representatives plus 740 million members) entries, respectively.

UniRef, a cluster from UniProt, is broadly acknowledged as a high-quality protein dataset often utilized in pretraining PLMs[6,7]. UniRef90 clusters are generated from the UniRef100 seed sequences with a 90% sequence identity threshold using the MMseqs2 (https://github.com/soedinglab/MMseqs2/) algorithm. ColabFoldDB is established through an amalgamation of various metagenomic databases including BFD (https://bfd.mmseqs.com/), MGnify[51], SMAG (eukaryotes)[52], MetaEuk (eukaryotes)[53], TOPAZ (eukaryotes)[54], MGV (DNA viruses)[55], GPD (bacteriophages)[56] and an updated version of the MetaClust[57] 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 introduce 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 pretraining dataset has been expanded by incorporating protein sequences sourced from ColabFoldDB in addition to those from the UniRef90 dataset.

Extended Data Fig. 8 illustrates the composition of the dataset used for pretraining the model. It 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 before 1 January 2023 are incorporated into the training set. Given its robustness and reliability, our training process also substantially prioritizes this dataset.

Training set. The complete dataset in ColabFoldDB initially contained approximately 950 million sequences. After initial deduplication and short-length filtering, which removed about 150 million duplicate sequences, and further refinement by cross-referencing and deduplicating with UniRef90, we narrowed down the dataset to 780 million unique sequences, ensuring diversity and representativeness for effective training. We conducted a composition analysis of each remaining sequence, excluding any that exhibited 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 200 million sequences. Finally, the pretrained dataset comprises approximately 939 million protein sequences with 200 billion tokens. Specifically, the UniRef90 dataset contains around 156 million protein sequences with 53 billion residue tokens. The ColabFoldDB cluster includes about 208 million protein sequences with 38B tokens, and the ColabFoldDB member contains 575 million sequences with 103 billion tokens. 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 1 January 2023 and 30 March 2023 were utilized as the validation datasets. The 18-million sequence increment was applied as a query to scrutinize the target database (that is, UniRef50 and the training dataset), and sequences over 90% or 0.5% similarity were 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 sequences after filtering were used as the validation set.

Pretraining data distribution. The bar charts in Extended Data Fig. 9 represent the distribution of sequence lengths within the UniRef90 and ColabFoldDB datasets. In both datasets, sequences in the '100–400' sequence length category predominate, followed by the '50–100' category. The '0–50' and '400+' categories contain substantially fewer sequences. Note the comparison between the distribution of UniRef90 and ColabFoldDB, indicating the variety of sequence lengths used for model training.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

All data used in this study are publicly available and the usages are illustrated in the Methods. The pretraining dataset of xTrimoPGLM-100B was curated from two extensive data repositories: UniRef90 (https://www.uniprot.org/help/downloads/; UniRef90 version preceding December 2022) and ColabFoldDB (https://colabfold.mmseqs.com/). Eighteen downstream task datasets are available online (https://huggingface.co/proteinglm/). All structure prediction datasets are from the AlphaFold database (https://alphafold.ebi.ac.uk/download/) and the PDB (https://www.rcsb.org/downloads/; May 2020 release).

Code availability

Trained weights for the xTrimoPGLM model, and downstream datasets, are available at https://huggingface.co/proteinglm/.
Model training used DeepSpeed v0.6.1 (https://github.com/microsoft/DeepSpeed/). Data analysis used Python v3.8 (https://www.python.org/), NumPy v1.16.4 (https://github.com/numpy/numpy/), SciPy v1.2.1 (https://www.scipy.org/), Seaborn v0.11.1 (https://github.com/mwaskom/seaborn/), Matplotlib v3.3.4 (https://github.com/matplotlib/matplotlib/) and Pandas v1.1.5 (https://github.com/pandas-dev/pandas/). TM-align v20190822 (https://zhanglab.dcmb.med.umich.edu/TM-align/) was used for computing TM-scores. Structure visualizations were created in Pymol v2.3.0 (https://github.com/schrodinger/pymol-open-source/). Protein 3D structures were predicted using AlphaFold2 with the official implementations (https://github.com/deepmind/alphafold/).

