ProFam: an open protein family language model for functional protein design

Jude Wells[*]
University College London

Alex Hawkins Hooker[*]
University College London

Micha Livne
NVIDIA

Weining Lin
University College London

David Miller
University College London

Christian Dallago
NVIDIA & Duke University

Nicola Bordin
University College London

Brooks Paige
University College London

Christine Orengo
University College London

Burkhard Rost
Technical University of Munich

Michael Heinzinger
Technical University of Munich

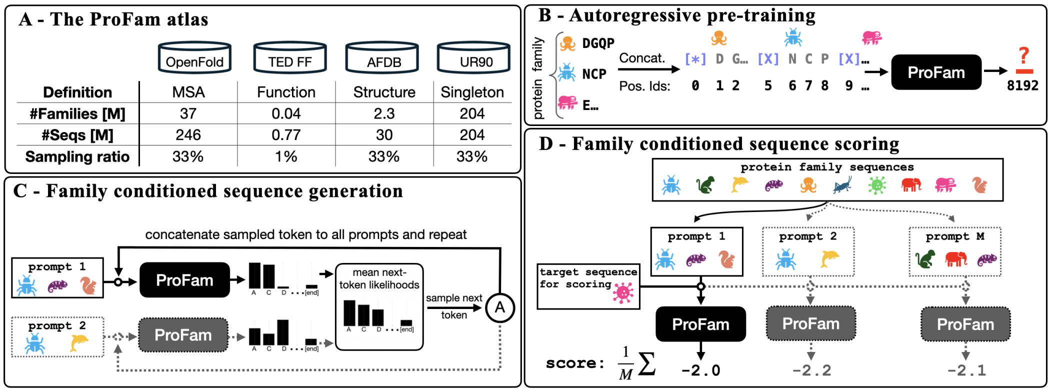
Abstract

Protein language models have become essential tools for engineering novel functional proteins. Within this domain, family-conditioned models use homologous sequences to steer protein design and enhance zero-shot fitness prediction. To provide an open foundation for this modelling strategy, we introduce *ProFam-1*, a 251M-parameter autoregressive protein family language model (pfLM) that conditions on sets of homologous sequences to guide sequence scoring and generation. *ProFam-1* achieves Spearman correlations of 0.47 for substitutions, and 0.48 for indels in ProteinGym zero-shot fitness prediction, competitive with state-of-the-art models. For homology-guided generation, *ProFam-1* generates diverse sequences with predicted structural similarity, while preserving residue conservation and covariance patterns. All of ProFam's training and inference pipelines, including a curated, large-scale training dataset *ProFam-atlas*, are released fully open source, lowering the barrier to future method development.

1 Introduction

Protein language models (pLMs) distill statistical patterns from vast sequence databases to predict protein properties [1–3], score variants [4, 5], and guide engineering [6, 7]. Dominant pre-training strategies include masked language models (MLMs) [1–3], sequence diffusion models [7–9], and autoregressive pLMs [10–12]. While MLM pLM likelihoods excel at ranking point mutations, they are ill-suited for scoring the relative fitness of insertions, deletions, and distant sequence variants. Conversely, autoregressive models offer a principled framework by decomposing the joint likelihood into a series of conditional probabilities, providing a likelihood score that is efficiently computed and that naturally accommodates insertions and deletions. Moreover, sampling from a learned autoregressively decomposed probability distribution supports *de novo* generation without specifying sequence length or alignment. However, unconditional autoregressive generation has limited practical utility, as it samples from the entire protein sequence space rather than targeting specific protein families or functions [13].

[*] Equal contribution



	OpenFold	TED FF	AFDB	UR90
Definition	MSA	Function	Structure	Singleton
#Families [M]	37	0.04	2.3	204
#Seqs [M]	246	0.77	30	204
Sampling ratio	33%	1%	33%	33%

Figure 1: **The ProFam-atlas & ProFam-1 Training and Inference.** The ProFam-atlas (Panel A) consists of protein families derived from the AlphaFold-Database (AFDB), TED, or OpenFold and individual proteins from UniRef90. ProFam-1 (Panel B) is autoregressively pre-trained on protein families by sampling at different ratios from the data sources defined in the ProFam-atlas. For each sample, unaligned protein sequences within one family are concatenated using special separator tokens (depicted here as [X]) up to a maximum of 8192 tokens/amino acids per family. During inference, ProFam-1 can leverage evolutionary information from homologs to either guide generation towards proteins with related function and structure (Panel C) or score variants (Panel D). Instead of expanding the model's input to ultra-long context, we show that both, scoring and generation, can be improved by ensembling large families into multiple prompts (indicated by dashed lines and gray coloring in panels C and D), each processed separately, and we average either the next-token probabilities (generation) or the final scores (variant scoring).

Early strategies for steering generative models relied on explicit functional labels, such as enzyme commission numbers [14], or fine-tuning on specific protein families [15]. Recent approaches instead condition directly on evolutionary context provided by sets of homologous sequences. This paradigm includes MSA-based MLMs [16, 17] and generative models [8, 18, 19] as well as alignment-free autoregressive models trained on concatenated sequences [20–22]. Unlike functional tags, conditioning on homologs guides generation via evolutionary constraints without requiring rigid functional priors. For scoring, the family context in the prompt provides evolutionary information that improves fitness prediction [17, 20, 22]. For sequence generation, family conditioning directs sampling toward a design goal via in-context exemplars, without the need for any additional model fine-tuning.

Here, we introduce ProFam, a suite of data, weights, and code for the training and inference of autoregressive protein family language models (pfLMs). First, we introduce a curated, large-scale and openly accessible training corpus called *ProFam-atlas*, incorporating single and multi-domain protein sets derived from sequence-, structure- and function-level relationships. To exploit this dataset, we introduce a 251M parameter autoregressive Transformer model, *ProFam-1*, trained on ProFam-atlas. We provide open-source training and inference code together with model weights.

In terms of capabilities, ProFam-1 is comparable to other leading sequence-based models, achieving performance competitive with state-of-the-art sequence-based methods on ProteinGym. Motivated by use-cases in evolution-based functional protein design, we also demonstrate ProFam's sequence generation capacities via a series of *in silico* evaluations. To maximise the signal available to the model via the evolutionary context in both settings, we experiment with sequence prompt ensembling schemes, finding that they improve both the diversity of the model's generations and its ability to follow evolutionary constraints.

2 Methods

We trained ProFam-1, a 251M-parameter autoregressive transformer, on the ProFam-atlas, a dataset of ≈40 million protein families aggregated from FoldSeek AFDB clusters [23], OpenProteinSet MSAs [24], TED functional domains [25], and UniRef90 sequences [26]. The model is trained with next-token-prediction on unaligned, concatenated family sequences up to a context length of 8192 tokens. For inference, we introduce an ensemble strategy that averages predictions across multiple subsampled prompts from the same family (Appendix A.5). We evaluated model performance through comprehensive in-silico benchmarks: capturing first- and higher-order sequence statistics on Enzyme Commission (EC) families, generating structurally consistent sequences for held-out FoldSeek clusters, and zero-shot variant fitness prediction on ProteinGym [27]. Full details on dataset construction, model architecture, training, and evaluation are provided in Appendix A.2. Throughout these benchmarks, we extensively compare ProFam-1 against PoET [20], an autoregressive pfLM trained on sets of homologous sequences.

