ONE PROTEIN IS ALL YOU NEED

Anonymous authors
Paper under double-blind review

ABSTRACT

Generalization beyond training data remains a central challenge in machine learning for biology. A common way to enhance generalization is self-supervised pre-training on large datasets. However, aiming to perform well on all possible proteins can limit a model's capacity to excel on any specific one, whereas practitioners typically need accurate predictions for individual proteins they study, often not covered in training data. To address this limitation, we propose a method that enables self-supervised customization of protein language models to one target protein at a time, on the fly, and without assuming any additional data. We show that our Protein Test-Time Training (ProteinTTT) method consistently enhances generalization across different models, their sizes, and datasets. ProteinTTT improves structure prediction for challenging targets, achieves new state-of-the-art results on protein fitness prediction, and enhances function prediction on two tasks. We also demonstrate ProteinTTT on two challenging case studies. We show that customization via ProteinTTT enables more accurate antibody–antigen loop modeling and improves 17% of structures in the Big Fantastic Virus Database, delivering improved predictions where general-purpose AlphaFold2 and ESMFold struggle.

1 INTRODUCTION

A comprehensive understanding of protein structure, function, and fitness is essential for advancing research in the life sciences (Subramaniam & Kleywegt, 2022; Tyers & Mann, 2003; Papkou et al., 2023). While machine learning models have shown remarkable potential in protein research, they are typically optimized for achieving the best average performance across large datasets (Jumper et al., 2021; Watson et al., 2023; Kouba et al., 2023). However, biologists often focus their research on individual proteins or protein complexes involved in, for example, metabolic disorders (Ashcroft et al., 2023; Gunn & Neher, 2023), oncogenic signaling (Hoxhaj & Manning, 2020; Keckesova et al., 2017), neurodegeneration (Gulen et al., 2023; oh Seo et al., 2023), and other biological phenomena (Gu et al., 2022). In these scenarios, detailed insights into a single protein can lead to significant scientific advances.

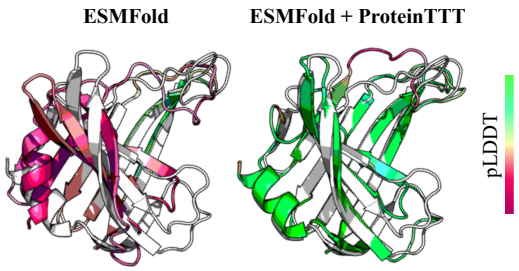


Figure 1: **Example of protein structure prediction after single-protein model customization via ProteinTTT.** ESMFold poorly predicts the structure of the CASP14 target T1074 (white) because the underlying language model ESM2 poorly fits the sequence, as indicated by the high perplexity (left and Fig. 2E in Lin et al. (2023)). Self-supervised test-time customization of ESM2 to the single sequence of T1074 reduces the perplexity, resulting in improved structure prediction (right).

However, general machine learning models for proteins often struggle to generalize to practically interesting individual cases due to data scarcity (Bushuiev et al., 2023; Chen & Gong, 2022) and distribution shifts (Škrinjar et al., 2025; Tagasovska et al., 2024; Feng et al., 2024). Bridging the gap between broad, dataset-wide optimization and precision needed to study single proteins of practical interest remains a key challenge in integrating machine learning into biological research (Sapoval et al., 2022). This challenge is particularly acute in computational biology, where accurate predictions for individual proteins are essential to guide resource-intensive wet-lab experiments, in contrast to

domains such as natural language processing or computer vision, where models are typically expected to flexibly handle diverse prompts from many users in real time (Brown, 2020; Ramesh et al., 2021).

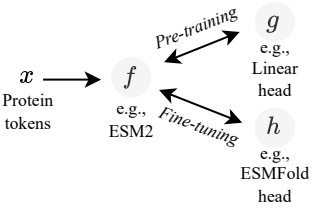
To address this challenge, we propose a test-time approach for generalization to one protein at a time, effectively enabling more accurate predictions for individual targets, particularly those poorly represented in training data. Our Protein Test-Time Training (ProteinTTT) method customizes protein language models (PLMs) to individual proteins on the fly and without assuming additional data. Our approach is based on a simple yet powerful premise: if a language model is less perplexed (surprised) by a protein sequence–or if it "understands" its unique patterns better–it will generate a more accurate representation for predicting its structure and function. Given a model pre-trained via masked language modeling, our method effectively minimizes perplexity on a target protein or its multiple sequence alignment (MSA) through self-supervised customization, improving downstream performance without updating the downstream task head. The widespread use of masked modeling as a pre-training paradigm makes ProteinTTT broadly applicable in computational biology.

In summary, this work demonstrates the surprising effectiveness of protein model customization and lays the foundation for exploring other test-time strategies and broader biological applications. The key contributions are: **(1)** We introduce ProteinTTT, to the best of our knowledge the first customization method in machine learning for proteins. We provide a user-friendly and easily extensible implementation[1] and provide insights into the effectiveness of protein model customization by linking it to perplexity minimization. **(2)** We empirically validate ProteinTTT, showing improvements in protein structure prediction capabilities of well-established models, achieving state-of-the-art results in protein fitness prediction, and enhancing protein function prediction on terpene synthase substrate classification and protein localization prediction. **(3)** We demonstrate the practical utility of focusing on one protein at a time through two challenging case studies. ProteinTTT enables more accurate prediction of antibody–antigen loops and improves 17% of structures in the Big Fantastic Virus Database, delivering accurate predictions where general-purpose AlphaFold2 and ESMFold struggle.

2 Background and related work

The broad adoption of Y-shaped architectures relying on masked modeling enables the development of a general method for customizing protein models at test time via masking-based self-supervision.

The Y-shaped paradigm of learning. In machine learning applied to proteins, architectures often follow a Y-shaped paradigm (Gandelsman et al., 2022), consisting of a backbone feature extractor f operating on protein tokens x, a self-supervised head g, and an alternative fine-tuning head h. During training, $g \circ f$ is first pre-trained, and the pre-trained backbone f is then reused to fine-tune $h \circ f$ toward a downstream task. Here, $\circ$ denotes a composition of two machine learning modules (e.g., g is applied on top of f in $g \circ f$). At test time, the final model $h \circ f$ is fixed. Generalization is achieved by leveraging the rich knowledge encoded in the backbone f and the task-specific priors embedded in the fine-tuning head h. This paradigm enables overcoming data scarcity during fine-tuning and underlies breakthrough approaches in protein structure prediction (Lin et al., 2023), protein design (Watson et al., 2023), protein function prediction (Yu et al., 2023), and other tasks (Hayes et al., 2024).



The backbone f is typically a large neural network pre-trained in a self-supervised way on a large dataset using a smaller pre-training projection head g (Hayes et al., 2024). The fine-tuning head h, however, depends on the application. In some cases, h is a large neural network, repurposing the pre-trained model entirely (Watson et al., 2023; Lin et al., 2023); in others, h is a minimal projection with few parameters (Cheng et al., 2023), or even without any parameters at all (i.e., a zero-shot setup; Meier et al. (2021); Dutton et al. (2024)). The fine-tuning head h can also be a machine learning algorithm other than a neural network (Samusevich et al., 2024).

Masked modeling. While the objective of fine-tuning $h \circ f$ is determined by the downstream application, the choice of pre-training objective for $g \circ f$ is less straightforward. Nevertheless, the

[1] https://anonymous.4open.science/r/ProteinTTT-anonymous-F585

dominant paradigm for protein pre-training is masked modeling, which optimizes model weights to reconstruct missing protein parts. This objective has proven effective across diverse tasks, including structure (Lin et al., 2023; Jumper et al., 2021), fitness (Meier et al., 2021; Su et al., 2023), and function prediction (Yu et al., 2023; Samusevich et al., 2024), as well as protein design (Hayes et al., 2024), and has been successfully applied to various protein representations such as sequences (Hayes et al., 2024), graphs (Dieckhaus et al., 2024; Bushuiev et al., 2023), and voxels (Diaz et al., 2023).

Model customization. Several studies have shown that machine learning models for proteins benefit from being fine-tuned on protein-specific (Notin et al., 2024; Kirjner et al., 2023; Rao et al., 2019) or protein family-specific (Sevgen et al., 2023; Samusevich et al., 2024) data. However, collecting additional data may be resource-intensive, and for many targets, relevant datasets or proteins may be limited or not available (Durairaj et al., 2023; Kim et al., 2025). In this paper, we propose a versatile method enabling customizing PLMs for a single target protein or its MSA in a self-supervised manner, on the fly, and without assuming any additional data. Customization methods have been developed in computer vision (Chi et al., 2024; Wang et al., 2023; Xiao et al., 2022; Karani et al., 2021) and natural language processing (Hardt & Sun, 2023; Ben-David et al., 2022; Banerjee et al., 2021). The paradigm of test-time training (TTT), developed to mitigate distribution shifts in computer vision applications (Gandelsman et al., 2022; Sun et al., 2020), is the main inspiration for our work. We demonstrate that customization via test-time training enhances the accuracy of PLMs across a wide range of downstream tasks even without the presence of explicit distribution shifts.

3 Protein model customization with ProteinTTT

In this section, we describe the proposed Protein Test-Time Training (ProteinTTT) approach (Section 3.1), followed by its applications to a range of well-established models and datasets (Section 3.2).

3.1 Self-supervised customization to a target protein

At test time, we assume a Y-shaped model with a backbone f that has been pre-trained via the self-supervised track $g \circ f$, followed by task-specific fine-tuning through the supervised track $h \circ f$. The goal of customization with ProteinTTT is to adapt the backbone f to a single protein x before making a prediction on a downstream task via the supervised track $h \circ f$. To achieve this, we customize the backbone f to the single example x:

$$\text{ProteinTTT} : (h \circ f(\cdot;\theta_0), x) \mapsto h \circ f(\cdot;\theta_x) \tag{1}$$

where θ_0 denotes pre-trained parameters and θ_x parameters optimized for the target protein x using the self-supervised track $g \circ f$, while the supervised head h remains frozen. Figure 2a illustrates our self-supervised customization approach, which is summarized in the following sections. Section C in the Appendix describes the extension of our approach to customization toward MSA sequences of a protein of interest, rather than its single sequence.

Customization training objective. We customize $g \circ f$ to a single target protein sequence x via minimizing the masked language modeling objective (Devlin, 2018; Rives et al., 2021):

$$\mathcal{L}(x;\theta) = \mathbb{E}_{M \sim p_{\text{mask}}(M)} \left[\sum_{i \in M} -\log p(x_i | x_{\setminus M}; \theta) \right], \tag{2}$$

where x denotes a sequence of protein tokens (typically amino acid types), and $\mathbb{E}_M$ represents the expectation over randomly sampled masking positions M. The objective function $\mathcal{L}(x;\theta)$ maximizes the log-probabilities $\log p(x_i | x_{\setminus M}; \theta) \doteq g(f(x_{\setminus M};\theta))_i$ of the true (i.e., wild-type) tokens x_i at the masked positions $i \in M$ in the partially masked sequence $x_{\setminus M}$, where θ denotes the parameters of the backbone f, and g is the masked language modeling head. Please note that here we focus on bi-directional masked modeling models, which employ random masking, but the method can be easily extended to models employing autoregressive masking.

To ensure consistency between the customization and pre-training, ProteinTTT adopts the same masking and data preprocessing strategies used during pre-training. Specifically, $p_{\text{mask}}(M)$ can follow different distributions, such as sampling a fixed proportion (e.g., 15%) of random amino acid

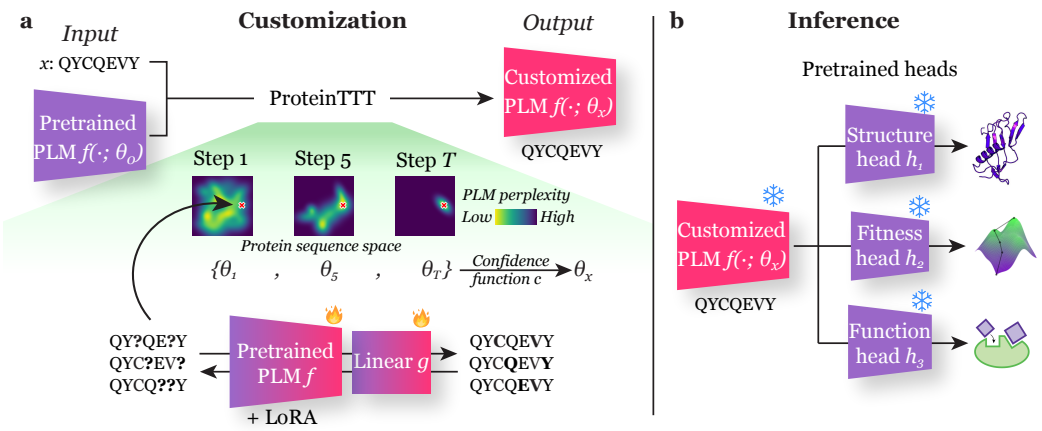


Figure 2: **Overview of protein language model (PLM) customization with ProteinTTT. (a)** Given a protein sequence of interest x and a pretrained PLM $f(\cdot;\theta_0)$, ProteinTTT yields a customized version of the PLM $f(\cdot;\theta_x)$ for that sequence. Customization is achieved by fine-tuning (fire icon) the pretrained parameters θ_0 via masked language modeling solely on the input sequence for T steps, selecting the optimal parameters θ_x using a confidence function c. This procedure adapts the model specifically to the input sequence, improving its internal representation as measured by model perplexity. **(b)** Once customized, the PLM can be used with pretrained task-specific heads, such as structure, fitness, or function prediction modules, h_1, h_2, and h_3, respectively, without modifying their parameters (snowflake icon). For example, the ESM2 PLM can be customized and then used with the pretrained ESMFold structure prediction head without modifying its 1.4-billion task-specific parameters, resulting in improved structure prediction for the given sequence (e.g., Figure 1).

tokens (Lin et al., 2023), or dynamically varying the number of sampled tokens based on another distribution (e.g., a beta distribution; Hayes et al. (2024)). During the customization, we replicate the masking distribution used during the pre-training. We also replicate other pre-training practices, such as replacing 10% of masked tokens with random tokens and another 10% with the original tokens (Devlin, 2018; Lin et al., 2023; Su et al., 2023) or cropping sequences to random 1024-token fragments (Lin et al., 2023; Su et al., 2023).

Optimization. Since customization with ProteinTTT does not assume more than a single sequence available, early stopping on validation data is not feasible. To address this, we first fine-tune the pretrained parameters θ_0 of a backbone f for a fixed number of steps T, yielding a set of parameters $\Theta = \{\theta_0, \theta_1, \ldots, \theta_T\}$. The final customized parameters θ^* are then selected as $\arg\max_{\theta\in\Theta} c(h(f(x;\theta)))$ where c is a confidence function. If c is not available, we set $\theta^* = \theta_T$. Section G.2 discusses how using pLDDT as the confidence function c for protein structure prediction makes ProteinTTT robust to hyperparameter selection and how the number of steps T can be kept fixed (e.g., $T = 30$) while optimizing learning rate and batch size effectively. Before customizing for the next target protein, the parameters are reset back to θ_0.

To make ProteinTTT easily applicable to large-scale models (e.g., the 3B-parameter ESM2 backbone), we leverage low-rank adaptation (LoRA; Hu et al. (2021)) and gradient accumulation during customization. Additionally, to improve the stability and predictability of customization, we use stochastic gradient descent (SGD; Ruder (2016)) instead of the commonly used Adam optimizer (Kingma & Ba, 2015), following (Gandelsman et al., 2022). Further details are provided in Section E.

3.2 Inference on downstream tasks

Once the backbone f is adapted to a target protein via self-supervised customization, it can be used in conjunction with a pre-trained downstream head h, as $h \circ f$. The key idea of customization with ProteinTTT is not to update the head h, but instead to leverage improved representations from f (Figure 2b). Section B provides a justification for why these customized representations generally enhance performance on downstream tasks.

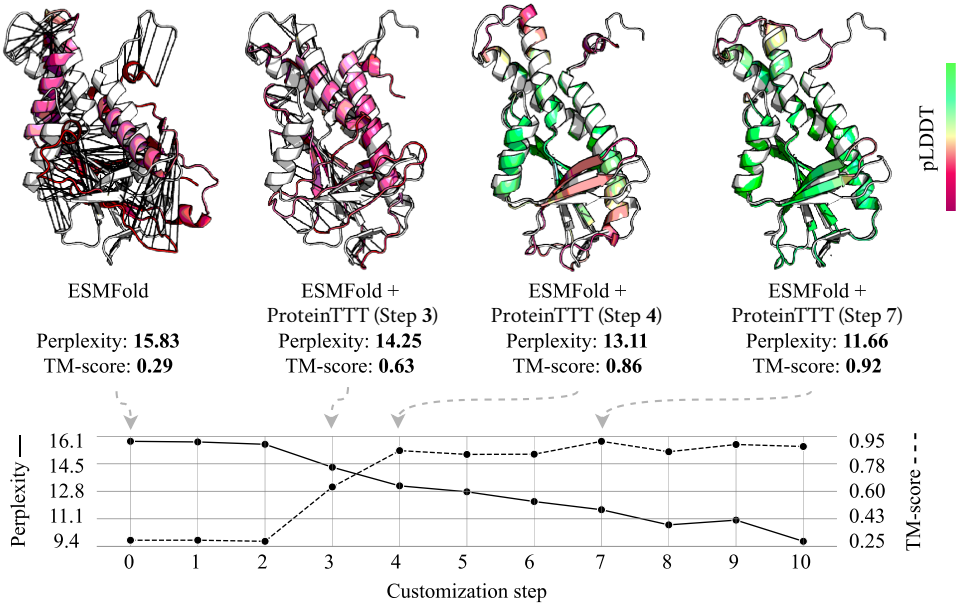


Figure 3: **Customization with ProteinTTT improves protein structure prediction by reducing protein sequence perplexity**. ESMFold fails to predict the structure of chain B from PDB entry 7EBL in the CAMEO validation set, as shown at customization step 0, where the perplexity is high and the TM-score is low. By applying customization with ProteinTTT for the single target sequence, the model iteratively improves the structure prediction quality, as demonstrated by the increasing TM-score, associated with reduced perplexity. At customization step 7, the predicted structure achieves the highest TM-score, as well as the highest predicted confidence metric pLDDT, enabling the selection of this step as the final prediction by the customized ESMFold + ProteinTTT.

Since Y-shaped architectures are prevalent in protein machine learning, ProteinTTT can be straightforwardly applied to numerous tasks. In this work, we consider three standard problems: protein structure, fitness, and function prediction, and apply our method to corresponding well-established models. For structure prediction, we apply ProteinTTT to ESMFold (Figure 3, Lin et al. (2023), HelixFold-Single (Fang et al., 2023), and ESM3 (Hayes et al., 2024); for fitness prediction, we use ESM2 (Lin et al., 2023), SaProt (Su et al., 2023), ProSST (Li et al., 2024), and MSA Transformer (Rao et al., 2021); and for function prediction, we apply ProteinTTT to ESM-1v-based (Meier et al., 2021) TerpeneMiner (Samusevich et al., 2024) and ESM-1b-based (Rives et al., 2021) Light attention (Stärk et al., 2021).

In all models we consider, f is a Transformer encoder that takes protein tokens as input, and g is a masked language modeling head (a layer mapping token embeddings to amino acid types). The downstream task heads h vary strongly across tasks. For structure prediction, h is a protein structure predictor: in ESMFold and HelixFold-Single, it is an AlphaFold2-inspired module (Jumper et al., 2021), while in ESM3, it is a VQ-VAE structure decoder (Razavi et al., 2019). For fitness prediction, h outputs a single score per sequence; ESM2, SaProt, and ProSST perform zero-shot inference using $h \circ f$ via log odds from g, with h functioning as a simple adaptation of g without introducing extra parameters. The function predictors are classification models: in TerpeneMiner (Samusevich et al., 2024), h is a random forest that outputs substrate probabilities, and in Light attention (Stärk et al., 2021), h is a light attention module predicting protein localization classes within a cell.

4 Experiments

In this section, we evaluate ProteinTTT on three well-established downstream tasks in protein machine learning: structure (Section 4.1), fitness (Section 4.2), and function (Section 4.3) prediction.

Table 1: **Customization with ProteinTTT improves protein structure prediction.** The metrics are averaged across 18 ESMFold low-confidence targets in the CAMEO test set, and standard deviations correspond to 5 random seeds. CoT and MP stand for the chain of thought and masked prediction baselines.

Method	TM-score ↑	LDDT ↑
ESM3 (Hayes et al., 2024)	0.3480 ± 0.0057	0.3723 ± 0.0055
ESM3 + CoT (Hayes et al., 2024)	0.3677 ± 0.0088	0.3835 ± 0.0024
ESM3 + ProteinTTT (Ours)	**0.3954 ± 0.0067**	**0.4214 ± 0.0054**
HelixFold-Single (Fang et al., 2023)	0.4709	0.4758
HelixFold-Single + ProteinTTT (Ours)	**0.4839 ± 0.0045**	**0.4840 ± 0.0061**
ESMFold (Lin et al., 2023)	0.4649	0.5194
ESMFold + MP (Lin et al., 2023)	0.4862 ± 0.0043	0.5375 ± 0.0070
ESMFold + ProteinTTT (Ours)	**0.5047 ± 0.0132**	**0.5478 ± 0.0058**

4.1 Protein structure prediction

Protein structure prediction is the task of predicting 3D coordinates of protein atoms given the amino acid sequence. It is arguably one of the best-established problems in computational biology (Jumper et al., 2021; Lin et al., 2023; Abramson et al., 2024).

Evaluation setup. To evaluate the performance of ProteinTTT, we employ CAMEO, a standard benchmark for protein folding. We use the validation and test folds from Lin et al. (2023), focusing only on targets with low-confidence predictions from the base ESMFold, as determined by pLDDT and perplexity (Section E.1). We use the standard TM-score (Zhang & Skolnick, 2004) and LDDT (Mariani et al., 2013) metrics to evaluate global and local structure prediction quality, respectively.

As baseline methods, we use techniques alternative to ProteinTTT for improving the performance of the pre-trained base models. In particular, the ESMFold paper proposes randomly masking 15% of amino acids in a protein sequence before inference, allowing for sampling multiple protein structure predictions from the regression ESMFold model (Lin et al., 2023). For each sequence, we sample a number of predictions equal to the total number of ProteinTTT steps and refer to this baseline as ESMFold + MP (Masked Prediction). As a baseline for ESM3, we use chain-of-thought iterative decoding, referred to as ESM3 + CoT, proposed in the ESM3 paper (Hayes et al., 2024).

Results. Customization with ProteinTTT consistently improves the performance of all the tested methods, ESMFold, HelixFold-Single, and ESM3, outperforming the masked prediction (ESMFold + MP) and chain-of-thought (ESM3 + CoT) baselines, as shown in Table 1. Among the 18 challenging CAMEO test proteins, ProteinTTT significantly improved the prediction of 7, 5, and 6 structures from ESMFold, HelixFold-Single, and ESM3, respectively, while only slightly disrupting the prediction of 2, 1, and 1 structures, respectively (Figure A6). Most notably, ProteinTTT enables accurate structure prediction for targets that are poorly predicted with the original models. For instance, Figure 1 presents a strongly improved structure predicted using ESMFold + ProteinTTT for the target that was part of the CASP14 competition and shown as an unsuccessful case in the original ESMFold publication (Lin et al. (2023), Fig. 2E). Another example is shown in Figure 3, where ProteinTTT refined the structure prediction from a low-quality prediction (TM-score $= 0.29$) to a nearly perfectly folded protein (TM-score $= 0.92$). Figure A4 shows that ESMFold + ProteinTTT maintains computational efficiency of ESMFold, being an order of magnitude faster than AlphaFold2. Figure A11 additionally demonstrates the robustness of ESM3 + ProteinTTT to the choice of hyperparameters.

4.2 Protein Fitness Prediction

The task of protein fitness prediction is to accurately order mutations of a protein based on their disruptive/favorable effects on protein functioning.

Evaluation Setup. We evaluate the models using ProteinGym, the state-of-the-art fitness prediction benchmark (Notin et al., 2024), focusing on its well-established zero-shot setup. Since the zero-shot setup only provides a test set without any data split, we also validate ProteinTTT on independent

Table 2: **Customization with ProteinTTT improves protein fitness prediction.** The right section of the table presents performance averaged across individual proteins and then across different protein phenotypes, as classified in the ProteinGym benchmark (Notin et al., 2024). The middle column shows the final performance, averaged across all five phenotype classes. In total, ProteinGym contains 2.5 million mutations across 217 proteins. Standard deviations are calculated over 5 random seeds and, for brevity, omitted in the right panel, where the maximum standard deviation does not exceed 0.0004.

	Avg. Spearman ↑	Spearman by phenotype ↑				
		Activity	Binding	Expression	Organismal Fitness	Stability
ESM2 (35M) (Lin et al., 2023)	0.3211	0.3137	0.2907	0.3435	0.2184	0.4392
ESM2 (35M) + ProteinTTT (Ours)	**0.3407** ± 0.00014	**0.3407**	**0.2942**	**0.3550**	**0.2403**	**0.4733**
SaProt (35M) (Su et al., 2023)	0.4062	0.3721	0.3568	0.4390	0.2879	0.5749
SaProt (35M) + ProteinTTT (Ours)	**0.4106** ± 0.00004	**0.3783**	**0.3569**	**0.4430**	**0.2955**	**0.5795**
ESM2 (650M) (Lin et al., 2023)	0.4139	0.4254	0.3366	0.4151	0.3691	**0.5233**
ESM2 (650M) + ProteinTTT (Ours)	**0.4153** ± 0.00003	**0.4323**	**0.3376**	**0.4168**	**0.3702**	0.5195
SaProt (650M) (Su et al., 2023)	0.4569	0.4584	0.3785	**0.4884**	0.3670	**0.5919**
SaProt (650M) + ProteinTTT (Ours)	**0.4583** ± 0.00001	**0.4593**	**0.3790**	0.4883	**0.3754**	0.5896
ProSST (K=2048) (Li et al., 2024)	0.5068	0.4758	0.4448	0.5302	0.4306	**0.6526**
ProSST (K=2048) + ProteinTTT (Ours)	**0.5087** ± 0.00004	**0.4822**	**0.4470**	**0.5321**	**0.4315**	0.6507

data. To achieve this, we create a new fitness prediction dataset mined from MaveDB, a public repository containing datasets from Multiplexed Assays of Variant Effect (MAVEs) (Esposito et al., 2019). Following ProteinGym, we measure performance on both datasets using Spearman correlation between predicted and experimental fitness values.

Results. ProteinTTT consistently enhances fitness prediction performance of all the tested models across varying model scales (35M and 650M parameters for both ESM2 and SaProt; 110M for ProSST) and both datasets, i.e., test ProteinGym (Table 2) and validation MaveDB (Table A5). Notably, ProSST + ProteinTTT sets a new state of the art on the ProteinGym benchmark (Spearman correlation coefficients calculated for individual deep mutational scanning experiments (DMSs) have statistically significant difference according to a paired t-test with $p < 0.05$).

We observe that ProteinTTT primarily improves performance for proteins with low MSA depth (i.e., the number of available homologous sequences), suggesting that single-sequence customization enhances predictions for proteins with fewer similar sequences in the training data (Table A4). The fact that ProteinTTT more effectively improves the performance of smaller ESM2 and SaProt models compared to their larger variants may indicate that the performance on the benchmark may be saturated for larger models, consistent we a recent observation (Notin, 2025). We provide a qualitative example showing how ESM2 (650M) + ProteinTTT significantly improves fitness prediction by capturing residues critical for protein stability (Figure A5). We also demonstrate that customization can be combined with evolutionary information from MSA to further boost fitness prediction performance (Section C).

4.3 Protein function prediction

Finally, we demonstrate a proof of concept for customization in the context of protein function prediction. We experiment with two tasks: predicting protein location within a cell (Stärk et al., 2021), and substrate classification for terpene synthases (TPS), enzymes producing the largest class of natural products (Samusevich et al., 2024). Section A shows that per-protein customization with ProteinTTT consistently enhances the performance of representative models on both tasks.

5 Case studies

ProteinTTT can be readily incorporated into structure, fitness, or function prediction pipelines by adding several lines of code (Section D). Here, we demonstrate two challenging structure prediction case studies: improving modeling of antibody–antigen loops (Section 5.1) and expanding known structures of viral proteins (Section 5.2).

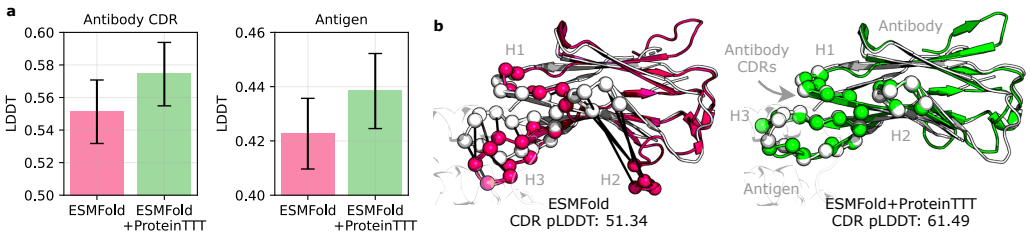


Figure 4: **ProteinTTT improves modeling of antibody–antigen loops**. **(a)** Average LDDT on the antibody complementarity-determining regions (CDRs, 175 structures) and antigens (814 structures) from the SAbDab dataset with ESMFold pLDDT < 70. Error bars indicate 95% confidence intervals estimated from 1000 bootstrap samples. **(b)** Example of improved structure prediction for CDRs in the 8K2W entry. The CDR regions H1, H2, and H3, i.e., the parts of the antibody that bind to the antigen, are highlighted with spheres, while black lines show the alignment error between the ground-truth CDR structure (white) and the predictions (colored).

5.1 Modeling antibody–antigen loops

Accurately predicting structures of antibodies (e.g., human defensive proteins) and antigens (e.g., viral proteins) enables rational design of new therapeutics (Bennett et al., 2025). However, the presence of highly variable loop regions makes modeling of these interactions a long-standing challenge. Here, we show that ProteinTTT substantially improves structure prediction for these loop-formed complementarity-determining regions (CDRs) of antibodies, i.e., the parts that bind antigens, as well as for antigens themselves, on the well-established SAbDab dataset (Dunbar et al., 2014).

We take the structures from SAbDab that are not predicted well by ESMFold (pLDDT < 70) and show that ProteinTTT improves the LDDT score for 115 of 175 antibody CDR substructures (66%) and 487 of 814 antigen chains (60%). As shown in Figure 4a, ESMFold + ProteinTTT achieves significantly higher average LDDT scores compared to general-purpose ESMFold. Figure 4b illustrates how ProteinTTT enables accurate prediction of all three CDRs in an antibody chain, providing an improved understanding of its binding interface with the corresponding antigen.

5.2 Expanding known structures of viral proteins

Predicting the structures of viral proteins is vital for vaccine development, antiviral design, and understanding infection (Bravi, 2024). Nevertheless, it remains challenging due to the high mutation rate, which often leaves viral proteins without close homologs or experimental structures in databases (Kim et al., 2025). Here, we demonstrate that per-protein customized predictions with ESMFold + ProteinTTT improve viral protein structure prediction, substantially expanding the Big Fantastic Virus Database—the repository of known viral protein structures (Kim et al., 2025).

Among all the entries in BFVD, predicted with AlphaFold2 through ColabFold (Mirdita et al., 2022) using MSAs constructed from Logan (Chikhi et al., 2024), only 55% have high-quality structure predictions (pLDDT > 70). We apply ESMFold and ESMFold + ProteinTTT to 70% of BFVD entries with the lowest AlphaFold2 pLDDT values to expand the database with higher-quality structures. This is achieved by applying all three methods to the specific protein and taking the predicted structure with the highest pLDDT. While ESMFold manages to improve the predicted structure (as measured by pLDDT) for 6% of these low-confidence structures, ESMFold + ProteinTTT leads to an improvement for 17% of these targets, substantially increasing the quality of known viral protein structures (Figure 5a).

We validate that the improved pLDDT confidence values from ESMFold + ProteinTTT correlate with the quality of the predicted structures, as measured by LDDT against reference AlphaFold2 structures having pLDDT > 90 (Pearson = 0.79; Figure A9). Notably, the largest improvements in pLDDT align with the largest improvements in LDDT (Figure 5b). We find that the benefit of customization saturates with the number of homologs available for a protein, indicating that ProteinTTT is most effective for challenging, out-of-distribution proteins (Figure 5c). Finally, Figure 5d–g shows examples where ProteinTTT enables high-confidence structure predictions in cases where general-purpose, uncustomized AlphaFold2 and ESMFold struggle.

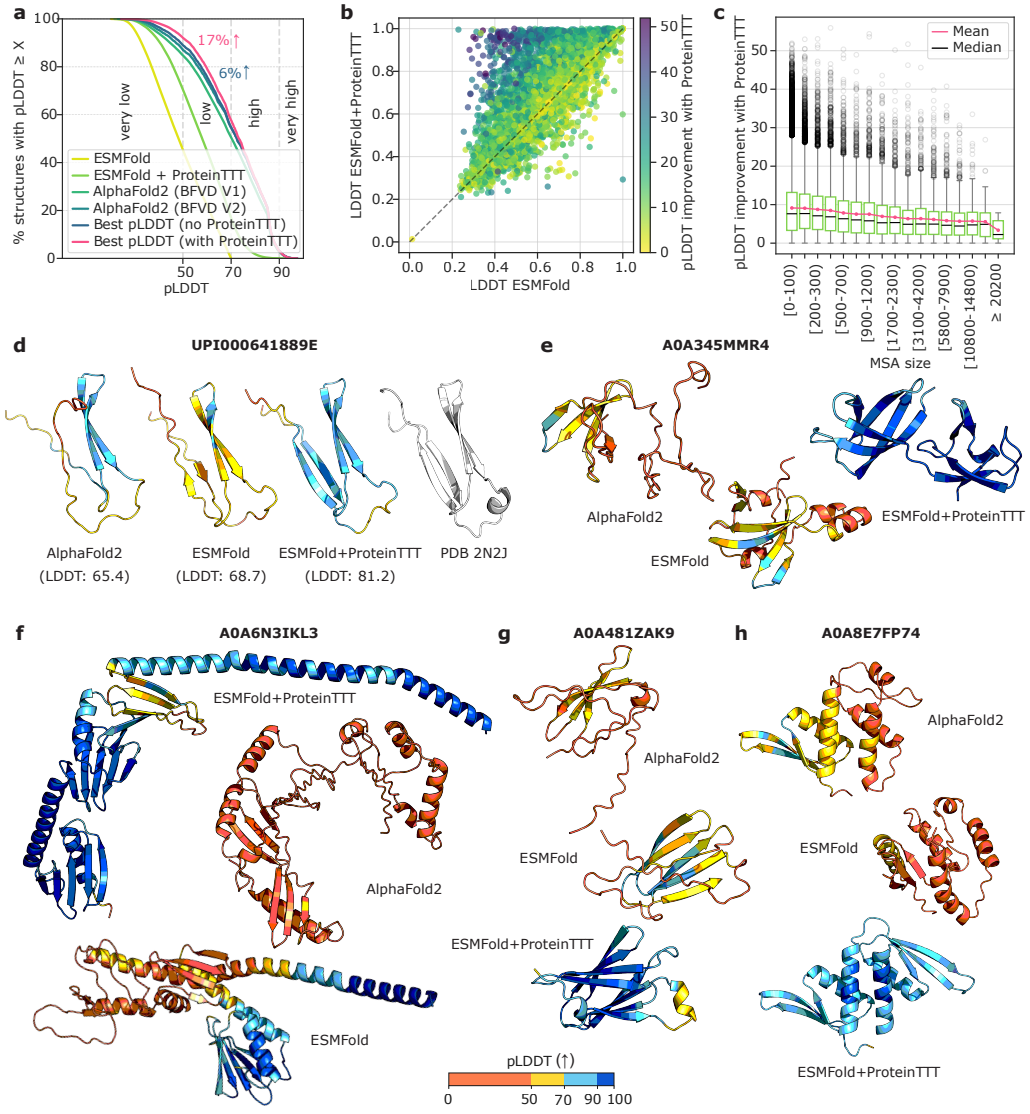


Figure 5: **ProteinTTT expands the Big Fantastic Virus Database (BFVD). (a)** ProteinTTT (light green) substantially improves the performance of ESMFold (yellow) on viral proteins, yielding better structures (pink) for 17% of BFVD entries compared to the original predictions by AlphaFold2 (green). **(b)** Improvements in pLDDT for ESMFold after ProteinTTT correspond to improvements in LDDT, as benchmarked against BFVD AlphaFold2 structures with pLDDT > 90. **(c)** ProteinTTT provides the largest pLDDT improvements (y-axis) for the most out-of-distribution proteins, i.e., those with the smallest MSAs (left on the x-axis) from the Logan database. **(d)** Structural comparison for BFVD entry UPI000641889E against the PDB structure 2N2J (100% sequence identity) shows that ESMFold + ProteinTTT yields a prediction closest to the ground truth (gray), as also measured by LDDT. **(e–g)** Additional examples of high-quality viral structures (as measured by pLDDT) predicted with ESMFold + ProteinTTT but not with ESMFold or AlphaFold2. Higher pLDDT values are better.

6 Discussion

We introduce ProteinTTT, a method for customizing protein language models to individual targets. ProteinTTT consistently improves performance across various models, their scales, and downstream tasks. It excels on challenging, out-of-distribution examples where general models often fail. We demonstrate its practical value through two case studies: enhancing the structural prediction of difficult antibody-antigen loops and improving 17% of low-confidence viral protein structures in the Big Fantastic Virus Database. Our work establishes per-protein customization as a powerful and practical tool for biological research.

REFERENCES

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*, pp. 1–3, 2024.

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.

Frances M. Ashcroft, Matthew Lloyd, and Elizabeth A. Haythorne. Glucokinase activity in diabetes: too much of a good thing? *Trends in Endocrinology & Metabolism*, 34(2):119–130, Feb 2023. ISSN 1043-2760. doi: 10.1016/j.tem.2022.12.007. URL https://doi.org/10.1016/j.tem.2022.12.007.

Pratyay Banerjee, Tejas Gokhale, and Chitta Baral. Self-supervised test-time learning for reading comprehension. *arXiv preprint arXiv:2103.11263*, 2021.

Eyal Ben-David, Nadav Oved, and Roi Reichart. Pada: Example-based prompt learning for on-the-fly adaptation to unseen domains. *Transactions of the Association for Computational Linguistics*, 10: 414–433, 2022.

Nathaniel R Bennett, Joseph L Watson, Robert J Ragotte, Andrew J Borst, DéJenaé L See, Connor Weidle, Riti Biswas, Yutong Yu, Ellen L Shrock, Russell Ault, et al. Atomically accurate de novo design of antibodies with rfdiffusion. *bioRxiv*, pp. 2024–03, 2025.

Barbara Bravi. Development and use of machine learning algorithms in vaccine target selection. *npj Vaccines*, 9(1):15, 2024.

Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Anton Bushuiev, Roman Bushuiev, Anatolii Filkin, Petr Kouba, Marketa Gabrielova, Michal Gabriel, Jiri Sedlar, Tomas Pluskal, Jiri Damborsky, Stanislav Mazurenko, et al. Learning to design protein-protein interactions with enhanced generalization. *arXiv preprint arXiv:2310.18515*, 2023.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*, 2013.

Tianlong Chen and Chengyue Gong. Hotprotein: A novel framework for protein thermostability prediction and editing. *NeurIPS 2022*, 2022.

Jun Cheng, Guido Novati, Joshua Pan, Clare Bycroft, Akvilė Žemgulytė, Taylor Applebaum, Alexander Pritzel, Lai Hong Wong, Michal Zielinski, Tobias Sargeant, et al. Accurate proteome-wide missense variant effect prediction with alphamissense. *Science*, 381(6664):eadg7492, 2023.

Zhixiang Chi, Li Gu, Tao Zhong, Huan Liu, Yuanhao Yu, Konstantinos N Plataniotis, and Yang Wang. Adapting to distribution shift by visual domain prompt generation. *arXiv preprint arXiv:2405.02797*, 2024.

Rayan Chikhi, Téo Lemane, Raphaël Loll-Krippleber, Mercè Montoliu-Nerin, Brice Raffestin, Antonio Pedro Camargo, Carson J Miller, Mateus Bernabe Fiamenghi, Daniel Paiva Agustinho, Sina Majidian, et al. Logan: planetary-scale genome assembly surveys life's diversity. *bioRxiv*, pp. 2024–07, 2024.

Cyrus Chothia and Arthur M Lesk. Canonical structures for the hypervariable regions of immunoglobulins. *Journal of molecular biology*, 196(4):901–917, 1987.

David W. Christianson. Structural and chemical biology of terpenoid cyclases. *Chemical Reviews*, 117(17):11570–11648, Sep 2017. ISSN 0009-2665. doi: 10.1021/acs.chemrev.7b00287. URL https://doi.org/10.1021/acs.chemrev.7b00287.

The UniProt Consortium. Uniprot: the universal protein knowledgebase in 2023. *Nucleic acids research*, 51(D1):D523–D531, 2023.

Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Daniel J Diaz, Chengyue Gong, Jeffrey Ouyang-Zhang, James M Loy, Jordan Wells, David Yang, Andrew D Ellington, Alex Dimakis, and Adam R Klivans. Stability oracle: a structure-based graph-transformer for identifying stabilizing mutations. *BioRxiv*, pp. 2023–05, 2023.

Henry Dieckhaus, Michael Brocidiacono, Nicholas Z Randolph, and Brian Kuhlman. Transfer learning to leverage larger datasets for improved prediction of protein stability changes. *Proceedings of the National Academy of Sciences*, 121(6):e2314853121, 2024.

James Dunbar, Konrad Krawczyk, Jinwoo Leem, Terry Baker, Angelika Fuchs, Guy Georges, Jiye Shi, and Charlotte M Deane. Sabdab: the structural antibody database. *Nucleic acids research*, 42 (D1):D1140–D1146, 2014.

Janani Durairaj, Andrew M Waterhouse, Toomas Mets, Tetiana Brodiazhenko, Minhal Abdullah, Gabriel Studer, Gerardo Tauriello, Mehmet Akdel, Antonina Andreeva, Alex Bateman, et al. Uncovering new families and folds in the natural protein universe. *Nature*, 622(7983):646–653, 2023.

Oliver Dutton, Sandro Bottaro, Istvan Redl, Michele Invernizzi, Albert Chung, Carlo Fisicaro, Falk Hoffmann, Stefano Ruschetta, Fabio Airoldi, Louie Henderson, et al. Improving inverse folding models at protein stability prediction without additional training or data. *bioRxiv*, pp. 2024–06, 2024.

Daniel Esposito, Jochen Weile, Jay Shendure, Lea M Starita, Anthony T Papenfuss, Frederick P Roth, Douglas M Fowler, and Alan F Rubin. Mavedb: an open-source platform to distribute and interpret data from multiplexed assays of variant effect. *Genome biology*, 20:1–11, 2019.

Xiaomin Fang, Fan Wang, Lihang Liu, Jingzhou He, Dayong Lin, Yingfei Xiang, Kunrui Zhu, Xiaonan Zhang, Hua Wu, Hui Li, et al. A method for multiple-sequence-alignment-free protein structure prediction using a protein language model. *Nature Machine Intelligence*, 5(10):1087–1096, 2023.

Tao Feng, Ziqi Gao, Jiaxuan You, Chenyi Zi, Yan Zhou, Chen Zhang, and Jia Li. Deep reinforcement learning for modelling protein complexes. *arXiv preprint arXiv:2405.02299*, 2024.

Yossi Gandelsman, Yu Sun, Xinlei Chen, and Alexei Efros. Test-time training with masked autoencoders. *Advances in Neural Information Processing Systems*, 35:29374–29385, 2022.

Jan Gorodkin. Comparing two k-category assignments by a k-category correlation coefficient. *Computational biology and chemistry*, 28(5-6):367–374, 2004.

Xin Gu, Patrick Jouandin, Pranav V. Lalgudi, Rich Binari, Max L. Valenstein, Michael A. Reid, Annamarie E. Allen, Nolan Kamitaki, Jason W. Locasale, Norbert Perrimon, and David M. Sabatini. Sestrin mediates detection of and adaptation to low-leucine diets in drosophila. *Nature*, 608(7921):209–216, Aug 2022. ISSN 1476-4687. doi: 10.1038/s41586-022-04960-2. URL https://doi.org/10.1038/s41586-022-04960-2.

Muhammet F. Gulen, Natasha Samson, Alexander Keller, Marius Schwabenland, Chong Liu, Selene Glück, Vivek V. Thacker, Lucie Favre, Bastien Mangeat, Lona J. Kroese, Paul Krimpenfort, Marco Prinz, and Andrea Ablasser. cgas–sting drives ageing-related inflammation and neurodegeneration. *Nature*, 620(7973):374–380, Aug 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06373-1. URL https://doi.org/10.1038/s41586-023-06373-1.

Kathryn H. Gunn and Saskia B. Neher. Structure of dimeric lipoprotein lipase reveals a pore adjacent to the active site. *Nature Communications*, 14(1):2569, May 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-38243-9. URL https://doi.org/10.1038/s41467-023-38243-9.

Moritz Hardt and Yu Sun. Test-time training on nearest neighbors for large language models. *arXiv preprint arXiv:2305.18466*, 2023.

Tomas 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. *bioRxiv*, pp. 2024–07, 2024.

Lucas Torroba Hennigen and Yoon Kim. Deriving language models from masked language models. *arXiv preprint arXiv:2305.15501*, 2023.

Thomas A Hopf, John B Ingraham, Frank J Poelwijk, Charlotta PI Schärfe, Michael Springer, Chris Sander, and Debora S Marks. Mutation effects predicted from sequence co-variation. *Nature biotechnology*, 35(2):128–135, 2017.

Gerta Hoxhaj and Brendan D. Manning. The pi3k–akt network at the interface of oncogenic signalling and cancer metabolism. *Nature Reviews Cancer*, 20(2):74–88, Feb 2020. ISSN 1474-1768. doi: 10.1038/s41568-019-0216-7. URL https://doi.org/10.1038/s41568-019-0216-7.

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.

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.

Pranav Kantroo, Gunter Wagner, and Benjamin Machta. Pseudo-perplexity in one fell swoop for protein fitness estimation. *bioRxiv*, pp. 2024–07, 2024.

Neerav Karani, Ertunc Erdil, Krishna Chaitanya, and Ender Konukoglu. Test-time adaptable neural networks for robust medical image segmentation. *Medical Image Analysis*, 68:101907, 2021.

Zuzana Keckesova, Joana Liu Donaher, Jasmine De Cock, Elizaveta Freinkman, Susanne Lingrell, Daniel A. Bachovchin, Brian Bierie, Verena Tischler, Aurelia Noske, Marian C. Okondo, Ferenc Reinhardt, Prathapan Thiru, Todd R. Golub, Jean E. Vance, and Robert A. Weinberg. Lactb is a tumour suppressor that modulates lipid metabolism and cell state. *Nature*, 543(7647):681–686, Mar 2017. ISSN 1476-4687. doi: 10.1038/nature21408. URL https://doi.org/10.1038/nature21408.

