LEVERAGING BIOLOGICAL PRIORS FOR PROTEIN FITNESS PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Accurately predicting the properties of proteins is a foundational problem in protein science and engineering. Models trained across experimental assays can learn transferable patterns, but their success depends on the quantity and quality of available data. On the other hand, biological priors informed by signals such as sequence and structure are valuable for predictions in the low-data regime, but they may encode restrictive assumptions that do not accurately reflect empirical evidence. We introduce PROFICIENT, a data-efficient protein fitness prediction framework that bridges these approaches. First, we incorporate sequence and structural information through protein language model embeddings and inverse folding features. Second, rather than relying solely on experimental data, we turn biological models into synthetic data generators, pretraining on the resulting fitness prediction tasks before finetuning on real assays. We find that synthetic pretraining improves downstream performance over training on experimental data alone, and our approach outperforms strong baselines across held-out ProteinGym assays.

1 INTRODUCTION

Protein fitness prediction aims to predict how changes to a protein's amino acid sequence will impact its biological function or biophysical properties. This task is important both for understanding basic biology as well as for drug discovery and development. Despite advances in high-throughput experimental assessment, many protein engineering tasks remain data-limited: only a small fraction of the combinatorial sequence space can be labeled, and models must often make useful predictions from hundreds, tens, or even fewer labeled variants (Notin et al., 2023a; Zhou et al., 2024).

One approach to this problem is to transfer knowledge from existing biological data. Pretrained protein language models (PLMs) have been shown to capture useful representations of protein sequences, and they can be used for zero-shot prediction or as features for supervised prediction (Lin et al., 2022; Madani et al., 2023; Notin et al., 2022). There are also growing datasets of labeled protein variants from high-throughput assays such as deep mutational scanning (DMS) (Fowler & Fields, 2014). Supervised protein fitness predictors can learn directly from these labeled protein variants, and they can incorporate additional features such as structure or evolutionary information (Gelman et al., 2021; Chen et al., 2023; Biswas et al., 2021). More recently, in-context and meta-learning approaches have been developed to adapt pretrained models to new assays with limited data (Beck et al., 2025; Teufel et al., 2025). These methods demonstrate that models can learn patterns that transfer across assays. However, the quantity, quality, and diversity of real protein fitness data can constrain the performance of the models.

A complementary approach focuses on developing strong inductive biases through biological priors. In the low-data regime, strong inductive biases are particularly valuable. Protein fitness landscapes are shaped by complex interactions between sequence, structure, and function, and models that encode these relationships can make effective predictions. Gaussian processes (GPs), for example, provide a principled framework for encoding prior knowledge by specifying a kernel over protein variants. Protein-specific kernels can incorporate many useful notions of similarity, such as sequence similarity or structural proximity, and recent work has developed sophisticated kernels which are strong predictors of protein fitness landscapes (Groth et al., 2024; Jokinen et al., 2018; Parkinson & Wang, 2023). These priors contain useful assumptions, but their form also limits the types of

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

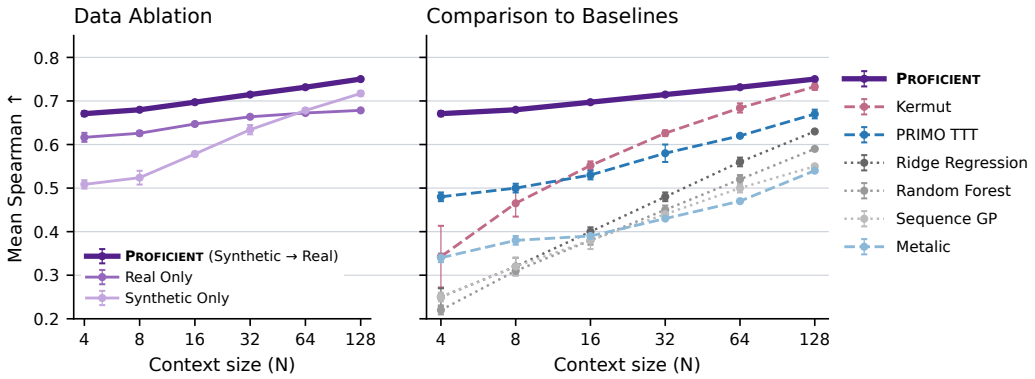


Figure 1: **Combining biological priors with real data enables accurate protein fitness prediction in low-data settings. (Left)** Synthetic pretraining on biological priors like Kermut followed by finetuning on real assay data improves performance over training on either synthetic or real data alone. **(Right)** PROFICIENT outperforms existing baselines at every context size, with the largest gains at small N. We report the mean Spearman correlation across 33 held-out ProteinGym assays, with error bars showing ± 1 standard deviation over five evaluation draws.

relationships they can capture. How can we combine the benefits of strong biological priors with the flexibility of learning from real data?

We introduce PROFICIENT, a data-efficient protein fitness prediction framework that incorporates biological knowledge through its inputs and its training procedure. First, we combine protein language model embeddings and inverse folding features in a flexible protein encoder. These inputs provide complementary sequence and structural information, and the encoder learns to combine them according to their relevance to the task. Second, we turn biological priors into synthetic data generators. Because experimental assays provide a limited number of examples which show the relationships between sequence and function, we first pretrain the model on synthetic tasks generated from these priors. We then finetune on real assay data, allowing the model to refine the biological relationships learned during pretraining to the specific structure of real fitness landscapes. Together, these components allow PROFICIENT to achieve strong performance across diverse protein fitness prediction tasks in the low-data regime, as seen in Figure 1.

2 Background and Related Work

2.1 Protein fitness prediction

A protein is a sequence of amino acids $x = (x_1, \ldots, x_L)$ over a discrete alphabet, which folds into a structure that determines its biological function. The fitness of a protein broadly refers to a scalar property which measures a specific attribute of the protein, such as thermostability or binding affinity. In supervised protein fitness prediction, we are given a labeled dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ of protein sequences and their corresponding fitnesses, and the goal is to predict y^* for test sequence x^*.

In practice, this labeled data typically comes from *in vitro* protein assays, sets of variants of a particular protein sequence and their corresponding fitness values as measured by high-throughput methods such as deep mutational scanning (DMS) (Fowler & Fields, 2014). Standardized benchmarks like ProteinGym (Notin et al., 2023a) aggregate assays, and we use this setting to evaluate our models.

2.2 Supervised and Pretrained Models

A complementary line of work aims to learn protein fitness directly from real data. Supervised models are trained on labeled protein variants and use features like structure or sequence (Gelman et al., 2021; Chen et al., 2023; Biswas et al., 2021). The methods can learn complex relationships, but may overfit in data-constrained settings. Protein language models (PLMs) instead pretrain on large corpora of unlabeled protein sequences. Masked language models such as ESM-2 (Lin et al., 2022) and

autoregressive models such as ProGen2 (Nijkamp et al., 2023) can be used to generate embeddings for supervised prediction or to perform zero-shot prediction by scoring variants according to their likelihood. Models such as Tranception (Notin et al., 2022) further combine PLM likelihoods with sequence information from related proteins. However, PLMs are not typically designed for fitness prediction, and it may be difficult for them to effectively leverage the limited labeled data.