3 Results

Our evaluation shows that ProFam-1 generates sequences that successfully capture evolutionary statistics of the family sequences in the prompt, even when conditioning on only a single protein sequence. Comparing ProFam-1 synthetic MSAs with the natural family, we see respectable correlations on position-specific conservation (Fig. 2a) and residue covariance (Fig. 6a, 6b). When combined with our ensemble inference strategy, the model achieves competitive zero-shot fitness prediction on ProteinGym (Fig. 3) and generates diverse sequences that retain the structural characteristics of their target families (Fig. 4). We release both the trained model and the ProFam-atlas dataset to the community.

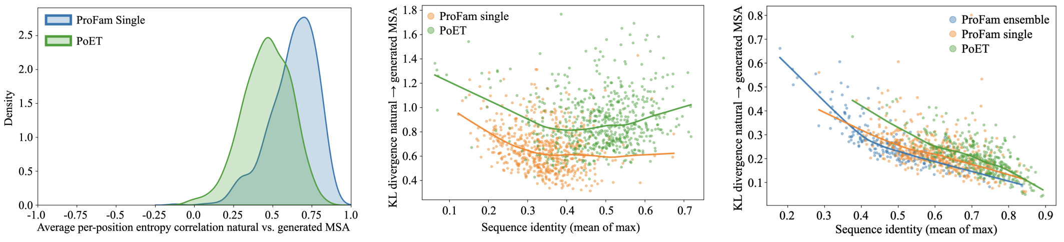
3.1 ProFam-1 captures family statistics.

We evaluated the performance of ProFam-1 relative to PoET in capturing key family statistics in EC families. Under single-sequence conditioning, ProFam-1's synthetic MSAs achieve higher conservation correlation and lower KL divergence (natural $\rightarrow$ synthetic MSA) in position-specific amino acid distributions (Figure 2a–b) compared to PoET. With multi-sequence conditioning, ProFam-1 generates a notably higher proportion of sequences within an acceptable length range (95.1% vs. PoET's 83.8%; Fig. 6c) and exhibits lower KL divergence when mean sequence identity is less than 80 (Fig. 2c). As expected, conditioning on multiple sequences generally reduces KL divergence compared to single sequence conditioning (Fig. 2c vs. 2b). We also assessed ProFam-1's ensemble mode (Appendix A.5). Ensemble generation yields sequences with lower average pairwise identity (PID) to natural homologs and typically achieves lower KL divergence at comparable PID levels (Fig. 2c). This increased diversity, however, slightly reduces the proportion of sequences within the acceptable length range (Fig. 6c).

Moving beyond single-site statistics, we evaluated the extent to which ProFam-1 captures higher-order dependencies as quantified by the Pearson Correlation Coefficient (PCC) of covariances derived from natural and synthetic MSAs. For this, we used the subset of 24 deep EC families, described in Appendix A.8, generated 1200 proteins conditioning on a single prompt per family and computed covariances as described in A.6.3. By focusing only on positions with sufficient support, we avoided introducing noise from positions that are not well supported in the natural MSA. Following this analysis, we see that both PoET and ProFam-1, learn higher-order correlations as indicated by a PCC of 0.86 and 0.88, respectively (Fig. 6b).

3.2 Improved Zero-Shot Fitness Prediction Using Test-Time-Scaling

Having established that ProFam-1 captures lower- and higher-order sequence statistics, we benchmarked to what extent its output probabilities capture protein fitness. Using deep-mutational scanning (DMS) data from ProteinGym [27], we observe two findings. First, prompt ensembling A.5 transforms ProFam from an average predictor (Spearman 0.42) into one that is competitive with state-of-the-art (Spearman 0.47) (Fig. 3): aggregating likelihoods across diverse prompts steadily raises Spearman correlation, with larger ensembles closing much of the gap to stronger zero-shot methods (Fig. 3, panel C). Second, fitness performance is not monotonically increasing with likelihood. Across assays, pushing the average variant log-likelihood too high often reduces Spearman; performance peaks in a mid-log-likelihood regime (roughly -1.8 to -1.1, often near -1.3). When using few prompts/ensembles, tuning the individual prompt(s) to target this likelihood range improves correlation; with larger ensembles, the benefit diminishes as ensemble diversity dominates (fig. 3C).



(a) (b) (c)

Figure 2: **ProFam-1 captures sequences statistics.** We tested how well ProFam-1 or PoET capture per-position conservation correlation (2a) and KL divergence of position-specific amino-acid distributions (natural → synthetic) under single-sequence EC prompting (2b); using the same set, we show how multi-sequence prompting changes KL divergence (2c). Additional evaluations on sequence length and covariance are shown in Appendix Figure 6.

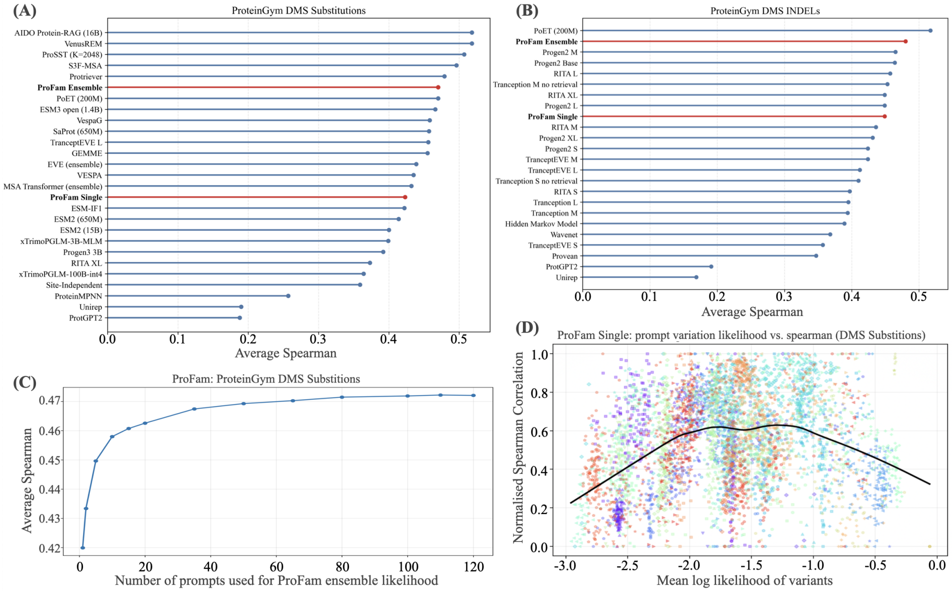
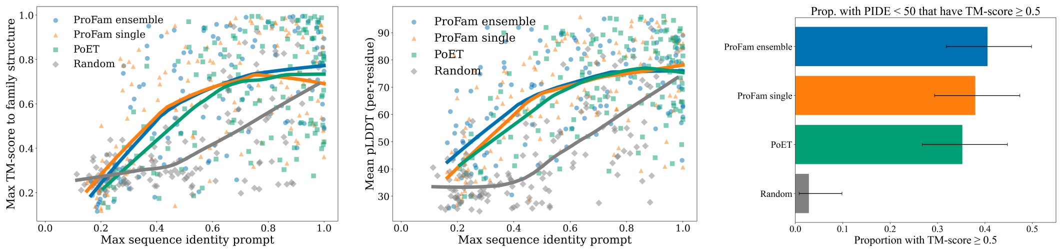


Figure 3: **Test-Time-Scaling Improves ProteinGym Zero-shot Performance**. Panel A and B show the average Spearman correlation between model predictions and experimental fitness scores across 217 and 65 deep mutational scanning experiments. Panel A shows substitutions (all assessed variants must be the same sequence length as the wild-type), while panel B shows performance for insertions and deletions. Panel C shows how ProFam's performance increases with additional prompts used for ensembling the likelihood score. Panel D plots results for 50 randomly selected ProteinGym DMS assays (no indels). Each assay is represented with a color-marker combination. For each assay, we show up to 150 results from different prompts (different context sequences) with the average log likelihood of the variants on the x-axis and the normalised Spearman correlation between the variant likelihood and the experimentally measured fitness score on the y-axis. We normalise so that the Spearman score has a minimum value of 0 and maximum value of 1 across all prompts. We see that the 'best' prompts are those which yield variant log-likelihood in the range -1.8 to -1.1.

