SaprotHub: Making Protein Modeling Accessible to All Biologists

Jin Su[1], Zhikai Li[1], Chenchen Han[1], Yuyang Zhou[1], Junjie Shan[1], Xibin Zhou[1], Dacheng Ma[2], The OPMC[4], Sergey Ovchinnikov[3], Fajie Yuan[1]

[1]Westlake University.
[2]Zhejiang Lab.
[3]MIT.
[4]Open Protein Modeling Consortium.

Correspondence to: yuanfajie@westlake.edu.cn;

Abstract

Training and deploying deep learning models pose challenges for users without machine learning (ML) expertise. SaprotHub offers a user-friendly platform that democratizes the training, utilization, and sharing of protein ML models, fostering collaboration within the biologist community—all achievable with just a few clicks, regardless of ML background. At its core, Saprot is a near-universal protein language model that, through its ColabSaprot framework, supports hundreds of protein training and prediction applications, enabling the co-construction and co-sharing of these trained models, thereby enhancing user engagement and community-driven innovation.

Proteins, the building blocks of life, are essential for a myriad of biological processes and pivotal in medical breakthroughs, pharmaceutical innovations, and genetic discoveries. Despite their importance, understanding protein structure and function has been a significant challenge in the scientific community. The success of AlphaFold2 [17] in CASP14 marked a groundbreaking era in structural biology by achieving atom-level protein structure prediction. Simultaneously, large-scale protein language models (PLMs) are driving significant advances in protein function prediction.

In this context, several notable PLMs have emerged, including the ESM models [29, 20, 45, 50], ProtTrans [5], UniRep [2], ProGen [21, 24], and ProtGPT2 [6], each demonstrating remarkable efficacy in their respective tasks. However, training and deploying these machine learning (ML) models for proteins pose significant challenges for researchers without ML expertise. These include selecting appropriate model architectures, managing coding intricacies, preprocessing large datasets, training models, and interpreting results. This complexity often deters non-ML researchers, hindering their active involvement in this domain, especially as AI models grow more sophisticated.

Given these challenges, ColabFold [23] emerges as a pioneering initiative by deploying AlphaFold2 predictions on the Google Colaboratory, making protein folding accessible to researchers of all backgrounds. However, ColabFold primarily emphasizes model inference, lacking support for researchers in more complex tasks such as training their own ML models. Furthermore, its focus is limited to protein structure tasks and does not encompass protein function prediction.

In response to this need, we present SaprotHub, an easy-to-use platform similar to ColabFold but specifically tailored for protein function prediction. SaprotHub empowers researchers by enabling them to create and train their own models without requiring advanced ML and coding expertise. Additionally, SaprotHub distinguishes itself by supporting a broad range of protein function predictions, rather than being confined to specific functions.

Furthermore, we introduce the concept of the *Open Protein Modeling Consortium* (OPMC) (Supplementary), which aims to unify resources and efforts across the entire protein research community. Within the framework of OPMC, researchers can easily share their individually trained models, facilitating the direct

Partial materials have been published in the conference [30].

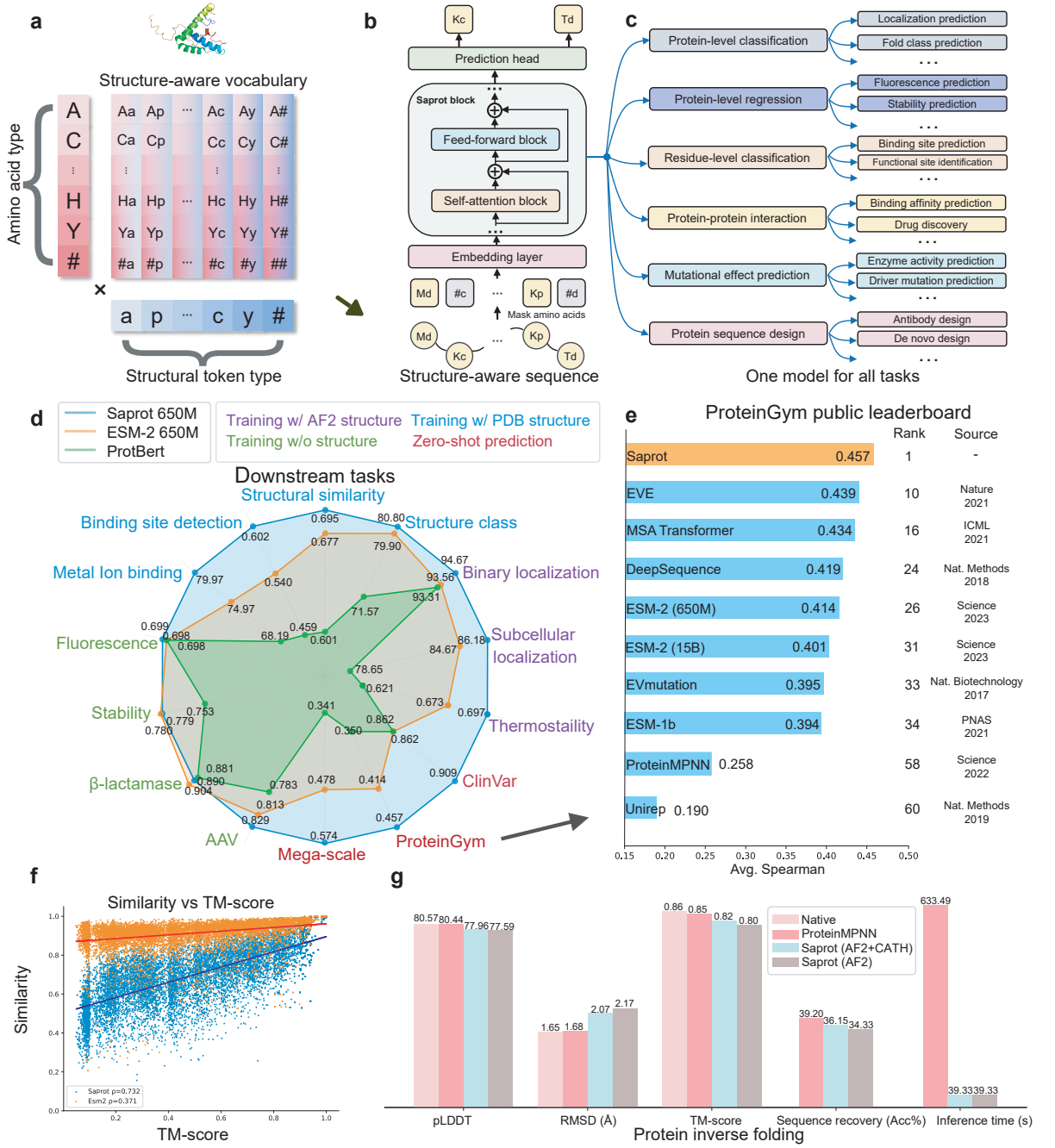


Fig. 1: **Illustration of Saprot. a,** The proposed SA vocabulary. #a, #p,...,#y represent only the 3Di token being visible, while A#, C#,..., Y# represent the AA token being visible. **b,** Architecture and pre-training of Saprot. **c,** Saprot supports numerous protein prediction tasks. **d,** Performance on 14 diverse prediction tasks. **e,** Saprot ranked *first* on ProteinGym leaderboard, outperforming 60 single and hybrid methods. Hybrid methods are not shown here (see Supplementary Fig. 1). **f,** Comparison of Saprot and ESM-2 in structural representation ability. The y-axis represents the cosine similarity score of the last layer representation for every pair of proteins, while the x-axis represents the TM-score of the protein pairs. Saprot shows higher Pearson correlation coefficient (ρ) between x and y axes, indicating more accurate protein structure encoding. **g,** Performance on protein sequence design. Although Saprot was not trained with a typical generation loss, its protein design performance rivals that of ProteinMPNN, with over 16 times faster inference speed. Saprot (AF2+CATH) indicates that it was pre-trained using AlphaFold2 predicted structures and fine-tuned with experimental structures from the CATH dataset. CATH is also used for training ProteinMPNN.

utilization of existing models and enabling continuous model learning and improvement through collaborative efforts. SaprotHub is designed to bring this vision to fruition, marking a significant stride towards the realization of OPMC.

SaprotHub consists of three parts: a novel protein language model called Saprot (Fig. 1a,b), ColabSaprot (Fig. 2a,b,c) for Saprot fine-tuning and inference by implementing adapter learning techniques [14] on the Colab platform, and an online community application store (Fig. 2d) for saving, sharing, retrieving,

and collaborating on fine-tuned Saprot models. Through the innovative integration of foundation protein models, Colab cloud computing technology, and adapter-based fine-tuning techniques, SaprotHub effectively addresses several key challenges. These include the difficulty of sharing and aggregating models due to their large parameter scales, the potential for catastrophic forgetting during continual learning, and the necessity of protecting proprietary data owned by biologists. Below, we describe these three modules in detail.

We developed Saprot, a large protein language model with universal capabilities that forms the foundation of SaprotHub. A distinctive feature of Saprot is its pioneering introduction of a structure-aware (SA) protein representation method, which departs from traditional amino acid (AA) sequences or 3D coordinate-based tertiary structure representations. Specifically, we introduce a novel SA vocabulary (SAV) for proteins, where $SAV = \{\mathcal{V}, \mathcal{F}\}$ represents the Cartesian product of $\mathcal{V}$ and $\mathcal{F}$, with $\mathcal{V}$ denoting the 20 amino acids and $\mathcal{F}$ denoting the 20 structural (a.k.a. 3Di) alphabet (Fig. 1a). The 3Di tokens are generated by encoding the 3D structures of proteins with a discrete autoencoder named Foldseek [32]. The SA token encompasses all combinations of AA and 3Di types in SAV. With SAV, proteins can be represented as a string of these newly developed SA tokens, capturing both their primary and tertiary structures. While intuitive and concise, the SA vocabulary proposed in Saprot has proven effective in addressing scalability issues when training on large-scale predicted protein structures [30, 4] and has garnered increasing attention in much subsequent research [8, 19, 26, 9].

Regarding the network architecture, we adopt the popular bidirectional Transformer [38] (Fig. 1b) for Saprot, similar to the ESM models. During the pre-training process, Saprot takes SA sequences (instead of AA sequences) as input, partially masks a few randomly selected SA tokens, and predicts the complete form of these masked tokens (Methods and Fig. 1b). With 650 million parameters, Saprot was trained from scratch on approximately 40 million protein SA sequences—the largest number of structures used for training PLMs at the time of the Saprot preprint release—over the course of three months, utilizing 64 NVIDIA 80G A100 GPUs. This substantial computational demand is similar to that used for training models like ESM-2 (650M) and AlphaFold3 [1]. The protein structure used to represent the SA sequence during training is predicted using AlphaFold2 [17].

After pre-training, Saprot has become a near-universal protein language model, adeptly addressing a wide array of prominent protein prediction tasks (Fig. 1c), including supervised training, zero-shot prediction, and protein sequence design. In supervised training, Saprot tackles a spectrum of classification and regression tasks, spanning both protein-level (e.g., fluorescence, thermostability, structure, and folding class prediction) and residue-level (e.g., binding site detection) predictions [15]. In zero-shot prediction scenarios, Saprot can predict enzyme activity [10], viral mutation fitness landscape [11], disease variant effects, and identify driver mutations [7] without fine-tuning (Methods: zero-shot datasets). Although not explicitly designed as a generative model, Saprot (with a masked language modeling loss) has demonstrated a surprising proficiency in the protein sequence design task [3], achieving a 16× speedup in inference compared to ProteinMPNN [3] (Fig. 1g).

Fig. 1d illustrates Saprot's clear dominance across over 10 diverse protein prediction tasks in comparison to ESM-2 [20] and ProtBert[5], two of the most prominent PLMs. When protein structures are available, Saprot consistently outperforms both ESM-2 and ProtBert, as shown by the tasks marked in purple, blue, and red. Even in scenarios where it is not feasible to input available protein structures (either experimentally determined or predicted by AlphaFold2), Saprot remains highly competitive, as indicated by the green color tasks. Notably, Saprot substantially outperforms ESM-2 in three zero-shot mutational effect prediction tasks (Mega-scale [31]: 0.574 vs. 0.478, ProteinGym [25]: 0.457 vs. 0.414, ClinVar [18]: 0.909 vs. 0.862).

At the time of our submission, Saprot had maintained the first position on the public blind ProteinGym leaderboard (Fig. 1e, Supplementary Fig. 1) for several consecutive months, despite its model parameters being released eight months earlier. Saprot surpasses over 60 most well-known methods in the field, including ESM-2 (650M, 3B, 15B), EVE [7], ESM-1v [45], MSA Transformer [50], ESM-IF [12], DeepSequence [28], UniRep [2], and some hybrid models. The 35M parameter version of Saprot even outperforms ESM-2 with 15B parameters. Supplementary Table 2-3 and Fig. 1, 3 provide more comparison results. The above results show that Saprot nearly has the 'one-model-fits-all' capability, allowing a single model framework to support numerous tasks, much like a tree bearing diverse fruits (Fig. 1c, d). This property is fundamental to underpinning SaprotHub for model co-construction and collaboration, as described below.

We developed ColabSaprot by integrating Saprot in Google Colab's infrastructure for seamless deployment and execution. This eliminates the need for environment setup and code debugging, enabling researchers to initiate training sessions effortlessly with just a few clicks (Fig. 2a). ColabSaprot supports all tasks within the Saprot framework, enabling direct prediction for tasks such as zero-shot mutation effect prediction [10, 7, 28] and protein sequence design [13, 3]. Specifically, for mutation effect prediction, we implemented single-site mutation prediction, multi-site mutation prediction, and all mutations prediction by scanning the entire protein sequence. For sequence design, we could either design sequences based on the backbone structure, or redesign specific regions of a known protein, such as designing the variable regions of antibodies while keeping the scaffold fixed (Methods).

For supervised tasks, users are able to fine-tune ColabSaprot using their own data. Here, we introduce a parameter-efficient fine-tuning technique by integrating lightweight adapter networks into ColabSaprot

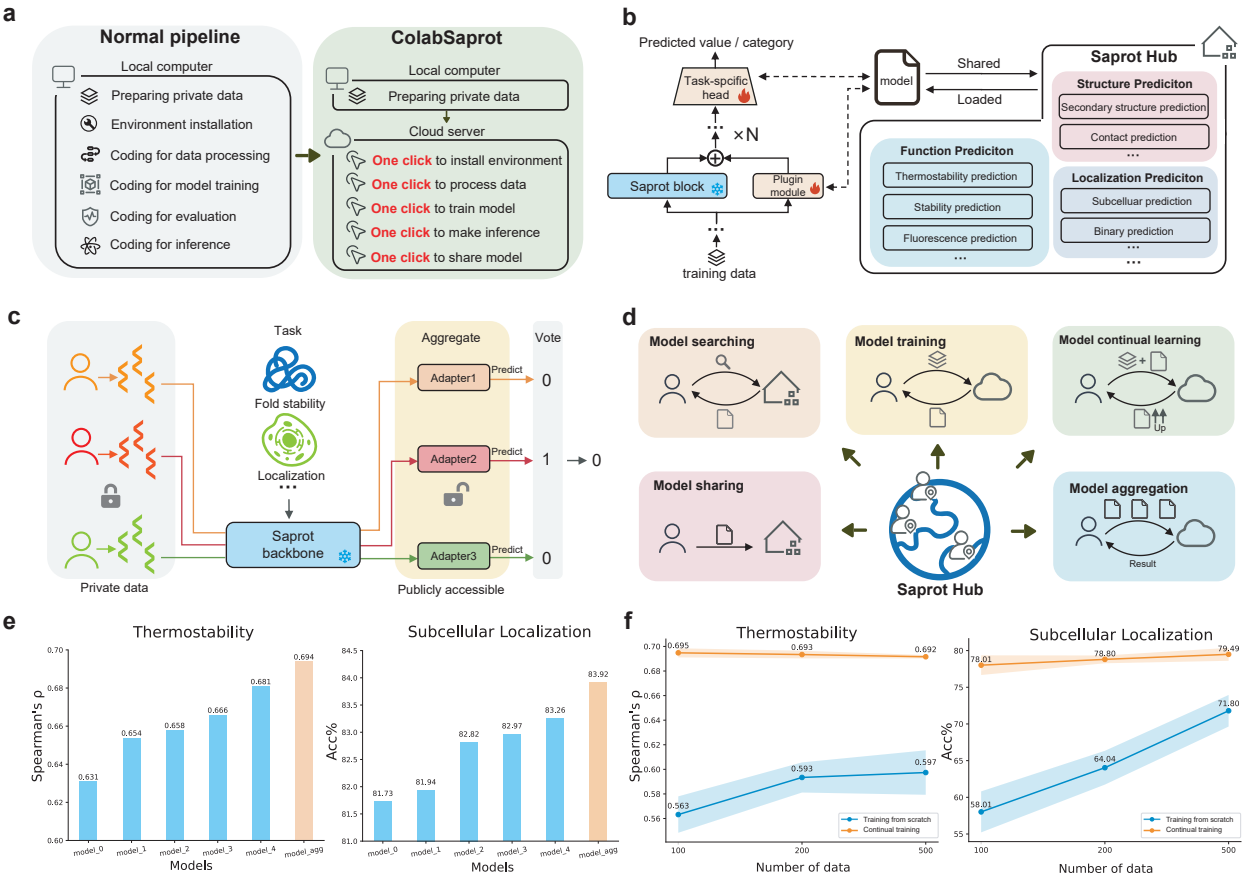


Fig. 2: **Illustration of ColabSaprot and SaprotHub. a,** comparison of model training and inferring between normal pipeline and ColabSaprot. ColabSaprot streamlines the process, offering simplified procedures for model training and inference with just a few clicks. **b,** a lightweight plugin module (i.e., adapter) is inserted into Saprot for efficient training, sharing and co-construction. **c,** ColabSaprot predicts protein functions by aggregating existing models (or adapters) in SaprotHub without requiring private data. **d,** SaprotHub enables biologists to share, co-build and collaborate within the community. **e,** ColabSaprot's community-based model collaboration mechanism (see **c**) allows it to achieve higher performance (orange bars) by aggregating individually trained models (blue bars). Each individual model is trained with its own data, which may or may not overlap. **f,** By continually learning from models trained and shared by other biologists, ColabSaprot significantly outperforms those training-from-scratch models, especially when users lack sufficient training data (x-axis represents the number of training examples).

(Fig. 2b). During training, only the adapter parameters in ColabSaprot are updated, resulting in nearly identical accuracy compared to fine-tuning all Saprot parameters (Supplementary Fig. 4). This design not only enhances optimization efficiency [33, 54], but more importantly, as proposed for the first time in this work, it enables collaborative and integrated mechanisms for these biologist fine-tuned, task-specific PLMs (or Saprot) within the community, especially in the cloud environment (Fig. 2c, d, e and Methods). Furthermore, with adapters, researchers can easily store and exchange their re-trained Saprot models in SaprotHub, accessing existing models by simply loading or uploading adapter networks instead of the full pre-trained model. Given that adapter networks contain much fewer parameters, often around 1% of the whole Saprot model, this method significantly reduces storage, communication, and management burdens on SaprotHub, making accessibility, co-construction, co-utilization and co-sharing possible for a broader scientific community (Fig. 2b, c, d).

Additionally, we developed many other features for ColabSaprot. Our interface supports nine types of data inputs, including SA sequences, AA sequences, UniProt IDs, and PDB files, etc. We offered automatic conversion of input file formats, automatic division of training and testing sets, overfitting checks, evaluation and visualization. To address GPU memory limitations, we implemented an adaptive batch size feature based on gradient accumulation [17], enhancing user convenience. We also provided advanced settings that allow researchers to extend code and customize training hyperparameters to suit their specific needs. By using ColabSaprot, researchers benefit from these specifically designed features.

Finally, we developed SaprotHub, an app store for seamlessly sharing and co-building peer-trained Saprot models within the biology community (Fig. 2d). Through the ColabSaprot interface, we implemented several advanced features, such as model storage, sharing, search, continual learning, and aggregation for better performance. This brings two important benefits to SaprotHub: (1) It allows biologists to share models they have trained themselves without compromising private data, thereby overcoming data sharing challenges

while enabling the sharing of research outcomes (Fig. 2c); (2) It enables biologists to continually train models based on shared models from peers (Fig. 2d). This is particularly valuable when data availability is limited, as fine-tuning on a superior pre-trained model typically improves predictive accuracy (Fig. 2f). In the future, as more biologists join SaprotHub, multiple Saprot models will be submitted for the same protein function. For a new user, our ColabSaprot can build upon existing Saprot models through adapter aggregation (Fig. 2c and Methods) or other more advanced ensemble learning [16], thereby enhancing predictive performance (Fig. 2e).

SaprotHub and ColabSaprot empower biology researchers to easily train and share advanced protein ML models for various prediction tasks, even without expertise in ML or coding. This capability enables the entire protein research community to contribute their models, fostering collaboration and promoting the dissemination of high-quality, peer-trained models. We have open-sourced the code for both Saprot and ColabSaprot, allowing other protein foundation models to adopt a similar approach or join SaprotHub.

We emphasize that this innovative, community-wide participation approach to protein modeling aligns with the OPMC vision. Our goal is to inspire and foster the collaborative development of protein modeling through SaprotHub. We envision SaprotHub as a catalyst for OPMC, driving innovation and cooperation in the field.

References

[1] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, pages 1–3, 2024.

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

[3] Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

[4] Diego del Alamo. The issues when training on the afdb structure data. *twitter: https://x.com/DdelAlamo/status/1795353297580445851*, 2024.

[5] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2021.

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

[7] Jonathan Frazer, Pascal Notin, Mafalda Dias, Aidan Gomez, Joseph K Min, Kelly Brock, Yarin Gal, and Debora S Marks. Disease variant prediction with deep generative models of evolutionary data. *Nature*, 599(7883):91–95, 2021.

[8] Benoit Gaujac, Jérémie Donà, Liviu Copoiu, Timothy Atkinson, Thomas Pierrot, and Thomas D Barrett. Learning the language of protein structure. *arXiv preprint arXiv:2405.15840*, 2024.

[9] Thomas Hayes, Roshan Rao, Halil Akin, Nicholas James Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Quy Tran, Jonathan Deaton, Marius Wiggert, Rohil Badkundri, Irhum Shafkat, Jun Gong, Alexander Derry, Raul Santiago Molina, Neil Thomas, Yousuf Khan, Chetan Mishra, Carolyn Kim, Liam J. Bartie, Patrick D. Hsu, Tom Sercu, Salvatore Candido, and Alexander Rives. Simulating 500 million years of evolution with a language model. *bioRxiv*, 2024.

[10] Yan He, Xibin Zhou, Chong Chang, Ge Chen, Weikuan Liu, Geng Li, Xiaoqi Fan, Mingsun Sun, Chensi Miao, Qianyue Huang, et al. Protein language models-assisted optimization of a uracil-n-glycosylase variant enables programmable t-to-g and t-to-c base editing. *Molecular Cell*.

[11] Brian Hie, Ellen D Zhong, Bonnie Berger, and Bryan Bryson. Learning the language of viral evolution and escape. *Science*, 371(6526):284–288, 2021.

[12] Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. In *International conference on machine learning*, pages 8946–8970. PMLR, 2022.

[13] Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. In *International conference on machine learning*, pages 8946–8970. PMLR, 2022.

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

[15] Mingyang Hu, Fajie Yuan, Kevin Yang, Fusong Ju, Jin Su, Hui Wang, Fei Yang, and Qiuyang Ding. Exploring evolution-aware &-free protein language models as protein function predictors. *Advances in Neural Information Processing Systems*, 35:38873–38884, 2022.

[16] Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *arXiv preprint arXiv:2306.02561*, 2023.

[17] 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.

[18] Melissa J Landrum, Jennifer M Lee, Mark Benson, Garth R Brown, Chen Chao, Shanmuga Chitipiralla, Baoshan Gu, Jennifer Hart, Douglas Hoffman, Wonhee Jang, et al. Clinvar: improving access to variant interpretations and supporting evidence. *Nucleic acids research*, 46(D1):D1062–D1067, 2018.

[19] Mingchen Li, Yang Tan, Xinzhu Ma, Bozitao Zhong, Huiqun Yu, Ziyi Zhou, Wanli Ouyang, Bingxin Zhou, Liang Hong, and Pan Tan. Prosst: Protein language modeling with quantized structure and disentangled attention. *bioRxiv*, pages 2024–04, 2024.

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

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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, 2024.

[26] Mahdi Pourmirzaei, Farzaneh Esmaili, Mohammadreza Pourmirzaei, Duolin Wang, and Dong Xu. Prot2token: A multi-task framework for protein language processing using autoregressive language modeling. *bioRxiv*, pages 2024–05, 2024.

[27] 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.

[28] Adam J Riesselman, John B Ingraham, and Debora S Marks. Deep generative models of genetic variation capture the effects of mutations. *Nature methods*, 15(10):816–822, 2018.

[29] 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.

[30] Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. Saprot: Protein language modeling with structure-aware vocabulary. In *The Twelfth International Conference on Learning Representations*, 2023.

[31] Kotaro Tsuboyama, Justas Dauparas, Jonathan Chen, Elodie Laine, Yasser Mohseni Behbahani, Jonathan J Weinstein, Niall M Mangan, Sergey Ovchinnikov, and Gabriel J Rocklin. Mega-scale experimental analysis of protein folding stability in biology and design. *Nature*, 620(7973):434–444, 2023.

[32] 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.

[33] Shuai Zeng, Duolin Wang, and Dong Xu. Peft-sp: Parameter-efficient fine-tuning on large protein language models improves signal peptide prediction. *bioRxiv*, pages 2023–11, 2023.

Methods

Constructing SA Protein Sequence

The Structure-Aware (SA) vocabulary encompasses both residue and structure information, as illustrated in Fig. 1**a**. Given a protein P, its primary sequence can be denoted as $(s_1, s_2, ..., s_n)$, where $s_i \in \mathcal{V}$ represents the residue at the i_{th} site, and $\mathcal{V}$ represents residue alphabet. Drawing inspiration from the vector quantization learning technique [55], we can can encode protein tertiary structures into discrete residue-like structural tokens. Here, we use Foldseek [32], a purpose-built software. Through Foldseek, we have a structure alphabet $\mathcal{F}$, wherein P is expressed as the sequence $(f_1, f_2, ..., f_n)$, with $f_j \in \mathcal{F}$ representing the 3Di token for the j_{th} residue site. To maintain simplicity, we adhere to the default configuration of Foldseek, which sets the size m of $\mathcal{F}$ to 20. We then combine the residue and structure tokens per residue site, generating a new structure-aware protein sequence $P = (s_1f_1, s_2f_2, ..., s_nf_n)$, where $s_if_i \in \mathcal{V} \times \mathcal{F}$ is the so-called SA token naturally fusing both residue and geometric conformation information. The SA-token protein sequence can then be fed into a standard Transformer encoder as basic input. It's important to note that we also introduce a mask signal "#" to both residue and structure alphabet, which results in "s_i#" and "#f_i" that indicate only residue or structure information is available. The size of the SA vocabulary is $21 \times 21 = 441$.

Saprot architecture and pre-training

Saprot employs the same network architecture and parameter size as the 650M version of ESM-2 [20], which is inspired by the BERT [38] model in the Natural Language Processing (NLP) field. The main distinction lies in the expanded embedding layer, which encompasses 441 structure-aware tokens instead of the original 20 residue tokens. This nearly identical architecture enables straightforward comparison with the ESM model.

Saprot is pre-trained with the Masked Language Modeling (MLM) objective, similar to ESM-2 and BERT. Formally, for a protein sequence $P = (s_1f_1, s_2f_2, ..., s_nf_n)$, the input and output can be represented as: $input : (s_1f_1, ..., \#f_i, ..., s_nf_n) \rightarrow output : s_if_i$ (Fig. 1b). Given that the 3Di token may not always be accurate for certain regions in predicted structures by AlphaFold2, f_i in $\#f_i$ is made visible during training to reduce the emphasis the model places on its predictions.

It is worth noting that the structures predicted by AlphaFold2 (AF2) are accompanied by confidence scores, referred to as pLDDT, which provide an assessment of the precision of atom coordinates. Therefore, we have implemented special handling for regions with low pLDDT scores. During the pre-training process, when regions with pLDDT scores lower than 70 are chosen for MLM prediction, we predict the "s_i#" token. Here, the input SA sequence is masked with the "##" token. This approach forces the model to focus on predicting the types of residues in these regions based on their context. In cases where regions with lower pLDDT scores are not selected for MLM prediction, the input still use the "s_i#" token. This ensures that the model solely relies on residue context (instead of the structural context) in these regions to assist in predicting other tokens. For downstream task prediction phases, we maintain consistency with the training data by handling regions with lower pLDDT scores (still with 70 as threshold). Specifically, tokens within these regions are represented as "s_i#", with only residue tokens being visible.

Processing pre-training data

We followed the procedures outlined in ESM-2 [20] to generate filtered protein data. Subsequently, we acquired all AF2 structures via the AlphaFoldDB website (https://alphafold.ebi.ac.uk/), utilizing the UniProt IDs of protein sequences. Proteins without structures in AlphaFoldDB were omitted for pre-training. This process yielded a collection of approximately 40 million structures. Employing Foldseek, we encoded these structures into 3Di tokens. Subsequently, we formulated structure-aware sequences by combining residue and 3Di tokens, treating them as a single SA token at each position.

Pre-training hyper-parameters

Following ESM-2 and BERT, during training, 15% of the SA tokens in each batch are masked. We replace the SA token s_if_i with the $\#f_i$ token 80% of the time, while 10% of the tokens are replaced with randomly selected tokens, and the other 10% tokens remain unchanged. For the optimization of Saprot, we adopt similar hyper-parameters to those employed in the ESM-2 training phase. Specifically, we employ the AdamW optimizer [44], setting $\beta_1 = 0.9, \beta_2 = 0.98$ and we utilize L_2 weight decay of 0.01. We gradually increase the learning rate from 0 to 4e-4 over the first 2000 steps and linearly lower it to 5e-4 from 150K steps to 1.5M steps. The overall training phase lasts approximately 3M steps. Like the ESM model, we also truncate them to a maximum of 1024 tokens, and our batch size consists of 512 sequences. Additionally, we also employ mixed precision training to train Saprot.

Baseline models

We compared Saprot to several prominent protein language models (see Supplementary Table 2,3). For supervised learning tasks, we compared Saprot with ESM-2 (the 650M version) [20], ProtBert (the BFD version) [5], MIF-ST [58], and GearNet [60]. The first two models utilize residue sequences as input, while the latter two models incorporate both residue and 3D structures as input. ESM-2 (650M) stands out as the primary baseline for comparison, given its remarkably similar model architecture, size, and training approach when compared to Saprot. ESM-2 also offers a 15 billion (B) parameter version, which can be challenging to fine-tune even on GPUs with 80G memory. Therefore, we only conducted comparisons with ESM-2 (15B) for zero-shot mutational effect prediction tasks, which can be achieved without the need for fine-tuning.

For the zero-shot mutational effect prediction task, we compare with ESM-2, ProtBert, ESM-1v [45], Tranception L (without MSA retrieval) [46], MSA Transformer [50], and EVE [7]. These models are also state-of-the-art according to the literature. For the protein inverse folding task, we compare with the well-known baseline ProteinMPNN [37] (see Fig. 1).

Zero-shot formula

Previous sequence-based protein language models like the ESM models predict mutational effects using the log odds ratio at the mutated position. The calculation can be formalized as follows:

$$\sum_{t \in T} [logP(x_t = s_t^{mt} | x_{\setminus T}) - logP(x_t = s_t^{wt} | x_{\setminus T})] \tag{1}$$

Here T represents all mutations and $s_t \in \mathcal{V}$ is the residue type for mutant and wild-type sequence. We slightly modify the formula above to adapt to the structure-aware vocabulary, where the probability assigned to each residue corresponds to the summation of tokens encompassing that specific residue type, as shown below:

$$\sum_{t \in T} [log \sum_{f \in \mathcal{F}} P(x_t = s_t^{mt} f | x_{\setminus T}) - log \sum_{f \in \mathcal{F}} P(x_t = s_t^{wt} f | x_{\setminus T})] \tag{2}$$

Here $f \in \mathcal{F}$ is the 3Di token generated by Foldseek and $s_t f \in \mathcal{V} \times \mathcal{F}$ is the SA token in our new vocabulary.

Zero-shot benchmarks

ProteinGym [25] is an extensive set of Deep Mutational Scanning (DMS) assays, enabling thorough comparison among zero-shot predictors. We assess all baselines on the substitution branch of the ProteinGym, utilizing its provided protein structures and following the standard evaluation pipeline from the original paper. For evaluation, We adopt Spearman's rank correlation as our metric.

ClinVar [18] serves as a freely accessible and publicly available repository containing information about human genetic variants and interpretations of their significance to disease. In our analysis, we harness the data sourced from EVE [7], and filter out proteins with length greater than 1024. To enhance the reliability of our data, we opt to consider proteins with labels 1 "Gold Stars" or higher, which indicate higher credibility. Following the methodology employed in EVE, we evaluate models' performance using the AUC metric.

Mega-scale [31] uses cDNA display proteolysis to measure the thermodynamic folding stability of protein domains. Its released dataset contains all single mutations and selected double mutants of natural and de novo designed proteins. We employ Spearman's rank correlation as our metric.

For each mutation dataset, we provide all variants with the wild-type structure, as AF2 cannot distinguish the structural changes induced by single mutations. The ClinVar dataset only provides UniProt IDs, so we manually downloaded all AF2 structures and eliminated proteins without structures in AlphaFoldDB. Both ProteinGym and Mega-scale datasets provide protein structures, either predicted from AF2 or derived from de novo design.

Supervised fine-tuning benchmarks

Fine-tuning Saprot with AF2 structure. For benchmarks lacking PDB structures, we retrieve all AF2-predicted structures using UniProt IDs. These benchmarks encompass the Thermostability task from FLIP [36] and the Localization Prediction task from DeepLoc [34]. DeepLoc consists of two branches for subcellular localization prediction: one involving 10 location categories and the other involving binary localization prediction with 2 location categories. We assess the models' performance on both branches.

Fine-tuning Saprot with PDB structure. These tasks provide experimentally determined structures as training data. We evaluate Metal Ion Binding task [41] and a series of tasks from ProteinShake [43], including Structure Class Prediction, Structural Similarity Prediction and Binding Site Detection.

Fine-tuning Saprot without structure. Saprot is specifically developed to incorporate protein structures as valuable input. However, it can still work even in scenarios where structures are not provided. In

such cases, the 3Di token is masked within the SA sequence. We evaluate the Fluorescence prediction and Stability prediction tasks from the TAPE [49] benchmark, the AAV dataset from the FLIP [36] benchmark and β-lactamase landscape prediction from the PEER [57] benchmark. For a given wild type protein, its variants do not offer additional structures.

Dataset splits

In previous literature, a common practice was to partition datasets based on sequence identity. However, a recent benchmark study, ProteinShake [43], mentioned that protein structures exhibit greater conservation than protein sequences. They argue that dissimilar sequences in training and test sets may share similar structures, leading to data leakage issues. Consequently, we adopt a more strict data splitting approach proposed by ProteinShake and split the data based on structure similarity. For all datasets with structures, we employ a 70% structure similarity threshold for splitting. For those without provided structures, we still follow the original data splits.

Supervised fine-tuning hyper-parameters

We employed the AdamW [44] optimizer during fine-tuning, setting $\beta_1 = 0.9, \beta_2 = 0.98$, along with L_2 weight decay of 0.01. We use a batch size of 64 for all datasets. We empirically found that the optimal learning rate for most baselines are in the range of 1e-5 to 5e-5. For **Training with AF2 structure** and **Training with PDB structure**, the optimal learning rate was set to 5e-5, whereas for **Training without structure**, it was set to 1e-5. We fine-tuned all model parameters until convergence and selected the best checkpoints based on their performance on the validation set.

Protein inverse folding

Given the 3D coordinates of a protein backbone, protein inverse folding aims to predict residue sequences that fold into this shape. This exhibits promising applications in protein de novo design. While not particularly targeted on this task, Saprot is endowed to predict reliable residue sequences from the backbone due to the large-scale pre-training tailored for predicting masked residues based on sequence and structure contexts. To utilize Saprot for inverse folding, we first encode protein backbone into 3Di tokens $(f_1, f_2, ..., f_n)$. Then we add masks to all residue parts of SA tokens, forming a SA sequence $(\#f_1, \#f_2, ..., \#f_n)$, which serves as input for Saprot to predict residue distributions at all positions. In contrast to ProteinMPNN [37] that generates residues in the auto-regressive manner, i.e. generating next token conditioned on all previous outputs, Saprot is able to simultaneously predict all residues with only one forward propagation. As shown in Fig. 1e, in CATH [47] test set, Saprot predicts 16 times faster than ProteinMPNN, while achieves comparable prediction accuracy, which demonstrates its huge potential in protein de novo design.

Embedding visualization

We employ t-SNE [56] to visualize the protein representations generated by Saprot and ESM-2 in three datasets (Supplementary Fig. 3). We use the non-redundant version ($PIDE < 40\%$) of the SCOPe [35] database, visualizing all alpha and beta proteins. We additionally visualize proteins in Subcellular Localization and Binary Localization datasets from DeepLoc [34]. To generate protein representations, we adopt the average pooling on the embeddings of the last layer for both Saprot and ESM-2.

Adapter learning

The adapter learning technique originated in the field of NLP [40, 39, 59] and has recently garnered attention in protein-related research. However, these recent preliminary preprints primarily focused on its ability in model accuracy or efficient GPU memory consumption [54, 53, 52, 51]. Through the application of adapters, researchers have observed their ability to mitigate overfitting issues or integrate domain-specific knowledge into protein language models. In this paper, we innovatively integrate it with Google Colaboratory and endow it with deeper capabilities, namely, to promote model sharing and co-construction within the biologist community in the online SaprotHub. This represents a novel perspective on the application of adapter learning. In the broad AI community, there are various types of adapters exist in the broader field of deep learning, including the well-known adapters like Houlsby [40], Pfeiffer [48], Compacter [42], and LoRA [14]. We choose LoRA for its capacity to deliver comparable results with fewer parameters. By integrating learnable low-rank matrices into each Saprot block while freezing the backbone, LoRA enables parameter-efficient fine-tuning and model sharing for these downstream tasks.

Community-wide collaboration

As the SaprotHub community expands, an increasing number of Saprot models will become available for the same protein function. This paves the way for collaborative co-construction among biologists. To illustrate

this advantage (Results are provided in Supplementary Fig **??**), we employ intuitive approach. In regression tasks, the final prediction is obtained by calculating the mean value of outputs from all models. For classification tasks, we aggregate predicted categories from all models and select the category with the highest prediction as the final outcome. The lightweight adapter technique plays a vital role in model aggregation, addressing the impracticality of loading and sharing multiple large pre-trained models.

Furthermore, more advanced functionalities can be developed. For instance, by injecting multiple adapters into Saprot via SaprotHub, continual learning can be conducted to further enhance performance. Adapters eliminate the need to modify the weights of the original Saprot model, effectively mitigating the issue of catastrophic forgetting. This study introduces a pioneering concept of integrating community wisdom by utilizing shared adapters, thereby ensuring the privacy of biologists' proprietary data. This design enables the realization of the OPMC vision.

ColabSaprot notebooks

ColabSaprot comprises three key components. The first component focuses on training and inference, enabling researchers to swiftly set up the runtime environment, preprocess training data, and fine-tune Saprot with ease. We offer advanced hyper-parameter settings, allowing researchers to customize their training strategy based on specific requirements, such as batch size, learning rate, and training steps. After completing the training process, researchers can share the model weights on the SaprotHub community store, benefiting other biologists.

Furthermore, researchers without their own training data can directly access a range of fine-tuned Saprot models from SaprotHub and perform inference for desired protein functions. These existing models also support continual training with additional data, leading to improved prediction accuracy.

The second component focuses on the zero-shot mutational effect prediction task, enabling users to leverage Saprot for predicting fitness changes resulting from residue mutations. We provide multiple options for users to predict single mutations, multi-site mutations, or saturation mutations on given proteins.

The final component of Saprot offers support for the protein inverse folding task based on the backbone structure. Saprot achieves an impressive 16 times faster inference speed than ProteinMPNN while maintaining competitive prediction performance. This capability empowers users to conduct precise protein designs on a large scale.

Data availability

The pre-training dataset for training Saprot is available at https://huggingface.co/datasets/westlake-repl/AF2_UniRef50. Downstream task datasets are all stored at https://huggingface.co/SaProtHub.

Code and Service availability

Saprot is an open-sourced model with MIT license. The code is available at https://github.com/westlake-repl/Saprot. The code implementation of ColabSaprot notebook is available at https://github.com/westlake-repl/SaProtHub. ColabSaprot service is available at https://colab.research.google.com/drive/1nxYBed32rkAvoUpt8996Ge6D--Su4_oK. Fine-tuned models can be obtained through SaprotHub https://huggingface.co/SaProtHub.

References

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

[35] John-Marc Chandonia, Naomi K Fox, and Steven E Brenner. SCOPe: classification of large macromolecular structures in the structural classification of proteins—extended database. *Nucleic Acids Research*, 47(D1):D475–D481, 11 2018.

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

[37] Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

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

[39] Junchen Fu, Fajie Yuan, Yu Song, Zheng Yuan, Mingyue Cheng, Shenghui Cheng, Jiaqi Zhang, Jie Wang, and Yunzhu Pan. Exploring adapter-based transfer learning for recommender systems: Empirical studies and practical insights. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 208–217, 2024.

[40] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019.

[41] Mingyang Hu, Fajie Yuan, Kevin Yang, Fusong Ju, Jin Su, Hui Wang, Fei Yang, and Qiuyang Ding. Exploring evolution-aware & -free protein language models as protein function predictors. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 38873–38884. Curran Associates, Inc., 2022.

[42] Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, 34:1022–1035, 2021.

[43] Tim Kucera, Carlos Oliver, Dexiong Chen, and Karsten Borgwardt. Proteinshake: Building datasets and benchmarks for deep learning on protein structures. *Advances in Neural Information Processing Systems*, 36, 2024.

[44] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017.

[45] 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.

[46] Pascal Notin, Mafalda Dias, Jonathan Frazer, Javier Marchena Hurtado, Aidan N Gomez, Debora Marks, and Yarin Gal. Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval. In *International Conference on Machine Learning*, pages 16990–17017. PMLR, 2022.

[47] Christine A Orengo, Alex D Michie, Susan Jones, David T Jones, Mark B Swindells, and Janet M Thornton. Cath–a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1109, 1997.

[48] Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. Adapterhub: A framework for adapting transformers. *arXiv preprint arXiv:2007.07779*, 2020.

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

[50] 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.

[51] Robert Schmirler, Michael Heinzinger, and Burkhard Rost. Fine-tuning protein language models boosts predictions across diverse tasks. *bioRxiv*, pages 2023–12, 2023.

[52] Amelie Schreiber. Esmbind and qbind: Lora, qlora, and esm-2 for predicting binding sites and post translational modification. *bioRxiv*, pages 2023–11, 2023.

[53] Samuel Sledzieski, Meghana Kshirsagar, Minkyung Baek, Bonnie Berger, Rahul Dodhia, and Juan Lavista Ferres. Democratizing protein language models with parameter-efficient fine-tuning. *bioRxiv*, 2023.

[54] Samuel Sledzieski, Meghana Kshirsagar, Bonnie Berger, Rahul Dodhia, and Juan Lavista Ferres. Parameter-efficient fine-tuning of protein language models improves prediction of protein-protein interactions.

[55] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[56] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[57] Minghao Xu, Zuobai Zhang, Jiarui Lu, Zhaocheng Zhu, Yangtian Zhang, Ma Chang, Runcheng Liu, and Jian Tang. Peer: a comprehensive and multi-task benchmark for protein sequence understanding. *Advances in Neural Information Processing Systems*, 35:35156–35173, 2022.

[58] Kevin K Yang, Niccolò Zanichelli, and Hugh Yeh. Masked inverse folding with sequence transfer for protein representation learning. *Protein Engineering, Design and Selection*, 36:gzad015, 2023.

[59] Fajie Yuan, Xiangnan He, Alexandros Karatzoglou, and Liguang Zhang. Parameter-efficient transfer from sequential behaviors for user modeling and recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 1469–1478, 2020.

[60] Zuobai Zhang, Minghao Xu, Arian Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. *arXiv preprint arXiv:2203.06125*, 2022.

Supplementary information

OPMC

The Open Protein Modeling Consortium (OPMC) is a collaborative initiative aimed at unifying the efforts of the protein research community. Its mission is to facilitate the sharing and co-construction of resources, particularly individually trained models, and to advance protein modeling through collective contributions. OPMC provides a platform that supports a wide range of protein function predictions, striving to make advanced protein modeling accessible to researchers, regardless of their machine learning expertise.

Here are some Q&A related to OPMC that we have gathered from our steering committee members and other researchers:https://github.com/westlake-repl/SaprotHub/blob/main/README.md

Be an OPMC member

The complete list of OPMC authors will be finalized in the last version of the paper. OPMC encompasses two membership categories: the steering committee and regular members.

Steering Committee: The steering committee comprises esteemed researchers with doctoral degrees who assume advisory roles within OPMC. Their primary responsibilities include:

- Providing constructive suggestions for OPMC and SaprotHub.
- Supporting academic activities such as organizing workshops, tutorials, and participating in peer review processes.
- Encouraging direct involvement of team members in the development of SaprotHub or OPMC.
- Participating in the review process for new members joining OPMC.
- Contributing high-quality open-source datasets and models to enhance OPMC's resources.
- Making other contributions or providing academic services that drive the progress of OPMC.

Regular Members: Regular members actively contribute to OPMC through practical means. This category includes independent researchers and professionals who:

- Participate in research, development, and collaborative efforts to advance OPMC's goals and objectives.
- Create innovative adapters for SaprotHub, trained on either private or publicly available data.
- Contributing more high-quality datasets to enrich the available resources.
- Aid in the development and administration of SaprotHub or ColabSaprot.

Membership Eligibility: We enthusiastically welcome individuals who make substantial contributions to OPMC and SaprotHub. The eligibility of regular members will be determined by the steering committee. Detailed requirements for joining will be provided at a later stage.

If you are interested in becoming a member of OPMC, please reach out to our corresponding author. Note that all OPMC members will be included as authors in the final version of this paper. The final list of members is expected to be finalized within the next 4 to 12 months.

Dataset	Evaluation Metric	Train	Valid	Test
Training with AF2 structure				
Thermostability [36]	Spearman's ρ	5310	706	706
Subcellular localization [34]	Acc	10414	1368	1368
Binary localization [34]	Acc	6707	698	807
Training with PDB structure				
Structural similarity [43]	Spearman's ρ	300699	4559	4850
Structure class [43]	ACC	7990	955	1005
Metal ion binding [41]	ACC	5797	719	719
Binding site detection [43]	Mcc	2368	442	464
Training without structure				
Fluorescence [49]	Spearman's ρ	20963	5235	25517
Stability [49]	Spearman's ρ	53614	2512	12851
AAV [36]	Spearman's ρ	22246	2462	50432
β-lactamase [57]	Spearman's ρ	4158	520	520

Supplementary Table 1: Downstream dataset descriptions after all data pre-processing. All datasets that require structures as input are split based on structure similarity with 70 as threshold. For **Training with AF2 structure**, proteins with structures not available in AlphaFoldDB were excluded from both training and testing evaluations.

Model	ProteinGym	Mega-scale	ClinVar
	Spearman's ρ	Spearman's ρ	AUC
ESM-2 (650M) [20] (Science, 2023)	0.414	0.478	0.862
ESM-2 (15B) [20] (Science, 2023)	0.401	0.467	0.843
ProtBert [5] (TPAMI, 2021)	0.350	0.341	0.862
ESM-1v [45] (NeurIPS, 2021)	0.374	0.374	0.891
Tranception L [46] (ICML, 2022)	0.374	0.307	0.845
MSA Transformer [50] (ICML, 2021)	0.421	0.423	0.854
EVE [7] (Nature, 2021)	0.433	0.260	0.878
Saprot	**0.457**	**0.574**	**0.909**

Supplementary Table 2: More baseline results on zero-shot mutational effect prediction.

Task	Category	Metric	ESM-2	ProtBert	MIF-ST [58]	GearNet[60]	**Saprot**
Thermostability	Training w/ AF2 structure	Spearman's ρ	0.673	0.621	0.678	0.454	0.697
Subcellular localization		Acc%	84.67	78.65	82.46	75.73	86.18
Binary localization		Acc%	93.56	93.31	92.08	90.72	94.67
Structure class	Training w/ PDB structure	Acc%	79.90	71.57	71.63	74.52	80.80
Structural similarity		Spearman's ρ	0.677	0.601	0.555	0.101	0.695
Binding site detection		Mcc	0.540	0.459	0.504	0.415	0.602
Metal Ion binding		Acc%	74.97	68.19	69.58	67.61	79.97
Fluorescence	Training w/o structure	Spearman's ρ	0.698	0.698	-	-	0.699
Stability		Spearman's ρ	0.780	0.753	-	-	0.779
β-lactamase		Spearman's ρ	0.904	0.881	-	-	0.890
AAV		Spearman's ρ	0.813	0.783	-	-	0.829

Supplementary Table 3: More baseline results on supervised learning tasks. MIF-ST and GearNet are not applicable when protein structures are not available as input.

Task / Dataset	**Metric**	ProstT5	**Saprot**
ClinVar	AUC	0.620	**0.909**
Mega-scale	Spearman's ρ	0.194	**0.574**
Subcellular localization	Acc%	83.26	**86.18**
Binary localization	Acc%	94.11	**94.67**
Protein inverse folding	pLDDT	76.24	**77.96**
	RMSD	2.28	**2.07**
	TM-score	0.78	**0.82**
	Sequence recovery	33.72	**36.15**

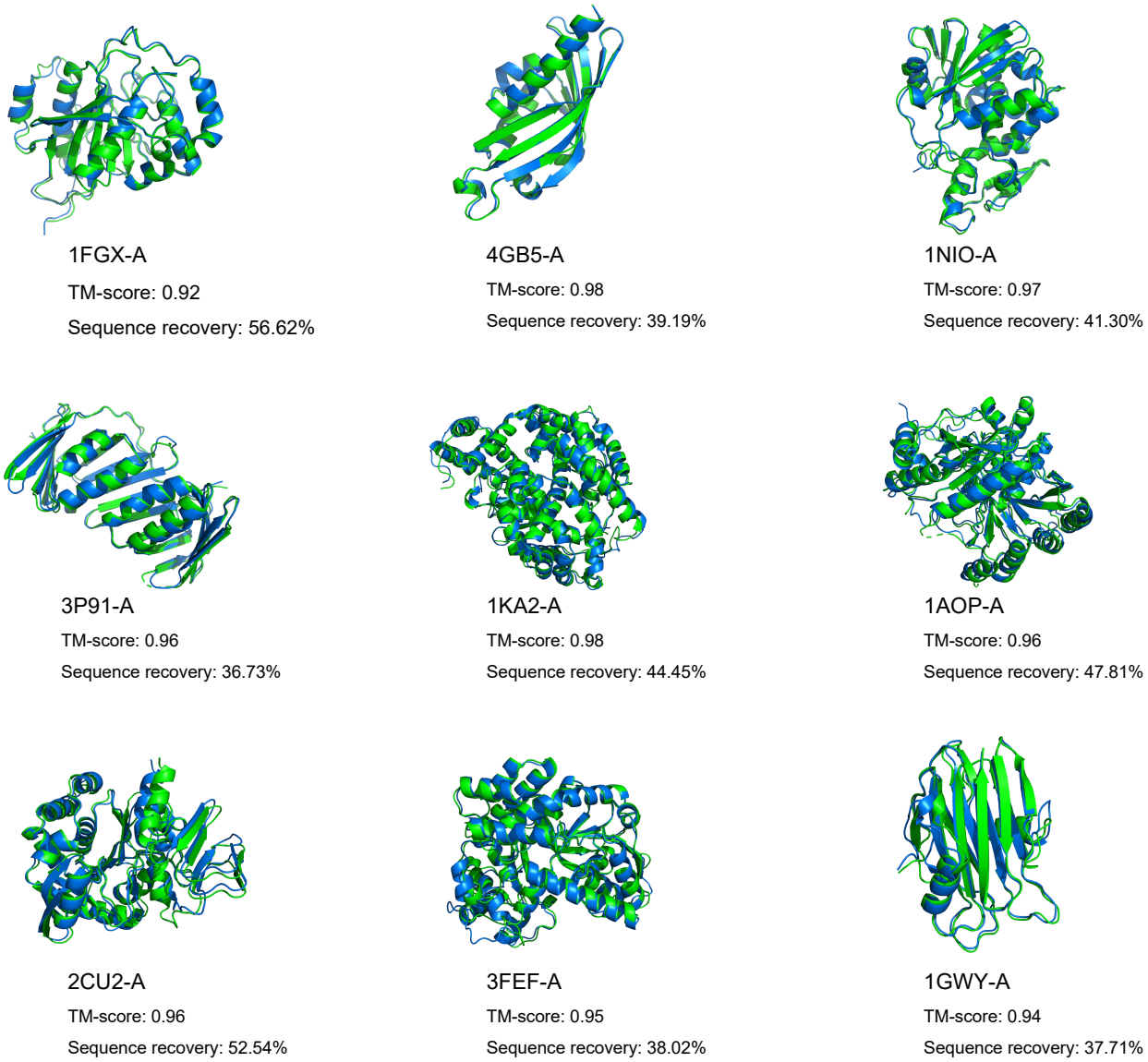
Supplementary Table 4: Comparison between ProstT5 and SaProt in several tasks. ProstT5 is a seq2seq T5 model where the input, if it is an amino acid sequence, corresponds to the output of the 3Di sequence, and vice versa. In general, Saprot is comparable to ProstT5 in the protein inverse folding task, but substantially outperform it on other tasks, e.g., zero-shot mutational effect prediction and supervised learning tasks.

Benchmark Scores

https://proteingym.org/benchmarks

Rank▲	Model name	Model type	Avg. Spearman
1	SaProt (650M)	Protein language model	0.457
2	TranceptEVE L	Hybrid - Alignment & PLM	0.456
3	TranceptEVE M	Hybrid - Alignment & PLM	0.455
4	GEMME	Alignment-based model	0.455
5	TranceptEVE S	Hybrid - Alignment & PLM	0.452
6	ProtSSN (ensemble)	Hybrid - Structure & PLM	0.449
7	ProtSSN (k=20 h=1280)	Hybrid - Structure & PLM	0.442
8	ProtSSN (k=20 h=512)	Hybrid - Structure & PLM	0.441
9	ProtSSN (k=20 h=768)	Hybrid - Structure & PLM	0.44
10	EVE (ensemble)	Alignment-based model	0.439
11	ProtSSN (k=30 h=512)	Hybrid - Structure & PLM	0.438
12	ProtSSN (k=30 h=768)	Hybrid - Structure & PLM	0.438
13	ProtSSN (k=30 h=1280)	Hybrid - Structure & PLM	0.438
14	VESPA	Protein language model	0.436
15	Tranception L	Hybrid - Alignment & PLM	0.434
16	MSA Transformer (ensemble)	Hybrid - Alignment & PLM	0.434
17	ProtSSN (k=10 h=1280)	Hybrid - Structure & PLM	0.433
18	EVE (single)	Alignment-based model	0.433
19	ProtSSN (k=10 h=512)	Hybrid - Structure & PLM	0.43
20	Tranception M	Hybrid - Alignment & PLM	0.427
21	ProtSSN (k=10 h=768)	Hybrid - Structure & PLM	0.423
22	ESM-IF1	Inverse folding model	0.422
23	MSA Transformer (single)	Hybrid - Alignment & PLM	0.421
24	DeepSequence (ensemble)	Alignment-based model	0.419
25	Tranception S	Hybrid - Alignment & PLM	0.418
26	ESM2 (650M)	Protein language model	0.414
27	DeepSequence (single)	Alignment-based model	0.407
28	ESM-1v (ensemble)	Protein language model	0.407
29	SaProt (35M)	Protein language model	0.406
30	ESM2 (3B)	Protein language model	0.406
31	ESM2 (15B)	Protein language model	0.401
32	MIF-ST	Hybrid - Structure & PLM	0.4
33	EVmutation	Alignment-based model	0.395
34	ESM-1b	Protein language model	0.394
35	VESPAl	Protein language model	0.394
36	Progen2 XL	Protein language model	0.391
37	ESM2 (150M)	Protein language model	0.387
38	MIF	Inverse folding model	0.383
39	Progen2 L	Protein language model	0.38
40	Progen2 M	Protein language model	0.379

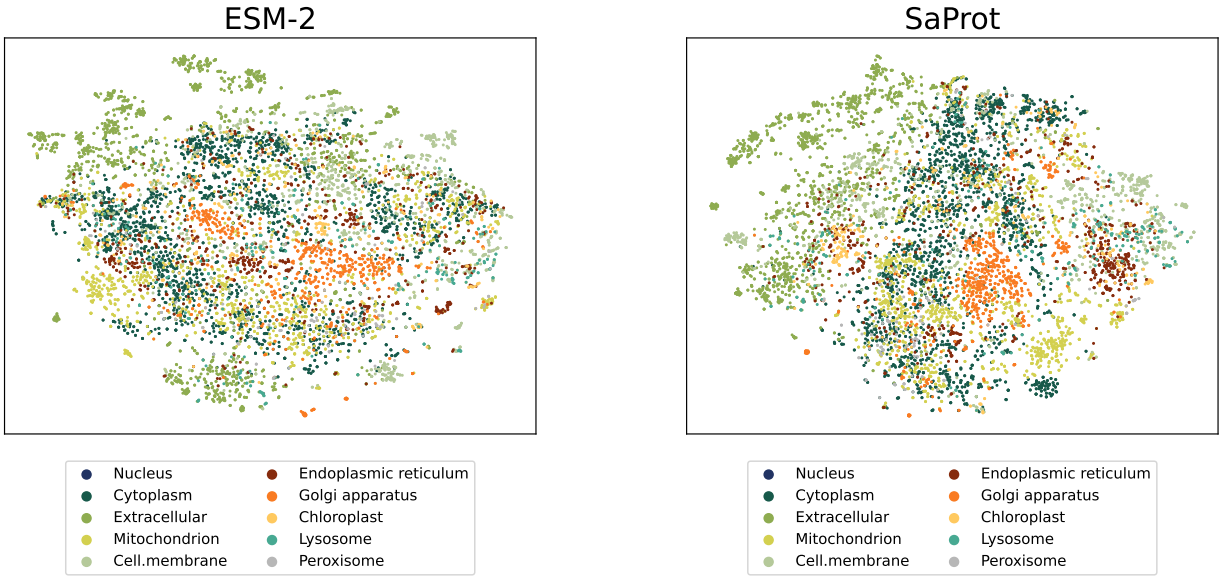
Supplementary Fig. 1: Saprot (checkpoint: https://huggingface.co/westlake-repl/SaProt_650M_AF2) ranked first on the public ProteinGym leaderboard until May 2024, surpassing over 60 well-known methods.



Supplementary Fig. 2: Case studies of protein inverse folding using Saprot. The blue-colored protein structures are predicted using ESMFold, with residue sequences generated by Saprot, and the green-colored structures are the experimentally determined structures. Given the backbone, Saprot is capable of predicting diverse sequences with few residue overlap while maintaining almost the same 3D structure.

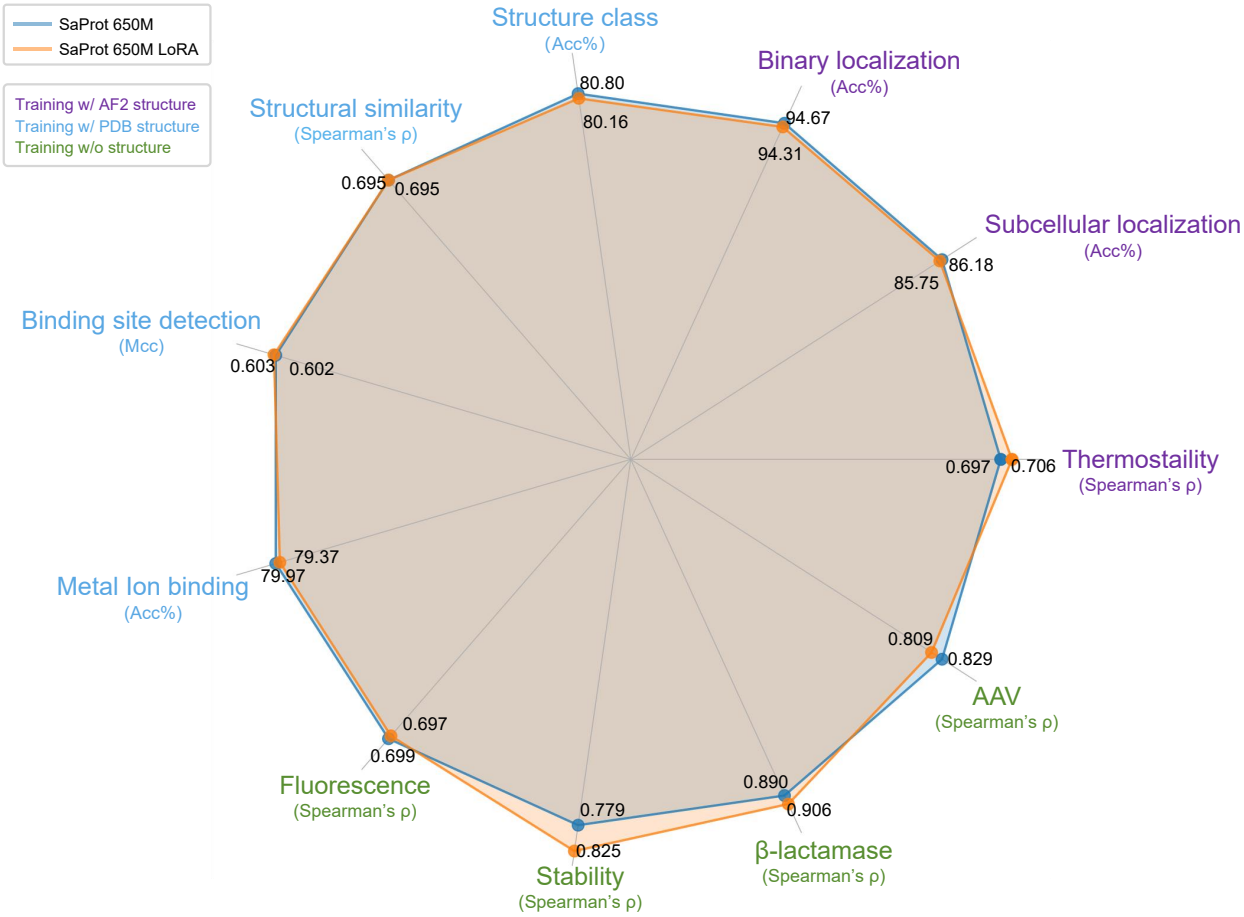
(a) Embedding visualizations of ESM-2 650M version and Saprot 650M version on SCOPe database. We use the non-redundant version ($PIDE < 40\%$) of the SCOPe database.

(b) Embedding visualizations of ESM-2 and SaProt on binary localization dataset.



(c) Embedding visualizations of ESM-2 and SaProt on subcellular localization dataset.

Supplementary Fig. 3: Embedding visualization



Supplementary Fig. 4: Comparison between full model fine-tuning and parameter efficient fine-tuning. While significantly reducing trainable parameters, fine-tuning Saprot using adapter module achieves competitive performance on all experimental settings, which demonstrates its applicability on various biological tasks.