References

40. Raffel, C. et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learning Res.* **21**, 1–67 (2020).
41. Zeng, A. et al. Glm-130b: An open bilingual pre-trained model. In *Eleventh International Conference on Learning Representations* https://openreview.net/pdf?id=-Aw0rrrPUF (ICLR, 2022).
42. Suzek, B. E. et al. UniRef clusters: a comprehensiÿve and scalable alternative for improving sequence similarity searches. *Bioinformatics* **31**, 926–932 (2015).
43. Chowdhery, A. et al. PaLM: scaling language modeling with pathways. Preprint at https://arxiv.org/abs/2204.02311 (2022).
44. Shoeybi, M. et al. Megatron-LM: training multi-billion parameter language models using model parallelism. Preprint at https://arxiv.org/abs/1909.08053 (2019).
45. Narayanan, D. et al. Memory-efficient pipeline-parallel DNN training. In *Proc. 38th International Conference on Machine Learning* 7937–7947 (PMLR, 2021).

46. Rasley, J., Rajbhandari, S., Ruwase, O. & He, Y. DeepSpeed: system optimizations enable training deep learning models with over 100 billion parameters. In *Proc. 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* 3505–3506 (Association for Computing Machinery, 2020).
47. Shazeer, N. Glu variants improve transformer. Preprint at https://arxiv.org/abs/2002.05202 (2020).
48. Wang, H. et al. DeepNet: scaling transformers to 1,000 layers. *IEEE Trans. Pattern Anal. Mach. Intell.* **46**, 6761–6774 (2024).
49. Loshchilov, I. & Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations* https://openreview.net/pdf?id=Bkg6RiCqY7 (ICLR, 2019).
50. Mirdita, M. et al. ColabFold: making protein folding accessible to all. *Nat. Methods* **19**, 679–682 (2022).
51. Mitchell, A. L. et al. MGnify: the microbiome analysis resource in 2020. *Nucleic Acids Res.* **48**, D570–D578 (2020).
52. Delmont, T. O. et al. Functional repertoire convergence of distantly related eukaryotic plankton lineages abundant in the sunlit ocean. *Cell Genom.* **28**, 100123 (2022).
53. Levy Karin, E., Mirdita, M. & Söding, J. Metaeuk-sensitive, high-throughput gene discovery, and annotation for large-scale eukaryotic metagenomics. *Microbiome* **8**, 48 (2020).
54. Alexander, H. et al. Eukaryotic genomes from a global metagenomic data set illuminate trophic modes and biogeography of ocean plankton. *mBio* **14**, e0167623 (2023).
55. Nayfach, S. et al. Metagenomic compendium of 189,680 DNA viruses from the human gut microbiome. *Nat. Microbiol.* **6**, 960–970 (2021).
56. Camarillo-Guerrero, L. F. et al. Massive expansion of human gut bacteriophage diversity. *Cell* **184**, 1098–1109 (2021).
57. Steinegger, M. & Söding, J. Clustering huge protein sequence sets in linear time. *Nat. Commun.* **9**, 2542 (2018).

Acknowledgements

We would like to express our deepest gratitude to all those who provided us with the possibility to complete this work. We are grateful to colleagues from THU for their consecutive suggestions about model training and inference, and to team members from BioMap for their help in data and biological guidance. Our heartfelt thanks are also extended to BioMap colleagues for their advice on antibody evaluations. Additionally, we would like to thank BioMap team members for their suggestion of infrastructure and operations support. We extend our gratitude to Zhipu AI for providing the computing resources. This work was supported by the National Science Fund for Distinguished Young Scholars (62425601 and 62276148), the New Cornerstone Science Foundation through the XPLORER PRIZE and a research fund from Zhipu AI.

Author contributions

B.C. conceived the methods, implemented and trained the model, investigated the scaling law and drafted the manuscript. X.C. led the project, managed workflows, prepared pretrained datasets, trained scaling law and xT-Fold and refined the manuscript. P.L. and Z.B. analyzed the generated protein sequences. Y.G., J.G., S.L. and B.W. evaluated the model on the protein understanding benchmark. X.T. and X.Z. helped to develop xT-Fold. A.Z. and C.L. helped implement the model training framework. L.S. and J.T. played a crucial role in integrating all resources, with L.S serving as an intermittent advisor throughout. All the authors read and approved the final manuscript. Team structure：B.C, Y.G, Z.B and B.W were interns at BioMap Research, California, USA; X.C was the Project Leader at BioMap Research, California, USA; J.T. and L.S. are the corresponding authors.

Competing interests

The authors declare no competing interests.