3.3 ProFam Generates Diverse Sequences Retaining Family Structure

To evaluate the design capabilities of ProFam-1, we conditioned on homologs within the 128 hold-out families (A.7), predicted the structure of the generated sequence and compared it to the AFDB entries of natural proteins within the same family using TM score [28]. Additionally, we report for each generated sequence the mean predicted local difference distance test (plDDT), an AlphaFold confidence metric [29]. When plotting these metrics against the maximum sequence identity across family members, including those sequences that were not sampled in the prompt (Fig. 4a, 4b), we



(a) (b) (c)

Figure 4: **ProFam Generates Diverse Sequences for Given Family Fold.** We compare sequences generated by ProFam-1 and PoET when conditioned on sequences from 128 held-out AFDB FoldSeek cluster families. We also show results for a random mutation baseline generated by randomly changing amino acids in natural sequences from the family. Structural coherence of the generated sequences was quantified by comparing ColabFold predictions of generated sequences with the family structures (taken from AFDB). 4a shows the relationship between maximum sequence identity and maximum TM-score when comparing the generated sequence with the natural family sequences and structures; similarly, 4b shows this relationship with pLDDT. 4c restricts the analysis to generated sequences with less than 50% maximum sequence identity to any natural family member, we report the proportion with TM-score greater than 0.5 (with 95% CI). Visual examples of structural overlays are provided in Appendix Figure 5. Sampling hyperparameters for PoET and ProFam are listed in Appendix Section A.9

find that ProFam-generated sequences (in both non-ensemble and ensemble mode) have similar TM and pLDDT scores to PoET, with a non-significant trend favouring ProFam when the PID is less than 70 (Fig. 4a). If we restrict our analysis to only consider generated sequences with PID less than 50 to any natural sequence in the family, we find that ProFam-1 sequences are more likely to have a TM score > 0.5 (Fig. 4c). Both ProFam-1 and PoET significantly outperform the random mutation baseline where random mutations are added to natural sequences at different rates.

4 Conclusion

We introduce ProFam-1, a fully open-source autoregressive protein family language model trained on the ProFam-atlas of 40-million protein families + single sequences. Our results demonstrate that ProFam-1 effectively internalizes biological constraints, reproducing evolutionary statistics and covariance structures found in natural alignments even when prompted with just a single sequence. Through our prompt ensembling strategy, ProFam-1 achieves zero-shot fitness prediction performance on ProteinGym that is comparable with state-of-the-art methods, showing that the model likelihood scores are correlated with fitness constraints. We further show that prompt ensemble sampling enables the generation of structurally valid sequences that exhibit low identity to natural homologs. By releasing ProFam-1, the ProFam-atlas, and our full codebase, we provide a robust, transparent foundation to accelerate community development in protein family language modelling and design.

5 Code and Data Availability

All components of the ProFam codebase, data pipelines, training procedures, inference utilities for variant scoring and family-conditioned generation, and detailed instructions for obtaining the ProFam-atlas dataset, are publicly available at `https://github.com/alex-hh/profam/`

Acknowledgments

JW, WL, DM, NB, BP, CO, BR, MH acknowledge funding from the BBSRC grant ProtFunAI (BB/Y514044/1). JW acknowledges the receipt of studentship awards from the Health Data Research UK-The Alan Turing Institute Wellcome PhD Programme in Health Data Science (Grant Ref: 218529/Z/19/Z). AHH was supported by the EPSRC Grant EP/S021566/1.

References

[1] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *PNAS*, 2019. doi: 10.1101/622803. URL `https://www.biorxiv.org/content/10.1101/622803v4`.

[2] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

[3] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: towards cracking the language of life's code through self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:7112–7127, 2021.

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

[5] Weining Lin, Jude Wells, Zeyuan Wang, Christine Orengo, and Andrew CR Martin. Enhancing missense variant pathogenicity prediction with protein language models using varipred. *Scientific Reports*, 14(1):8136, 2024.

[6] Brian L. Hie, Varun R. Shanker, Duo Xu, Theodora U. J. Bruun, Payton A. Weidenbacher, Shaogeng Tang, Wesley Wu, John E. Pak, and Peter S. Kim. Efficient evolution of human antibodies from general protein language models. *Nature Biotechnology*, pages 1–9, 2023. URL `https://www.nature.com/articles/s41587-023-01763-2`.

[7] Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, et al. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025.

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

[9] Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Diffusion language models are versatile protein learners. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[10] Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R Eguchi, Po-Ssu Huang, and Richard Socher. Progen: Language modeling for protein generation. *arXiv preprint arXiv:2004.03497*, 2020.

[11] N Ferruz, S Schmidt, and B Höcker. Protgpt2 is a deep unsupervised language model for protein design. nat commun. 2022; 13 (1): 4348.

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

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

[14] Geraldene Munsamy, Sebastian Lindner, Philipp Lorenz, and Noelia Ferruz. Zymctrl: a conditional language model for the controllable generation of artificial enzymes. In *NeurIPS machine learning in structural biology workshop*. NeurIPS, 2022.

[15] Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos Jr, Caiming Xiong, Zachary Z Sun, Richard Socher, et al. Large language models generate functional protein sequences across diverse families. *Nature biotechnology*, 41 (8):1099–1106, 2023.

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

[17] Yo Akiyama, Zhidian Zhang, Milot Mirdita, Martin Steinegger, and Sergey Ovchinnikov. Scaling down protein language modeling with msa pairformer. *bioRxiv*, pages 2025–08, 2025.

[18] Alex Hawkins-Hooker, David T Jones, and Brooks Paige. MSA-Conditioned Generative Protein Language Models for Fitness Landscape Modelling and Design. In *Machine Learning in Structural Biology Workshop at NeurIPS*, 2021.

[19] Bo Chen, Zhilei Bei, Xingyi Cheng, Pan Li, Jie Tang, and Le Song. Msagpt: Neural prompting protein structure prediction via msa generative pre-training. *Advances in Neural Information Processing Systems*, 37:37504–37534, 2024.

[20] Timothy Truong Jr and Tristan Bepler. Poet: A generative model of protein families as sequences-of-sequences. *Advances in Neural Information Processing Systems*, 36:77379–77415, 2023.

[21] Damiano Sgarbossa, Cyril Malbranke, and Anne-Florence Bitbol. Protmamba: a homology-aware but alignment-free protein state space model. *Bioinformatics*, 41(6), 2025.

[22] Ruben Weitzman, Peter Mørch Groth, Lood Van Niekerk, Aoi Otani, Yarin Gal, Debora Marks, and Pascal Notin. Protriever: End-to-end differentiable protein homology search for fitness prediction. *arXiv preprint arXiv:2506.08954*, 2025.

[23] Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, et al. Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research*, 50(D1):D439–D444, 2022.

[24] Gustaf Ahdritz, Nazim Bouatta, Sachin Kadyan, Lukas Jarosch, Dan Berenberg, Ian Fisk, Andrew Watkins, Stephen Ra, Richard Bonneau, and Mohammed AlQuraishi. Openproteinset: Training data for structural biology at scale. *Advances in Neural Information Processing Systems*, 36:4597–4609, 2023.