Recent methods address low-data fitness prediction more directly. ProteinNPT (Notin et al., 2023b) uses pretrained protein language-model embeddings and attention across variants and observed labels, but trains its network separately for each assay. Metalic (Beck et al., 2025) uses meta-learning to learn across assays and adapt to new assays with limited data; and PRIMO (Teufel et al., 2025) uses in-context learning and test-time training to adapt PLMs to new assays with limited data. These methods demonstrate that models are able to adapt to new prediction tasks with limited data. However, the experimental assays used to learn this adaptation cover a limited range of fitness landscapes, and it may be difficult to learn the full range of relationships between sequence, structure, and fitness.

2.3 Biological Priors with Gaussian Processes

Gaussian processes (GPs) provide a framework for supervised prediction in which assumptions about the fitness landscape are encoded through a kernel. For proteins, the kernel defines the similarity between variants. Early work showed that GPs can guide protein discovery by modeling sequence-function landscapes under limited data (Romero et al., 2013). More recent work has developed kernels which incorporate biological structure more directly, including sequence or structural similarity (Groth et al., 2024; Zhou et al., 2025). Biological knowledge can also be incorporated through simulated measurements, and mGPfusion combines simulated and experimental data within a GP model (Jokinen et al., 2018).

These models enable biological knowledge to be useful for protein fitness prediction. However, their performance depends on the assumptions encoded in the kernel, which may not fully capture the structure of experimental fitness landscapes.

2.4 Learning from Synthetic Data

Synthetic data provides another way to incorporate biological knowledge. We can construct training tasks that reflect useful assumptions about the prediction problem by specifying a data generation procedure. Prior-data fitted networks (Müller et al., 2022) follow this approach. A model is trained on tasks sampled from a prior distribution to predict query labels given a context set, and it learns to approximate the posterior predictive distribution under that prior. Models trained on purely synthetic data have been shown to achieve strong performance in domains like tabular data (Hollmann et al., 2025) and time series (Taga et al., 2025), but there has been limited exploration of this approach for protein fitness prediction.

Synthetic training enables models to learn from many types of tasks without requiring additional experimental measurements. However, these tasks are only as useful as the assumptions encoded in the data generation procedure. In this work, we combine synthetic training with real data to learn from both biological priors and empirical evidence.

3 Learning from Real Protein Data

We first consider the setting where we train a model directly on experimental assays. Rather than training a different model for each assay, we instead train a shared model to predict protein fitness values conditioned on a context set of labeled variants. This approach allows the model to learn patterns that transfer across assays, and it enables the model to adapt to new assays without updating model parameters at inference.

3.1 Model Architecture

Our model combines a protein encoder with an in-context learning backbone. The protein encoder is designed to integrate sequence and structural information, and it produces a single embedding for

each variant. The in-context learning backbone then takes the context set of labeled variants and the query variants as input, and it predicts a distribution over the fitness values for the query variants.

Biological Inputs We adapt the combination of sequence and structural features from Kermut (Groth et al., 2024), a Gaussian process kernel which demonstrates strong performance on protein fitness prediction tasks.

First, we obtain *sequence information* for each variant by mean-pooling the residue-level embeddings from the frozen ESM-2 650M model (Lin et al., 2022). This representation summarizes the sequence-level information of the variant, and it captures evolutionary and functional signals learned from large corpora of protein sequences. We also incorporate a zero-shot fitness score from ESM-2, which contains additional evolutionary information about the variant.

We also want to capture *structural information*. We use inverse-folding features from ProteinMPNN (Dauparas et al., 2022), where we extract the predicted amino acid distribution and the log-likelihood of the observed mutant at each mutated residue. We also provide the three-dimensional coordinates of each mutated residue from an AlphaFold2 structure of the wild-type protein (Jumper et al., 2021). These features provide complementary information about the local structural environment of the mutation, and they can help the model learn how structural context influences fitness.

Combining Features We combine the sequence and structural features through a learned gating mechanism. We first project the pooled ESM-2 embeddings through a sequence encoder, and a separate encoder jointly encodes the remaining features into a representation of the same size. The model uses both representations to compute an elementwise sigmoid gate, which is then used to add the gated sequence and structural representations together to produce a single embedding for each variant. This gating mechanism lets the contribution of the sequence and structural features vary across variants.

In-context Learning The embeddings for each sequence are then passed to an in-context learning backbone, which uses masked self-attention to learn relationships between the context and query variants, similar to the setup used in prior-data fitted networks (Müller et al., 2022). We first process the variant embeddings through transformer blocks. We then add a linear embedding of the observed fitness values for the context variants, and we use additional transformer blocks to learn relationships between the context and query variants. Specifically, the context variants self-attend to one another, while the query variants attend to the context variants but not to one another. This allows the model to learn how to use the context set to make predictions for each query variant independently. A final linear layer outputs the mean and log-variance of a Gaussian predictive distribution for each query.

3.2 Training and Evaluation

We use measured protein fitness values from the ProteinGym benchmark (Notin et al., 2023a), which aggregates deep mutational scanning (DMS) assays from multiple sources. We use PRIMO's split; training, validation, and evaluation details are provided in Appendix D. Each training task consists of samples from a single assay, which we split into a context set of $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ and a query set of $\mathcal{Q} = \{(x_i, y_i)\}_{i=1}^{M}$. During training, we sample random tasks with varying context sizes $N \in [2, 128]$ to expose the model to a range of low-data settings. The model, which takes the context set and query inputs as input, predicts a distribution over the fitness values for the query points. We minimize the negative log-likelihood of the true fitness values for the query set $\mathcal{L}_{\text{NLL}}(\theta) = -\sum_{i=1}^{M} \log p_\theta(y_i \mid x_i, \mathcal{D})$.

For evaluation, we provide $N \in \{4, 8, 16, 32, 64, 128\}$ labeled variants from each held-out assay and evaluate model predictions on all remaining variants. We repeat this evaluation for five randomly-sampled context sets for each assay. We average the Spearman correlation across assays within each evaluation and report the mean across the five draws, with error bars showing ± 1 standard deviation.