Additional information

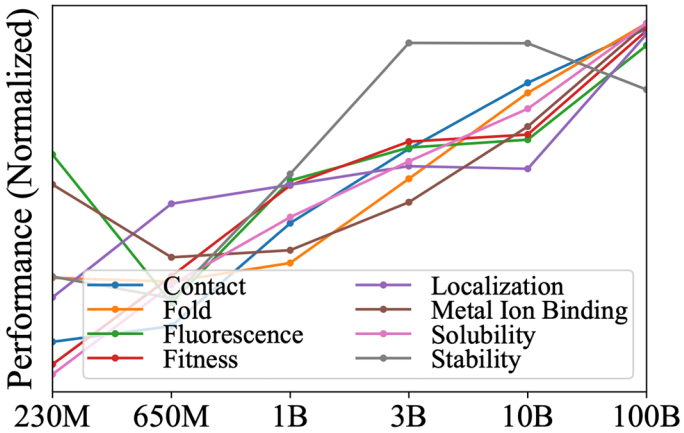
Extended data is available for this paper at https://doi.org/10.1038/s41592-025-02636-z.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41592-025-02636-z.

Correspondence and requests for materials should be addressed to Bo Chen, Xingyi Cheng, Jie Tang or Le Song.

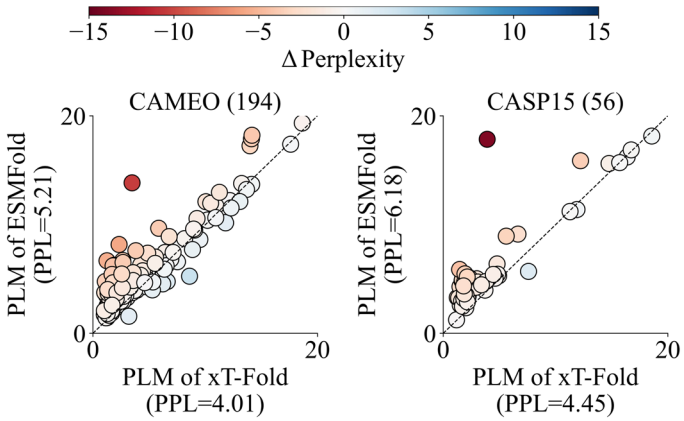
Peer review information *Nature Methods* thanks Noelia Capapey and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Primary Handling Editor: Arunima Singh, in collaboration with the *Nature Methods* team. Peer reviewer reports are available.

https://doi.org/10.1038/s41592-025-02636-z



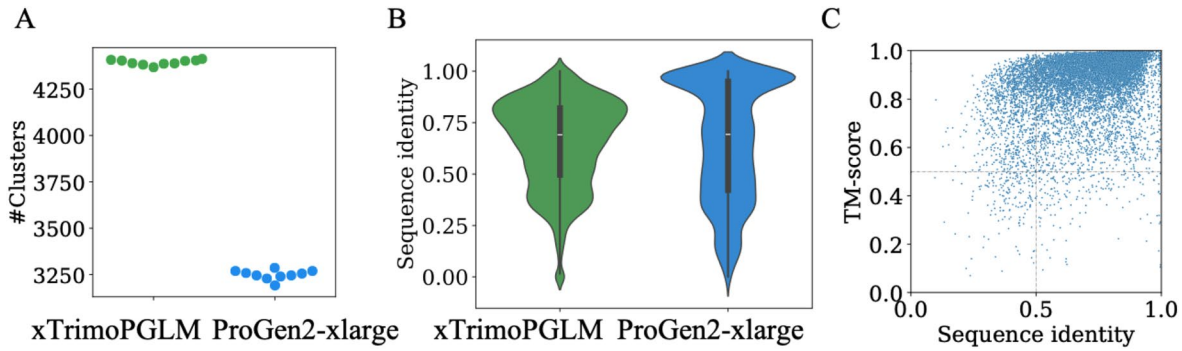
Extended Data Fig. 1 | Comparison of xTrimoPGLM family models with varying sizes and training FLOPs. Including 230M (FLOPs: 3×10^{20}), 650M (FLOPs: 3.8×10^{21}), 1B (FLOPs: 1.2×10^{21}), 3B (FLOPs: 1.8×10^{22}), 10B (FLOPs: 1.8×10^{22}), and 100B (FLOPs: 6.2×10^{23}). Models are evaluated across eight downstream tasks using the Linear Probing fine-tuning approach. Results indicate that most tasks exhibit a positive correlation between performance and both training FLOPs and model size. Unnormalized performance metrics are provided in Supplementary Table 6.