[25] Andy M Lau, Nicola Bordin, Shaun M Kandathil, Ian Sillitoe, Vaishali P Waman, Jude Wells, Christine A Orengo, and David T Jones. Exploring structural diversity across the protein universe with the encyclopedia of domains. *Science*, 386(6721):eadq4946, 2024.

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

[27] Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood Van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, et al. Proteingym: Large-scale benchmarks for protein fitness prediction and design. *Advances in Neural Information Processing Systems*, 36:64331–64379, 2023.

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

[29] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

[30] Eli Weinstein, Alan Amin, Jonathan Frazer, and Debora Marks. Non-identifiability and the blessings of misspecification in models of molecular fitness. *Advances in neural information processing systems*, 35:5484–5497, 2022.

[31] Chao Hou, Di Liu, Aziz Zafar, and Yufeng Shen. Understanding language model scaling on protein fitness prediction. *bioRxiv*, 2025.

[32] Inigo Barrio-Hernandez, Jingi Yeo, Jürgen Jänes, Milot Mirdita, Cameron LM Gilchrist, Tanita Wein, Mihaly Varadi, Sameer Velankar, Pedro Beltrao, and Martin Steinegger. Clustering predicted structures at the scale of the known protein universe. *Nature*, 622(7983):637–645, 2023.

[33] Gustaf Ahdritz, Nazim Bouatta, Christina Floristean, Sachin Kadyan, Qinghui Xia, William Gerecke, Timothy J O'Donnell, Daniel Berenberg, Ian Fisk, Niccolò Zanichelli, et al. Openfold: Retraining alphafold2 yields new insights into its learning mechanisms and capacity for generalization. *Nature methods*, 21(8):1514–1524, 2024.

[34] Maria Hauser, Martin Steinegger, and Johannes Söding. Mmseqs software suite for fast and deep clustering and searching of large protein sequence sets. *Bioinformatics*, 32(9):1323–1330, 2016.

[35] Michel van Kempen, Stephanie S Kim, Charlotte Tumescheit, Milot Mirdita, Cameron LM Gilchrist, Johannes Söding, and Martin Steinegger. Foldseek: fast and accurate protein structure search. *Biorxiv*, pages 2022–02, 2022.

[36] Ian Sillitoe, Nicola Bordin, Natalie Dawson, Vaishali P Waman, Paul Ashford, Harry M Scholes, Camilla S M Pang, Laurel Woodridge, Clemens Rauer, Neeladri Sen, Mahnaz Abbasian, Sean Le Cornu, Su Datt Lam, Karel Berka, Ivana Hutařová Varekova, Radka Svobodova, Jon Lees, and Christine A Orengo. CATH: increased structural coverage of functional space. *Nucleic Acids Research*, 49(D1):D266–D273, November 2020. ISSN 0305-1048. doi: 10.1093/nar/gkaa1079. URL `https://doi.org/10.1093/nar/gkaa1079`. _eprint: https://academic.oup.com/nar/article-pdf/49/D1/D266/35364652/gkaa1079.pdf.

[37] Sean R. Eddy. Accelerated profile hmm searches. *PLOS Computational Biology*, 7(10):1–16, 10 2011. doi: 10.1371/journal.pcbi.1002195. URL `https://doi.org/10.1371/journal.pcbi.1002195`.

[38] T E Lewis, I Sillitoe, and J G Lees. cath-resolve-hits: a new tool that resolves domain matches suspiciously quickly. *Bioinformatics*, 35(10):1766–1767, 10 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty863. URL `https://doi.org/10.1093/bioinformatics/bty863`.

[39] Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.

[40] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[41] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

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

[43] William P. Russ, Drew M. Lowery, Prashant Mishra, Michael B. Yaffe, and Rama Ranganathan. Natural-like function in artificial WW domains. *Nature*, 437(7058):579–583, September 2005. ISSN 1476-4687. doi: 10.1038/nature03990. URL `https://doi.org/10.1038/nature03990`.

[44] Steve W Lockless and Rama Ranganathan. Evolutionarily conserved pathways of energetic connectivity in protein families. *Science*, 286(5438):295–299, 1999.

[45] Martin Weigt, Robert A White, Hendrik Szurmant, James A Hoch, and Terence Hwa. Identification of direct residue contacts in protein–protein interaction by message passing. *Proceedings of the National Academy of Sciences*, 106(1):67–72, 2009.

[46] Michael Socolich, Steve W Lockless, William P Russ, Heather Lee, Kevin H Gardner, and Rama Ranganathan. Evolutionary information for specifying a protein fold. *Nature*, 437(7058):512–518, 2005.

[47] William P Russ, Matteo Figliuzzi, Christian Stocker, Pierre Barrat-Charlaix, Michael Socolich, Peter Kast, Donald Hilvert, Remi Monasson, Simona Cocco, Martin Weigt, et al. An evolution-based model for designing chorismate mutase enzymes. *Science*, 369(6502):440–445, 2020.

[48] Matteo Figliuzzi, Pierre Barrat-Charlaix, and Martin Weigt. How pairwise coevolutionary models capture the collective residue variability in proteins? *Molecular Biology and Evolution*, 35(4):1018–1027, 01 2018. ISSN 0737-4038. doi: 10.1093/molbev/msy007. URL `https://doi.org/10.1093/molbev/msy007`.

[49] Alex Hawkins-Hooker, Florence Depardieu, Sebastien Baur, Guillaume Couairon, Arthur Chen, and David Bikard. Generating functional protein variants with variational autoencoders. *PLoS Computational Biology*, 17, 2020. URL `https://api.semanticscholar.org/CorpusID:215790693`.

[50] Mario Michael Krell, Matej Kosec, Sergio P Perez, and Andrew Fitzgibbon. Efficient sequence packing without cross-contamination: Accelerating large language models without impacting performance. *arXiv preprint arXiv:2107.02027*, 2021.

[51] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022.

A Appendix

A.1 Additional results

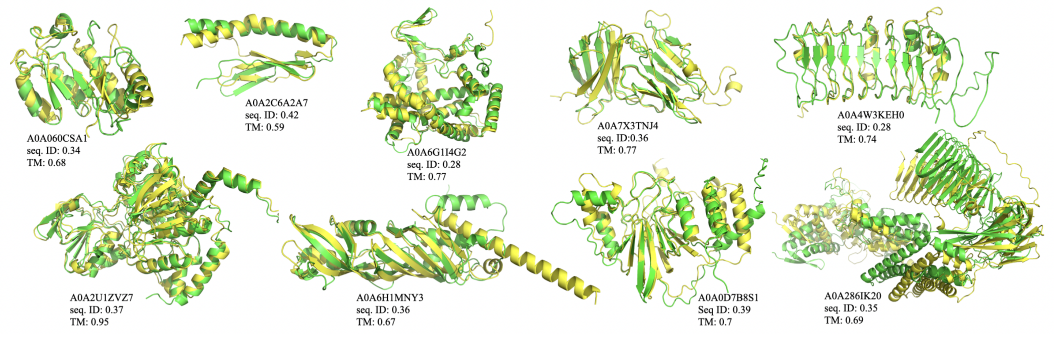
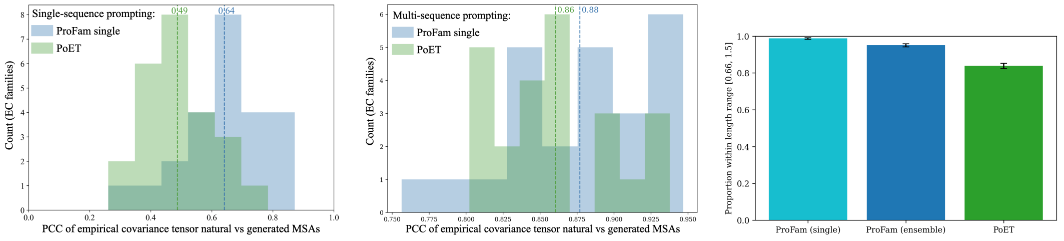


Figure 5: **ProFam-sampled structure overlays.** We highlight selected samples from the FoldSeek AFDB held-out set of 128 families. Predicted structures of ProFam-sampled sequences (green) overlaid with the predicted structure of the most similar natural sequence (yellow). For each pair, we report the UniProt ID, sequence identity and TM score.