We compare PROFICIENT-Real, our model trained on real assay data, against several baselines in Figure 2 (right). We compare against methods that learn across real protein fitness assays like PRIMO Test-Time Training (PRIMO-TTT) and Metalic, shown in blue, as well as conventional predictors that are fit to each assay independently, shown in gray. We also include three conventional predictors fitted independently to the labeled support set of each assay, using the released PRIMO implementations

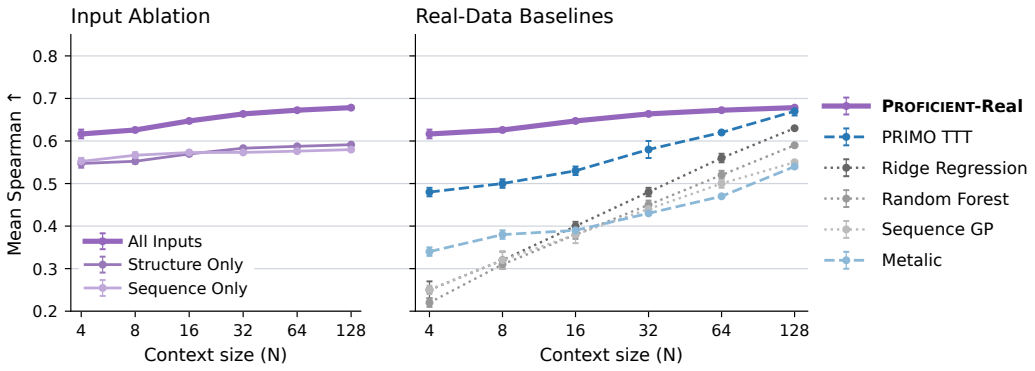


Figure 2: **Combining sequence and structural inputs improves learning**, leading to strong performance compared to other real-data baselines like PRIMO and Metalic. **(Left)** Using both sequence and structural inputs outperforms using either input alone across all contexts. **(Right)** PROFICIENT-Real outperforms the compared baselines, with the largest improvements at small context sizes. Error bars show $\pm$ 1 standard deviation over five evaluation draws.

(Teufel et al., 2025). *Sequence GP* uses an RBF kernel on frozen ESM-2 embeddings, and a learned affine transformation of the ProGen2 zero-shot score as its mean function. *Random forest* averages predictions from 100 regression trees trained on the ESM-2 embeddings. *Ridge regression* combines a regularized linear predictor on ESM-2 embeddings with a learned linear contribution from ProGen2. These baselines test the effectiveness of just the pretrained sequence representations without learning across assays. We leave ProteinNPT evaluation to future work, since a matched comparison would require separately training its predictor on each of the 990 support sets.

Our model outperforms these baselines at every evaluated context size, with the largest improvements in the small-N regime. Its performance with only 4 labeled variants is competitive with the best baseline PRIMO-TTT with 64 labels. Furthermore, unlike PRIMO-TTT, PROFICIENT-Real does not require any additional optimization at inference time, and it can make predictions for new assays without updating model parameters.

3.3 CONTRIBUTIONS OF BIOLOGICAL INPUTS

To assess how sequence and structural information contribute to our prediction, we ablate the inputs to PROFICIENT-Real and evaluate the performance of the resulting models. We compare against a model that uses only sequence information from ESM-2, and a model that uses only structural information from ProteinMPNN and AlphaFold2.

We see in Figure 2 (left) that combining these inputs significantly improves the performance over either input group alone. These gains show that the sequence and structural information provide complementary information, and our combined PROFICIENT-Real model is able to effectively leverage both sources of information to make accurate predictions.

3.4 LIMITATIONS OF REAL-ONLY TRAINING

Despite its strong performance against the preceding baselines, PROFICIENT-Real does not consistently outperform Kermut, a Gaussian process with a kernel that encodes explicit biological priors. Figure 3 (left) shows that the relative performance of these approaches depends on the available context. Across the evaluation panel, PROFICIENT-Real achieves higher mean Spearman correlation at small contexts, demonstrating the value of learning across assays. However, its performance plateaus as the context size increases, while Kermut continues to improve.

The right panel of Figure 3 shows the performance of PROFICIENT-Real and Kermut when evaluated on the six binding assays. We average Spearman correlations over the five evaluations within each assay, and show the median and interquartile range across the binding assays. In this setting, even for smaller contexts $N = 16$ and $N = 32$, Kermut generally achieves a higher correlation than

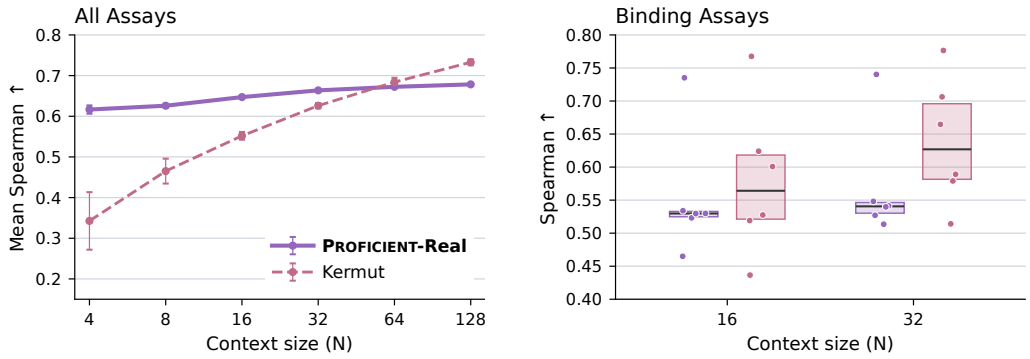


Figure 3: **Strong biological priors can outperform real-data training. (Left)** Kermut outperforms PROFICIENT-Real at larger context sizes, and real-only performance plateaus. Error bars show $\pm$ 1 standard deviation over five evaluation draws. **(Right)** On binding assays, Kermut achieves higher correlation than PROFICIENT-Real even at smaller context sizes $N = 16$ and $N = 32$. Boxes show the median and interquartile range across six assays.

PROFICIENT-Real, suggesting that the biological priors encoded in Kermut are particularly relevant for this phenotype.

These results motivate our next step: we want to combine the benefits of both approaches. Rather than relying solely on real data, we ask whether we can combine the strong biological priors encoded in Kermut with the flexibility of learning from real data.

4 BIOLOGICAL PRIORS AS SYNTHETIC TRAINING TASKS

Biological priors specify assumptions about how protein sequences and structures relate to fitness. We use this domain knowledge to generate synthetic fitness prediction tasks. Each task presents a sampled fitness landscape, and the model is trained to predict fitness values for query variants given a context set of labeled variants.

4.1 CONSTRUCTING SYNTHETIC TASKS

We construct synthetic tasks by combining real protein variants with sampled fitness values. Real variants preserve the realistic sequence and structural relationships between variants, while synthetic labels allow us to generate a diverse set of fitness landscapes.

Gaussian processes provide a natural framework for specifying these relationships, where a specified kernel defines the similarity between variants. For our experiments, we sample from a GP with an ESM-2 mean function and the Kermut kernel, which combines structural and sequence-level similarity. We then divide the variants into context and query sets. The full details of the kernel are described by Groth et al. (2024), and we summarize the relevant details in Appendix C.2.