https://doi.org/10.1038/s41592-025-02636-z



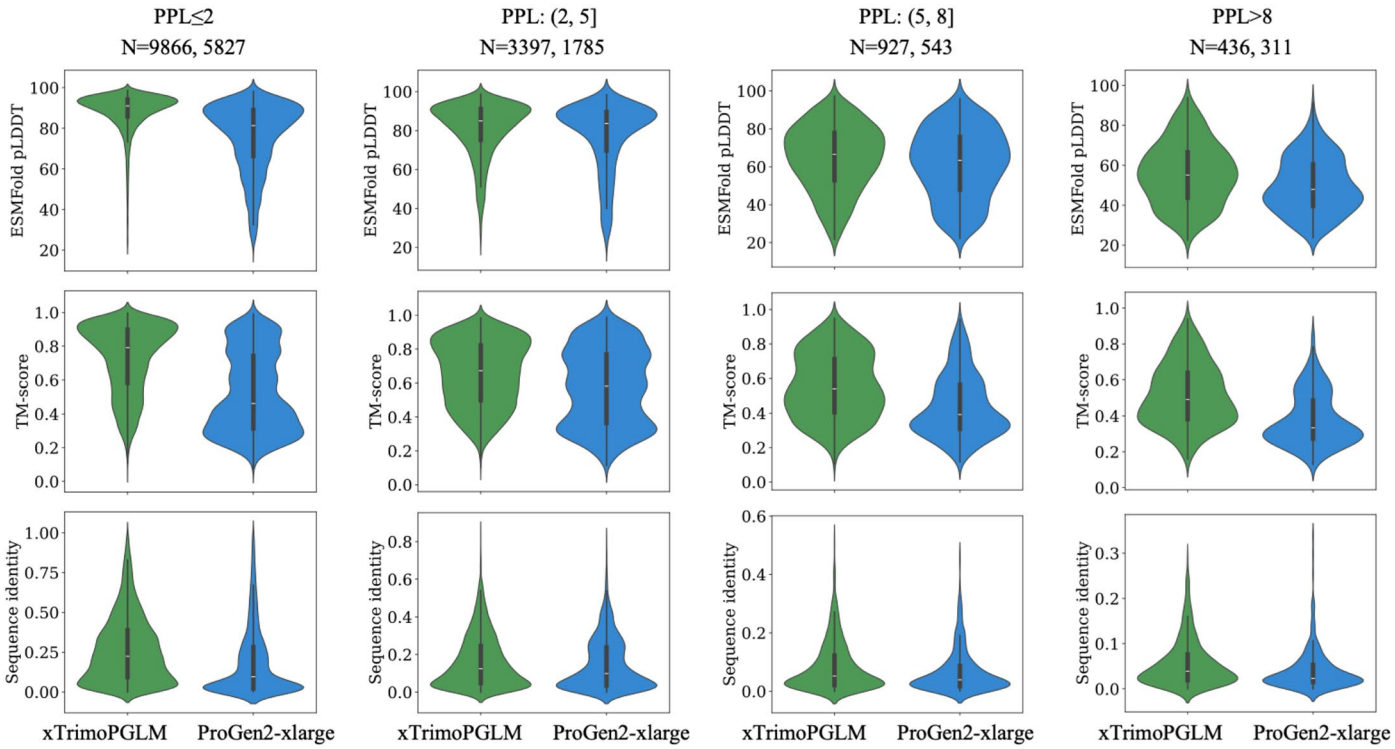
Extended Data Fig. 2 | Perplexity delta comparison on CAMEO and CASP15. Scatter plots show delta perplexity for CAMEO and CASP15. The perplexities (PPL) are from the PLM modules of xT-Fold and ESMFold. Points represent proteins, with color gradients indicating perplexity delta by the x-axis PPL minus the y-axis PPL.