(a) (b) (c)

Figure 6: **Assessing generated sequences when conditioning on sequences from EC families** In figures 6a and 6b we show the level of correlation between residue pair covariances in synthetic MSAs and natural MSAs (Sec. A.6.3) for 24 large EC families. Covariance correlation is strongest when the prompt contains multiple sequences from the family 6b; however, significant co-variation signal still remains in the synthetic MSAs even when we condition only on a single sequence from the family 6a. In 6c we show the proportion of sequences generated by ProFam-1 (and PoET) that fall within an acceptable length range, when conditioning on sequences from EC families. We define an acceptable length range for a generated sequence as being no less than $0.66 \times$ shortest seq. in family and no greater than $1.5 \times$ longest seq. in family

A.1.1 ProFam Synthetic MSAs for AlphFold on CASP

To quantify the utility of generated sequences, we input ProFam-1 synthetic MSAs (generated by conditioning on the target sequence only) for hard (free-modelling) CASP15/16 targets into ColabFold and measured structure prediction accuracy using lDDT and TM-score against experimental structures (Section A.6.4). Although synthetic MSAs did not match the performance of natural MSAs (lDDT=0.9), they improved over single-sequence inputs (lDDT=0.7 vs. 0.5). However, gains in TM-score were more modest, showing only a non-significant upward trend compared to single sequences, with natural MSAs remaining significantly superior (Fig. 7).

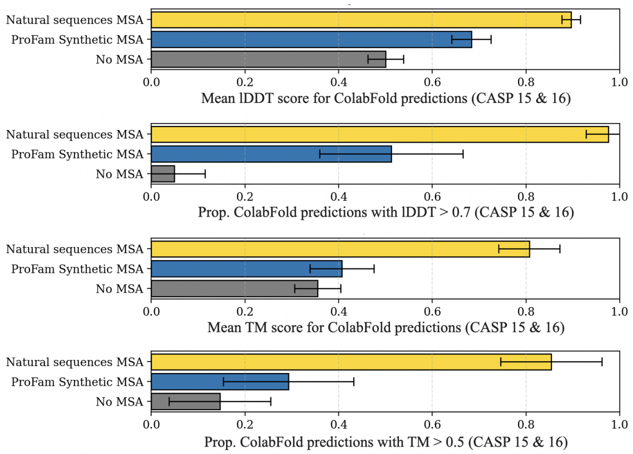
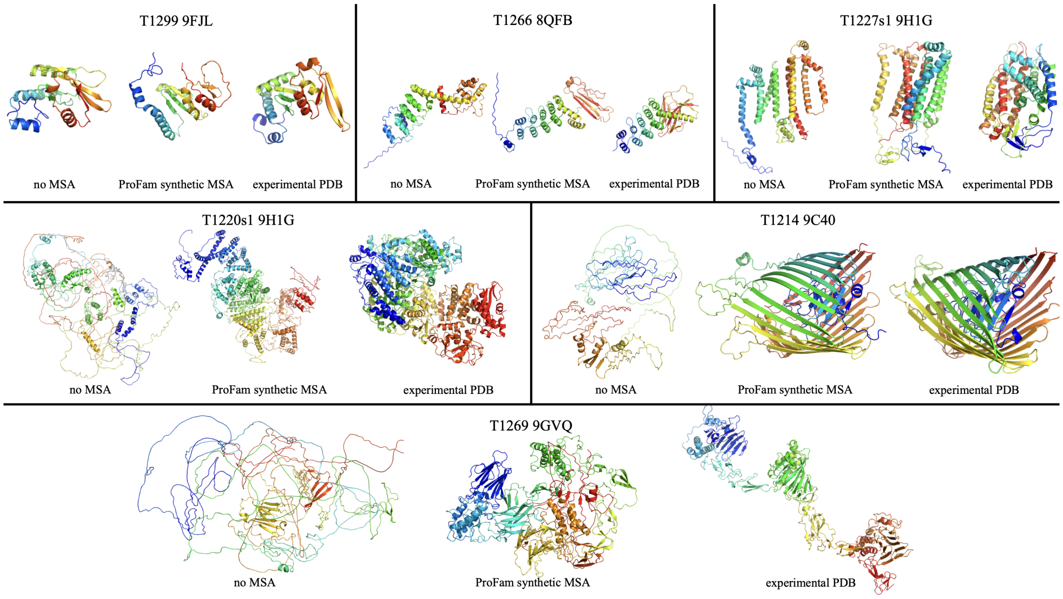


Figure 7: **ColabFold performance on targets from CASP15/16 using ProFam-1 synthetic MSAs.** We compare single-sequence inputs (no MSA), ProFam-1 synthetic MSAs, and natural MSAs. The plots show the mean (with 95% CI) for lDDT and TM-score, as well as the proportion (with 95% CI) of predictions exceeding standard quality thresholds (lDDT $>$ 0.7, TM-score $>$ 0.5).



(a)

Figure 8: **ProFam Synthetic MSAs Improve Structure Prediction.** (8a) AlphaFold2 predicted structure comparison for six CASP16 targets using: no MSA (left), and a ProFam synthetic MSA generated from the target sequence alone (middle). ProFam synthetic MSAs improve over no MSA in 5 of 6 visualised cases (T1266, T1227s1, T1220s1, T1214, T1269) and are worse in one case (T1299); Experimental PDB structures are shown on the right. MSA using real homologous sequences remains best across all targets.

A.1.2 Prompt Engineering for Improved Fitness Prediction.

As shown previously [30, 31], higher average log-likelihood does not monotonically translate into better zero-shot fitness prediction. As shown in Figure 3 D, performance improves as the variant log-likelihood increases up to roughly -1.8, but beyond approximately -1.1 the trend reverses: prompts that further raise likelihood tend to yield lower Spearman correlations with experimental fitness. Figure 3 D shows correlations scaled so that values for each assay span the [0, 1] range. This highlights the trend, but not the scale. In contrast, figure 9 displays the unnormalised Spearman values for each prompt-assay combination for 25 assays, with one point per prompt (i.e., per subsample of the family). Two patterns emerge. First, the attainable Spearman correlation varies widely across prompts for the same assay: the gap between the best and worst prompt commonly exceeds 0.3. Second, the average variant log-likelihood also spans a broad range, and the optimal likelihood differs by assay.

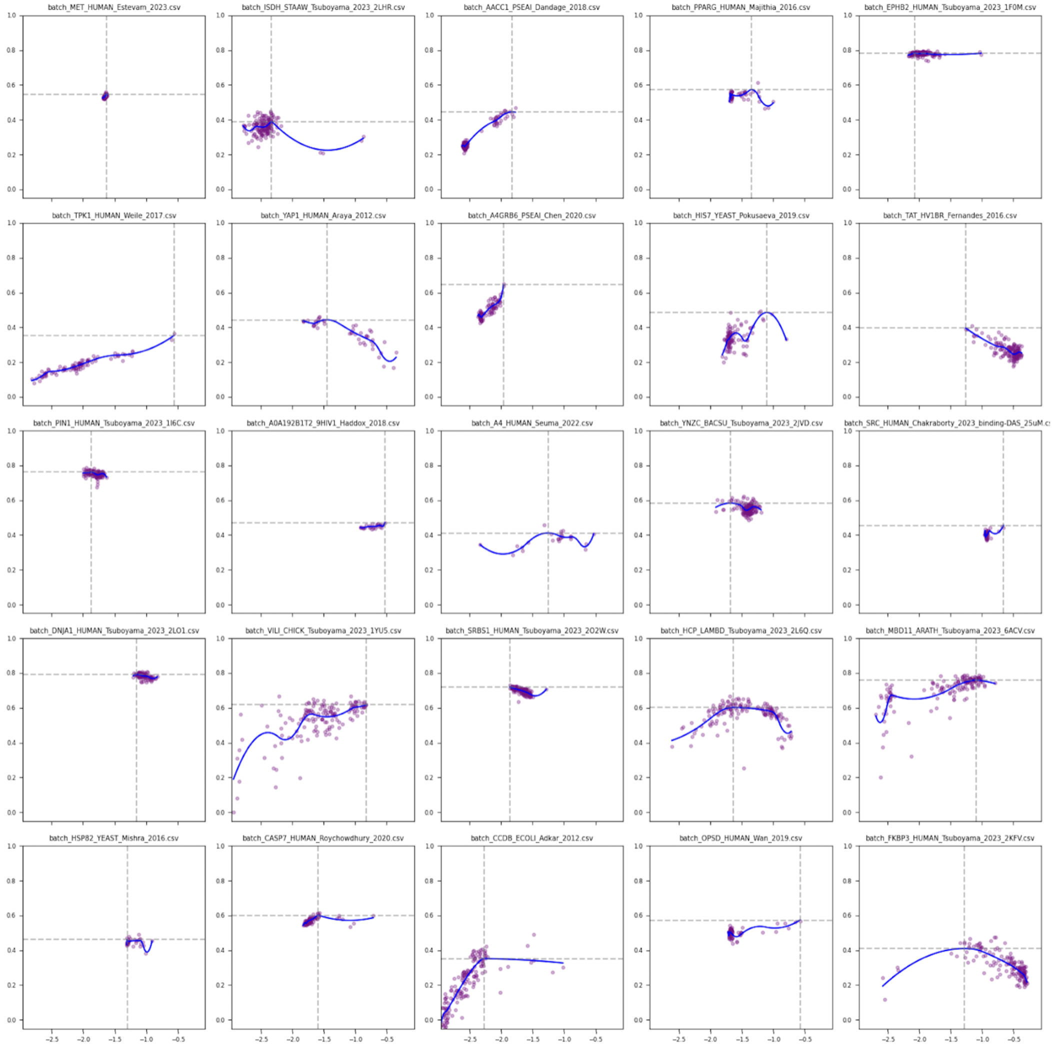


Figure 9: log-likelihood and Spearman correlation for 25 randomly selected individual assays. Each point in the scatter plot is a different prompt (set of context sequences). Here we observe that several assays exhibit a decline in Spearman correlation once the log-likelihood exceeds a certain threshold.

A.2 Methods

A.3 The ProFam-Atlas Dataset

We introduce a curated, large-scale dataset of protein sequences grouped by varying definitions of *family*. These family definitions, detailed in the following, range from structure-centric multi-domain proteins in the AlphaFold Database (AFDB) [23], over structure and function-centric single-domains in The Encyclopedia of Domains (TED) [25], to deep Multiple Sequence Alignments (MSAs) from the maximally diverse 270k-representative subset of the OpenFold OpenProteinSet [24]. Training dataset sampling probabilities and dataset sizes can be found in table A.6.5 (Appendix).

A.3.1 FoldSeek AlphaFold Database Clusters

To include a structure-centric definition of protein family, we source data from FoldSeek AlphaFold Database (AFDB) [23] clusters. Prior work applied first sequence (50% sequence identity at 90% coverage) and then structure (E-value<=0.01 at 90% coverage) based clustering to the AFDB, which resulted in 2.3 million non-singleton clusters [32]. We treat each of these clusters as a protein family from which we can sample sequences to create a single training example, which we refer to as a *document*. For validation, we constructed a leakage-free set of 128 protein families and filtered the training data against it to ensure no sequence in any of the training datasets has above or equal 30 percent of pairwise identical residues (PID) against *any* sequence in one of the 128 held-out families (Appendix section A.7). After removing validation proteins, this set contains 2.3M protein families comprising 30M sequences.

A.3.2 OpenProteinSet MSAs

The OpenProteinSet [24] is a widely used resource for defining protein families, notably for training structure prediction methods such as OpenFold [33]. This dataset originated from >16 million MSAs produced by HHblits aligning all-against-all on Uniclust30, from which the OpenFold authors filtered 270,000 maximally diverse MSAs. These MSAs may not be suitable for training protein family LMs, as sampling entries from the alignment can yield non-overlapping sequence segments with no pairwise similarity, or sequences with significant length differences. To remedy this and generate cleaner, length-consistent families, we re-processed the MSAs through a segmentation and clustering pipeline. We first split aligned sequences at contiguous gaps of > 10 residues, discarding fragments shorter than 90 residues. The remaining subsequences were pooled and re-clustered using MMseqs2 [34] at 30% sequence identity and 70% coverage; clusters containing at least two sequences were defined as valid families. This pipeline yielded 37M protein families comprising 246M sequences.

A.3.3 TED FunFams

While the FoldSeek AFDB clusters [35] provide a useful partitioning of the AFDB, sequences in this dataset are only retained if they achieve 90% overlap at the whole-chain level. To expand on this, we considered homology at the domain level and incorporated alignments from fine-grained functional groupings. Specifically, we mapped all continuous domains from The Encyclopedia of Domains [25] that had a CATH assignment onto a library of 212,872 Hidden Markov Model profiles for the CATH Functional Families (v4.3.0) [36]. TED domain sequences were scanned with `hmmsearch` from the HMMER3 suite [37] using per-profile threshold cutoffs (`cut_tc`). Resulting matches were further refined with `cath-resolve-hits` [38] using default parameters to identify optimal domain boundaries. TED domains matching existing FunFam signatures were redundancy-reduced at 50% sequence identity, 90% overlap with MMseqs2 [39]. To ensure sequence and length consistency within these families, we clustered within families (30% sequence identity, 80% overlap with MMseqs2) and keep all members in the largest cluster only. This resulted in 38k families and 765k sequences.

A.3.4 UniRef90 Single Sequences

To add coverage to regions of sequence space not covered by other family-based groupings, we included 205M single sequences from UniRef90 (release 2025_01) [26].

A.4 The ProFam-1 Model

ProFam-1 is an autoregressive transformer language model built on the Llama architecture [40] with 16 layers, 1024 hidden dimensions, and approximately 251 million parameters. The protein family Language Model (pfLM) is trained on a mix of single sequences and unaligned, concatenated protein sequences (up to 8192 tokens) from the ProFam-atlas families. Since many protein families substantially exceed this context window, each family-level training example is constructed by randomly subsampling sequences from the family, until the 8192-token budget is reached. We employ a residue-level tokenization strategy where each amino acid corresponds to a single token, supplemented by special tokens for sequence initiation and separators. We sampled training data uniformly (33% each) from AFDB, OpenProteinSet and UniRef90 but down-weighted TED FunFams to avoid oversampling of the relatively few families in this set. For positional encoding, we use the default Llama RoPE encoding scheme for positions within the concatenated sequence-of-sequences string. To separate sequences, we add a special separator token between concatenated sequences. This design preserves both within-sequence and between-sequence positional information. During training, we randomized the order of sequences within each sample to encourage invariance with respect to sequence order. With this setup, the model was trained for 592,619 steps covering 236 billion tokens over approximately 14 days on four NVIDIA A100 GPUs.

A.5 Ensemble Sequence Generation and Scoring

Borrowing from the success of test-time-scaling in LLMs [41], which leveraged more inferences to gain better performance downstream, we introduce ensemble sampling both as a strategy to handle context length, as well as a methodology to increase performance in downstream tasks. As such, rather than conditioning autoregressive inference on all available family members in a single forward pass, which would exceed typical transformer context windows, we construct M *prompts*, with M being a user-set hyperparameter, each containing a different subsample of sequences from the family. We sample prompt sequences non-exhaustively and with replacement, such that the same sequence cannot be repeated within a prompt, but may be repeated across different prompts. During generation, we average the M predicted probability distributions across prompts at each autoregressive step before sampling the next amino acid. When evaluating ensemble sequence generation with ProFam-1 we always used 8 independently sampled prompts ($M = 8$) to constitute the ensemble.

Formally:

- let M be the number of prompt variants in the ensemble.
- For each variant $m \in \{1, \dots, M\}$ let $z^{(m)} \in \mathbb{R}^{|V|}$ be the model logits for the next token over tokens $j \in V$, and define the per-variant next-token distribution:

$$p_j^{(m)} = \text{softmax}(z^{(m)})_j$$

The ensemble aggregates these M distributions into a single distribution over j:

$$\bar{p}_j = \frac{1}{M} \sum_{m=1}^{M} p_j^{(m)}$$

from which the next token is sampled and concatenated onto each of the prompts. For simplicity, here we omit top-p (aka nucleus sampling) and assume that the temperature hyperparameter of 1 is implied by omission.

For scoring, we also average across prompts to compute family-conditioned likelihoods, similar to [20]. This approach allows ProFam-1 to leverage information from diverse family members while keeping each forward pass within practical token budgets. Unlike in ensemble sampling, for scoring, the ensemble mean is computed over the model logits (pre-softmax) as opposed to the probabilities (post-softmax).

A.6 Benchmarks

To evaluate ProFam-1's capabilities in sequence generation, family statistic capture, and variant scoring, we utilized four distinct sets of benchmarks. First, we employed 128 held-out FoldSeek families to assess sequence generation quality and structural consistency. Second, we used Enzyme Commission (EC) families to evaluate the model's ability to capture sequence statistics within functionally related groups; specifically, 460 EC families were used for first-order statistic evaluation and a subset of 24 deep families was used for Statistical Coupling Analysis (SCA). Finally, to test the utility of ProFam-generated synthetic MSAs, we benchmarked structure prediction performance on CASP 15 and 16 targets. Additionally, we utilized the ProteinGym benchmark [27] for zero-shot variant scoring. Throughout these benchmarks, we extensively compare ProFam-1 against PoET [20], an autoregressive pfLM trained on sets of homologous sequences.

A.6.1 Sequence Generation and Structural Consistency

To evaluate the structural plausibility of generated sequences, we utilized the 128 held-out families described in Section A.3.1. For each family, we sampled 20 sequences from ProFam-1 and PoET while conditioning on a random subset of the family's sequences up to a maximum of 8192 tokens. For ProFam-1 in non-ensemble mode, we condition on the exact same prompt as PoET. For ProFam-1 in ensemble mode, we allow the prompts to differ. We selected the generated sequence with median length to predict its structure using ColabFold with default settings [42].

A.6.2 Family Sequence Statistics

We ascertained whether ProFam-1 captures sequence statistics such as conservation and single-site variations from both minimal context (single sequences) and extended context (multiple sequences). For this evaluation, we utilized 460 Enzyme Commission (EC) families (construction details in Appendix A.8), chosen because they group proteins by shared catalytic function.

We used a single representative sequence per EC family to independently generate sequences using the sampling parameters described in Appendix A.9. We compared the per-position conservation correlation and KL divergence between the natural family and the synthetic families generated by PoET and ProFam-1. KL divergence is measured at each position in the MSAs, and we report the average. We further evaluated the model using multi-sequence prompts to assess how additional evolutionary context influences the fidelity of generated statistics. For these experiments, we define an acceptable length range for a generated sequence as being no less than $0.66 \times$ shortest seq. in family and no greater than $1.5 \times$ longest seq. in family.

A.6.3 Higher-Order Statistics of Synthetic MSAs

First-order statistics derived from multiple sequence alignments (such as per-column conservation) are necessary but not sufficient to uniquely specify fold and activity [43]. Incorporating second-order statistics via pairwise couplings and associated statistical models (e.g. Statistical Coupling Analysis and Potts models [44, 45]) captures residue–residue constraints aligned with structural contacts [45], and enables the design of sequences that retain fold and function [46, 47]. To determine if ProFam-1 generated sequences correctly capture the pairwise couplings of protein families, we measure the level of correlation between residue pair covariances in synthetic MSAs and MSAs constructed by searching against natural sequence databases, similar to [48, 49].

Let L be the aligned sequence length and S the alphabet size (here $S = 20$). For the natural and synthetic MSAs, we compute SCA-style rank 4 covariance tensors C where indices i, j index aligned positions $\in \{1, ..., L\}$ and indices a, b index the amino acid identities at positions i and j respectively:

$$C^{\text{nat}},\, C^{\text{syn}} \in \mathbb{R}^{L\times L\times S\times S}, \quad C_{ijab} \;=\; P_{ij}(a,b) \;-\; P_i(a)\,P_j(b),$$

where probabilities are estimated from amino acid counts with a pseudocount of 1 added to each position, and gaps/unknowns excluded.

An entry (i, j, a, b) is kept if and only if all of the following hold:

1. Pair support filter: both MSAs have at least 10 sequences with non-gaps at positions i and j
2. i != j; (the diagonal is excluded)
3. Non-symmetric row filter at i: the count of residue a at position i is at least 10 in both MSAs; no residue-specific constraint is applied on b at position j.

We then calculate the Pearson correlation between the flattened covariance tensors, only for the retained entries. The covariance analysis was conducted on a subset of 24 deep EC families (from the 460 families described above) that possess sufficient homologs (>200 sequences) to accurately estimate covariance.

A.6.4 Synthetic MSA Generation for CASP Targets

As co-variation strongly influences structure prediction, to demonstrate the utility of ProFam-1 we benchmark synthetic MSA generation to predict the structure of nine protein targets from CASP16 and 45 protein-domain targets from CASP15, comprising all targets that had resolved PDB structures at the time of writing. CASP16 predictions were made at the whole chain level, while CASP15 was predicted at the domain level, due to source data formatting. Towards this end, we provided ProFam-1 only with the target sequence (no additional family information) to generate 1200 sequences (sampled independently of each other) for each target protein in CASP15/16. After alignment of the generated sequences, the resulting MSA was input to ColabFold/AlphaFold2 [42]. For comparison, we also compute scores for predictions made without an MSA (single-sequence input) and for predictions using an MSA of natural homologs generated with the default ColabFold protocol.

A.6.5 Zero-Shot Variant Fitness Prediction

Recent work has shown that the likelihood assigned to a variant sequence by a pLM is positively correlated with the variant's empirical fitness across many functional assays [27]. This enables the use of pretrained pLMs as zero-shot fitness predictors, without requiring any assay-specific training data. Here, we use "zero-shot variant scoring" to mean that the model's (optionally family-conditioned) sequence log-likelihood is used as a surrogate for experimentally measured fitness, without any training or fine-tuning on assay-specific labels. For an autoregressive pLM, such as ProGen [13], or RITA [12], the likelihood of a variant sequence with L amino acids is $S = (s_1, ..., s_L)$ is factorized as the product of conditional probabilities over amino acid tokens:

$$P(S = \{s_1, ..., s_L\}) = \prod_{i=1}^{L} P(s_i | s_{i-1}, s_{i-2}, ..., s_1)$$

The log-likelihood can thus be computed as the sum of conditional log-probabilities and to obtain a length-normalised metric, we define the fitness score as the mean conditional log-likelihood per token.

$$\hat{F}(S = \{s_1, ..., s_L\}) = \frac{1}{L} \sum_{i=1}^{L} \log P(s_i | s_{i-1}, s_{i-2}, ..., s_1)$$

Autoregressive protein *family* models, such as ProFam-1 and PoET [20], condition the probability of each amino acid not only on the preceding tokens in the variant sequence, but also on homologous sequences $\{R_1, R_2, ..., R_n\}$ from the same protein family:

$$\hat{F}(S = \{s_1, ..., s_L\} | R_1, R_2, ..., R_n) = \frac{1}{L} \sum_{i=1}^{L} \log P(s_i | s_{i-1}, s_{i-2}, ..., s_1, R_1, R_2, ..., R_n)$$

Conditioning on evolutionarily related sequences allows these models to leverage evolutionary information when estimating variant effects, which has been shown to improve fitness prediction accuracy compared to single-sequence models [20].

We benchmark protein-variant fitness prediction using the ProteinGym dataset [27], a collection of deep-mutational scanning (DMS) experiments spanning approximately 200 diverse proteins. For each protein, the dataset contains an experimentally derived fitness measurement for hundreds of variants across various functional assays, including thermal stability, enzymatic activity, and ligand binding. Computational methods are assessed based on the Spearman rank correlation between the predicted fitness score and the fitness assay result. To aid comparison with PoET [20], we used the same MSAs used by PoET.

Training data

Dataset	Sample Probability	Max tokens per example	Families	Sequences
FoldSeek S50	0.33	8192	2,300,407	30,033,967
Uniref90	0.33	8192	203,929,771	203,929,771
OpenFold	0.33	320	37,245,744	246,251,236
TED FunFams	0.01	8192	38,097	765,136

A.7 Held out dataset construction

We constructed the held-out validation dataset of 128 FoldSeek AFDB clusters and removed homologs from the training data based on sequence similarity:

1. **Select held-out families and remove them from training:** We selected 128 families from the FoldSeek clustering of the AlphaFold Database (AFDB) [23, 32, 35] and excluded all sequences belonging to these families from the candidate training pool.
2. **Build a non-redundant target set from held-out families:** We extracted all sequences from the 128 families and clustered them with MMseqs2 `easy-cluster` [34, 39] at 90% sequence identity and 80% alignment coverage. One representative per cluster was retained to form the held-out *target* database.
3. **Assemble the query set from all training sources:** We created a *query* database containing all remaining candidate training sequences across datasets (UniRef90, reprocessed OpenFold MSAs [24], and TED FunFams).
4. **Search and filter by homology:** We ran MMseqs2 `easy-search` from the query database against the held-out target database and removed any query sequence with a hit to any target representative at $\geq$30% sequence identity and $\geq$80% alignment coverage. The filtered remainder constitutes the final training set used in our experiments.

A.8 EC Clustered Validation Dataset

EC family annotations were downloaded from `https://ftp.expasy.org/databases/enzyme/enzyme.dat` on 30 May 2024, and corresponding proxtein sequences were retrieved using the UniProt API. For each level-4 EC family, sequences were clustered with MMseqs2 at a minimum of 30% sequence identity and 70% coverage. To ensure family homogeneity, only sequences belonging to the largest cluster in each family were retained, and families whose largest cluster contained fewer than 50 sequences were discarded.

Sequences in each retained cluster were aligned with MAFFT (v7.526). The resulting MSAs were filtered using HHfilter (v3.3.0) with default parameters and a maximum sequence identity threshold of 90% to remove highly redundant sequences. Families with fewer than 50 sequences after filtering were removed, yielding a final set of 460 EC families.

These 460 families were used for sequence-based evaluation of first-order family statistics (Section A.6.2). For Statistical Covariance Analysis (SCA; Section A.6.3), which requires deep MSAs to robustly estimate covariance tensors, we restricted the analysis to the 24 families whose filtered MSAs contained at least 200 sequences.

A.9 Sequence generation hyper-parameters

When sampling from sequences from PoET and ProFam-1, we limit the number of tokens in the prompt to the maximum sequence length seen by ProFam during training (8192 tokens) minus 1.2× the maximum sequence length in the prompt. This ensures that ProFam can generate a sequence that is 20% longer than any other sequence in the prompt while remaining in the supported token length range of 8192. We use a temperature of 1.0 for both ProFam-1 and PoET. ProFam-1 uses a top-p (nucleus sampling probability) of 0.95; PoET uses 0.9.

A.10 Diversity weighting for prompt sequence sampling

To mitigate the risk that the prompt might sample highly similar sequences from a large cluster within the family, we employ a diversity weighting scheme taken from [20]. The homology-based weighting scheme assigns each sequence in a multiple sequence alignment (MSA) a weight inversely proportional to the number of similar sequences in the MSA.

Given an MSA with sequences $\{x_i\}_{i=1}^{N}$, each of aligned length L, let the similarity between sequences i and j be

$$s_{ij} = \frac{1}{L_i} \sum_{k \in \mathcal{P}_i} \mathbf{1}\{x_{ik} = x_{jk}\},$$

where $\mathcal{P}_i$ is the set of positions in sequence i that are not gaps or non-standard residues, and $L_i = |\mathcal{P}_i|$ is the number of such positions. The corresponding distance is

$$d_{ij} = 1 - s_{ij}.$$

For a fixed threshold θ (in our case $\theta = 0.2$), the neighbor count for sequence i is

$$n_i = \left|\{j \in \{1, \ldots, N\} : d_{ij} \leq \theta\}\right|.$$

The unnormalised weight of sequence i is

$$w_i = \frac{1}{n_i},$$

and the final normalised weight returned by the function is

$$p_i = \frac{w_i}{\sum_{k=1}^{N} w_k},$$

so that $\sum_{i=1}^{N} p_i = 1$.

Table 1: Model architecture hyperparameter details.

Field	Value
vocab_size	68
hidden_size	1024
intermediate_size	4096
num_hidden_layers	16
num_attention_heads	16
num_key_value_heads	8
max_position_embeddings (rope)	131072
rope_theta	500000
attn_implementation	flash_attention_2
attention_bias	false
attention_dropout	0.0
rms_norm_eps	1.0e-05
hidden_act	silu
torch_dtype	bfloat16
pretraining_tp	1

Table 2: Training setup hyperparameter details.

Field	Value
optimizer	adamw
learning rate	0.001
scheduler	constant_with_warmup
num_warmup_steps	200
num_training_steps	592619
precision	bf16-true
accelerator	gpu
strategy	ddp
devices	auto
sync_batchnorm	true
seed	12345
accumulate_grad_batches	2
tokens_per_document	30000
pack_to_max_tokens	52000
batch_size	1*

* with sequence packing

We used sequence packing [50] with Flash Attention [51], allowing multiple documents to be packed into a single batch dimension with no cross-document attention, thereby eliminating the need for any padding tokens. We used memory-mapped datasets to load the data efficiently and achieved a throughput of 212,000 tokens per second (53,000 tokens per GPU per second). The average number of tokens in a packed batch for a single GPU was 50,000 and we used data-parallelism with 4 GPUs and 2 gradient accumulation steps per update resulting in an effective batch size of 400,000 tokens per training step.