Rachel Seongeun Kim, Eli Levy Karin, Milot Mirdita, Rayan Chikhi, and Martin Steinegger. Bfvd—a large repository of predicted viral protein structures. *Nucleic Acids Research*, 53(D1):D340–D347, 2025.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Andrew Kirjner, Jason Yim, Raman Samusevich, Shahar Bracha, Tommi S Jaakkola, Regina Barzilay, and Ila R Fiete. Improving protein optimization with smoothed fitness landscapes. In *The Twelfth International Conference on Learning Representations*, 2023.

Petr Kouba, Pavel Kohout, Faraneh Haddadi, Anton Bushuiev, Raman Samusevich, Jiri Sedlar, Jiri Damborsky, Tomas Pluskal, Josef Sivic, and Stanislav Mazurenko. Machine learning-guided protein engineering. *ACS catalysis*, 13(21):13863–13895, 2023.

Elodie Laine, Yasaman Karami, and Alessandra Carbone. Gemme: a simple and fast global epistatic model predicting mutational effects. *Molecular biology and evolution*, 36(11):2604–2619, 2019.

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*, pp. 2024–04, 2024.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574. URL https://www.science.org/doi/abs/10.1126/science.ade2574.

Yuejiang Liu, Parth Kothari, Bastien van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. TTT++: when does self-supervised test-time training fail or thrive? In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 21808–21820, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/b618c3210e934362ac261db280128c22-Abstract.html.

Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2): 129–137, 1982.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Valerio Mariani, Marco Biasini, Alessandro Barbato, and Torsten Schwede. lddt: a local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics*, 29(21):2722–2728, 2013.

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.

Peter Mikhael, Itamar Chinn, and Regina Barzilay. Clipzyme: Reaction-conditioned virtual screening of enzymes. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=0mYAK6Yhhm.

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.

Pascal Notin. Have we hit the scaling wall for protein language models? Substack blog post, May 7 2025. URL https://pascalnotin.substack.com/p/have-we-hit-the-scaling-wall-for.

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.

Dong oh Seo, David O'Donnell, Nimansha Jain, Jason D. Ulrich, Jasmin Herz, Yuhao Li, Mackenzie Lemieux, Jiye Cheng, Hao Hu, Javier R. Serrano, Xin Bao, Emily Franke, Maria Karlsson, Martin Meier, Su Deng, Chandani Desai, Hemraj Dodiya, Janaki Lelwala-Guruge, Scott A. Handley, Jonathan Kipnis, Sangram S. Sisodia, Jeffrey I. Gordon, and David M. Holtzman. Apoe isoform– and microbiota-dependent progression of neurodegeneration in a mouse model of tauopathy. *Science*, 379(6628):eadd1236, 2023. doi: 10.1126/science.add1236. URL https://www.science.org/doi/abs/10.1126/science.add1236.

Andrei Papkou, Lucia Garcia-Pastor, José Antonio Escudero, and Andreas Wagner. A rugged yet easily navigable fitness landscape. *Science*, 382(6673):eadh3860, 2023. doi: 10.1126/science.adh3860. URL https://www.science.org/doi/abs/10.1126/science.adh3860.

A Paszke. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

Predrag Radivojac and et al. A large-scale evaluation of computational protein function prediction. *Nature Methods*, 10(3):221–227, Mar 2013. ISSN 1548-7105. doi: 10.1038/nmeth.2340. URL https://doi.org/10.1038/nmeth.2340.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pp. 8821–8831. Pmlr, 2021.

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.

Roshan Rao, Joshua Meier, Tom Sercu, Sergey Ovchinnikov, and Alexander Rives. Transformer protein language models are unsupervised structure learners. *Biorxiv*, pp. 2020–12, 2020.

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*, pp. 8844–8856. PMLR, 2021.

Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 14837–14847, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/5f8e2fa1718d1bbcadf1cd9c7a54fb8c-Abstract.html.

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.

Xavier Robin, Juergen Haas, Rafal Gumienny, Anna Smolinski, Gerardo Tauriello, and Torsten Schwede. Continuous automated model evaluation (cameo)—perspectives on the future of fully automated evaluation of structure prediction methods. *Proteins: Structure, Function, and Bioinformatics*, 89(12):1977–1986, 2021.

Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

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

Raman Samusevich, Téo Hebra, Roman Bushuiev, Anton Bushuiev, Tereza Čalounová, Helena Smrčková, Ratthachat Chatpatanasiri, Jonáš Kulhánek, Milana Perković, Martin Engst, Adéla Tajovská, Josef Sivic, and Tomáš Pluskal. Highly accurate discovery of terpene synthases powered by machine learning reveals functional terpene cyclization in archaea. *bioRxiv*, 2024. doi: 10.1101/2024.01.29.577750. URL https://www.biorxiv.org/content/early/2024/04/25/2024.01.29.577750.

Nicolae Sapoval, Amirali Aghazadeh, Michael G. Nute, Dinler A. Antunes, Advait Balaji, Richard Baraniuk, C. J. Barberan, Ruth Dannenfelser, Chen Dun, Mohammadamin Edrisi, R. A. Leo Elworth, Bryce Kille, Anastasios Kyrillidis, Luay Nakhleh, Cameron R. Wolfe, Zhi Yan, Vicky Yao, and Todd J. Treangen. Current progress and open challenges for applying deep learning across the biosciences. *Nature Communications*, 13(1):1728, Apr 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-29268-7. URL https://doi.org/10.1038/s41467-022-29268-7.

Emre Sevgen, Joshua Moller, Adrian Lange, John Parker, Sean Quigley, Jeff Mayer, Poonam Srivastava, Sitaram Gayatri, David Hosfield, Maria Korshunova, et al. Prot-vae: protein transformer variational autoencoder for functional protein design. *bioRxiv*, pp. 2023–01, 2023.

Peter Škrinjar, Jérôme Eberhardt, Janani Durairaj, and Torsten Schwede. Have protein-ligand co-folding methods moved beyond memorisation? *BioRxiv*, pp. 2025–02, 2025.

Yidong Song, Qianmu Yuan, Sheng Chen, Yuansong Zeng, Huiying Zhao, and Yuedong Yang. Accurately predicting enzyme functions through geometric graph learning on esmfold-predicted structures. *Nature Communications*, 15(1):8180, 2024.

Hannes Stärk, Christian Dallago, Michael Heinzinger, and Burkhard Rost. Light attention predicts protein location from the language of life. *Bioinformatics Advances*, 1(1):vbab035, 11 2021. ISSN 2635-0041. doi: 10.1093/bioadv/vbab035. URL https://doi.org/10.1093/bioadv/vbab035.

Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. Saprot: Protein language modeling with structure-aware vocabulary. *bioRxiv*, pp. 2023–10, 2023.

Sriram Subramaniam and Gerard J. Kleywegt. A paradigm shift in structural biology. *Nature Methods*, 19(1):20–23, Jan 2022. ISSN 1548-7105. doi: 10.1038/s41592-021-01361-7. URL https://doi.org/10.1038/s41592-021-01361-7.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pp. 9229–9248. PMLR, 2020.

Nataša Tagasovska, Ji Won Park, Matthieu Kirchmeyer, Nathan C Frey, Andrew Martin Watkins, Aya Abdelsalam Ismail, Arian Rokkum Jamasb, Edith Lee, Tyler Bryson, Stephen Ra, et al. Antibody domainbed: Out-of-distribution generalization in therapeutic protein design. *arXiv preprint arXiv:2407.21028*, 2024.

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.

Mike Tyers and Matthias Mann. From genomics to proteomics. *Nature*, 422(6928):193–197, Mar 2003. ISSN 1476-4687. doi: 10.1038/nature01510. URL https://doi.org/10.1038/nature01510.

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*, pp. 2022–02, 2022.

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.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Renhao Wang, Yu Sun, Yossi Gandelsman, Xinlei Chen, Alexei A Efros, and Xiaolong Wang. Test-time training on video streams. *arXiv preprint arXiv:2307.05014*, 2023.

Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

Zehao Xiao, Xiantong Zhen, Ling Shao, and Cees GM Snoek. Learning to generalize across domains on single test samples. *arXiv preprint arXiv:2202.08045*, 2022.

Tianhao Yu, Haiyang Cui, Jianan Canal Li, Yunan Luo, Guangde Jiang, and Huimin Zhao. Enzyme function prediction using contrastive learning. *Science*, 379(6639):1358–1363, 2023. doi: 10.1126/science.adf2465. URL https://www.science.org/doi/abs/10.1126/science.adf2465.

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.

Hao Zhao, Yuejiang Liu, Alexandre Alahi, and Tao Lin. On pitfalls of test-time adaptation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 42058–42080. PMLR, 2023. URL https://proceedings.mlr.press/v202/zhao23d.html.

APPENDIX

CONTENTS

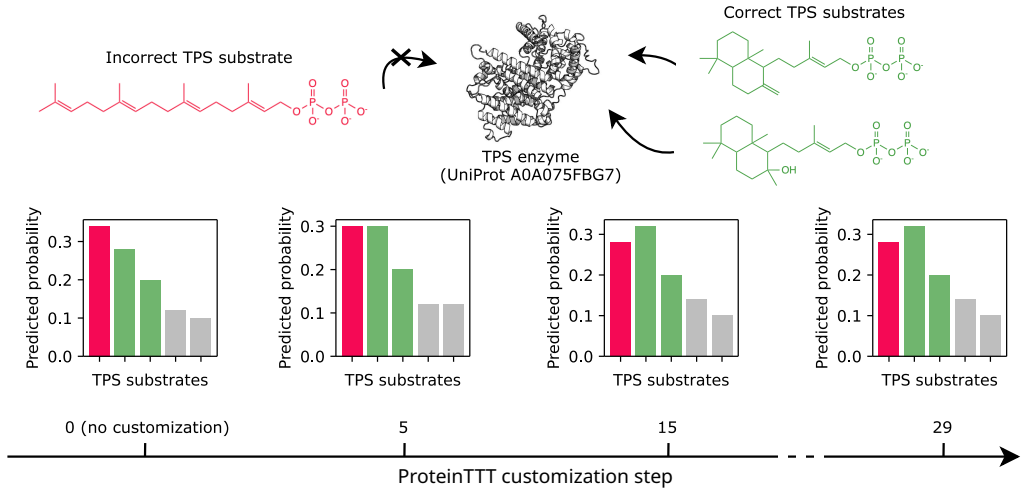


Figure A1: **Customization with ProteinTTT enables the correct substrate classification for a terpene synthase (TPS) enzyme.** With progressive customization steps of TerpeneMiner + ProteinTTT, the probability of the initially misclassified substrate (red) decreases, while the probability of the true substrates (green) increases. The bar plots also display the predicted probabilities for other substrates with non-zero values (grey).

A Customization for protein function prediction

Protein function prediction is essential for understanding biological processes and guiding bioengineering, but is challenging due to its vague definition and limited data (Yu et al., 2023; Radivojac & et al., 2013; Stärk et al., 2021; Mikhael et al., 2024; Samusevich et al., 2024). While improved structure prediction with ProteinTTT (Section 4.1) can already enhance function prediction (Song et al., 2024), we also evaluate our customization method directly on two function classification tasks: subcellular localization, predicting protein location within a cell (Stärk et al., 2021), and substrate classification for terpene synthases (TPS), enzymes producing the largest class of natural products (Christianson, 2017; Samusevich et al., 2024). Using ProteinTTT with TerpeneMiner (Samusevich et al., 2024) for TPS detection and Light attention (Stärk et al., 2021) for subcellular localization, we achieve consistent performance gains.

Evaluation setup. For the terpene substrate classification, we use the largest available dataset of characterized TPS from Samusevich et al. (2024) and reuse the original cross-validation schema. In the case of protein localization prediction, we use a standard DeepLoc dataset (Almagro Armenteros et al., 2017) as a validation set and setHard from Stärk et al. (2021) as the test set.

Given a protein, the goal of function prediction is to correctly classify it into one of the predefined functional annotations. We assess the quality of the TPS substrate prediction using standard multi-label classification metrics used in the TerpeneMiner paper (Samusevich et al., 2024): mean average

Table A1: **Customization with ProteinTTT improves protein function prediction.** For the terpene syntase (TPS) substrate classification task, the metrics are computed on the 512 TPS sequences based on the cross-validation schema of the TPS dataset (Samusevich et al., 2024). Subcellular localization prediction performance is reported for 432 protein sequences from the setHard test set (Stärk et al., 2021). The error bars show standard deviations across five random seeds.

TPS substrate classification

Method	mAP ↑	AUROC ↑
TerpeneMiner (Samusevich et al., 2024)	0.805	0.948
TerpeneMiner + ProteinTTT (Ours)	**0.811 ± 0.0011**	**0.950 ± 0.0002**

Subcellular localization prediction

Method	Accuracy ↑	MCC ↑	F1-score ↑
Light attention (Stärk et al., 2021)	0.627	0.549	0.618
Light attention + ProteinTTT (Ours)	**0.634 ± 0.004**	**0.557 ± 0.005**	**0.627 ± 0.004**

precision (mAP) and area under the receiver operating characteristic curve (AUROC). In the case of protein localization prediction, we similarly use the classification metrics from the original paper (Stärk et al., 2021): accuracy, multi-class Matthews correlation coefficient (MCC), and F1-score.

Results. Customization with ProteinTTT improves model performance on both of the protein function prediction tasks and across all considered metrics (Table A1). Figure A1 provides a qualitative result, where customization with ProteinTTT iteratively refines the prediction of TerpeneMiner toward a correct TPS substrate class. We hypothesize that improvement with customization is more challenging in classification tasks, as opposed to regression problems, because a larger change in the latent space is required to shift the top-class probability.

B JUSTIFICATION OF CUSTOMIZATION VIA PERPLEXITY MINIMIZATION

While the paradigm of test-time customization has been investigated in other domains, the reasons behind its surprising effectiveness are not completely clear (Liu et al., 2021; Zhao et al., 2023). Here, we offer a potential justification for the effectiveness of ProteinTTT by linking it to perplexity minimization.

Perplexity has traditionally been used in natural language processing to evaluate how well models comprehend sentences (Brown, 2020; Chelba et al., 2013). Protein language modeling has adopted this metric to assess how effectively models "understand" amino acid sequences (Hayes et al., 2024; Lin et al., 2023). For bidirectional, random masking language models, which are the focus of this study, we consider the following definition of perplexity[2]:

$$\text{Perplexity}(x) = \exp\left(\frac{1}{|x|}\sum_{i=1}^{|x|} -\log p(x_i|x_{\setminus i};\theta)\right), \quad (3)$$

where $|x|$ is the length of the input protein sequence x and $p(x_i|x_{\setminus i};\theta)$ represents the probability that the model correctly predicts the token x_i at position i when it is masked on the input $x_{\setminus i}$. Perplexity ranges from 1 to infinity (the lower, the better), providing an intuitive measure of how well a model fits, on average, tokens in a given sequence. A perplexity value of 1 indicates that the model perfectly fits the sequence, accurately predicting all the true tokens.

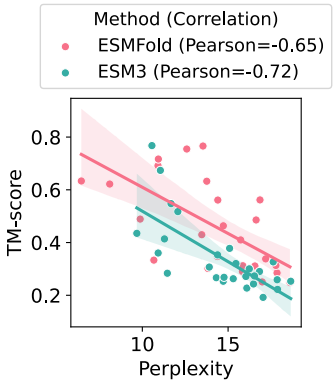


Figure A2: Quality of protein structure prediction, as measured by TM-score, correlates with perplexity of the underlying language model on the challenging targets from the CAMEO validation set. Higher TM-scores are associated with lower perplexity, indicating that better predictions are linked to lower uncertainty in the language model's understanding of the protein sequence.

Several studies have shown that lower perplexity on held-out protein sequences (calculated through the self-supervised track $g \circ f$) correlates with better performance on downstream tasks (via the supervised track $h \circ f$), such as predicting protein contacts (Rao et al., 2020), structure (Lin et al., 2023), or fitness (Kantroo et al., 2024). To give an example, we analyze the correlation between perplexity and structure prediction quality (Figure A2; see Section 4.1 for experimental details). A notable correlation suggests that reducing a model's perplexity on a single target sample x (applied independently to all test samples) can lead to improved predictions on the downstream task (Figure 3; Figure A10).

Since we assume only a single target example x, the minimization of the masked language modeling loss $\mathcal{L}(x;\theta)$ (Equation (2)) on this example is directly linked to minimizing the perplexity $\text{Perplexity}(x)$ (Equation (3)). For instance, in the case of a single masked position (i.e., $|M| = 1$), the loss is equal to the logarithm of perplexity. More generally, it can be shown formally that by minimizing the masked language modeling objective, the model learns to approximate the conditional marginals of the language (of proteins), including the leave-one-out probabilities evaluated in perplexity (Hennigen & Kim, 2023). As a result, applying self-supervised test-time customization

[2]Please note that this is an approximation of perplexity, which is computationally intractable for bidirectional models, and is often referred to as pseudo-perplexity (Lin et al., 2023; Salazar et al., 2019).

Table A2: **ProteinTTT can be used with MSA when available.** Please see Table 2 for evaluation details.

Method	Avg. Spearman ↑
ESM2 (Lin et al., 2023)	0.4139
ESM2 + ProteinTTT$_{\text{MSA}}$ (Ours)	**0.4299** $_{\pm 0.00099}$
MSA Transformer (Rao et al., 2021)	0.4319
MSA Transformer + ProteinTTT (Ours)	**0.4326** $_{\pm 0.00003}$

on x through $g \circ f$ enhances the representation of the target protein in the backbone f, leading to improved downstream performance via the fine-tuning track $h \circ f$.

C Customization with multiple sequence alignment (MSA)

Customization training objective. Since many target proteins may not have homologous sequences (Rao et al., 2021) and finding such homologs may be time-consuming (Lin et al., 2023), the ProteinTTT customization objective (Equation (2)) only assumes a single target sequence for customization. However, we also extend the loss function to the case when a multiple sequence alignment (MSA) is available:

$$\mathcal{L}_{\text{MSA}}(x;\theta) = \mathbb{E}_{x' \sim p_{\text{MSA}}(x'|x)}\big[\mathcal{L}(x';\theta)\big], \tag{4}$$

where $p_{\text{MSA}}(x'|x)$ is the distribution of sequences x' homologous to the target protein x, $\mathcal{L}$ is the single-sequence loss function defined in Equation (2), and θ denotes the tunable parameters of the model backbone f. We refer to customization using Equation (4) as ProteinTTT$_{\text{MSA}}$.

Results for fitness prediction. It is known that evolutionary information is important for protein fitness prediction (Laine et al., 2019). Therefore, we demonstrate how ProteinTTT$_{\text{MSA}}$ and ProteinTTT can enhance the performance of PLMs on the ProteinGym benchmark (Notin et al., 2024). Table A2 shows that using ProteinTTT$_{\text{MSA}}$ with high-quality MSAs curated by Notin et al. (2024) strongly enhances the performance of ESM2, approaching that of MSA Transformer, pre-trained on MSAs. Moreover, we find that MSA Transformer slightly benefits from single-sequence customization with ProteinTTT, while customization to whole or subsampled MSAs disrupts the performance (Table A3 in Section G.2).

D Implementation details

Infrastructure. All experiments with ProteinTTT are conducted on machines equipped with a single NVIDIA A100 40GB GPU, an 8-core AMD processor, and 128 GB of physical memory.

Source code. We provide a user-friendly and easily extensible PyTorch (Paszke, 2019) implementation of ProteinTTT, available as the `proteinttt` Python package[3]. We provide two Python code snippets Listing 1 and Listing 2 to demonstrate the implementation of inference and customization with ProteinTTT, respectively. Listing 1 demonstrates how inference with ESMFold can be enhanced with ProteinTTT by adding just a few lines of code to enable customization. Next, Listing 2 shows how ProteinTTT can be easily implemented for a PLM of interest by inheriting from the abstract `TTTModule` class. To integrate ProteinTTT within a model (e.g., ESM2), the user needs to implement methods that define the model's vocabulary, an interface for predicting logits, and a specification of which modules need to be fine-tuned or remain frozen. The rest, i.e., the test-time training logic itself, is implemented within the unified `TTTModule` class.

[3] `https://anonymous.4open.science/r/ProteinTTT-anonymous-F585`

```python
import esm
from proteinttt.models.esmfold import ESMFoldTTT, DEFAULT_ESMFOLD_TTT_CFG


# Set protein sequence
sequence = (
    "GIHLGELGLLPSTVLAIGYFENLVNIICESLNMLPKLEVSGKEYKKFKFTIVIPKDLDANIKKRAKIY"
    "FKQKSLIEIEIPTSSRNYPIHIQFDENSTDDILHLYDMPTTIGGIDKAIEMFMRKGHIGKTDQQKLLE"
    "ERELRNFKTTLENLIATDAFAKEMVEVIIEE"
)

# Load model
model = esm.pretrained.esmfold_v1()
model = model.eval().cuda()

predict_structure(model, sequence)
# pLDDT: 38.43025

# ============================ ProteinTTT ================================
# Customize model to sequence
model = ESMFoldTTT.ttt_from_pretrained(
    model, ttt_cfg=DEFAULT_ESMFOLD_TTT_CFG, esmfold_config=model.cfg
)
model.ttt(sequence)
# ========================================================================

predict_structure(model, sequence)
# pLDDT: 78.69619

# ============================ ProteinTTT ================================
# Reset model to original state (after this model.ttt can be called with
# another protein)
model.ttt_reset()
# ========================================================================
```

Listing 1: Incorporation of ProteinTTT into an ESMFold structure prediction pipeline using the `proteinttt` package.

Optimization. We minimize the loss defined in Equation (2) using stochastic gradient descent (SGD) with zero momentum and zero weight decay (Ruder, 2016). While a more straightforward option might be to use the optimizer state from the final pre-training step, this approach is often impractical because the optimizer parameters are usually not provided with the pre-trained model (Hayes et al., 2024; Lin et al., 2023). Moreover, many models are pre-trained using the Adam optimizer (Kingma & Ba, 2015) or its variants (Loshchilov & Hutter, 2019). However, it was shown that Adam results in less predictable behavior of test-time training compared to the SGD optimizer, possibly due to its more exploratory behavior (Gandelsman et al., 2022).

Customizing large models. We aim for customization to be applicable on the fly, i.e., without the need for any pre-computation and on a single GPU with a minimum computational overhead. Since state-of-the-art models for many protein-oriented tasks are typically large, with up to billions of parameters, our aim presents two key challenges. First, when using pre-trained Transformers on a single GPU, even for the forward pass, the batch size is typically limited to only several samples due to the quadratic complexity of the inference (Vaswani, 2017). Second, for the backward pass, even a batch size of one is not always feasible for large models. To address the first challenge, we perform forward and backward passes through a small number of training examples and accumulate gradients to simulate updates with any batch size. We address the second challenge by employing low-rank adaptation (LoRA; Hu et al. (2021)), which in practice enables fine-tuning of any model for which a forward pass on a single sample is feasible, due to a low number of trainable parameters. Section G.3 details how ESMFold (Lin et al., 2023), with its 3B-parameter ESM2 backbone f, can be efficiently customized, retaining its speed advantage while enhancing performance.

```python
import torch
import esm
from esm.model.esm2 import ESM2
from proteinttt.base import TTTModule


class ESM2TTT(TTTModule, ESM2):
    def __init__(self, ttt_cfg: TTTConfig, **kwargs):
        ESM2.__init__(self, **kwargs)
        TTTModule.__init__(self, ttt_cfg=ttt_cfg)
        self.ttt_alphabet = esm.Alphabet.from_architecture("ESM-1b")
        self.ttt_batch_converter = self.ttt_alphabet.get_batch_converter()

    def _ttt_tokenize(self, seq: str, **kwargs):
        batch_labels, batch_strs, batch_tokens = self.ttt_batch_converter(
            [(None, seq)]
        )
        return batch_tokens

    def _ttt_get_frozen_modules(self) -> list[torch.nn.Module]:
        return [self.embed_tokens]

    def _ttt_mask_token(self, token: int) -> int:
        return self.ttt_alphabet.mask_idx

    def _ttt_get_padding_token(self) -> int:
        return self.ttt_alphabet.padding_idx

    def _ttt_token_to_str(self, token: int) -> str:
        return self.ttt_alphabet.all_toks[token]

    def _ttt_get_all_tokens(self) -> list[int]:
        return [
            self.ttt_alphabet.tok_to_idx[t]
            for t in self.ttt_alphabet.all_toks
        ]

    def _ttt_get_non_special_tokens(self) -> list[int]:
        return [
            self.ttt_alphabet.tok_to_idx[t]
            for t in self.ttt_alphabet.standard_toks
        ]

    def _ttt_predict_logits(
        self, batch: torch.Tensor, start_indices: torch.Tensor = None
    ) -> torch.Tensor:
        return self(batch)["logits"]
```

Listing 2: Implementation of ESM2 + ProteinTTT within the `proteinttt` package.

E EXPERIMENTAL DETAILS

In this section, we describe the proposed benchmark suite for the three customization tasks considered in this work: protein structure prediction (Section E.1), protein fitness prediction (Section E.2), and protein function prediction (Section E.3). Each subsection describes the application of ProteinTTT to the respective models, along with details on the data, metrics, and models. Table A3 additionally summarizes the hyperparameters used for the application of ProteinTTT to individual models.

E.1 PROTEIN STRUCTURE PREDICTION

E.1.1 DATASETS

CAMEO dataset. To evaluate the capabilities of ProteinTTT on protein folding, we employ the CAMEO validation and test sets as described in Lin et al. (2023). Specifically, the validation set was obtained by querying the CAMEO (Continuous Automated Model Evaluation) web server[4] (Robin et al., 2021) for entries between August 2021 and January 2022, while the CAMEO test set consists of entries from April 1, 2022, to June 25, 2022. Most of the entries in the CAMEO sets are predicted with high accuracy and confidence (Lin et al., 2023). Therefore, we subselect the challenging validation and test sets where customization with ProteinTTT is suitable.

Specifically, we apply two standard criteria: (1) preserving entries with ESMFold pLDDT scores below 70 to filter out high-confidence predictions (Jumper et al., 2021), and (2) selecting entries with ESM2 perplexity scores greater than or equal to 6, ensuring that the predictions are challenging due to poor sequence understanding rather than other factors. Additionally, most structures with perplexity scores below 6 are already associated with high-confidence predictions (Figure S5 in Lin et al. (2023)). After filtering, the resulting challenging validation and test sets consist of 27 (out of 378) and 18 (out of 194) targets, respectively.

E.1.2 METRICS

To assess the quality of the predicted protein structures with respect to the ground truth structures, we use two standard metrics averaged across the test dataset: TM-score (Zhang & Skolnick, 2004) and LDDT (Mariani et al., 2013).

TM-score. The TM-score (Template Modeling score) is a metric used to assess the quality of the global 3D alignment between the predicted and target protein structures. It evaluates the structural similarity by comparing the distance between corresponding residues after superposition. The TM-score ranges from 0 to 1, where higher values indicate better alignment.

LDDT. The Local Distance Difference Test (LDDT) is an alignment-free metric used to assess the accuracy of predicted protein structures. Unlike global metrics, LDDT focuses on local structural differences by measuring the deviation in distances between atom pairs in the predicted structure compared to the target structure. It is particularly useful for evaluating the accuracy of local regions, such as secondary structure elements. LDDT scores range from 0 to 100, with higher values indicating better local structural agreement.

E.1.3 MODELS

ESMFold. The ESMFold architecture comprises two key components: a protein language model, ESM2, which, given a protein sequence, generates embeddings for individual amino acids, and a folding block that, using these embeddings and the sequence, predicts the protein 3D structure along with per-amino-acid confidence scores, known as pLDDT scores. In our experiments, we use the `esmfold_v0` model from the publicly available ESMFold checkpoints[5]. Please note that we use `esmfold_v0` and not `esmfold_v1` to avoid data leakage with respect to the CAMEO test set.

ESMFold + ProteinTTT. Since the ESM2 backbone of ESMFold was pre-trained in a self-supervised masked modeling regime, the application of ProteinTTT to ESMFold is straightforward. We treat ESM2 as the backbone f, the language modeling head predicting amino acid classes from their embeddings as the self-supervised head g, and the folding trunk along with the structure modules as the downstream task head h. After each ProteinTTT step, we run $h \circ f$ to compute the pLDDT scores, which allows us to estimate the optimal number of customization steps for each protein based on the highest pLDDT score.

[4] `https://www.cameo3d.org/modeling`

[5] `https://github.com/facebookresearch/esm/blob/main/esm/esmfold/v1/pretrained.py`

Since the backbone f is given by the ESM2 model containing 3 billion parameters, we apply LoRA (Hu et al., 2021) to all matrices involved in self-attention. This enables fine-tuning ESMFold + ProteinTTT on a single GPU.

ESMFold + ME. Since ESMFold is a regression model, it only predicts one solution and does not have a straightforward mechanism for sampling multiple structure predictions. Nevertheless, the authors of ESMFold propose a way to sample multiple candidates (Section A.3.2 in Lin et al. (2023)). To sample more predictions, the masking prediction (ME) method randomly masks 15% (same ratio as during masked language modeling pre-training) of the amino acids before passing them to the language model. Selecting the solution with the highest pLDDT may lead to improved predicted structure. Since sampling multiple solutions with ESMFold + ME and selecting the best one via pLDDT is analogous to ESMFold + ProteinTTT, we employ the former as a baseline, running the method for the same number of steps.

ESM3. Unlike ESMFold, ESM3 is a fully multiple-track, BERT-like model (Devlin, 2018), pre-trained to unmask both protein sequence and structure tokens simultaneously (along with the function tokens). The structure tokens in ESM3 are generated via a separately pre-trained VQ-VAE (Razavi et al., 2019) operating on the protein geometry. In our experiments, we use the smallest, publicly available version of the ESM3 model (`ESM3_sm_open_v0`)[6].

ESM3 + ProteinTTT. We treat the Transformer encoder of ESM3 as f, the language modeling head decoding amino acid classes as g, and the VQ-VAE decoder, which maps structure tokens to the 3D protein structure, as h. During the customization steps, we train the model to unmask a protein sequence while keeping the structural track fully padded. During the inference, we provide the model with a protein sequence and run it to unmask the structural tokens, which are subsequently decoded with the VQ-VAE decoder. After each customization step, we run $h \circ f$ to compute the pLDDT scores, which allows us to estimate the optimal number of customization steps for each protein based on the highest pLDDT score. We choose the optimal hyperparameters by maximizing the difference in TM-score after and before applying ProteinTTT across the validation dataset.

Despite the fact that the model contains 1.4 billion parameters, even without using LoRA, ESM3 + ProteinTTT can be fine-tuned on a single NVIDIA A100 GPU. Therefore, we do not employ LoRA for fine-tuning ESM3, while this can also be possible.

ESM3 + CoT. To improve the generalization and protein-specific performance of ESM3, the original ESM3 paper employs a chain of thought (CoT) procedure. The procedure unfolds in n steps as follows. At each step, $1/n$ of the masked tokens with the lowest entropy after softmax on logits are unmasked. Then, the partially unmasked sequence is fed back into the model, and the process repeats until the entire sequence is unmasked. In our experiments, we set $n = 8$, which is the default value provided in the official GitHub repository.

HelixFold-Single. HelixFold-Single is an MSA-free protein structure prediction model that combines representations from a pretrained protein language model with adapted AlphaFold2 geometric modules (EvoformerS and Structure) to directly predict atomic coordinates (Fang et al., 2023). We use the official implementation[7]

HelixFold-Single + ProteinTTT. HelixFold-Single shares the main concept with ESMFold, and we combine it with ProteinTTT in the same way as in ESMFold + ProteinTTT.

[6] `https://github.com/evolutionaryscale/esm`

[7] `https://github.com/PaddlePaddle/PaddleHelix/tree/dev/apps/protein_folding/helixfold-single`

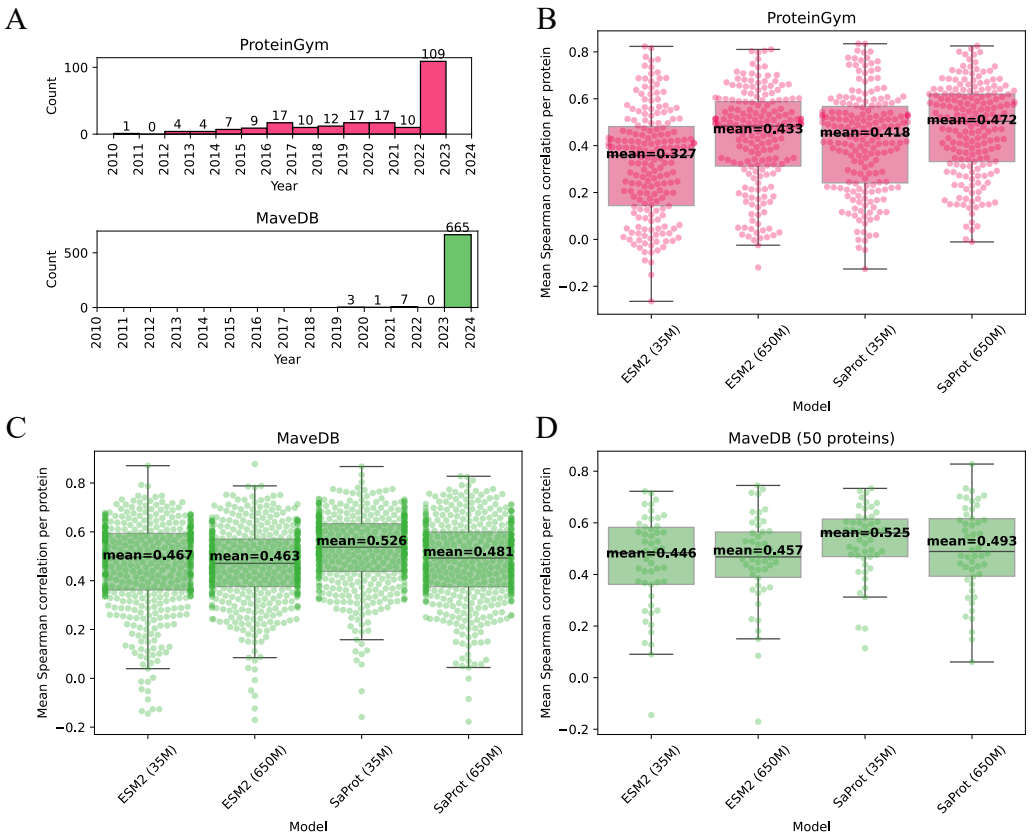


Figure A3: **Comparison of the standard ProteinGym dataset with the MaveDB dataset constructed in this work**. A) MaveDB, mined from Esposito et al. (2019), includes novel assays even after filtering to ensure distinct proteins from the comprehensive ProteinGym dataset. This is largely because most MaveDB assays post-filtering date to 2024, whereas the latest assays in ProteinGym date to 2023. B, C, D) MaveDB is of sufficient quality for model evaluation. Representative baselines, ESM2 and SaProt with both 35 million and 650 million parameters, evaluated on ProteinGym generalize effectively to MaveDB, following a similar distribution of predictions. Panel D illustrates the random subset of 50 proteins used for hyperparameter tuning for fitness prediction. Each point in the plots represents one protein and shows the Spearman correlation averaged across all assays corresponding to the protein (typically one assay per protein). The box plots standardly depict quartiles, medians, and outliers.

E.2 Protein fitness prediction

E.2.1 Datasets

ProteinGym. ProteinGym[8] is the standard benchmark for protein fitness prediction (Notin et al., 2024). The latest, second version of the dataset includes 217 deep mutation scanning experiments (DMSs) across different proteins. We focus on the well-established zero-shot setup of the benchmark and do not experiment with the supervised setup, as it has not yet been fully incorporated into the official codebase at the time of this study. In total, the dataset contains 2.5M mutants with annotated ground-truth fitness. Since ProteinGym does not contain a data split for the zero-shot setup, which was employed in this work, we use the whole dataset as the test set.

[8] https://github.com/OATML-Markslab/ProteinGym

MaveDB dataset. To establish a validation set disjoint from ProteinGym (Notin et al., 2024), we mined MaveDB[9] (Esposito et al., 2019). As of August 1, 2024, the database contains 1178 Multiplexed Assays of Variant Effects (MAVEs), where each assay corresponds to a single protein, measuring the experimental fitness of its variants. We applied quality control filters to remove potentially noisy data. Specifically, we ensured that the UniProt identifier (Consortium, 2023) is valid and has a predicted structure available in the AlphaFold DB (Varadi et al., 2022). We also excluded assays with fewer than 100 variants, as well as those where at least one mutation had a wrongly annotated wild type or where most mutations failed during parsing. Additionally, to ensure no overlap between datasets, we removed any assays whose UniProt identifier matched with those in ProteinGym, ensuring that the validation and test sets contain different proteins.

The described methodology resulted in the MaveDB dataset comprising 676 assays (out of 1178 in the entire MaveDB) with experimental fitness annotations. This corresponds to 483 unique protein sequences and 867 thousand mutations in total. The large size of the dataset, despite the comprehensiveness of ProteinGym containing 217 assays, can be attributed to the fact that many assays in MaveDB were released after the ProteinGym construction (Figure A3A). To ensure the quality of the constructed MaveDB dataset, we validated that representative baselines from ProteinGym generalize to the new assays, following similar distributions of predictions (Figure A3B,C). Finally, for efficiently tuning hyperparameters for fitness prediction models, we sampled 50 proteins (Figure A3D), corresponding to 83 assays comprising 134 thousand variants.

E.2.2 METRICS

Protein fitness labels are not standardized and can vary across different proteins. Nevertheless, the ranking of mutations for a single protein, as defined by fitness labels, can be used to assess the mutation-scoring capabilities of machine learning models. As a result, Spearman correlation is a standard metric for evaluation.

Spearman by phenotype. When computing Spearman correlations, we follow the evaluation protocol proposed in ProteinGym (Notin et al., 2024). First, for each protein, we compute Spearman correlation scores between the predicted ranks of mutations and their corresponding labels. Then, we average the scores across five categories of assayed phenotypes, measuring the effects of mutations: catalytic activity ("Activity"), binding affinity to a target ("Binding"), protein expression levels in a cell ("Expression"), organism growth rate ("Organismal Fitness"), and protein thermostability ("Stability").

Avg. Spearman. We refer to the mean score across the five phenotype categories as "Avg. Spearman". We report the "Avg. Spearman" metric as the mean and standard deviation across five random seeds (Table 2, Table A4).

Spearman by MSA Depth. Following (Notin et al., 2024), we split the performance by the depth of available multiple sequence alignment (MSA), i.e., the number of homologous sequences available, as provided in ProteinGym: "Low depth", "Medium depth", and "High depth", and report the Spearman correlation for each subset individually (Table A4). Specifically, the MSA depth categories in ProteinGym are determined using the following thresholds from Hopf et al. (2017): "Low" is defined as $N_{eff}/L < 1$, "Medium" as $1 < N_{eff}/L < 100$, and "High" as $N_{eff}/L > 100$, where N_{eff} represents the normalized number of effective sequences in the MSA, and L is the sequence length covered in the MSA.

E.2.3 MODELS

ESM2. The ESM2 model is a bidirectional, BERT-like (Devlin, 2018) Transformer trained on millions of protein sequences using masked modeling (Lin et al., 2023). The goal of protein fitness prediction is to predict the effects of mutations, and PLMs are often adapted to this task using zero-shot transfer via log odds ratio (Notin et al., 2024; Meier et al., 2021). Specifically, for a given single- or multi-point mutation, where certain amino acids T are substituted from x_i to x_i^m for each

[9] https://www.mavedb.org

$i \in T$, the fitness prediction via the log odds ratio is defined as:

$$\sum_{i \in T} \Big(\log p(x_i^m | x_{\setminus i}) - \log p(x_i | x_{\setminus i}) \Big), \tag{5}$$

where the sum iterates over mutated positions $i \in T$ with $p(x_i^m | x_{\setminus i})$ and $p(x_i | x_{\setminus i})$ denoting the predicted probabilities of the mutated amino acid and the original one (i.e., wild type), respectively. The conditionals $x_{\setminus i}$ indicate that the input sequence to the model has the position i masked. In this setup, the native (unmutated) sequence, where $T = \emptyset$, has a predicted fitness of 0. Mutations with negative values represent favorable mutations, while positive values correspond to disruptive mutations. We follow the ProteinGym benchmark and use this formula (Notin et al., 2024) to evaluate the fitness prediction capabilities of ESM2. We use the implementation of ESM2 from ProteinGym.

ESM2 + ProteinTTT. ESM2 can be straightforwardly customized with ProteinTTT. Specifically, we treat the Transformer encoder as the backbone f, and the language modeling head, which projects token embeddings to amino acid probabilities, as the pre-training head g. The log odds ratio given by Equation (5) serves as the task-specific head h, which in this case involves the pre-training head g that predicts log probabilities. Overall, we apply ProteinTTT to the pre-trained ESM2 model and, after a pre-defined number of self-supervised fine-tuning steps, score mutations using Equation (5). During customization, we fine-tune all parameters in $g \circ f$ end-to-end except for token and position embeddings. When evaluating ESM2 + ProteinTTT$_{\text{MSA}}$, we use the MSAs curated by the authors of ProteinGym (Notin et al., 2024).

SaProt. We also experiment with a structure-aware protein language model, SaProt (Su et al., 2023). SaProt builds off the ESM2 model but incorporates structural information from predicted protein structures. Specifically, SaProt uses the same Transformer architecture but expands its vocabulary by combining the 20 standard amino acid tokens with 20 structural tokens from the 3Di vocabulary, increasing the total alphabet size to 400. The 3Di tokens capture the geometry of the protein backbone and are generated using VQ-VAE (Razavi et al., 2019), which projects continuous geometric information into discrete tokens and was trained as part of the Foldseek method (van Kempen et al., 2022).

Since SaProt is also a protein language model, it also uses Equation (5) to score variants. However, please note that SaProt, as implemented in ProteinGym (Notin et al., 2024), uses a slightly different version of the log odds ratio. In SaProt, the conditions in the log probabilities in Equation (5) are replaced with $x_{\setminus T}$ instead of $x_{\setminus i}$, not assuming the independence of substitutions. During customization with ProteinTTT, we only mask sequential information and leave the structural part of the tokens unchanged, reflecting the original pre-training setup. We use the implementation of SaProt from ProteinGym[8].

SaProt + ProteinTTT. Since the architecture of SaProt is based on ESM2, the ProteinTTT components f, g, and h remain the same. It means that customization can be applied to the model in the same way as in the case of ESM2 + ProteinTTT discussed above.

ProSST. Finally, we experiment with the state-of-the-art fitness predictor, ProSST (Li et al., 2024). ProSST primarily improves upon SaProt (Su et al., 2023) by incorporating a larger vocabulary of structural tokens and employing disentangled attention mechanisms. Instead of relying on the 3Di alphabet optimized for protein structure search with Foldseek (van Kempen et al., 2022), Li et al. (2024) pre-train a new autoencoder to denoise corrupted protein backbones and cluster the resulting latent space using the K-means algorithm (Lloyd, 1982). Notably, optimal performance for fitness prediction is achieved with $K = 2048$ tokens, compared to just 20 in the 3Di vocabulary used by SaProt. We adopt this model in our experiments. Additionally, disentangled attention in ProSST enhances information propagation between sequence and structure within its Transformer blocks, further improving prediction performance. The model has 110M parameters in total.

ProSST, similarly to ESM2 and SaProt, is pre-trained using masked language modeling applied to protein sequence tokens. To score mutations on the ProteinGym benchmark (Notin et al., 2024), ProSST also uses the log-odds ratio, but in a slightly different way compared to ESM2 and SaProt. Specifically, ProSST performs a single forward pass to predict log probabilities, which are then used to score all mutations. Formally, this approach modifies the log probability condition in Equation (5), replacing $x_{\setminus i}$ with x.

ProSST + ProteinTTT. Similarly to ESM2 and SaProt, we treat the Transformer encoder in ProSST as the backbone f, the masked language modeling head as the pre-training head g, and the log-odds ratio formula as the task-specific head h.

MSA Transformer. Finally, we experiment with MSA Transformer for fitness prediction (Rao et al., 2021). Similar to ESM2 (Lin et al., 2023), MSA Transformer is pre-trained on large protein sequence datasets; however, it is trained on multiple sequence alignments (MSAs) rather than individual sequences.

Since MSA Transformer is also a protein language model, it can be used for fitness prediction in the same way as ESM2, as discussed above, by computing the log-odds ratio over the first sequence in the MSA in this case. We reproduce the results of MSA Transformer on the ProteinGym benchmark with two modifications: (1) we sample a weighted subset of 32 sequences from each MSA instead of 400, and (2) we use only one random seed instead of five for ensembling. These changes significantly reduce computational time while also slightly improving performance compared to the results reported in ProteinGym. This improvement may be explained by the fact that the performance of MSA Transformer saturates with increasing MSA input size (Figure 4 in Rao et al. (2021)).

MSA Transformer + ProteinTTT. We experiment with customizing MSA Transformer to MSA subsamples of varying sizes, ranging from a single target sequence (i.e., customization via Equation (2) with ProteinTTT) to the full MSA subset of 32 sequences (i.e., customization via Equation (4) with ProteinTTT$_{\text{MSA}}$). We observe that applying ProteinTTT$_{\text{MSA}}$ to MSA Transformer with a batch size of 32 disrupts performance, while reducing the input MSA subsample size mitigates this effect. Ultimately, MSA Transformer + ProteinTTT results in a slight performance improvement.

E.3 Protein function prediction

E.3.1 Datasets

TPS dataset. For the evaluation of terpene substrate classification, we use the largest available dataset of characterized TPS enzymes from Samusevich et al. (2024) and repurpose the original 5-fold cross-validation schema. We focus on the most challenging TPS sequences, defined as those predicted by the TPS detector, proposed by the dataset authors, with confidence scores below 0.8. This filtering results in 104, 98, 113, 100, 97 examples in the individual folds.

setHard. For the test evaluation of subcellular location prediction, we use the setHard dataset constructed by Stärk et al. (2021). The dataset was redundancy-reduced, both within itself and relative to all proteins in DeepLoc (Almagro Armenteros et al. (2017); next paragraph), a standard dataset used for training and validating machine learning models. The setHard dataset contains 490 protein sequences, each annotated with one of ten subcellular location classes, such as "Cytoplasm" or "Nucleus". Since we use ESM-1b (Rives et al., 2021) in our experiments with the dataset, we further filter the data to 432 sequences that do not exceed a length of 1022 amino acids. This step, consistent with Stärk et al. (2021), ensures that ESM-1b can generate embeddings for all proteins.

DeepLoc. For hyperparameter tuning in the subcellular location prediction task, we use the test set from the DeepLoc dataset (Almagro Armenteros et al., 2017). Similar to setHard, DeepLoc assigns labels from one of ten subcellular location classes. The dataset contains 2768 proteins, which we further filter to 2457 sequences that do not exceed a length of 1022 amino acids, ensuring compatibility with the embedding capabilities of ESM-1b. Since setHard was constructed to be independent of DeepLoc, setHard provides a leakage-free source of data for validation.

E.3.2 Metrics

mAP, AUROC. The TPS substrate prediction problem is a 12-class multi-label classification task over possible TPS substrates. Therefore, we assess the quality of the predictions using standard multi-label classification metrics such as mean average precision (mAP) and area under the receiver operating characteristic curve (AUROC) averaged across individual classes. These metrics were used in the original TerpeneMiner paper (Samusevich et al., 2024). We report the performance by

averaging the metric values concatenated across all validation folds from the 5-fold cross-validation schema.

Accuracy, MCC, F1-score. To evaluate the performance of subcellular location prediction methods, we use standard classification metrics as employed in Stärk et al. (2021). Accuracy standardly measures the ratio of correctly classified proteins, while Matthew's correlation coefficient for multiple classes (MCC) serves as an alternative to the Pearson correlation coefficient for classification tasks (Gorodkin, 2004). The F1-score, the harmonic mean of precision and recall, evaluates performance from a retrieval perspective, balancing the trade-off between false positives and false negatives.

E.3.3 MODELS

TerpeneMiner. TerpeneMiner is a state-of-the-art method for the classification of terpene synthase (TPS) substrates (Samusevich et al., 2024). The model consists of two parallel tracks. Given a protein sequence, TerpeneMiner first computes its ESM-1v embedding (Meier et al., 2021) and a vector of similarities to the functional domains of proteins from the training dataset, based on unsupervised domain segmentation of AlphaFold2-predicted structures (Jumper et al., 2021). The ESM-1v embedding and the similarity vector are then concatenated and processed by a separately trained random forest, which predicts TPS substrate class probabilities.

In our experiments, we use the "PLM only" version of the model, which leverages only ESM-1v embeddings. This version exhibits a minor performance decrease compared to the full model but exactly follows a Y-shaped architecture, allowing us to validate the effectiveness of ProteinTTT for predicting TPS substrates. We use the implementation of TerpeneMiner available at the official GitHub page [10].

TerpeneMiner + ProteinTTT. When applying ProteinTTT to TerpeneMiner, we treat the frozen ESM-1v model as a backbone f, its language modeling head as a self-supervised head g, and the random forest classifying TPS substrates as a downstream supervised head h.

Light Attention. We use Light attention (Stärk et al., 2021) as a representative baseline for subcellular location prediction. Light attention leverages protein embeddings from a language model, which in our case is ESM-1b (Rives et al., 2021). The model processes per-residue embeddings via a softmax-weighted aggregation mechanism, referred to as light attention, which operates with linear complexity relative to sequence length and enables richer aggregation of per-residue information, as opposed to standard mean pooling. We re-train the model using ESM-1b embeddings on the DeepLoc dataset (Almagro Armenteros et al., 2017) using the code from the official GitHub page[11].

Light attention + ProteinTTT. When applying ProteinTTT to Light attention, we treat the frozen ESM-1b as the backbone f, the language modeling head of ESM-1b as the self-supervised head g, and the Light attention block as the fine-tuning head h.

F CASE STUDY DETAILS

F.1 MODELING ANTIBODY-ANTIGEN LOOPS

We download the SAbDab dataset from the official website[12](Dunbar et al., 2014). We apply ProteinTTT to targets with low-confidence ESMFold predictions (pLDDT < 70) and remove sequences longer than 400 residues due to GPU memory limitations. This results in a final set of 175 antibody and 814 antigen chains. We predict the full structures using ESMFold+ProteinTTT (with the same hyperparameters tuned on the CAMEO validation set specified in Table A3) and compute LDDT scores against the corresponding PDB structures to assess local errors, which are particularly relevant for loop regions. For antibodies, we evaluate the complete structures, while for complementarity-determining regions (CDRs), we extract the CDR substructures as annotated in SAbDab according to Chothia numbering (Chothia & Lesk, 1987) and calculate LDDT on these regions.

[10] https://github.com/pluskal-lab/TerpeneMiner

[11] https://github.com/HannesStark/protein-localization

[12] https://opig.stats.ox.ac.uk/webapps/sabdab-sabpred/sabdab

Table A3: **Hyperparameters used for adapting ProteinTTT to individual models.** The optimal hyperparameters were estimated using validation datasets corresponding to each of the considered tasks: *Fitness prediction*, *Structure prediction*, and *Function prediction*. Comma-separated lists show the values used for hyperparameter grid search, while the final values selected for computing the test results are highlighted in **bold**. Low-rank adaptation (LoRA) was only used with ESMFold, containing 3 billion parameters in the ESM2 backbone. Please note that we did not tune the number of customization steps, as adjusting the learning rate and batch size effectively controls the expected performance under the fixed number of steps, as shown in Figure A10. Therefore, we used 30 steps in all our experiments. The only exception was ESM3 + ProteinTTT, where the number of steps was set to 50 during initial experiments with different models/tasks conducted in parallel before standardizing the number of steps to 30. Methods marked with an asterisk ("*") used a slightly different calculation of the loss function. Specifically, the loss was propagated over all tokens, including special and non-masking tokens, while averaging the loss across all tokens simultaneously, rather than first averaging over sequences. This approach was used in the early stages of development, and we provide it in our codebase via `loss_kind = "unnormalized_cross_entropy"`. Please note that MSA Transformer always uses 1 MSA in a batch and the "Batch size" represents the number of sequences in this MSA with the target sequence always present as the first one.

	Learning rate	Batch size	Grad. acc. steps	Steps (Conf. func. c)	LoRA rank r	LoRa α
Fitness prediction						
ESM2 (35M) + ProteinTTT *	4e-5, **4e-4**, 4e-3	**4**	4, 8, **16**, 32, 64	**30**	-	-
ESM2 (650M) + ProteinTTT *	**4e-5**, 4e-4, 4e-3	**4**	4, 8, **16**, 32	**30**	-	-
SaProt (35M) + ProteinTTT *	4e-5, **4e-4**, 4e-3	**4**	4, **8**, 16, 32	**30**	-	-
SaProt (650M) + ProteinTTT *	**4e-5**, 4e-4, 4e-3	**2**, 4	4, 8, **16**, 32	**30**	-	-
ProSST (K=2048) + ProteinTTT *	**1e-5**, 4e-5, 4e-4, 4e-3	**4**	4, **8**, 16, 32	**30**	-	-
ESM2 (650M) + ProteinTTT$_{\text{MSA}}$ *	4e-6, 1e-5, 4e-5, 4e-4, **4e-3**	**4**	**2**, 4	50, **100**	-	-
MSA Transformer + ProteinTTT	1e-6, 3e-6, 1e-5, 3e-5, **1e-4**	**1**, 4, 8, 16, 32	1, 2, 4, **8**	**30**	-	-
Structure prediction						
ESMFold + ProteinTTT	**4e-4**	**4**	**4**, 8, 32, 64	**30 (pLDDT)**	4, **8**, 32	8, 16, **32**
HelixFold-Single + ProteinTTT	**4e-4, 1e-3**	**4, 8, 16**	**1**	**30 (pLDDT)**	-	-
ESM3 + ProteinTTT	1e-4, 4e-4, **1e-3**	**2**	**1**, 4, 16	**50 (pLDDT)**	-	-
Function prediction						
TerpeneMiner + ProteinTTT	**4e-4**, 1e-3	**2**	**2**, 4, 8	**30**	-	-
Light attention + ProteinTTT	4e-4, 1e-3, **3e-3**	**2**	**2**, 4	**30**	-	-

F.2 Expanding known structures of viral proteins

We use BFVD version `archived/2023_02_v2`[13]. This version contains maximum-pLDDT structures from predictions generated by two strategies: (i) ColabFold (Mirdita et al., 2022) with MSAs constructed using Logan (Chikhi et al., 2024), and (ii) ColabFold with 12 additional recycle steps and MSAs constructed using Logan. In Figure 5, we also report pLDDT values for BFVD version `archived/2023_02_v1`, where structures are simply obtained from ColabFold with MSAs from Logan. We re-predict structures using ESMFold and then ESMFold+ProteinTTT for cases where the original ESMFold predictions have pLDDT < 70 (with the same hyperparameters tuned on the CAMEO validation set, as specified in Table A3).

G Extended results

In this section, we provide additional results on test sets (Section G.1), discuss validation performance (Section G.2), and analyze the runtime performance of customization (Section G.3).

G.1 Detailed test performance

In this section, we provide details on the test performance. Specifically, Table A4 shows that customization with ProteinTTT primarily enhances performance on challenging targets, characterized by a low number of similar proteins in sequence databases, as measured by MSA depth. Additionally, we provide an example illustrating how ProteinTTT substantially improves the correlation between ESM2-predicted fitness and ground-truth stability by better identifying disruptive mutations in the protein core (Figure A5).

[13] `https://bfvd.steineggerlab.workers.dev`

Next, Figure A6 shows the distribution of ProteinTTT effects: in many cases, customization has minimal impact on performance; often, it leads to substantial improvements; and in rare cases, customization results in a decrease in performance. This positions ProteinTTT as a method for enhancing prediction accuracy, while a comprehensive analysis of its failure modes remains an important direction for future research. While we demonstrate these effects using a protein folding example, we observe a similar distribution of ProteinTTT impact across the tasks.

We also observe that the overall trend of customization with ProteinTTT generally leads to improved performance, with robust consistency across random seeds. However, the progression of the performance curve can be rugged, particularly in classification tasks, where substantial changes in the underlying representations are required to shift the top-predicted class in the discrete probability distribution (Figure A8).

G.2 Validation performance

This section discusses the performance of ProteinTTT on validation data. Table A5 illustrates the validation performance of the tested methods for fitness prediction on our newly constructed MaveDB dataset. ProteinTTT enhances the performance of all the methods.

This section discusses hyperparameter tuning on validation data. Table A3 provides the grid of hyperparameters explored for each model and its size, as well as specifies the optimal hyperparameters suitable for downstream applications. Figure A10 demonstrates the trend of hyperparameter tuning with optimal hyperparameter combination balancing underfitting and overfitting to a single target protein. While most hyperparameter configurations lead to overall improvements when using customization with ProteinTTT, poorly chosen hyperparameters can have detrimental effects due to rapid overfitting. However, with a reliable predicted confidence measure, such as pLDDT, the appropriate customization step can be selected to mitigate overfitting. Figure A11 demonstrates that when using ESM3 + ProteinTTT with pLDDT-based step selection for protein folding, all hyperparameter configurations result in improved performance compared to the base ESM3 model.

G.3 Runtime performance

In this section, we demonstrate that customization with ProteinTTT can be done efficiently, with an acceptable computational overhead. Specifically, we show that ESMFold, known for being a faster alternative to more performant methods such as AlphaFold2 (Jumper et al., 2021) or AlphaFold3 (Abramson et al., 2024), still remains in the category of lightweight methods even with ProteinTTT customization (Figure A4).

This observation highlights the practical utility of ProteinTTT. For example, ESMFold enabled structural characterization of large metagenomics data (>617 million metagenomic sequences), which would be infeasible with AlphaFold2 (Lin et al., 2023). Nevertheless, the original ESMFold has high confidence predictions only for 36% of sequences from the metagenomic database, while the other 392 million sequences remain with low or medium confidence predictions. At the same time, ESMFold + ProteinTTT enables more accurate predictions compared to the original ESMFold (Figure A6 suggests that ESMFold + ProteinTTT significantly improves predictions in almost 40% of challenging sequences). It means that applying ESMFold + ProteinTTT to these remaining sequences could significantly expand the metagenomic atlas characterized by ESMFold.

Table A4: **ProteinTTT performance on ProteinGym depending on MSA depth.** MSA depth reflects the number of available proteins similar to the target protein and, when using large protein language models, can be interpreted as a measure of the representation of similar proteins in the training data (Section E.2.2). Customization with ProteinTTT primarily improves performance on difficult targets, with low MSA depth. Standard deviations are calculated over 5 random seeds but are omitted in the right panel for brevity, where the maximum standard deviation does not exceed 0.0004.

	Avg. Spearman ↑	Spearman by MSA depth ↑		
		Low depth	Medium depth	High depth
ESM2 (35M) (Lin et al., 2023)	0.3211	0.2394	0.2707	0.451
ESM2 (35M) + ProteinTTT (Ours)	**0.3407 ± 0.00014**	**0.2445**	**0.3144**	**0.4598**
SaProt (35M) (Su et al., 2023)	0.4062	0.3234	0.3921	0.5057
SaProt (35M) + ProteinTTT (Ours)	**0.4106 ± 0.00004**	**0.3253**	**0.3972**	**0.5091**
ESM2 (650M) (Lin et al., 2023)	0.4139	0.3346	0.4063	**0.5153**
ESM2 (650M) + ProteinTTT (Ours)	**0.4153 ± 0.00003**	**0.3363**	**0.4126**	0.5075
SaProt (650M) (Su et al., 2023)	0.4569	0.3947	**0.4502**	**0.5448**
SaProt (650M) + ProteinTTT (Ours)	**0.4583 ± 0.00001**	**0.3954**	0.4501	0.5439
ProSST (K=2048) (Li et al., 2024)	0.5068	0.4731	**0.5107**	0.5749
ProSST (K=2048) + ProteinTTT (Ours)	**0.5087 ± 0.00004**	**0.4809**	0.5104	**0.5750**

Table A5: **Performance of ProteinTTT on the MaveDB dataset.** In this work, we use our newly constructed MaveDB dataset as a validation fold for tuning the ProteinTTT hyper-parameters for fitness prediction. For computational efficiency, we only select a subset of 50 proteins (Section E.2.1) and do not run customization across multiple random seeds to estimate standard deviations. The performance shown was calculated by first aggregating correlations per assay, and then per protein (some assays correspond to the same protein).

	Avg. Spearman ↑
ESM2 (35M) (Lin et al., 2023)	0.4458
ESM2 (35M) + ProteinTTT (Ours)	**0.4593**
ESM2 (650M) (Lin et al., 2023)	0.4568
ESM2 (650M) + ProteinTTT (Ours)	**0.4604**
SaProt (650M) (Su et al., 2023)	**0.4926**
SaProt (650M) + ProteinTTT (Ours)	**0.4926**
SaProt (35M) (Su et al., 2023)	0.5251
SaProt (35M) + ProteinTTT (Ours)	**0.5271**
ProSST (K=2048) (Li et al., 2024)	0.5444
ProSST (K=2048) + ProteinTTT (Ours)	**0.5462**

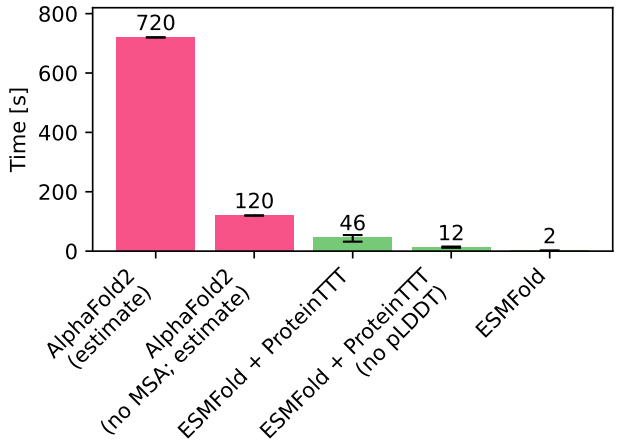


Figure A4: **Running time of ESMFold + ProteinTTT.** For ESMFold and its variants, the median and interquartile ranges of running times on the CAMEO test set are shown using a single NVIDIA A100 GPU. For AlphaFold2, we use estimates from Lin et al. (2023). Specifically, a forward pass through AlphaFold2 is approximately 60 times more computationally expensive than ESMFold (e.g., `AlphaFold2 (no MSA; estimate)`: $2 \times 60 = 120$ seconds), with additional MSA construction taking at least 10 minutes using standard pipelines (`AlphaFold2 (estimate)`: $2 \times 60 + 10 \times 60 = 720$ seconds). ESMFold + ProteinTTT (30 steps) involves LoRA parameter updates, along with forward passes at each customization step to estimate pLDDT and select the structure with the highest predicted confidence. Disabling pLDDT significantly reduces computational overhead (`ESMFold + ProteinTTT (no pLDDT)` compared to `ESMFold + ProteinTTT`), but may require careful parameter tuning (Section G.2). Overall, ESMFold + ProteinTTT maintains the speed advantage of ESMFold, and is at least an order of magnitude faster than AlphaFold2.

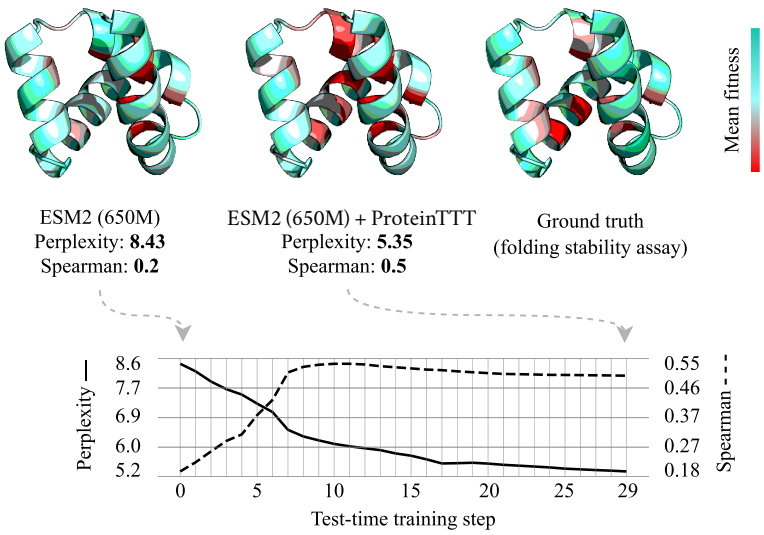


Figure A5: **Example of protein fitness prediction upon single-sequence model customization with ProteinTTT.** Fitness predictions from ESM2 (650M) show poor correlation with experimental fitness values in the ProteinGym test set measured by the stability assay "UBR5_HUMAN_Tsuboyama_2023_1I2T" (Tsuboyama et al., 2023) (left). ESM2 + ProteinTTT achieves significantly higher correlation, likely due to improved detection of disruptive mutations in the protein core that impact protein stability (middle). The ground-truth fitness data aligns with the customized model, showing that residues crucial for stability (i.e., having negative mean fitness) are concentrated in the protein core (right). Residue colors represent the mean fitness upon all single-point substitutions (with the exception of several missing mutations in the ground-truth data), with red indicating residues where mutations have detrimental effects on average.

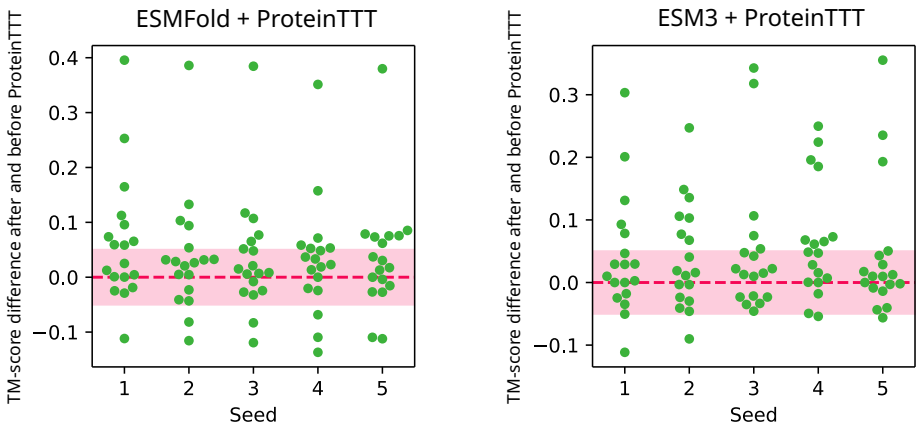


Figure A6: **Per-protein performance of ESMFold + ProteinTTT and ESM3 + ProteinTTT on the CAMEO test set.** The y-axis shows the change in TM-score after applying customization with ProteinTTT, with higher values indicating improvement. The x-axis represents performance across five random seeds. The red dashed line marks no change in TM-score (TM-score difference $= 0$), and the pink band represents minor changes in TM-score ($-0.05 <$ TM-score difference < 0.05), which we do not consider significant. Each point in the swarm plot corresponds to a single protein from the CAMEO test set. On average, applying ProteinTTT to ESMFold improves the structure predictions for 7 out of 18 proteins, with 2 showing degradation. The rest of the proteins are not significantly affected. Similarly, applying ProteinTTT to ESM3 results in 6 improvements out of 18 proteins, with 1 case of degradation.

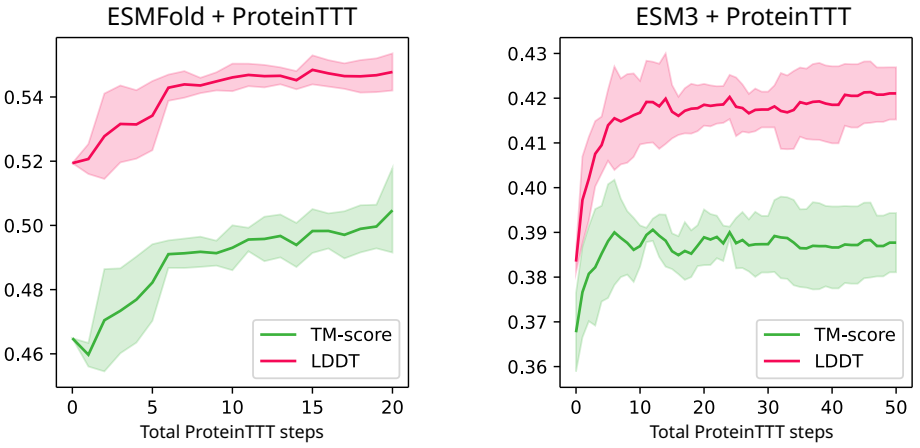


Figure A7: **Test performance of ESMFold + ProteinTTT and ESM3 + ProteinTTT on the CAMEO test set depending on the total number of customization steps.** The x-axis shows the averaged performance across all test proteins, with error bars representing the standard deviation across five random seeds. The y-axis metrics correspond to the structure with the highest pLDDT score up to the given step. While an increased number of ProteinTTT steps generally enhances performance, only a few steps (e.g., five) may suffice to achieve significant performance improvement.

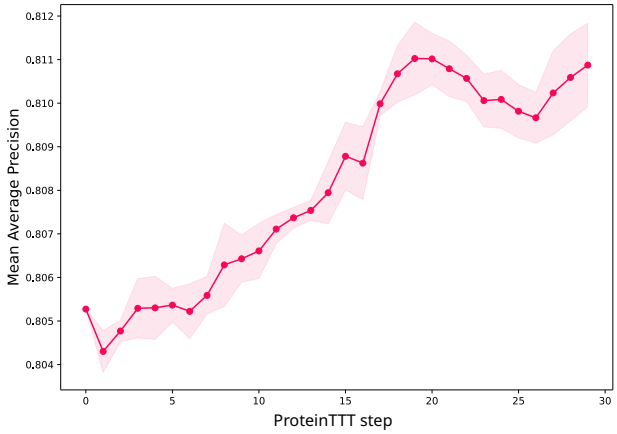


Figure A8: **Test performance of TerpeneMiner + ProteinTTT across customization steps.** The performance is averaged across all 512 proteins in the dataset, with error bars representing the standard deviation across 5 random seeds.

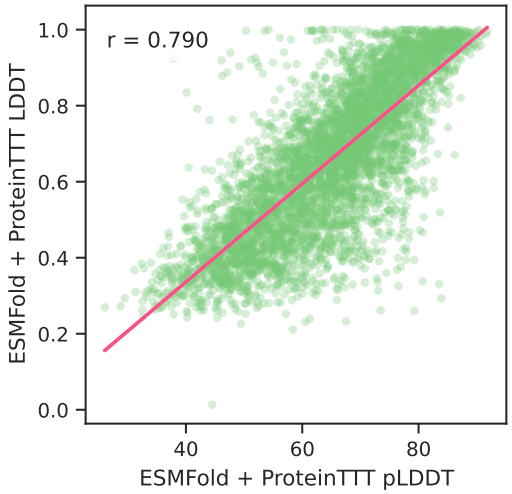


Figure A9: **ESMFold + ProteinTTT pLDDT correlates with ESMFold + ProteinTTT LDDT.** The evaluation was performed on 4,894 AlphaFold2 reference structures from the BFVD database with pLDDT > 90. Here, $r = 0.790$ denotes the Pearson correlation coefficient.

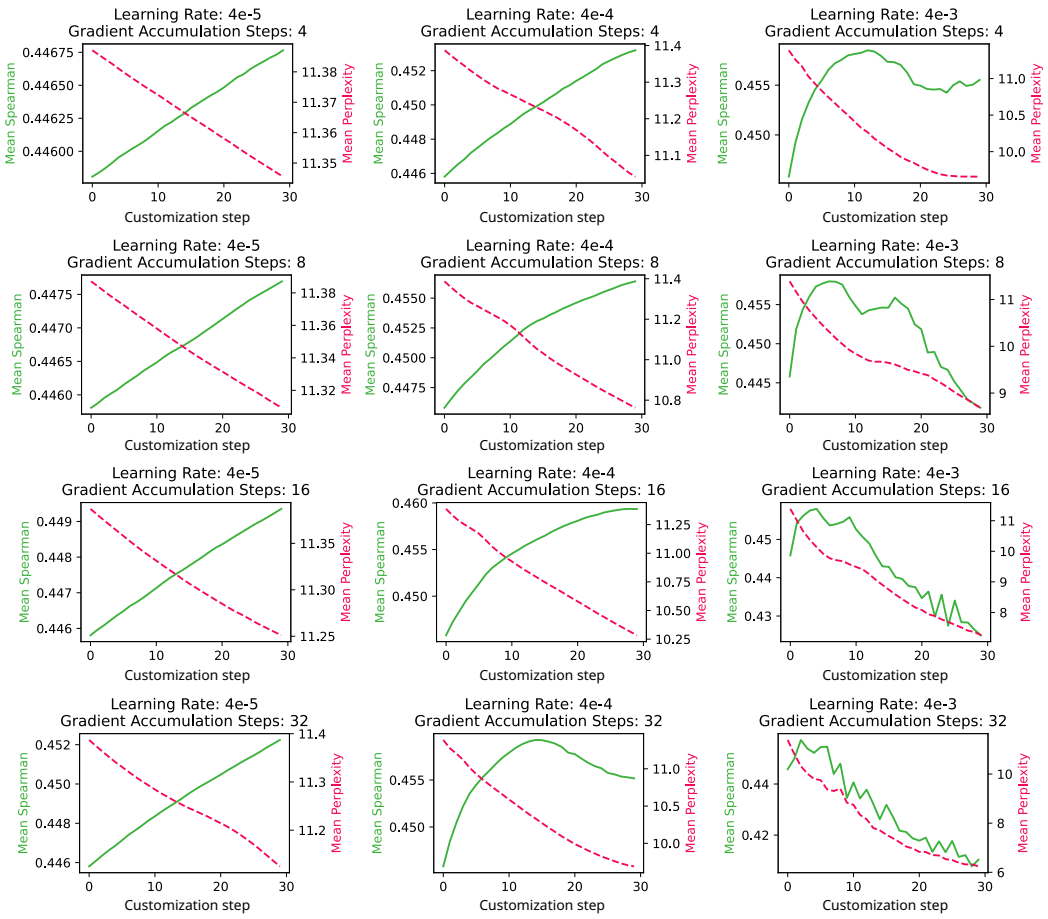


Figure A10: **Dependence on ProteinTTT hyperparameters for customized fitness prediction.** Each plot shows the progression of Spearman correlation (green) increasing alongside a decrease in perplexity (pink) for each customization step, averaged across all assays in the MaveDB validation dataset. The model used is ESM2 (35M) + ProteinTTT, and the grid displays the combinations of different numbers of gradient accumulation steps (i.e., effective batch sizes; shown in rows, increasing from top to bottom) and learning rates (columns, increasing from left to right). As the learning rate increases and the number of gradient accumulation steps grows, the model reaches peak performance more quickly but begins to overfit to a target protein. The optimal hyperparameter combination (learning rate = 4e-4, gradient accumulation steps = 16) lies near the center of the grid, balancing between underfitting and overfitting to a target protein. Notably, the figure demonstrates that, although ProteinTTT involves three main hyperparameters (batch size, learning rate, and the number of steps), there are effectively only two degrees of freedom controlling the performance of the model. In other words, by keeping the number of steps constant (e.g., 30), the expected performance can be controlled by adjusting the learning rate and the batch size.

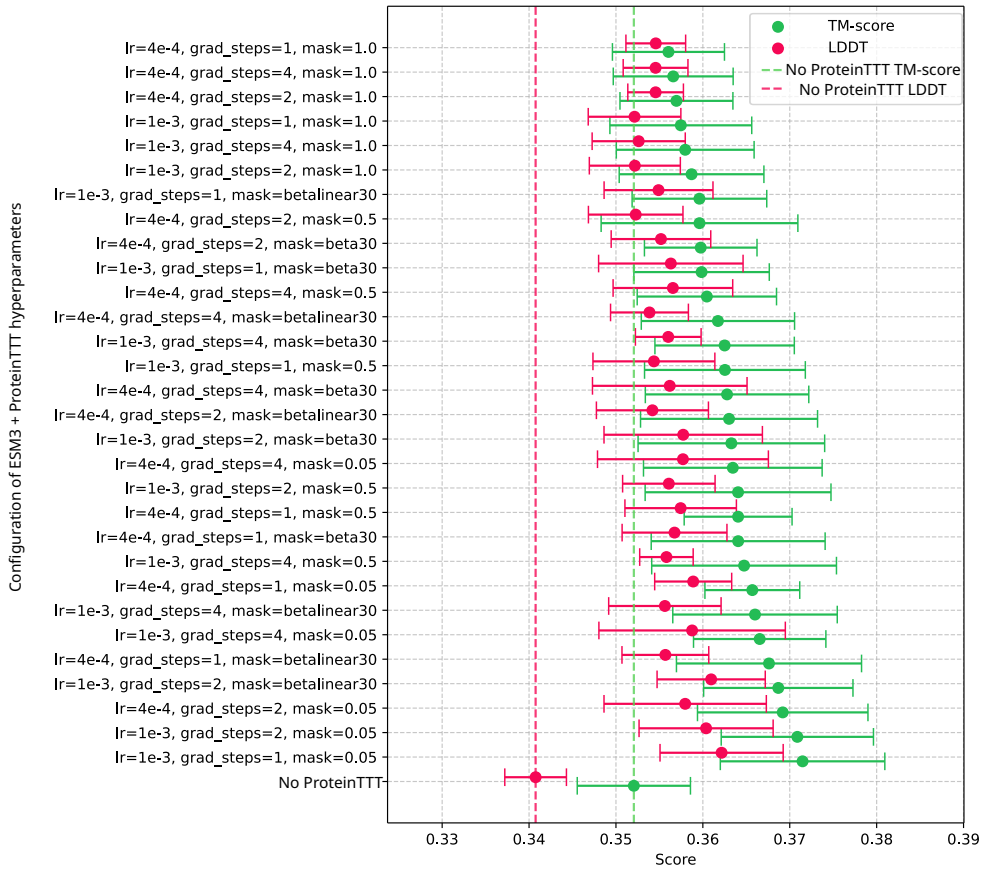


Figure A11: **Hyperparameter search for protein structure prediction with ESM3 + ProteinTTT.** We conducted a comprehensive grid search based on three key hyperparameters: learning rate (denoted as "lr"), number of gradient accumulation steps (denoted as "grad_steps"; with the batch size of two), and masking strategy (denoted as "mask"). We explored two learning rates, 4e-4 and 1e-3, three gradient accumulation step values of 1, 4, and 16, and five different masking strategies: uniform sampling of 0.05, 0.5, and 1.0 fractions of amino acids, as well as the "beta30" and "betalinear30" distributions proposed in the ESM3 paper (Hayes et al., 2024). Each row in the table presents the mean TM-score and LDDT metrics with standard deviations across five random seeds on the CAMEO validation fold. The last row, denoted as "No ProteinTTT", shows the performance of ESM3 without customization. The results indicate that ESM3 + ProteinTTT is robust to the choice of hyperparameters and consistently outperforms the base model across all configurations. We selected the configuration from the last row (excluding "No ProteinTTT") to compute the results on the test fold. For the hyperparameter search, we used 30 customization steps instead of 50 to reduce computation time.