https://doi.org/10.1038/s41592-025-02636-z



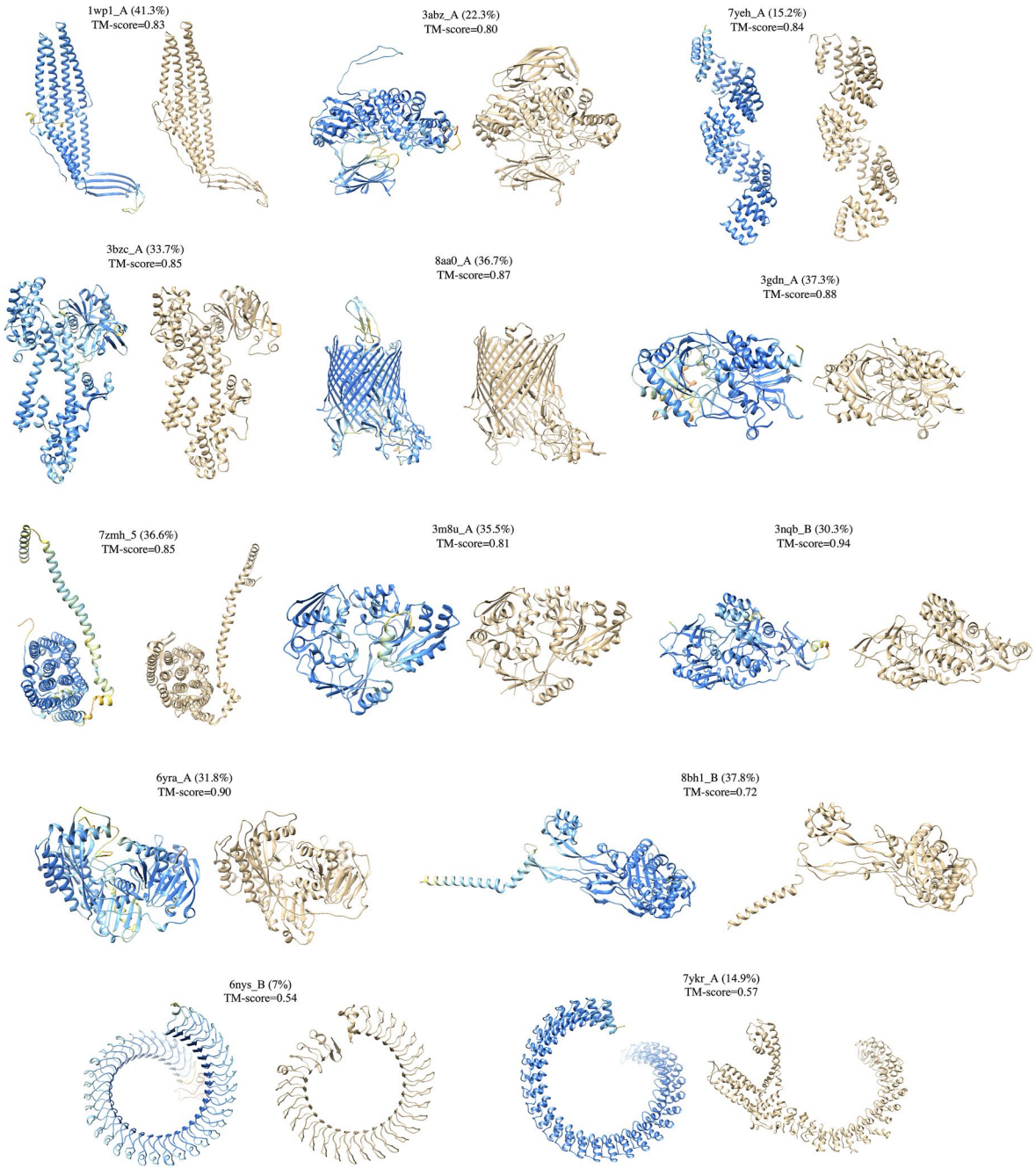
Extended Data Fig. 3 | Sequence identity to natural sequence space and diversity analysis. (**A**: 5,000 sequences are randomly selected from two sequence set and are clustered using MMseqs2 (–min-seq-id 0.5 -c 0.8). The Y-axis show the number of clusters of 10 repetitions. **B**: Maximum identity of generated sequences (N = 14,626 and 8,466) to UniClust30, UniProt, and BFD databases. **C**: Scatter plot of maximum sequence identity to natural sequence space and maximum structure similarity to AlphaFold database (UniProt50 subset). The bars in the violin plot indicate the median and interquartile range (IQR) for each group with whiskers extending 1.5 × IQR past the upper and lower quartiles.