Data-informed hyperparameters. The hyperparameters of the GP control properties of the synthetic landscapes, such as the smoothness of the function and the relative importance of sequence and structural similarity. However, especially in the low-data regime, the appropriate hyperparameters of the GP are often unknown. With a small number of labeled variants, multiple settings could plausibly explain the observed data, and accounting for this uncertainty may improve model predictions. We therefore train our model across a data-informed distribution of GP hyperparameters. In each batch, we sample a random assay from the training set and a random subset of its variants. We then fit the GP hyperparameters ϕ by taking 25 gradient steps to maximize the marginal likelihood of the measured fitness values. Given the variants $\{x_i\}_{i=1}^n$ and hyperparameters ϕ, we compute the kernel matrix K using the Kermut kernel, and we jointly sample fitness values

$$\tilde{\mathbf{y}}_{\text{GP}} \mid X, \phi \sim \mathcal{N}\big(m_\phi(X),\, K_{\text{Kermut},\phi}(X, X) + \sigma_\phi^2 I\big). \tag{1}$$

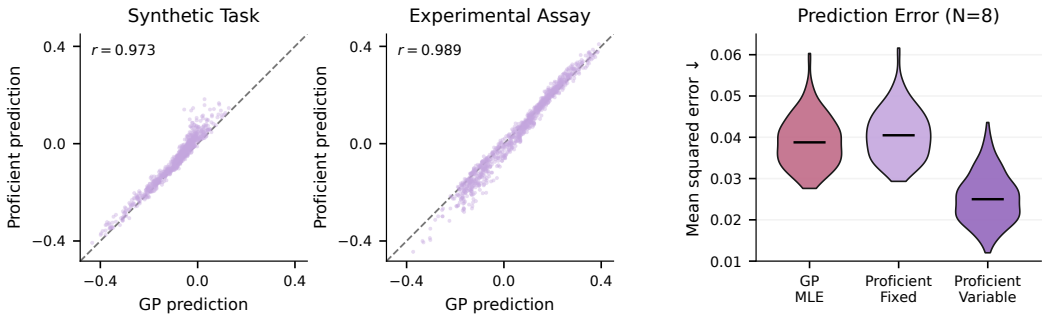


Figure 4: **Synthetic training closely approximates GP predictions, while varying hyperparameters can improve performance. (Left)** A model trained with fixed GP hyperparameters closely matches GP predictions under synthetic and experimental labels ($N = 32$) **(Right)** On experimental TCRG1 labels ($N = 8$), a model trained on tasks with variable hyperparameters achieves lower MSE than a model trained on tasks with fixed hyperparameters. Violins show median and distribution across 200 sampled query sets sharing one context set.

This allows us to generate tasks over a distribution of GP settings which are informed by experimental data. The measured fitness values are used only to fit ϕ, and are never used as training labels.

Zero-shot informed tasks. At very small contexts, fitting a GP can lead to worse predictions than using the zero-shot score alone. In Figure 5, we see that Kermut achieves lower Spearman's than zero-shot ESM-2 at $N \leq 8$, even though its mean function incorporates the zero-shot score. This suggests that the GP is overfitting to the small number of labeled variants, and it motivates us to train our model to use the zero-shot information more effectively. We therefore augment our synthetic training tasks by blending GP-sampled labels with the zero-shot score, and we include normalization and scaling details in Appendix C.2.

4.2 Learning from Synthetic Landscapes

We use the same model architecture and training objective as in Section 3, replacing the measured fitness values with synthetic labels. This training procedure follows the framework from prior-data fitted networks (PFNs) (Müller et al., 2022): a model learns conditional distributions over query labels given a context set of labeled variants, and it is trained by minimizing the expected negative log-likelihood of the sampled labels across synthetic tasks. This approach allows the model to learn the inductive biases encoded in the biological priors. We refer to the model trained on these tasks as Proficient-Synth.

Sanity check: fixed hyperparameters. Holding the GP hyperparameters fixed enables us to first evaluate the model's ability to capture the behavior of a specific GP. We therefore test whether a model trained on synthetic data can recover the predictive means of the underlying GP. We train with fixed Kermut hyperparameters and compare the model's predictions with those of the same GP on variants from the stability assay `TCRG1_MOUSE_Tsuboyama_2023_1E0L`. We consider two prediction tasks with $N = 32$ variants: one with labels sampled from the GP, and one with experimentally measured labels. In Figure 4 (left), we compare the model's predictions to those of the GP and see that the predictions closely agree under both synthetic and real labels.

Data-informed hyperparameters improve low-data predictions. Varying the hyperparameters across tasks provides a more diverse training distribution and also exposes the model to uncertainty in the underlying GP hyperparameters. We evaluate the models on experimentally measured fitness values from the same stability assay with $N = 8$. In Figure 4 (right), we compare a GP whose hyperparameters are fit by maximum likelihood (GP MLE), a model trained on synthetic tasks with fixed hyperparameters, and a model trained on synthetic tasks with data-informed hyperparameters. The violin plots show the distribution of MSE across 200 randomly sampled query sets. We see the

fixed-hyperparameter model closely matches the GP MLE, while the data-informed model achieves lower MSE, demonstrating that training on diverse GP settings can improve predictions when few labeled variants are available.

4.3 Transfer to Experimental Assays

We evaluate PROFICIENT-Synth on the same panel of held-out ProteinGym assays. Both PROFICIENT-Synth and Kermut use the same ESM-2 zero-shot information: our method uses it as an input feature, while Kermut uses it as the mean function of the GP. In Figure 5, we see that PROFICIENT-Synth achieves higher mean Spearman than Kermut at small context sizes, and it achieves similar performance at larger contexts. Our method also outperforms the zero-shot baseline at every context, indicating that our zero-shot informed tasks enabled the model to effectively leverage the zero-shot information. These results show that our data generation procedure was effective, and that training on synthetic landscapes can produce a model that generalizes well to real protein fitness landscapes.

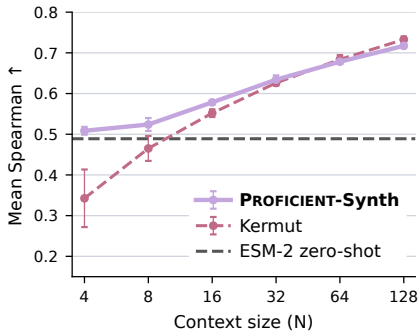


Figure 5: **Synthetic pretraining generalizes well to experimental assays.** Error bars show $\pm$ 1 std over five evaluation draws.

5 Combining Synthetic Pretraining with Real Fitness Landscapes

The previous sections explore two different sources of information: training on real assays allows PROFICIENT-Real to learn patterns that transfer across assays, while synthetic pretraining allows PROFICIENT-Synth to learn strong inductive biases from biological priors. We now combine these approaches by pretraining on synthetic tasks and then finetuning on real assay data.

5.1 Combining Synthetic and Real Training

We initialize PROFICIENT with the synthetically pretrained model from the previous section, and then finetune it on real assay data. We continue to use the same architecture and training objective, replacing the synthetic labels with measured fitness values. During finetuning, we use the same assay-level split and evaluation protocol as before. We use a smaller learning rate of 3×10^{-5}, and we select the final checkpoint with the highest mean Spearman correlation across the validation assays.

We study the impact of synthetic pretraining in Figure 1 (left), where we compare the performance of PROFICIENT-Synth, our synthetically pretrained PFN, PROFICIENT-Real, our model trained on real assay data, and PROFICIENT, our model which is first pretrained on synthetic tasks and then finetuned on real assay data. We see that finetuning on real assay data improves performance over training on synthetic tasks alone, and it also improves performance over training on real assay data alone. Compared to synthetic pretraining, PROFICIENT is better at smaller contexts; however, compared to real-only training, the gains are largest at larger contexts. This suggests that synthetic pretraining and real finetuning provide complementary information, and that combining these approaches allows the model to achieve strong performance across a range of context sizes.

5.2 Evaluation

Comparison with Existing Methods We compare PROFICIENT with representative supervised baselines in Figure 1 (right). We see that PROFICIENT achieves higher mean Spearman correlation than the baselines at every context size, with the largest gains at small contexts. Even with only four labeled variants, PROFICIENT achieves a mean Spearman correlation of 0.671, outperforming all compared supervised baselines even when they receive 32 labeled variants.

Zero-shot predictors can rank variants without any labeled measurements from the target assay, providing a useful reference for whether the limited labels are beneficial. In Figure 8, we compare PROFICIENT with the strongest zero-shot methods on our evaluation panel. We see that, unlike many

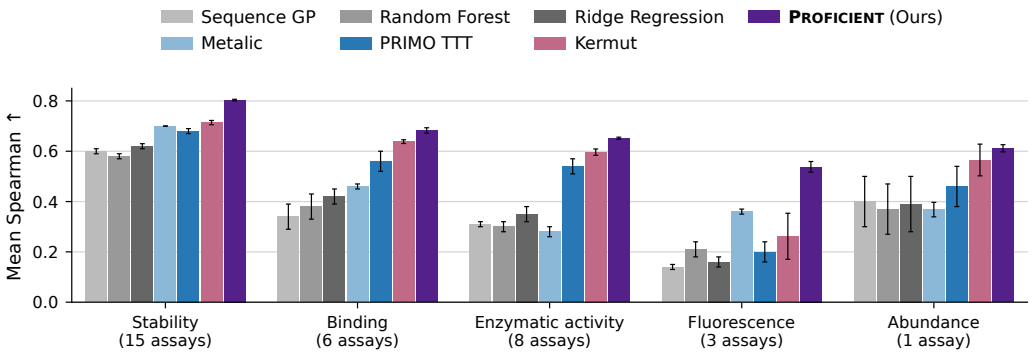


Figure 6: **PROFICIENT is competitive across different phenotypes.** We evaluate the performance with $N = 32$ labeled variants and report the mean and standard deviation over five evaluation draws.

supervised baselines, PROFICIENT outperforms the zero-shot predictors, effectively leveraging the limited labeled data to improve predictive accuracy.

Performance across Phenotypes We further break down the performance of PROFICIENT and existing baselines by phenotype in Figure 6. We see that PROFICIENT achieves strong performance across all phenotypes, with particularly large gains on fluorescence. Furthermore, we see that PROFICIENT now outperforms Kermut on binding assays, addressing the weakness of real-only training that we identified in Section 3.4.

High-Fitness Variant Selection We simulate a protein engineering setting where the goal is to use a few labeled variants to select high-fitness candidates from a larger set of unlabeled variants. Given a context set of N labeled variants, we use each model to predict the fitness of all remaining variants and select the top 100. We then measure the selection quality by counting how many of the selected variants are in the top 10% of true fitness values. We see in Figure 7 that PROFICIENT is able to select more high-fitness variants at every context size, suggesting that our method can help prioritize valuable candidates early on in the protein engineering process.

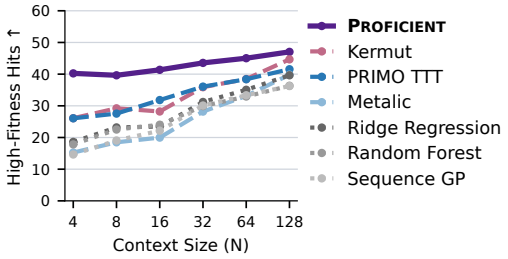


Figure 7: **PROFICIENT selects more high-fitness variants** compared to other baselines.

6 DISCUSSION

Despite the success of synthetic pretraining in domains such as tabular learning, its application to protein fitness prediction remains relatively underexplored. In this work, we show how protein-specific biological priors can be combined with learning from experimental data. Specifically, sequence and structural features provide complementary information about protein variants, while synthetic pretraining followed by real-assay finetuning combines the benefits of strong biological priors with empirical supervision. Together, these components improve predictive performance when few labeled variants are available. Our work suggests a broader opportunity to use biological knowledge both to inform model inputs and to generate synthetic training tasks, especially in settings where experimental data is limited.

REFERENCES

Jacob Beck, Shikha Surana, Manus McAuliffe, Oliver Bent, Thomas D Barrett, Juan Jose Garau-Luis, and Paul Duckworth. Metalic: Meta-learning in-context with protein language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Surojit Biswas, Grigory Khimulya, Ethan C Alley, Kevin M Esvelt, and George M Church. Low-n protein engineering with data-efficient deep learning. *Nature methods*, 18(4):389–396, 2021.

Lin Chen, Zehong Zhang, Zhenghao Li, Rui Li, Ruifeng Huo, Lifan Chen, Dingyan Wang, Xiaomin Luo, Kaixian Chen, Cangsong Liao, et al. Learning protein fitness landscapes with deep mutational scanning data from multiple sources. *Cell Systems*, 14(8):706–721, 2023.

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

Douglas M Fowler and Stanley Fields. Deep mutational scanning: a new style of protein science. *Nature methods*, 11(8):801–807, 2014.

Sam Gelman, Sarah A Fahlberg, Pete Heinzelman, Philip A Romero, and Anthony Gitter. Neural networks to learn protein sequence–function relationships from deep mutational scanning data. *Proceedings of the National Academy of Sciences*, 118(48):e2104878118, 2021.

Peter M Groth, Mads H Kerrn, Lars Olsen, Jesper Salomon, and Wouter Boomsma. Kermut: Composite kernel regression for protein variant effects. *Advances in Neural Information Processing Systems*, 37:29514–29565, 2024.

Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J. Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q. Tran, Jonathan Deaton, Marius Wiggert, Rohil Badkundri, Irhum Shafkat, Jun Gong, Alexander Derry, Raul S. Molina, Neil Thomas, Yousuf A. Khan, Chetan Mishra, Carolyn Kim, Liam J. Bartie, Matthew Nemeth, Patrick D. Hsu, Tom Sercu, Salvatore Candido, and Alexander Rives. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025. doi: 10.1126/science.ads0018.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Emmi Jokinen, Markus Heinonen, and Harri Lähdesmäki. mgpfusion: predicting protein stability changes with gaussian process kernel learning and data fusion. *Bioinformatics*, 34(13):i274–i283, 2018.

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.

Mingchen Li, Yang Tan, Xinzhu Ma, Bozitao Zhong, Huiqun Yu, Ziyi Zhou, Wanli Ouyang, Bingxin Zhou, Pan Tan, and Liang Hong. ProSST: Protein language modeling with quantized structure and disentangled attention. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022.

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

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2022.

Erik Nijkamp, Jeffrey A. Ruffolo, Eli N. Weinstein, Nikhil Naik, and Ali Madani. ProGen2: Exploring the boundaries of protein language models. *Cell Systems*, 14(11):968–978, 2023.

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

Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood Van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, et al. Proteingym: Large-scale benchmarks for protein fitness prediction and design. *Advances in neural information processing systems*, 36:64331–64379, 2023a.

Pascal Notin, Ruben Weitzman, Debora Marks, and Yarin Gal. Proteinnpt: Improving protein property prediction and design with non-parametric transformers. *Advances in Neural Information Processing Systems*, 36: 33529–33563, 2023b.

Jonathan Parkinson and Wei Wang. Linear-scaling kernels for protein sequences and small molecules outperform deep learning while providing uncertainty quantitation and improved interpretability. *Journal of Chemical Information and Modeling*, 63(15):4589–4601, 2023.

Philip A Romero, Andreas Krause, and Frances H Arnold. Navigating the protein fitness landscape with gaussian processes. *Proceedings of the National Academy of Sciences*, 110(3):E193–E201, 2013.

Ning Sun, Shuxian Zou, Tianhua Tao, Sazan Mahbub, Dian Li, Yonghao Zhuang, Hongyi Wang, Xingyi Cheng, Le Song, and Eric P. Xing. Mixture of experts enable efficient and effective protein understanding and design. *bioRxiv*, 2024. doi: 10.1101/2024.11.29.625425.

Ege Onur Taga, Muhammed Emrullah Ildiz, and Samet Oymak. Timepfn: Effective multivariate time series forecasting with synthetic data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 20761–20769, 2025.

Yang Tan, Ruilin Wang, Banghao Wu, Liang Hong, and Bingxin Zhou. From high-throughput evaluation to wet-lab studies: advancing mutation effect prediction with a retrieval-enhanced model. *Bioinformatics*, 41 (Supplement_1):i401, 2025.

Felix Teufel, Aaron W. Kollasch, Yining Huang, Ole Winther, Kevin K. Yang, Pascal Notin, and Debora S. Marks. Few-shot protein fitness prediction via in-context learning and test-time training. In *AI for Science Workshop at the 39th Conference on Neural Information Processing Systems (NeurIPS 2025)*, 2025. URL https://arxiv.org/abs/2512.02315.

Zuobai Zhang, Pascal Notin, Yining Huang, Aurélie C. Lozano, Vijil Chenthamarakshan, Debora Marks, Payel Das, and Jian Tang. Multi-scale representation learning for protein fitness prediction. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Juannan Zhou, Carlos Martí-Gómez, Samantha Petti, and David M McCandlish. Learning sequence-function relationships with scalable, interpretable gaussian processes. *bioRxiv*, 2025.

Ziyi Zhou, Liang Zhang, Yuanxi Yu, Banghao Wu, Mingchen Li, Liang Hong, and Pan Tan. Enhancing efficiency of protein language models with minimal wet-lab data through few-shot learning. *Nature Communications*, 15(1):5566, 2024.

A Additional Results

Comparison with Zero-Shot Predictors Zero-shot predictors provide a reference for how well a model can rank variants without any labeled measurements from the target assay. We compare PROFICIENT with the strongest zero-shot methods on our evaluation panel in Figure 8 (Li et al., 2024; Sun et al., 2024; Tan et al., 2025; Zhang et al., 2024; Hayes et al., 2025). We see that, unlike many supervised baselines, PROFICIENT outperforms the zero-shot predictors with $N = 4$, effectively leveraging the limited labeled data to improve predictive accuracy.

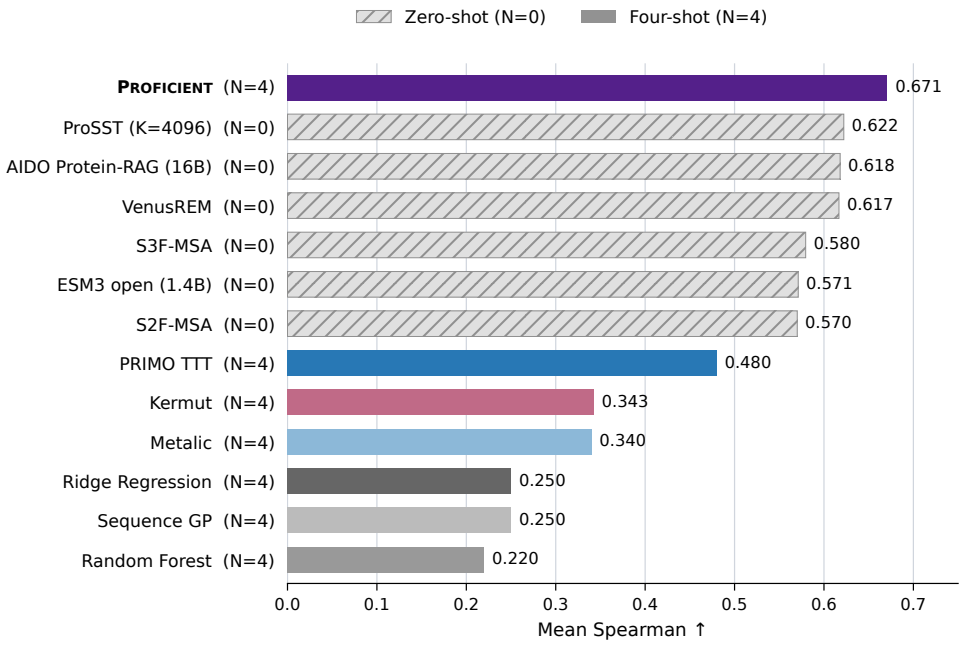


Figure 8: Zero-shot reference scores and PROFICIENT with four labeled variants per assay. Bars show mean Spearman across 33 assays. Zero-shot scores use full-assay evaluations; PROFICIENT uses the held-out query sets of five context draws.

NDCG We also evaluate full-rank normalized discounted cumulative gain (NDCG), which gives greater weight to placing high-fitness variants near the top of the ranking. In Figure 9, PROFICIENT outperforms the compared baselines at every context size, with the largest gains at small contexts.

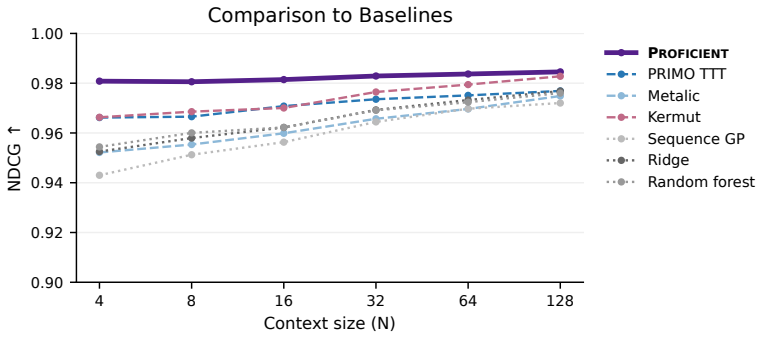


Figure 9: Mean NDCG across the evaluation assays using one shared context draw

B Model Architecture and Biological Inputs

We describe the checkpoints used for PROFICIENT-Synth, PROFICIENT-Real, and PROFICIENT in Figure 1. They share the architecture below but differ in training targets.

Protein features. Frozen ESM-2 650M produces a mean-pooled, 1,280-dimensional embedding for each variant. The dense feature stream contains the Kermut sequence and mutation features, including ProteinMPNN amino-acid probabilities and mutant log-likelihoods and AlphaFold2 residue coordinates. Two additional channels supply a zero-shot score and its binary availability indicator. The protein feature extractors remain frozen throughout all three training regimes.

Feature fusion. The dense stream uses a linear projection followed by GELU and layer normalization to produce $d \in \mathbb{R}^{512}$. The sequence stream applies layer normalization, a $1280 \to 512 \to 512$ multilayer perceptron with GELU between its linear layers, and output layer normalization to produce $e \in \mathbb{R}^{512}$. An elementwise gate combines them as

$$g = \mathrm{sigmoid}(W_g[d; e] + b_g), \qquad h = \mathrm{LayerNorm}(d + g \odot e).$$

Attention and label conditioning. Two transformer blocks process variant features before label conditioning. A linear $1 \to 512$ projection embeds each observed context label; query positions instead receive a learned query token. Six further transformer blocks process these conditioned representations. All eight blocks use hidden dimension 512, eight attention heads, pre-layer normalization, residual connections, and a GELU feed-forward network of width 1,024, with zero dropout. In both stages, context variants attend to context variants and query variants attend only to context variants; query variants never attend to one another. After a final layer normalization, a linear $512 \to 2$ head predicts the Gaussian mean and log-variance, with predictive variance bounded below by 10^{-4}.

C Training Procedures and Synthetic Priors

Optimization and task sampling All three regimes minimize Gaussian negative log-likelihood on query labels using AdamW with zero weight decay and constant learning rate of 3×10^{-4}. An update uses tasks from one assay; each task partitions its sampled variants into N context rows, with the sampler drawing N uniformly from the integers $2, \ldots, 128$.

C.1 Real-only training

PROFICIENT-Real starts from random initialization, and we use the validation assays to select the checkpoint with the highest mean validation Spearman correlation. Training has a 100,000-update cap and stops after 10,000 updates without a validation improvement of at least 0.001, subject to a 10,000-update minimum. The selected checkpoint is at update 12,500.

C.2 Synthetic pretraining

We pretrain from random initialization for 100,000 updates with 128 tasks per update, using seed 123. Each task contains 192 variants, split into N context variants. We uniformly sample $N \in 2, \ldots, 128$ Each batch uses one assay and a 1,024-variant pool, fitting Kermut GP hyperparameters for 25 marginal-likelihood optimization steps at learning rate 0.1. We draw synthetic GP labels jointly for each task, and we augment the sampled labels with zero-shot scores as described below and minimize Gaussian query negative log-likelihood using AdamW with a constant learning rate of 3×10^{-4} and zero weight decay. We use the final checkpoint at 100,000 updates for evaluation and subsequent finetuning.

Kermut Kermut (Groth et al., 2024) is a representative example of a GP kernel which combines structural and sequence-level similarity, and which we use as a reference kernel throughout this work. Its structure kernel compares mutations by combining three terms: the similarity of local structural environments $k_H(x, x')$, the similarity of mutation probabilities under an inverse-folding model $k_p(x, x')$, and the spatial distance of the mutation sites $k_d(x, x')$. For single-mutant variants, Kermut

defines the structure kernel as the product of these three terms,

$$k^{(1)}_{\text{struct}}(x, x') = \lambda\, k_H(x, x')\, k_p(x, x')\, k_d(x, x'),$$

and for multi-mutant variants, the kernel sums over all pairs of mutated sites,

$$k_{\text{struct}}(x, x') = \sum_{i \in M} \sum_{j \in M'} k^{(1)}_{\text{struct}}(x_i, x'_j).$$

Kermut also incorporates a sequence kernel to capture effects such as epistasis, by including a squared exponential kernel over pretrained protein language model embeddings,

$$k_{\text{seq}}(x, x') = \exp\left(-\frac{\|f_{\text{PLM}}(x) - f_{\text{PLM}}(x')\|_2^2}{2\sigma^2}\right).$$

The structure and sequence kernels are then combined to form the final Kermut kernel,

$$k_{\text{Kermut}}(x, x') = \pi\, k_{\text{struct}}(x, x') + (1 - \pi)\, k_{\text{seq}}(x, x').$$

Zero-shot Informed priors For each synthetic task, we blend our ESM-2 score with the GP-sampled labels:

$$\tilde{\mathbf{y}} = S((1 - \alpha)S(\mathbf{y}_{\text{GP}}) + \alpha S(\mathbf{s}))\,, \qquad \alpha \sim \text{Beta}(2, 6),$$

where S min–max scales values across the task's context and query variants to $[-0.5, 0.5]$. This produces landscapes with varying agreement with the zero-shot predictor while retaining variation from the GP.

C.3 Real-assay finetuning

PROFICIENT initializes from the selected synthetic checkpoint and updates all encoder and transformer parameters using measured, assay-normalized DMS labels. A fresh AdamW optimizer uses learning rate 3×10^{-5}. Training has a 5,000-update cap and stops after 1,000 updates without an improvement of at least 0.001 in mean validation Spearman. The selected checkpoint is at update 2,100; training stopped at update 3,100.

D Data and Evaluation Protocol

We use the train/test split from PRIMO (Teufel et al., 2025), which is chosen so that no close homolog or identical protein appears in training and test. We further divide the train split by reserving 10 assays for validation, and we filter the remaining training assays by requiring at least 256 aligned variants.

D.1 Training assays

Table 1: Complete training population.

Assay
A4GRB6_PSEAI_Chen_2020
ACE2_HUMAN_Chan_2020
AMIE_PSEAE_Wrenbeck_2017
ANCSZ_Hobbs_2022
BBC1_YEAST_Tsuboyama_2023_1TG0
BCHB_CHLTE_Tsuboyama_2023_2KRU
CALM1_HUMAN_Weile_2017
CAS9_STRP1_Spencer_2017_positive
CASP7_HUMAN_Roychowdhury_2020
CATR_CHLRE_Tsuboyama_2023_2AMI
CBPA2_HUMAN_Tsuboyama_2023_1O6X
CBS_HUMAN_Sun_2020
CBX4_HUMAN_Tsuboyama_2023_2K28
CD19_HUMAN_Klesmith_2019_FMC_singles
CP2C9_HUMAN_Amorosi_2021_abundance
DN7A_SACS2_Tsuboyama_2023_1JIC

Assay
DOCK1_MOUSE_Tsuboyama_2023_2M0Y
ERBB2_HUMAN_Elazar_2016
F7YBW8_MESOW_Aakre_2015
FKBP3_HUMAN_Tsuboyama_2023_2KFV
HCP_LAMBD_Tsuboyama_2023_2L6Q
HECD1_HUMAN_Tsuboyama_2023_3DKM
HMDH_HUMAN_Jiang_2019
HXK4_HUMAN_Gersing_2022_activity
HXK4_HUMAN_Gersing_2023_abundance
ILF3_HUMAN_Tsuboyama_2023_2L33
ISDH_STAAW_Tsuboyama_2023_2LHR
LGK_LIPST_Klesmith_2015
LYAM1_HUMAN_Elazar_2016
MBD11_ARATH_Tsuboyama_2023_6ACV
MET_HUMAN_Estevam_2023
MK01_HUMAN_Brenan_2016
MSH2_HUMAN_Jia_2020
MTH3_HAEAE_RockahShmuel_2015
MTHR_HUMAN_Weile_2021
MYO3_YEAST_Tsuboyama_2023_2BTT
NUD15_HUMAN_Suiter_2020
NUSA_ECOLI_Tsuboyama_2023_1WCL
NUSG_MYCTU_Tsuboyama_2023_2MI6
OBSCN_HUMAN_Tsuboyama_2023_1V1C
ODP2_GEOSE_Tsuboyama_2023_1W4G
OTC_HUMAN_Lo_2023
OTU7A_HUMAN_Tsuboyama_2023_2L2D
OXDA_RHOTO_Vanella_2023_activity
P84126_THETH_Chan_2017
PAI1_HUMAN_Huttinger_2021
PIN1_HUMAN_Tsuboyama_2023_1I6C
PITX2_HUMAN_Tsuboyama_2023_2L7M
POLG_PESV_Tsuboyama_2023_2MXD
PR40A_HUMAN_Tsuboyama_2023_1UZC
PRKN_HUMAN_Clausen_2023
PTEN_HUMAN_Mighell_2018
Q53Z42_HUMAN_McShan_2019_binding-TAPBPR
Q59976_STRSQ_Romero_2015
RAD_ANTMA_Tsuboyama_2023_2CJJ
RAF1_HUMAN_Zinkus-Boltz_2019
RASK_HUMAN_Weng_2022_abundance
RASK_HUMAN_Weng_2022_binding-DARPin_K55
RBP1_HUMAN_Tsuboyama_2023_2KWH
RCRO_LAMBD_Tsuboyama_2023_1ORC
RD23A_HUMAN_Tsuboyama_2023_1IFY
RFAH_ECOLI_Tsuboyama_2023_2LCL
RL20_AQUAE_Tsuboyama_2023_1GYZ
RNC_ECOLI_Weeks_2023
RPC1_BP434_Tsuboyama_2023_1R69
RS15_GEOSE_Tsuboyama_2023_1A32
SAV1_MOUSE_Tsuboyama_2023_2YSB
SBI_STAAM_Tsuboyama_2023_2JVG
SDA_BACSU_Tsuboyama_2023_1PV0
SERC_HUMAN_Xie_2023
SHOC2_HUMAN_Kwon_2022
SOX30_HUMAN_Tsuboyama_2023_7JJK
SPG1_STRSG_Olson_2014
SPG2_STRSG_Tsuboyama_2023_5UBS
SPIKE_SARS2_Starr_2020_binding
SPIKE_SARS2_Starr_2020_expression
SPTN1_CHICK_Tsuboyama_2023_1TUD
SRBS1_HUMAN_Tsuboyama_2023_2O2W
SRC_HUMAN_Ahler_2019
TCRG1_MOUSE_Tsuboyama_2023_1E0L
THO1_YEAST_Tsuboyama_2023_2WQG
TNKS2_HUMAN_Tsuboyama_2023_5JRT
TPK1_HUMAN_Weile_2017
TPMT_HUMAN_Matreyek_2018
UBC9_HUMAN_Weile_2017
UBE4B_HUMAN_Tsuboyama_2023_3L1X
UBE4B_MOUSE_Starita_2013
UBR5_HUMAN_Tsuboyama_2023_1I2T
VG08_BPP22_Tsuboyama_2023_2GP8
VILI_CHICK_Tsuboyama_2023_1YU5
YAIA_ECOLI_Tsuboyama_2023_2KVT
YAP1_HUMAN_Araya_2012
YNZC_BACSU_Tsuboyama_2023_2JVD

D.2 VALIDATION ASSAYS

These ten assays are used for checkpoint selection and are excluded from training and evaluation.

Table 2: Complete validation population (10 assays).

Assay
AACC1_PSEAI_Dandage_2018
ARGR_ECOLI_Tsuboyama_2023_1AOY
CP2C9_HUMAN_Amorosi_2021_activity
CUE1_YEAST_Tsuboyama_2023_2MYX
ENVZ_ECOLI_Ghose_2023
GDIA_HUMAN_Silverstein_2021
KKA2_KLEPN_Melnikov_2014
PPARG_HUMAN_Majithia_2016
RASH_HUMAN_Bandaru_2017
SRC_HUMAN_Chakraborty_2023_binding-DAS_25uM

D.3 EVALUATION ASSAYS

Table 3: Complete evaluation population (33 assays).

Assay
AMFR_HUMAN_Tsuboyama_2023_4G3O
RCD1_ARATH_Tsuboyama_2023_5OAO
SR43C_ARATH_Tsuboyama_2023_2N88
FECA_ECOLI_Tsuboyama_2023_2D1U
PKN1_HUMAN_Tsuboyama_2023_1URF
CSN4_MOUSE_Tsuboyama_2023_1UFM
SPA_STAAU_Tsuboyama_2023_1LP1
NKX31_HUMAN_Tsuboyama_2023_2L9R
EPHB2_HUMAN_Tsuboyama_2023_1F0M
SQSTM_MOUSE_Tsuboyama_2023_2RRU
MAFG_MOUSE_Tsuboyama_2023_1K1V
SCIN_STAAR_Tsuboyama_2023_2QFF
DNJA1_HUMAN_Tsuboyama_2023_2LO1
VRPI_BPT7_Tsuboyama_2023_2WNM
ESTA_BACSU_Nutschel_2020
CASP3_HUMAN_Roychowdhury_2020
BLAT_ECOLX_Deng_2012
BLAT_ECOLX_Jacquier_2013
BLAT_ECOLX_Stiffler_2015
BLAT_ECOLX_Firnberg_2014
VKOR1_HUMAN_Chiasson_2020_activity
VKOR1_HUMAN_Chiasson_2020_abundance
Q8WTC7_9CNID_Somermeyer_2022
D7PM05_CLYGR_Somermeyer_2022
GFP_AEQVI_Sarkisyan_2016
DLG4_RAT_McLaughlin_2012
RL40A_YEAST_Roscoe_2013
GRB2_HUMAN_Faure_2021
DYR_ECOLI_Thompson_2019
DLG4_HUMAN_Faure_2021
RL40A_YEAST_Mavor_2016
DYR_ECOLI_Nguyen_2023
RL40A_YEAST_Roscoe_2014

E AI ASSISTANCE

We used Codex and Claude throughout the research process to assist with writing and debugging code, monitoring training and evaluation runs, and editing the manuscript for clarity and errors. We reviewed and validated all AI-assisted code. The experimental design and scientific conclusions were determined by the authors.