https://doi.org/10.1038/s41592-025-02636-z



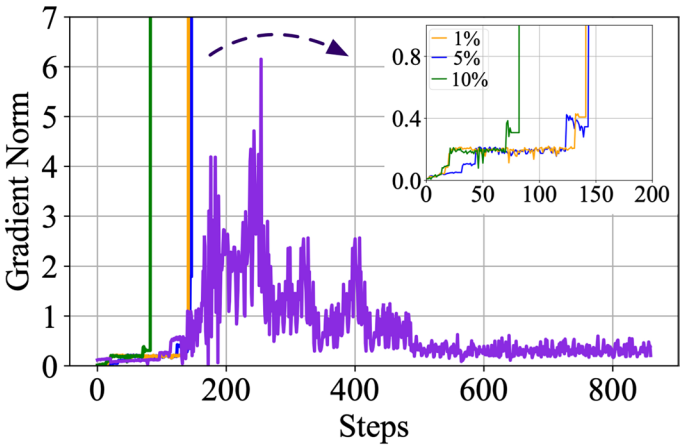
Extended Data Fig. 4 | Comparison of sequences generated by xTrimoPGLM and PROGEN2-xlarge. At different PPL ranges. ESMFold predicted confidence (pLDDT scores), the resemblance to proteins cataloged in the Protein Data Bank (TM-score and sequence identity) of generated sequences by xTrimoPGLM (green) and PROGEN2-xlarge (blue). The sequences are divided into four groups according to the perplexity values (< 2, 2-5, 5−8 and > 8). The number of sequences of both models in each group are shown in the upper of the figure. The bars in the violin plot indicate the median and interquartile range (IQR) for each group with whiskers extending 1.5 × IQR past the upper and lower quartiles.

https://doi.org/10.1038/s41592-025-02636-z



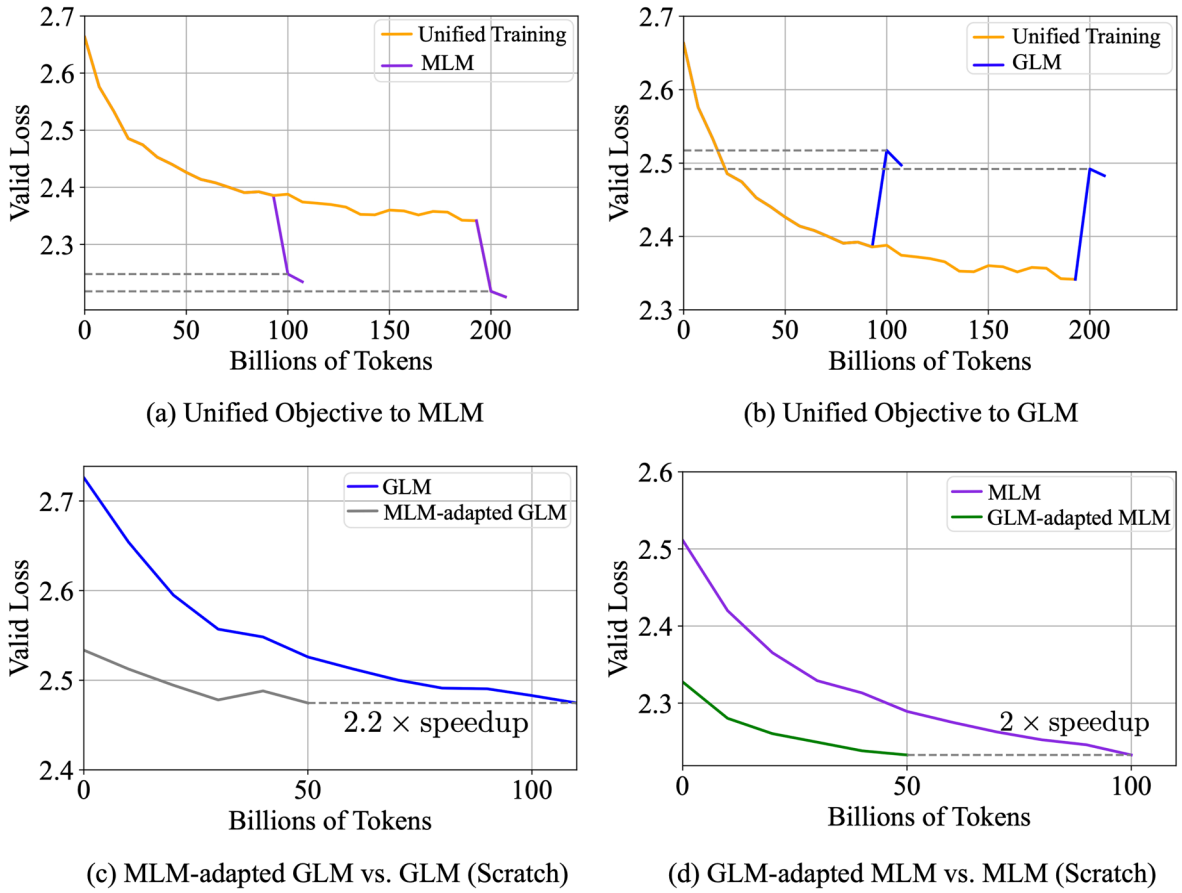
Extended Data Fig. 5 | Generated protein sequences cases. More protein sequences visualizations generated by xTrimoPGLM-100B. (Left: Generated by xTrimoPGLM-100B. Right: Natural Proteins).

https://doi.org/10.1038/s41592-025-02636-z



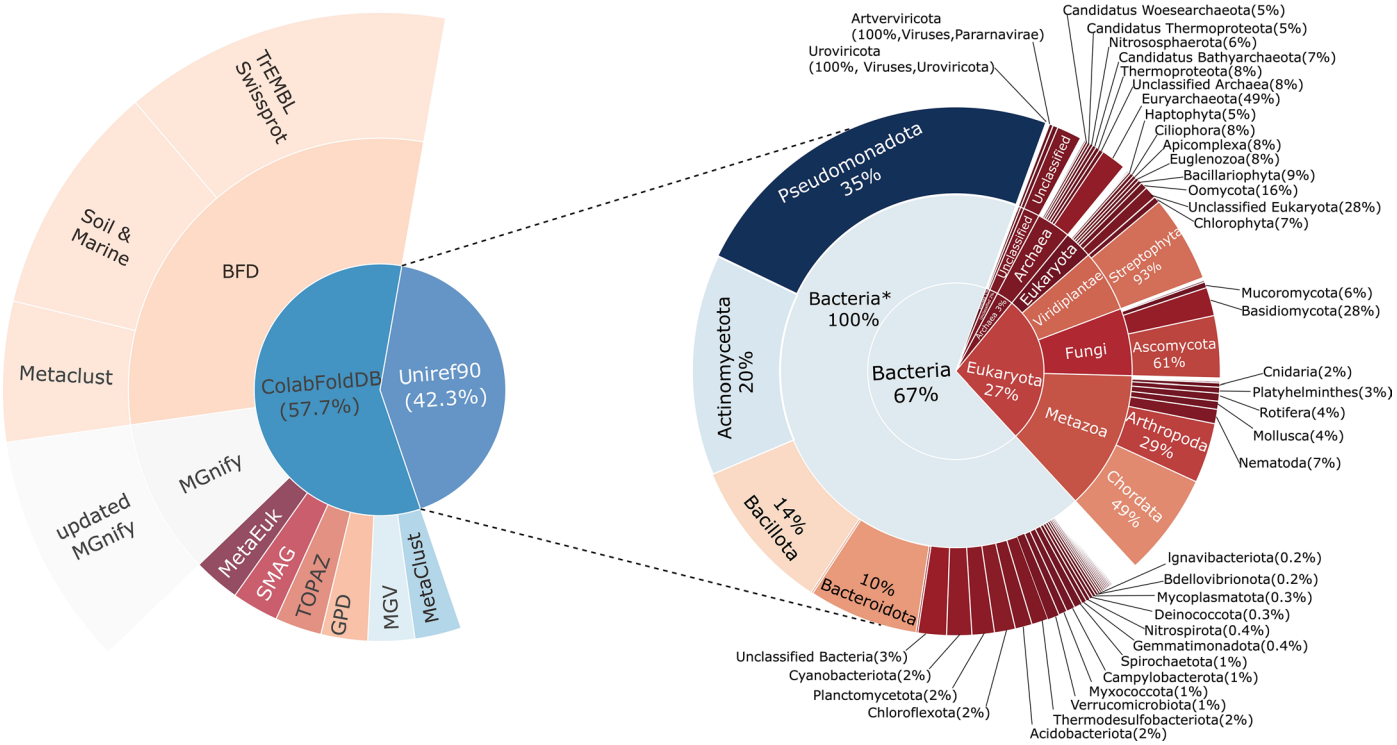
Extended Data Fig. 6 | Trials on different strategies for transition from Stage-1 to Stage-2. Compared to other strategies, the Smooth Transition Strategy demonstrates greater success in maintaining stability throughout the transition stage.

https://doi.org/10.1038/s41592-025-02636-z



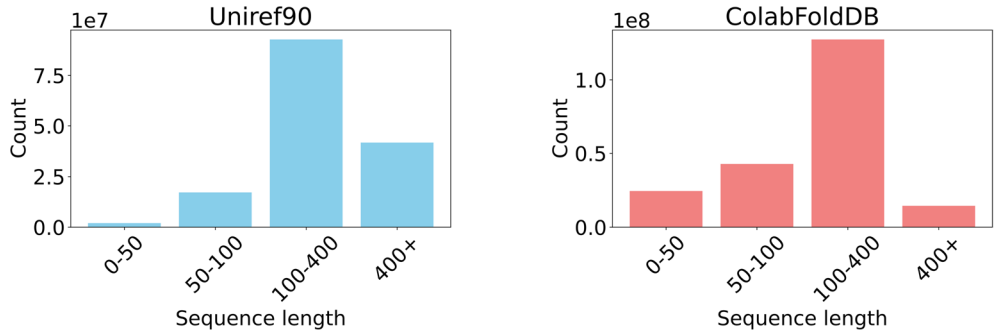
Extended Data Fig. 7 | The empirical analysis of unified training. (**a**)(**b**) The MLM and GLM objectives are optimized simultaneously. (**c**)(**d**) Adapting the model from the pre-trained one substantially accelerates convergence compared to that trained from scratch.

https://doi.org/10.1038/s41592-025-02636-z



Extended Data Fig. 8 | The pre-training dataset. 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.

https://doi.org/10.1038/s41592-025-02636-z



Extended Data Fig. 9 | Pre-training data distribution. The distribution of sequence lengths within the Uniref90 and ColabFoldDB datasets.

nature portfolio

Corresponding author(s): Bo Chen, Xingyi Cheng, Jie Tang, Le Song

Last updated by author(s): Nov 25, 2024

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
☐	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☒	☐	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐	A description of all covariates tested
☒	☐	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☐	☒	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	We did not use any software for data collection. All the data used in this project is directly downloaded from open sources.
Data analysis	Trained weight for the \smodel model, and downstream datasets are available at \url{https://huggingface.co/proteinglm}. Model training used DeepSpeed v0.6.1~\url{https://github.com/microsoft/DeepSpeed}. Data analysis used Python v3.8 (https://www.python.org/), NumPy v1.16.4 (https://github.com/numpy/numpy), SciPy v1.2.1 (https:// www.scipy.org/), seaborn v0.11.1 (https://github.com/mwaskom/seaborn), Matplotlib v3.3.4 (https://github.com/matplotlib/matplotlib), pandas v1.1.5 (https://github.com/pandas-dev/pandas), TM-align v20190822 (https://zhanglab.dcmb.med.umich.edu/TM-align/) was used for computing TM-scores. Structure visualizations were created in Pymol v2.3.0 (https://github.com/schrodinger/pymol-open-source).

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

All data used in this study are publicly available and the usages are illustrated in our methods. 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 (https://colabfold.mmseqs.com). 18 downstream task datasets are all available online (https://huggingface.co/proteinglm). All structure prediction datasets are from AlphaFold Database (https://alphafold.ebi.ac.uk/download PDB database (https://www.rcsb.org/downloads) that released date is less than May 2020).

Human research participants

Policy information about studies involving human research participants and Sex and Gender in Research.

Reporting on sex and gender	N/A
Population characteristics	N/A
Recruitment	N/A
Ethics oversight	N/A

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	We have clearly described the number of samples at each experimental stage in the corresponding sections of the main manuscript.
Data exclusions	None
Replication	The Source code for the xTrimoPGLM model and inference script on downstream tasks are available in the supplementary zip files to ensure the reproducibility of the method. Moreover, the model weight are avaliable at https://huggingface.co/proteinglm.
Randomization	Not applicable, we are not making a comparison between two groups.
Blinding	Not applicable, we are not making a comparison between two groups.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging