Stabilizing protein fitness predictors via the PCS framework

Omer Ronen [1] **Alex Zhao** [2] **Ron Boger** [3 4 5] **Chengzhong Ye** [1] **Bin Yu** [1 6 2]

Abstract

We improve protein fitness prediction by addressing an often-overlooked source of instability in machine learning models: the choice of data representation. Guided by the Predictability–Computability–Stability (PCS) framework for veridical (truthful) data science, we construct *Stable* predictors by applying a prediction-based screening procedure (pred-check in PCS) to select predictive representations, followed by ensembling models trained on each—thereby leveraging representation-level diversity. This approach improves predictive accuracy, out-of-distribution generalization, and uncertainty quantification across a range of model classes. Our *Stable* variant of the recently introduced kernel regression method, Kermut, achieves state-of-the-art performance on the ProteinGym supervised fitness prediction benchmark: it reduces mean squared error by up to 20% and improves Spearman correlation by up to 10%, with the largest improvements on splits representing a distribution shift. We further demonstrate that *Stable* predictors yield statistically significant improvements in in-silico protein design tasks. Our results highlight the critical role of representation-level variability in fitness prediction and, more broadly, underscore the need to address instability throughout the entire data science lifecycle to advance protein design.

1. Introduction

Improving the ability of machine learning (ML) models to predict the effects of mutations on protein fitness is a central challenge in computational biology, with broad implications for protein design, disease understanding, and more. In protein engineering, ML has shown promise in reducing experimental costs by prioritizing high-fitness sequences for wet-lab testing (Yang et al., 2025).

However, applying ML to guide experiments presents key challenges. First, models must generalize to out-of-distribution (OOD) sequences—typically mutations of a reference—since the goal is to discover novel functional proteins. Second, uncertainty quantification (UQ) is crucial for assessing prediction reliability, as no existing model generalizes well across the vast, combinatorial space of protein sequences.

Current approaches to improving OOD generalization (Tagasovska et al., 2024) and UQ (Greenman et al., 2025) focus largely on the modeling stage of the data science life-cycle (DSLC) (Yu & Kumbier, 2020). Bayesian methods provide uncertainty estimates via posterior distributions (Greenman et al., 2025; Notin et al., 2023b), while ensembling neural networks captures variability from random model initializations (Gruver et al., 2021). Frequentist methods estimate predictive variance through probabilistic modelling (Nix & Weigend, 1994; Greenman et al., 2025). Conformal prediction, which provides UQ in the form of prediction intervals, have been used for model selection and function prediction (Fannjiang et al., 2022; Boger et al., 2025; Fannjiang & Park, 2025).

This work emphasizes the importance of stability under *reasonable* perturbations to data processing choices, with a focus on protein engineering. Guided by the Predictability–Computability–Stability (PCS) framework for veridical data science (Yu & Kumbier, 2020; Yu & Barter, 2024), we propose a simple and broadly applicable procedure to improve UQ and OOD generalization by leveraging multiple reasonable representations of the same protein sequence—for example, embeddings from different pretrained protein language models (PLMs) or zero-shot evolutionary scores from diverse methods. Our key contributions are:

- We identify a critical but underexplored source of instability in protein fitness prediction: the choice of data representation (e.g., embeddings). State-of-the-art

[1]Department of Statistics, University of California, Berkeley [2]Center for Computational Biology, UC Berkeley [3]Biophysics Graduate Group, UC Berkeley [4]Innovative Genomics Institute, UC Berkeley [5]California Institute for Quantitative Biosciences, UC Berkeley [6]Department of Computer Science and Electrical Engineering, University of California, Berkeley. Correspondence to: Omer Ronen <omer_ronen@berkeley.edu>.

Proceedings of the ICML 2025 Workshop on Multi-modal Foundation Models and Large Language Models for Life Sciences, Vancouver, Canada. 2025.

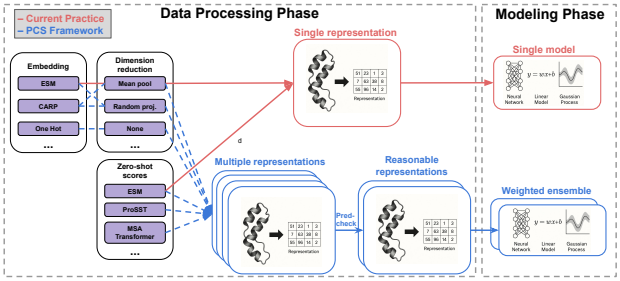

Figure 1. We stabilize protein fitness prediction via the PCS framework by leveraging representation-level variability: predictive representations are selected through a pred-check step and ensembled. This leads to substantial gains in accuracy and uncertainty quantification, underscoring the value of considering multiple reasonable representations for a protein sequence.

model performance varies significantly across representations (Section 4, Appendix 6.4).

- We introduce a simple and intuitive two-step procedure (Figure 1, Section 3) inspired by the PCS framework: (1) a **pred-check** step to select and weight predictive representations, and (2) an **ensembling** step that fits a base model on selected representations, yielding a **Stable** fitness predictor.
- On the supervised ProteinGym substitution benchmark, Stable predictors outperform their base models—including Gaussian Processes, linear models, and CNNs—in both accuracy and uncertainty estimation. Stable Kermut improves over standard Kermut (the current state-of-the-art) by reducing MSE by up to 20% and increasing Spearman correlation by up to 10%, with the greatest gains under distribution shift. For uncertainty, Stable Kermut improves correlation between predicted uncertainty and true errors by up to 70% (Section 4).
- Our Stable variants of Bayesian Ridge and Kermut enhance performance in in-silico iterative protein design, recovering 6% more top-fitness sequences than the previous best method Kermut (Section 4).

2. Background and related works

Supervised fitness prediction We represent a protein sequence as $p = (a_1, \ldots, a_{l^{(p)}})$, where $l^{(p)}$ is the sequence length and each a_i is one of the 20 canonical amino acids. A *mutated sequence* relative to p is defined as a sequence of the same length that differs from p at one or more positions, denoted by $m^{(p)}$. We denote the set of all such mutated sequences as $\mathbb{M}_p$. The goal of supervised fitness prediction is to learn a predictor (model) $\hat{f}^{(p)} : \mathbb{M}_p \to \mathbb{R}$ that maps a mutated sequence to its corresponding fitness value y—a scalar quantity that may reflect properties such as thermostability or binding affinity. We assume access to a labeled dataset of n mutations,, $(m_1^{(p)}, y_1), \ldots, (m_n^{(p)}, y_n)$, where y_i is the experimentally measured fitness of $m_i^{(p)}$.

We consider models that output both point predictions and uncertainty estimates. In this work, we study Bayesian Ridge regression, Kermut (a GP) and an ensemble of convolutional neural networks, each one of these models have been used for fitness prediction in previous works (Gruver et al., 2021; Groth et al., 2024; Greenman et al., 2025). Details are provided in Appendix 6.5.

Representations of protein sequences Predicting protein fitness with machine learning requires converting sequences into numerical representations, or *embeddings*. The simplest choice is one-hot encoding, which maps a sequence of length l to a $20 \times l$ vector by concatenating one-hot vectors for the 20 canonical amino acids.

Recent advances leverage embeddings from pre-trained PLMs (Rives et al., 2021; Rao et al., 2021; Lin et al., 2023; Yang et al., 2024), which capture patterns learned during pre-training on large-scale sequence data. These models typically represent a protein sequence as $l \times h$ matrices that are mean-pooled into fixed-size vectors for downstream prediction tasks (Li et al., 2024a). PLMs can also provide *zero-shot* scores—log-likelihood ratios between mutant and reference sequences—that correlate with fitness without supervised training (Meier et al., 2021; Notin et al., 2023a).

It is important to emphasize that the space of *reasonable*[1] choices for representing a single protein sequence is prohibitively large (Table 1). For example, Li et al. (2024a) show that different PLMs can produce embeddings that perform equally well in prediction tasks, and that increasing model size or training data does not necessarily lead to better results. In addition, while many computational methods incorporate a zero-shot score into the representation, there is no universally optimal score: different scores perform best for different tasks (Notin et al., 2023a). Moreover, although the penultimate layer is often used by default, Li et al. (2024a) report that embeddings from alternative layers can offer competitive performance. Finally, while mean pooling is the most common dimension reduction strategy, (Li et al., 2024b) have shown that alternative pooling methods can perform similarly well. Table 1 provides a non-exhaustive list of different and reasonable choices for embedding a single protein sequence.

We define a protein representation as a triplet compris-

[1]We define a representation as reasonable if there is no clear justification, a priori, to expect it will perform poorly for the purpose of fitness prediction.

ing: (1) a PLM embedding, (2) a dimensionality reduction method, and (3) a zero-shot score. To ensure tractability, we select a representative subset. For embeddings, we use ESM1v (Meier et al., 2021), ESM2 (Lin et al., 2023), and CARP (Yang et al., 2024). Dimensionality reduction is performed via standard mean pooling (Dallago et al., 2021) and three randomized variants that perturb pooling weights. For zero-shot scores, we include MSA Transformer (Rao et al., 2021), ProSST (Li et al., 2024c), ESM2 (Lin et al., 2023), and TranceptEVE (Notin et al., 2022b). Full details are provided in Appendices 6.2.1 and 6.2.2.

3. Stable fitness predictors via the PCS framework

Stabilizing fitness predictors via the PCS framework We adopt a two-step procedure to explore reasonable protein representations following Agarwal et al. (2025), and quantify associated uncertainty. First, in the **prediction check (pred-check)** step, we select a subset of informative zero-shot scores based on their fit to the training data (details in Appendix 6.3). We find it effective to select the top three zero-shot scores, weighted by Mean Decrease in Impurity (MDI), normalized to sum to one, from a random forest trained to predict the training fitness labels using the zero-shot scores as features (Appendix 6.4).

Secondly, in the **ensembling** step, we form 36 representations by taking the Cartesian product of the 3 selected zero-shot scores, 3 embeddings, and 4 dimensionality reduction methods. A base model is trained on each representation, and predictions are a weighted average of the ensemble outputs (weighted by the normalized MDI scores association with the zero-shot score). Uncertainty is estimated by combining (1) the average uncertainty reported by the base models, and (2) the empirical standard deviation of the predictions across different representations.

Our method builds on the PCS-based UQ framework (Agarwal et al., 2025), but focuses on representation-level sensitivity rather than algorithmic variability. Key changes include filtering representations (not algorithms) and perturbing reasonable data representations instead of using bootstrap resampling.

4. Experiments

4.1. ProteinGym benchmark performance

Setup We evaluate our methods on the ProteinGym supervised benchmark (Notin et al., 2023a), which includes 217 single-substitution assays across diverse proteins, organisms, and fitness definitions (see Section 2). Results on double mutants are deferred to Appendix 6.7.2. While other benchmarks exist (e.g., FLIP (Dallago et al., 2021)), we focus on ProteinGym for its scale and diversity.

For each dataset, we use 80% of the mutated sequences for training and evaluate predictive performance on the remaining 20%. ProteinGym provides three types of data splits:

- **Random split**: sequences are randomly assigned to the training and test sets.
- **Modulo split**: sequences with mutations at every fifth residue are assigned to the test set.
- **Contiguous split**: the sequence is divided into five equal segments, and each fold contains sequences with mutations in residues from one segment.

Methods As baselines, we consider three models adopted from prior work: (1) a Bayesian Ridge regression model (Greenman et al., 2025), (2) a Kermut regressor (Groth et al., 2024), and (3) an ensemble of four CNNs (Gruver et al., 2021; Greenman et al., 2025) (details for the models are provided in Appendix 6.5). For each of these models, we use ESM2 embeddings with mean pooling across the sequence dimension, combined with the ESM2 zero-shot score as additional input. These same models also serve as the base models for our Stable predictors, where they are trained separately on each selected representation before being ensembled. We also report results from ProteinNPT (Notin et al., 2023b) which is a strong baseline, though we do not reimplement a Stable version of it due to its high computational cost (Groth et al., 2024). For each one of the three models (Bayesian Ridge, Kermut and CNN), we implement a Stable version following the procedure described in Section 3. These stable versions are ensembles of the base models fitted on the same training data with different representations of the protein sequences. Each Stable version consists of 36 models.

Results The results, presented in Figure 2 (a) and in Appendix 6.7, show that Stable predictors consistently outperform their base estimators across all models and data splits. Among them, Stable Kermut achieves the best performance, followed by Stable Bayesian Ridge. The improvements are particularly notable on the more challenging *modulo* and *contiguous* splits, which introduce covariate shifts between training and test sets. On average, across the two evaluation splits, Stable CNN increases Spearman correlation by 22%, Stable Bayesian Ridge by 30%, and Stable Kermut by 9% relative to their respective base models. Remarkably, Stable Bayesian Ridge outperforms ProteinNPT across all splits and metrics, achieving performance on par with Kermut. These results demonstrate that mitigating instability due to data representation can yield improvements comparable to—or even greater than—those achieved through

the development of new algorithms, underscoring the critical role of representation choice. In addition to improved accuracy, Stable estimators also provide more reliable uncertainty quantification, as evidenced by stronger Spearman correlations between predicted uncertainty and absolute error of a model—exceeding a 50% gain for Kermut, and yielding several-fold improvements for CNN and Bayesian Ridge. Appendix 6.4 presents a detailed ablation of the Stable pipeline. Each component—ensembling, pred-check, and others—shows statistically significant gains in at least one setting (i.e., base model and split), and none reduce performance in any case.

4.2. In-silico protein engineering

Setup We conduct an in-silico protein engineering campaign with the goal of identifying high-fitness sequences using as few evaluations as possible. We implement an iterative Bayesian Optimization (BO) procedure, beginning with an initial batch of labeled sequences used to train a predictive model. At each iteration, we select the next batch of k sequences $(m_1, \ldots, m_k)$ using the upper confidence bound (UCB) acquisition function (which is commonly used in these settings (Gruver et al., 2021; Notin et al., 2023b; Greenman et al., 2025; Yang et al., 2025)), defined as:

$$\hat{f}(m) + \lambda \hat{\sigma}(m), \tag{1}$$

where $\hat{f}(m)$ denotes the predicted fitness, $\hat{\sigma}(m)$ represents the model's uncertainty and λ controls the exploration-exploitation tradeoff. We evaluate two choices for λ, $\{0.1, 2\}$, as proposed by Notin et al. (2023b); Yang et al. (2025) respectively, and find that $\lambda = 2$ yields stronger performance in recovering the highest-fitness sequences, while $\lambda = 0.1$ yields better performance in recovering a larger number of high fitness sequences (i.e., their fitness values belong to 70th or 90th percentile of all measured fitness values within their assay).

For each dataset, we initialize the BO loop with 50 randomly selected sequences (two datasets with less than 50 sequences are removed) and acquire 50 additional sequences per round over 5 rounds, resulting in a total of 250 labeled sequences. This setup mirrors the structure and scale of real-world protein engineering campaigns (Yang et al., 2025). We evaluate three base predictors—CNN ensemble, Bayesian Ridge, and Kermut—along with their Stable variants.

Results Stable variants of all base methods—Kermut, Bayesian Ridge, and CNN—consistently outperform their non-Stable counterparts. Figure 2 (b) reports the cumulative fraction of assays in which the highest-fitness sequence is recovered across five steps of Bayesian optimization (BO), averaged over three independent runs with different random initializations. At every step, Stable Kermut and Stable Bayesian Ridge recover the top sequence in more assays than their base versions. By step five, Stable Kermut shows a 6% absolute improvement over base Kermut.

Appendix 6.6 presents additional results, including comparisons at $\lambda = 0.1$ and further analysis on recovering multiple high-fitness sequences, measured using quantiles of each assay's fitness distribution.

In summary, for both top-sequence recovery and high-percentile recovery, Stable Kermut achieves the best overall performance—with $\lambda = 2$ performing best for identifying the top sequence, and $\lambda = 0.1$ performing best recovering a large number of high fitness sequences.

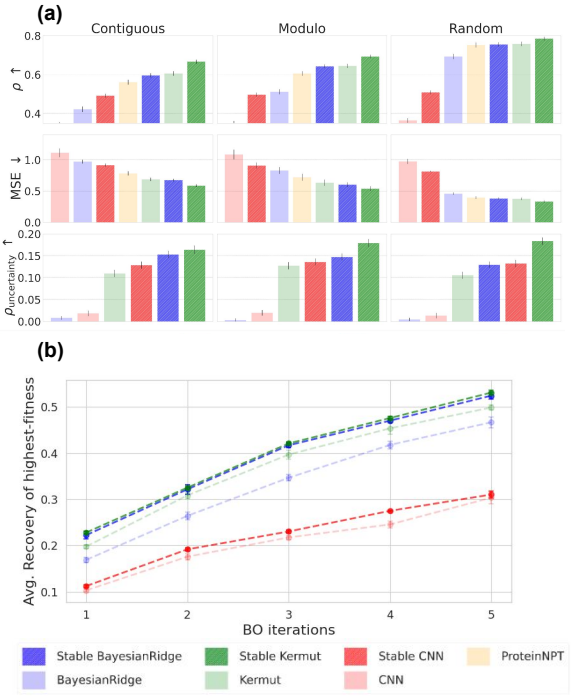

Figure 2. (a) ProteinGym benchmark. We report the average Spearman correlation (ρ), mean squared error (MSE), and the Spearman correlation between the uncertainty scores and the absolute errors of the predictions ($\rho_{\text{uncertainty}}$). Each column corresponds to a different ProteinGym split (b) Percent of assays for which the highest fitness sequence was recovered for across 5 BO iterations.

5. Discussion

This work highlights the impact of representation-induced instability in protein fitness prediction and shows that leveraging this variability—guided by the PCS framework—can improve performance, especially under distribution shift. While we studied a subset of reasonable representations (e.g., zero-shot scores and pre-trained embeddings), many choices (e.g., network layer) remain unexplored. Stabilizing predictions across these dimensions is a promising direction. Though Stable predictors improve accuracy, they incur

computational cost; our ensemble of 36 models is modest, and techniques like pred-checks or dimensionality reduction may help mitigate overhead. The PCS-guided approach may generalize to tasks in computational chemistry and beyond, where multiple featurization choices exist, to improve prediction and UQ.

References

Agarwal, A., Xiao, M., Barter, R., Ronen, O., Fan, B., and Yu, B. Pcs-uq: Uncertainty quantification via the predictability-computability-stability framework. *arXiv preprint arXiv:2505.08784*, 2025.

Baek, M., DiMaio, F., Anishchenko, I., Dauparas, J., Ovchinnikov, S., Lee, G. R., Wang, J., Cong, Q., Kinch, L. N., Schaeffer, R. D., et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. The protein data bank. *Nucleic acids research*, 28(1): 235–242, 2000.

Bhattacharya, N., Thomas, N., Rao, R., Dauparas, J., Koo, P. K., Baker, D., Song, Y. S., and Ovchinnikov, S. Single layers of attention suffice to predict protein contacts. *Biorxiv*, pp. 2020–12, 2020.

Boger, R. S., Chithrananda, S., Angelopoulos, A. N., Yoon, P. H., Jordan, M. I., and Doudna, J. A. Functional protein mining with conformal guarantees. *Nature Communications*, 16(1):85, 2025.

Dallago, C., Mou, J., Johnston, K. E., Wittmann, B. J., Bhattacharya, N., Goldman, S., Madani, A., and Yang, K. K. Flip: Benchmark tasks in fitness landscape inference for proteins. *bioRxiv*, pp. 2021–11, 2021.

Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R. J., Milles, L. F., Wicky, B. I. M., Courbet, A., de Haas, R. J., Bethel, N., Leung, P. J. Y., Huddy, T. F., Pellock, S., Tischer, D., Chan, F., Koepnick, B., Nguyen, H., Kang, A., Sankaran, B., Bera, A. K., King, N. P., and Baker, D. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022. doi: 10.1126/science.add2187. URL https://www.science.org/doi/abs/10.1126/science.add2187.

Fannjiang, C. and Park, J. W. Reliable algorithm selection for machine learning-guided design. *arXiv preprint arXiv:2503.20767*, 2025.

Fannjiang, C., Bates, S., Angelopoulos, A. N., Listgarten, J., and Jordan, M. I. Conformal prediction under feedback covariate shift for biomolecular design. *Proceedings of the National Academy of Sciences*, 119(43):e2204569119, 2022.

Greenman, K. P., Amini, A. P., and Yang, K. K. Benchmarking uncertainty quantification for protein engineering. *PLOS Computational Biology*, 21(1):e1012639, 2025.

Groth, P. M., Kerrn, M., Olsen, L., Salomon, J., and Boomsma, W. Kermut: Composite kernel regression for protein variant effects. *Advances in Neural Information Processing Systems*, 37:29514–29565, 2024.

Gruver, N., Stanton, S., Kirichenko, P., Finzi, M., Maffettone, P., Myers, V., Delaney, E., Greenside, P., and Wilson, A. G. Effective surrogate models for protein design with bayesian optimization. In *ICML Workshop on Computational Biology*, volume 198, 2021.

Hsu, C., Nisonoff, H., Fannjiang, C., and Listgarten, J. Learning protein fitness models from evolutionary and assay-labeled data. *Nature biotechnology*, 40(7):1114–1122, 2022a.

Hsu, C., Verkuil, R., Liu, J., Lin, Z., Hie, B., Sercu, T., Lerer, A., and Rives, A. Learning inverse folding from millions of predicted structures. In *International conference on machine learning*, pp. 8946–8970. PMLR, 2022b.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

Li, F.-Z., Amini, A. P., Yue, Y., Yang, K. K., and Lu, A. X. Feature reuse and scaling: Understanding transfer learning with protein language models. *bioRxiv*, pp. 2024–02, 2024a.

Li, F.-Z., Yang, J., Johnston, K. E., Gürsoy, E., Yue, Y., and Arnold, F. H. Evaluation of machine learning-assisted directed evolution across diverse combinatorial landscapes. *bioRxiv*, pp. 2024–10, 2024b.

Li, M., Tan, Y., Ma, X., Zhong, B., Yu, H., Zhou, Z., Ouyang, W., Zhou, B., Hong, L., and Tan, P. Prosst: Protein language modeling with quantized structure and disentangled attention. biorxiv. 2024c.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637): 1123–1130, 2023.

Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., and Rives, A. Language models enable zero-shot prediction of the

effects of mutations on protein function. *Advances in neural information processing systems*, 34:29287–29303, 2021.

Nix, D. A. and Weigend, A. S. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 ieee international conference on neural networks (ICNN'94)*, volume 1, pp. 55–60. IEEE, 1994.

Notin, P., Dias, M., Frazer, J., Marchena-Hurtado, J., Gomez, A. N., Marks, D., and Gal, Y. Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval. In *International Conference on Machine Learning*, pp. 16990–17017. PMLR, 2022a.

Notin, P., Van Niekerk, L., Kollasch, A. W., Ritter, D., Gal, Y., and Marks, D. S. Trancepteve: Combining family-specific and family-agnostic models of protein sequences for improved fitness prediction. *bioRxiv*, pp. 2022–12, 2022b.

Notin, P., Kollasch, A., Ritter, D., Van Niekerk, L., Paul, S., Spinner, H., Rollins, N., Shaw, A., Orenbuch, R., Weitzman, R., et al. Proteingym: Large-scale benchmarks for protein fitness prediction and design. *Advances in Neural Information Processing Systems*, 36:64331–64379, 2023a.

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

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Rao, R. M., Liu, J., Verkuil, R., Meier, J., Canny, J., Abbeel, P., Sercu, T., and Rives, A. Msa transformer. In *International Conference on Machine Learning*, pp. 8844–8856. PMLR, 2021.

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., 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.

Sillitoe, I., Bordin, N., Dawson, N., Waman, V. P., Ashford, P., Scholes, H. M., Pang, C. S., Woodridge, L., Rauer, C., Sen, N., et al. Cath: increased structural coverage of functional space. *Nucleic acids research*, 49(D1):D266–D273, 2021.

Suzek, B. E., Huang, H., McGarvey, P., Mazumder, R., and Wu, C. H. Uniref: comprehensive and non-redundant uniprot reference clusters. *Bioinformatics*, 23(10):1282–1288, 2007.

Tagasovska, N., Park, J. W., Kirchmeyer, M., Frey, N. C., Watkins, A. M., Ismail, A. A., Jamasb, A. R., Lee, E., Bryson, T., Ra, S., et al. Antibody domainbed: Out-of-distribution generalization in therapeutic protein design. *arXiv preprint arXiv:2407.21028*, 2024.

Varadi, M., Anyango, S., Deshpande, M., Nair, S., Natassia, C., Yordanova, G., Yuan, D., Stroe, O., Wood, G., Laydon, A., 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.

Varadi, M., Bertoni, D., Magana, P., Paramval, U., Pidruchna, I., Radhakrishnan, M., Tsenkov, M., Nair, S., Mirdita, M., Yeo, J., et al. Alphafold protein structure database in 2024: providing structure coverage for over 214 million protein sequences. *Nucleic acids research*, 52(D1):D368–D375, 2024.

Yang, J., Lal, R. G., Bowden, J. C., Astudillo, R., Hameedi, M. A., Kaur, S., Hill, M., Yue, Y., and Arnold, F. H. Active learning-assisted directed evolution. *Nature Communications*, 16(1):714, 2025.

Yang, K. K., Fusi, N., and Lu, A. X. Convolutions are competitive with transformers for protein sequence pretraining. *Cell Systems*, 15(3):286–294, 2024.

Yu, B. and Barter, R. L. *Veridical data science: The practice of responsible data analysis and decision making*. MIT Press, 2024.

Yu, B. and Kumbier, K. Veridical data science. *Proceedings of the National Academy of Sciences*, 117(8): 3920–3929, 2020. doi: 10.1073/pnas.1901326117. URL https://www.pnas.org/doi/abs/10.1073/pnas.1901326117.

6. Appendix

6.1. Reasonable choices for embedding a protein sequence

Table 1. A list of reasonable choices for the representation of a single protein. A single protein sequence can represented by any combination of zero-shot scores, embeddings from a PLM (with any choice of dimension reduction, layer or model) as well as structure and inverse folding information based on this structure. This represents a huge space of potential reasonable ways to represent a single protein sequence.

Category	Component	Reasonable choices
Zero-shot score	Method / Model	ESM1v (Meier et al., 2021), ESM2 (Lin et al., 2023), MSA Transformer (Rao et al., 2021), ProSST (Li et al., 2024c)
Embedding	Method / Model	ESM1v (Meier et al., 2021), ESM2 (Lin et al., 2023), CARP (Yang et al., 2024), MSA Transformer (Rao et al., 2021), One-hot (Hsu et al., 2022a)
	Layer	Penultimate, first/middle layers, or learned combinations across attention layers (Bhattacharya et al., 2020)
	Dimension reduction	Mean pooling, max pooling, pool over sequence dimension, pool over hidden dimension, mean pooling over mutated sites, flattening (Dallago et al., 2021; Li et al., 2024b).
Structure	3D coordinates	AlphaFold2 (Jumper et al., 2021), RosettaFold (Baek et al., 2021), Experimental (PDB, (Berman et al., 2000))
	Inverse folding	ESM-IF (Hsu et al., 2022b), ProteinMPNN (Dauparas et al., 2022)

6.2. List of embeddings and zero shot score

6.2.1. Selected representative zero-shot scores

We describe below the different methods included in our study for zero-shot score prediction. Our focus was to include a set of diverse methods, which include PLMs, MSA based models and structure based models.

ESM2 ESM2 (Lin et al., 2023) is a transformer-based protein language model. We use the 650 million parameters version consisting of 33 layers. It follows a BERT style encoder only transformer architecture and is trained on the UniRef50 (Suzek et al., 2007) protein sequence database using a masked language modeling objective.

MSA Transformer MSA Transformer (Rao et al., 2021) uses a specialized transformer architecture with interleaved row and column (axial) self-attention layers to process multiple sequence alignments (MSAs). It has approximately 100 million parameters and 12 layers, and is trained on 26 million MSAs. An MSA is generated for each UniRef50 sequence.

TranceptEVE TranceptEVE (Notin et al., 2022b) is a hybrid model that integrates an autoregressive transformer (Tranception (Notin et al., 2022a)) with a variational autoencoder (EVE) trained on family-specific MSAs. Tranception offers three model variants; the best-performing TranceptionEVE configuration on the ProteinGym indel benchmark uses the medium-sized Tranception model (16 attention heads, 24 layers, 1024-dimensional embeddings) trained on UniRef100. EVE is trained on MSAs built from UniRef100 for 3,219 clinically relevant proteins.

ProSST ProSST (Li et al., 2024c) consists of a transformer model integrated with a geometric vector perceptron (GVP) encoder that encodes 3D structural information. The model has around 250 million parameters, with the transformer comprising 12 layers and the GVP encoder using approximately six layers. This model is pretrained on data collected from AlphaFoldDB (Jumper et al., 2021; Varadi et al., 2022; 2024), which contained more than 214 million structures. The dataset used for training the structure encoder is extracted from CATH43-S40 (Sillitoe et al., 2021), a dataset of manually annotated protein crystal structural domains.

6.2.2. Selected representative embedding models

We describe below the embeddings models we used, our focus was to include at least two models with different architecture. ESM2 and ESM1v are similar and were included for ease of implementation.

ESM2 ESM2 (Lin et al., 2023) is a transformer-based protein language model. We use the 650 million parameters version consisting of 33 layers. It follows a BERT style encoder only transformer architecture and is trained on the UniRef50 protein sequence database using a masked language modeling objective. ESM2's hidden dimension is 1280.

CARP CARP (Convolutional Autoencoding Representations of Proteins (Yang et al., 2024)) is a convolutional neural network model based on a ByteNet encoder architecture with dilated convolutions. We used the CAPR model with approximately 640 million parameters and 56 layers. The model is trained on UniRef50 using a masked language modeling task. Unlike transformers, CARP captures long-range dependencies via convolution, scaling linearly with sequence length. Embeddings are taken from the final hidden representations, with a hidden dimension of 1280.

ESM1v ESM1v (Meier et al., 2021) is a transformer model based on the ESM1b architecture, optimized for zero-shot variant effect prediction. It has 650 million parameters and 33 layers, with a hidden dimension of 1,280. The final model is an ensemble of five independently trained networks, each trained on UniRef90. We extract embeddings from the last hidden layer (layer 33) of the first model in the ensemble.

6.3. Pred-check procedure

Following the PCS framework, we apply a **prediction check (Pred-Check)** step to filter representations that are less predictive for a particular fitness prediction problem.

The pred-check procedure has two main goals:

- **Filtering:** Remove the worst-performing zero-shot scores.
- **Weighting:** Assign weights to the remaining scores based on their predictive quality.

Procedure Let $\{y_i\}_{i=1}^n$ denote the training labels and $\{z_i^{(j)}\}_{i=1}^n$ for $j = 1, \ldots, k$ denote the values of the zero-shot scores. Our procedure involves the following steps:

1. Assign a prediction score ($s^{(i)}$) for every zero-shot scores vector ($\{z_i^{(j)}\}_{i=1}^n$) using the training labels ($\{y_i\}_{i=1}^n$).
2. Keep the top k scores (assuming higher is better). We set $k = 3$, but find that similar performance is obtained with $k = 4$ or $k = 2$.
3. Obtain the weights by normalizing the prediction score via soft-max transformation.

We consider the following prediction scores:

- **Correlation (Corr)** — For each zero-shot score vector, we compute the Spearman correlation with the training fitness labels.

- **LASSO** — All zero-shot score vectors are concatenated into a feature matrix. We then fit a LASSO regression model (using 5-fold cross-validation to select the regularization strength (Pedregosa et al., 2011)). The prediction score for each zero-shot score is the absolute value of its corresponding LASSO coefficient, reflecting its importance in predicting fitness. This can be viewed as a form of feature selection.

- **RF-MDI** — As with LASSO, the zero-shot scores are concatenated into a feature matrix. A Random Forest model is trained to predict fitness, and the prediction score for each zero-shot score is its Mean Decrease in Impurity (MDI) importance value.

We highlight that the above pred-check procedure does not require any held-out calibration set, and is done using the training set only.

6.4. Ablation studies

We perform an ablation study to quantify the contribution of each component in our StaPred fitness predictors. Specifically, we ablate the following elements: (1) ensembling across multiple embeddings, (2) ensembling across dimensionality reduction methods, and (3) the pred-check procedure (i.e., use of RF-MDI and its advantage over using a single score or just an average).

Each ablation is evaluated by measuring the change in Spearman correlation across 217 ProteinGym assays. For each component, we report box plots and p-values from two-sample t-tests assessing whether the component improves average correlation across assays. The specific ablations are described below:

Embedding To assess the benefit of ensembling multiple embeddings (ESM1v (Meier et al., 2021), ESM2 (Lin et al., 2023), and CARP (Yang et al., 2024)), we compare the performance of the full ensemble (after pred-check) to that of a variant that uses only a single embedding model. Both versions use the same weighting scheme based on RF-MDI.

Pred-Check To isolate the impact of the pred-check step, we compare our RF-MDI procedure to the three alternative methods described in Section 6.3. We also include a baseline that uses each zero-shot score individually in the ensemble. All settings use the same embeddings and dimensionality reduction methods; only the selection and weighting of zero-shot scores differ.

Dimensionality Reduction To evaluate the effect of ensembling across dimensionality reduction methods, we compare the full ensemble to a variant that uses only mean pooling for each embedding. Both use the same zero-shot selection and weighting via RF-MDI.

Results The results for StaPred Kermut, StaPred Bayesian Ridge, and StaPred CNN are shown in Figures 3, 4, and 5, respectively. For all three base models, the pred-check step and dimensionality reduction ensemble lead to statistically significant improvements. Ensembling embeddings yields additional gains for StaPred Kermut and StaPred Bayesian Ridge. For StaPred CNN—the weakest overall model—embedding ensembling performs similarly to using ESM2 alone, but not worse.

6.5. Detailed description of fitness predictors

Bayesian Ridge The Bayesian Ridge model assumes that the fitness label follows a Gaussian likelihood given the d-dimensional representation of sequence ($\mathbf{x}(m)$) (which we consider as the concetanation of the mean-pooled embeddings with a zero-shot score):

$$y|\mathbf{x}(m) \sim N\left(\sum_{i=1}^{d} x(m)_i \beta_i, \sigma^2\right) \tag{2}$$

where β_i is the weight of the i-th feature, and σ^2 is the variance of the noise.

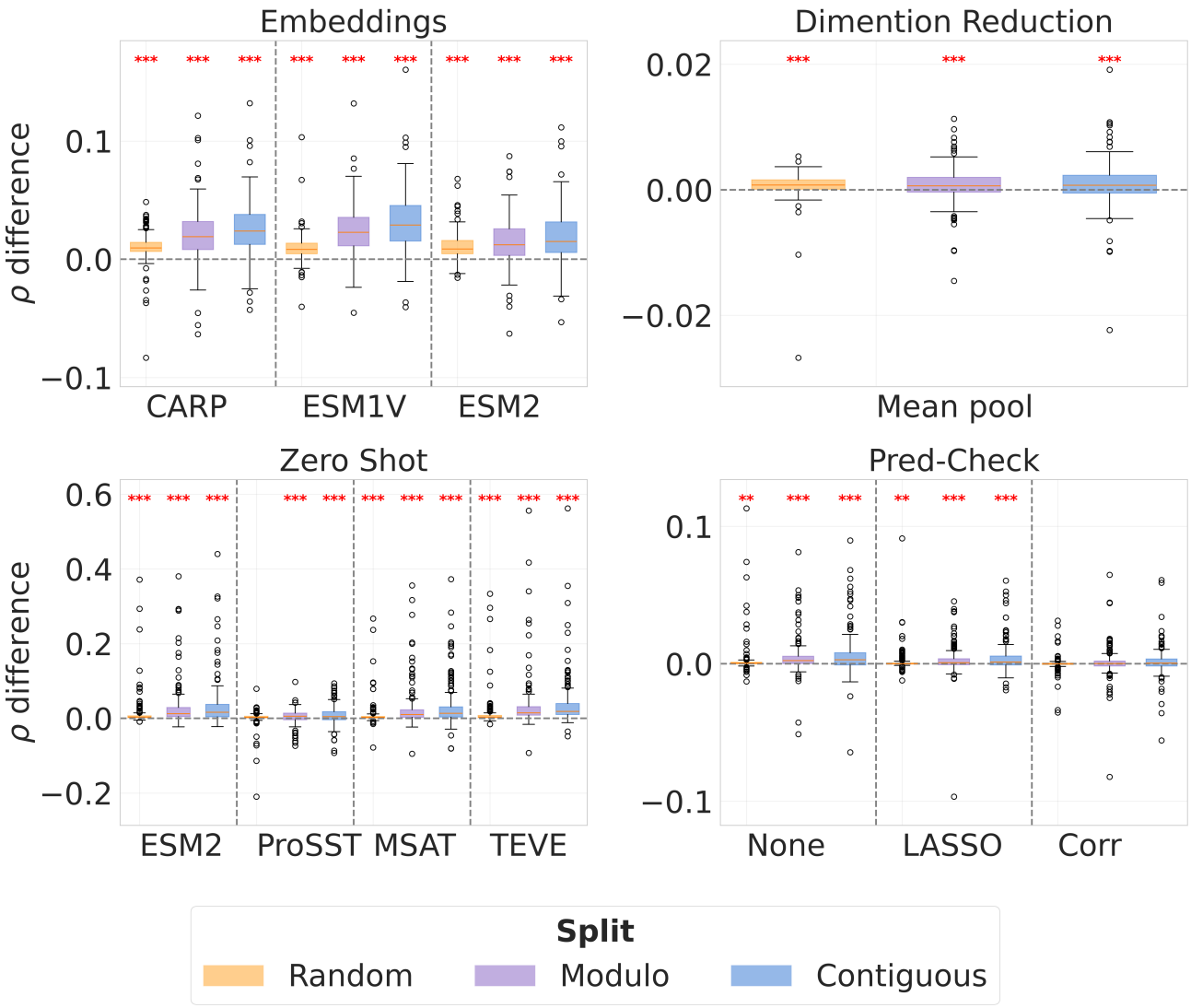

Figure 3. Each component of StaPred Kermut improves prediction accuracy. Box plots show the change in test set Spearman correlation when ablations are applied to individual components. Asterisks indicate significance levels from one-sided two-sample t-tests (*¡0.05, **¡0.01, ***¡0.001), and colors denote different data splits. The top left panel compares StaPred Kermut to a model using only a single embedding; the top right to a model with only mean pooling instead of multiple randomized dimensionality reduction; the bottom left to a model using only a single zero-shot score (MSAT is short for MSA Transformer and TEVE is a short for TranceptEVE) ; and the bottom right compares the RF-MDI pred-check to alternative pred-check procedures. Ensembling embeddings and random dimensionality reduction both significantly improve performance. The RF-MDI pred-check outperforms each individual zero-shot score and their ensemble, and performs comparably to the correlation-based method (Corr).

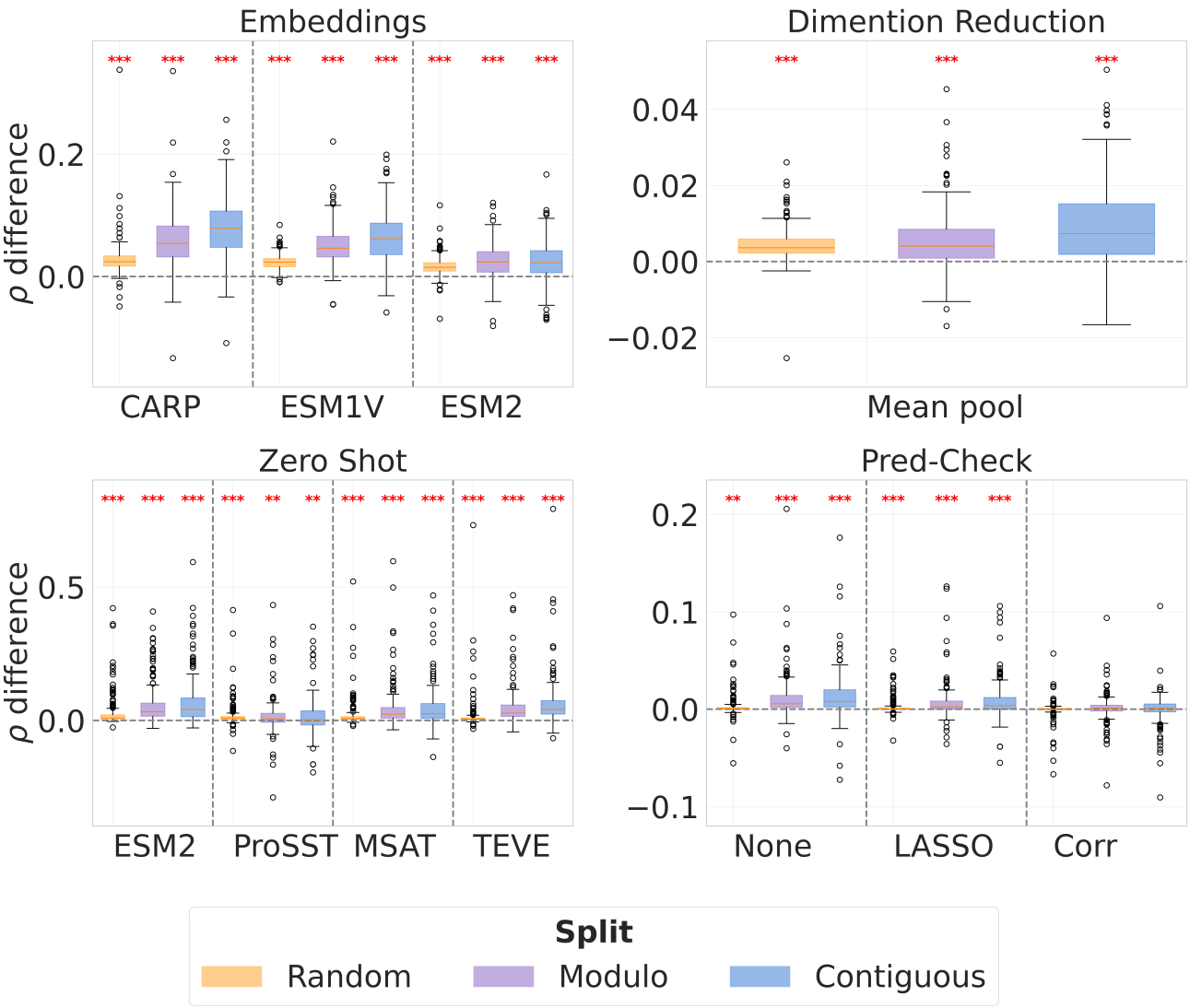

Figure 4. Each component of StaPred Bayesian Ridge improves prediction accuracy. Box plots show the change in test set Spearman correlation when ablations are applied to individual components. Asterisks indicate significance levels from one sided two-sample t-tests (*¡0.05, **¡0.01, ***¡0.001), and colors denote different data splits. The top left panel compares StaPred Bayesian Ridge to a model using only a single embedding; the top right to a model with only mean pooling instead of multiple randomized dimensionality reduction; the bottom left to a model using only a single zero-shot score (MSAT is short for MSA Transformer and TEVE is a short for TranceptEVE); and the bottom right compares the RF-MDI pred-check to alternative pred-check procedures. Ensembling embeddings and random dimensionality reduction both significantly improve performance. The RF-MDI pred-check outperforms each individual zero-shot score and their ensemble, and performs comparably to the correlation-based method (Corr).

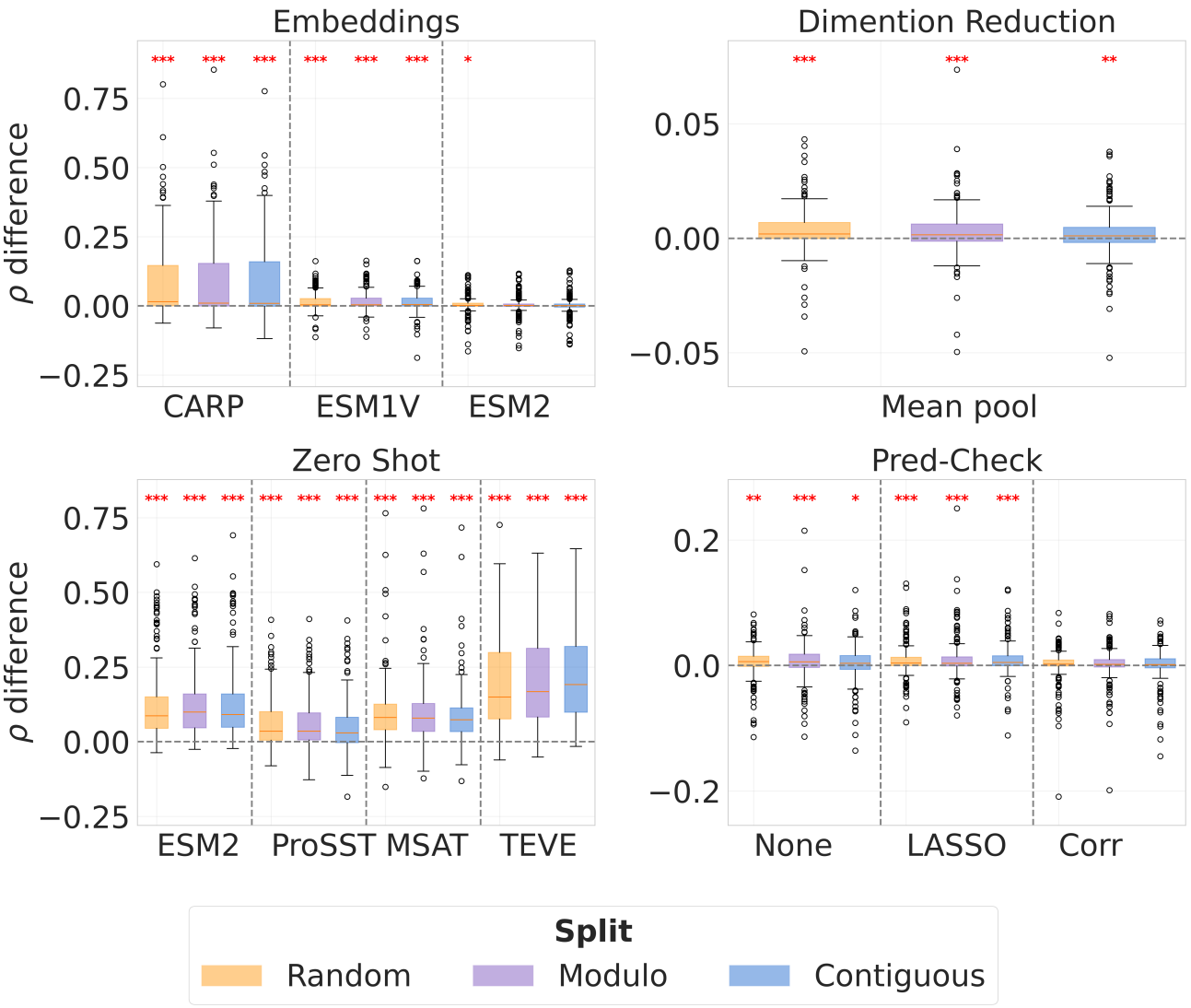

Figure 5. Each component of StaPred CNN improves or does not hurt prediction accuracy. Box plots show the change in test set Spearman correlation when ablations are applied to individual components. Asterisks indicate significance levels from one-sided two-sample t-tests (*¡0.05, **¡0.01, ***¡0.001), and colors denote different data splits. The top left panel compares StaPred CNN to a model using only a single embedding; the top right to a model with only mean pooling instead of multiple randomized dimensionality reduction; the bottom left to a model using only a single zero-shot score (MSAT is short for MSA Transformer and TEVE is a short for TranceptEVE); and the bottom right compares the RF-MDI pred-check to alternative pred-check procedures. Random dimensionality reduction significantly improves performance, while ensembeling of embedding is on par with using ESM2. The RF-MDI pred-check outperforms each individual zero-shot score and their ensemble, and performs comparably to the correlation-based method (Corr).

The weights are assumed to follow a Gaussian prior:

$$\beta_i|\alpha_i \sim N(0, \alpha_i^{-1}), \tag{3}$$

where α_i is the precision parameter for the i-th weight. The model also places gamma priors on α_i and the noise precision $\tau = 1/\sigma^2$:

$$\alpha_i \sim \mathit{Gamma}(a_0, b_0), \quad \tau \sim \mathit{Gamma}(c_0, d_0) \tag{4}$$

The hyperparameters a_0, b_0, c_0, and d_0 are set to small values (10^{-6}) following the default scikit-learn (Pedregosa et al., 2011) implementation.

Kermut The Kermut method (Groth et al., 2024) defines a Gaussian Process for distribution of the fitness given the sequence. Its mean function is defined using a zero-shot score, while its kernel is defined as the following product:

$$K_{\mathit{Kermut}}(x, x') = \pi K_{\mathit{seq}}(x, x') + (1 - \pi) K_{\mathit{struc}}(x, x'), \tag{5}$$

where $K_{\mathit{seq}}(x, x')$ represents a sequence kernel defined using RBF Kernel applied to mean pooled embeddings of a PLM. and $K_{\mathit{struc}}(x, x')$ is a structure kernel defined as the multipication of three parts (1) A Hellinger Kernel which is a negative exponential of the Hellinger distance between the inverse folding probabilities between the mutated sites. (2) An exponential kernel in the abolute difference between the log-probabilites of the specific amino acid mutations assinged by an inverse folding model, and (3) a distance kernel which is the negative exponential of the physical distance between the mutation sites. We set $\pi = 0.5$ in our experiements as it is the default values in the Kermut implementation.

CNN Model We use a CNN architecture with the following structure:

- Three 1D convolutional layers with [4, 4, 6] filters respectively, each with kernel size 8
- Each conv layer is followed by ReLU activation, batch normalization, and dropout (p=0.1)
- Global average pooling after the final conv layer
- A tanh activation followed by a linear layer that outputs a single value

The model is trained using Adam optimizer with learning rate 1e-3 and batch size 128 for 100 epochs, using the concetanation of the mean-pooled embeddings with a zero-shot score.

6.6. Additional BO results

We report additional BO results. For $\lambda = 0.1$, Figure 6 shows, at each BO iteration, the cumulative fraction of evaluated sequences whose fitness exceeds the 90th and 70th percentiles, averaged over all datasets, with $\pm$1,s.d. computed from three independent runs (each initialized with a different random subset). Figure 8 presents the same analysis for $\lambda = 2$. Finally, Figure 7 plots the running percentage of datasets in which at least one highest-fitness sequence has been recovered for $\lambda = 0.1$.

StaPred variants outperform Kermut (and Bayesian Ridge) on the high-percentile metrics at both $\lambda = 0.1$ and $\lambda = 2$, with the larger gains at $\lambda = 0.1$. For recovery of the highest-fitness sequence, StaPred Kermut and StaPred Bayesian Ridge match Kermut at $\lambda = 0.1$, but both show clear improvements at $\lambda = 2$, surpassing their own and Kermut's $\lambda = 0.1$ performance.

6.7. Additional ProteinGym benchmark results

6.7.1. Results on single mutants

We report the numerical results for the random in Table 4, modulo Table 3 and contiguous in Table 2

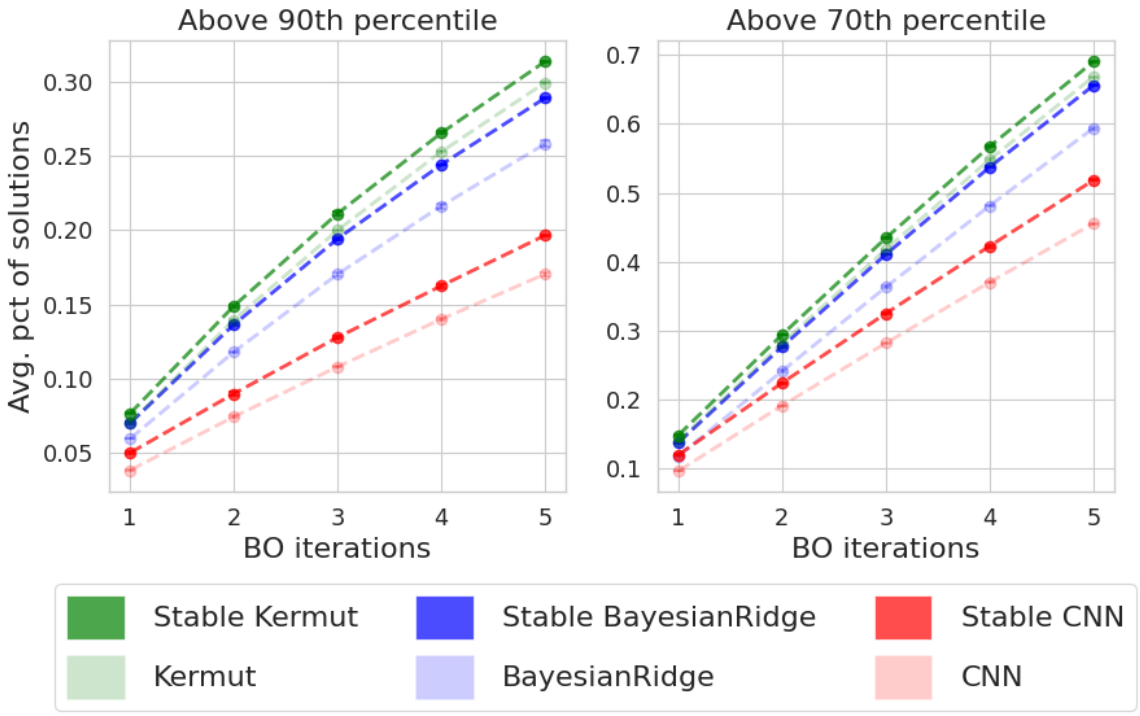

Figure 6. StaPred predictors recover high-fitness sequences more often than their base models when using BO with $\lambda = 0.1$. The y-axis shows the cumulative percentage of solutions that are above the 90th (left) and 70th (right) percentiles of the assay's fitness distribution. The x-axis shows the BO step. StaPred Kermut provides the strongest results under this setting.

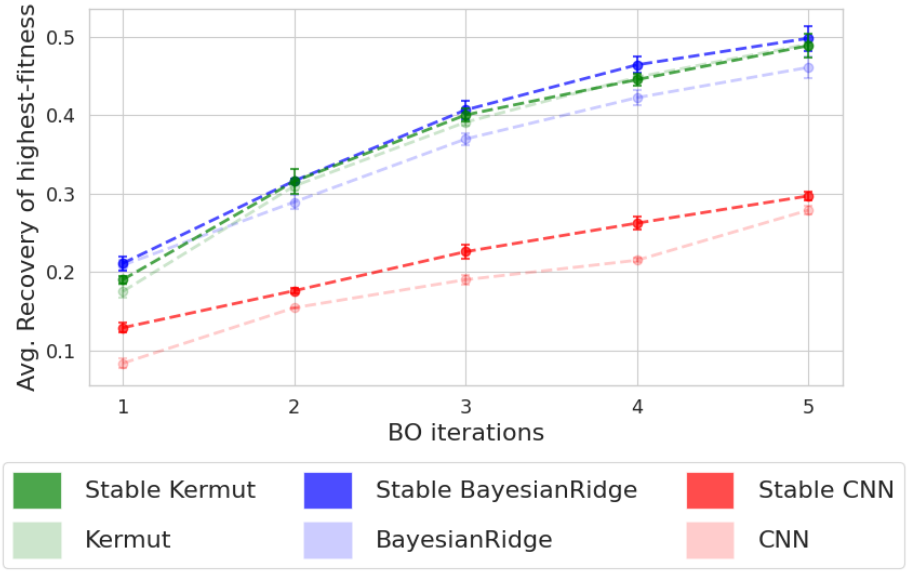

Figure 7. StaPred predictors recover the highest-fitness sequence more often—or at least as often—as their base models when using BO with $\lambda = 0.1$. The y-axis shows the percentage of experiments in which the top-fitness sequence is recovered; the x-axis shows the number of BO steps. Results are averaged over three random training set initializations, with error bars showing standard deviation (often too small to be visible). Both StaPred Kermut and StaPred Bayesian Ridge perform worse under $\lambda = 0.1$ compared to $\lambda = 2$.

6.7.2. Results on multiple mutants

We consider the same setup as in (Groth et al., 2024), where we analyze the 51 datasets with less than 7500 sequences (due to Kermut's memory requirements which require to store the covariance matrix). We consider two splitting strategies (1) *random* split where 80% of the data point are assigned to the training set and 20% to the test set and (2) *one vs. two* where all single mutations are assigned to the training set and all double mutations are assigned to the test set. Similar to (Groth et al., 2024) we find that using zero-shot score hurts performance in the one vs. two setting and does provide significant improvement in the random setting. We therefore report the results for all method without using any zero-shot scores (i.e., mean function of Kermut and StaPred Kermut is zero).

Results The results are presented in Figure 9 and Tables 6 and 5 for one vs. two and random, respectively. On the challenging one vs. two split, StaPred Bayesian Ridge provides the strongest spearman results with an average 0.72 compared with 0.69 for Bayesian Ridge and 0.67 for both Kermut and StaPred Kermut. For the random split StaPred Kermut achieves a spearman value of 0.95 surpassing Kermut with Spearman of 0.94. StaPred Bayesian Ridge is able to match Kermut's performance with Spearman of 0.94.

On both random and one vs. two splits, StaPred Kermut and StaPred Bayesian Ridge predictors provide uncertainty values whose correlation with the model errors are higher compared with their base models.

6.8. Computational Resources

All experiments in this work were carried out using a single A100 GPU.

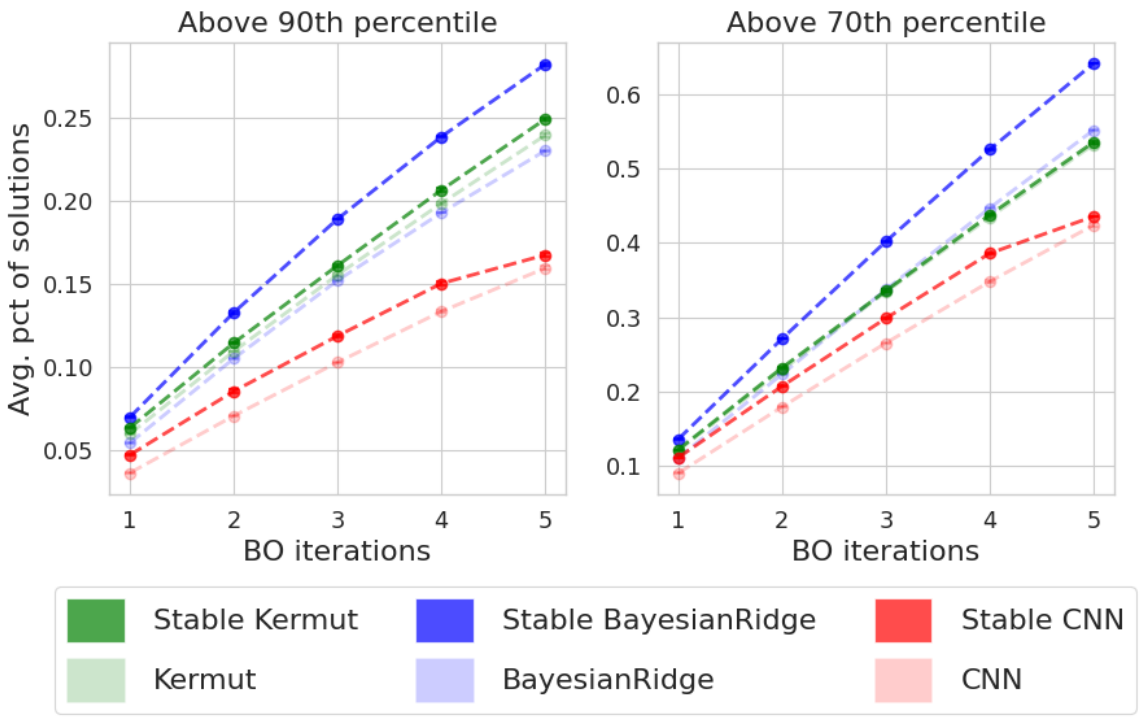

Figure 8. StaPred predictors recover high-fitness sequences more often—or at least as often—as their base models when using BO with $\lambda = 2$. The y-axis shows the cumulative percentage of solutions that are above the 90th (left) and 70th (right) percentiles of the assay's fitness distribution. The x-axis shows the BO step. StaPred Bayesian Ridge provides the strongest results under this setting; however, it underperforms compared to both Kermut and StaPred Kermut with $\lambda = 0.1$.

Figure 9. ProteinGym benchmark results on 51 multiples mutation datasets studied in (Groth et al., 2024). We report the average Spearman correlation (ρ), mean squared error (MSE), and the Spearman correlation between the uncertainty scores and the absolute errors of the predictions ($\rho_{\text{uncertainty}}$). Each column corresponds to a different ProteinGym split. StaPred predictors outperform their base models on Spearman correlation and on the correlation of the uncertainty with the absolute error.

Table 2. Benchmark results on ProteinGym single-substitution assays for the contiguous train / test split. ProteinNPT results are taken from the ProteinGym website, which does not provide uncertainty estimates.

Model	$\rho \uparrow$	MSE $\downarrow$	$\rho_{uncertainty} \uparrow$
CNN	0.344 ± 0.013	1.113 ± 0.070	0.019 ± 0.006
Stable CNN	0.492 ± 0.010	0.916 ± 0.026	0.129 ± 0.008
Bayesian Ridge	0.422 ± 0.014	0.973 ± 0.031	0.008 ± 0.004
Stable Bayesian Ridge	0.597 ± 0.011	0.677 ± 0.022	0.153 ± 0.008
Kermut	0.606 ± 0.012	0.688 ± 0.028	0.110 ± 0.008
Stable Kermut	0.667 ± 0.010	0.587 ± 0.020	0.164 ± 0.010
ProteinNPT	0.561 ± 0.013	0.784 ± 0.034	-

Table 3. Benchmark results on ProteinGym single-substitution assays for the modulo train / test split. ProteinNPT results are taken from the ProteinGym website, which does not provide uncertainty estimates.

Model	$\rho \uparrow$	MSE $\downarrow$	$\rho_{uncertainty} \uparrow$
CNN	0.344 ± 0.013	1.113 ± 0.070	0.019 ± 0.006
StaPred CNN	0.492 ± 0.010	0.916 ± 0.026	0.129 ± 0.008
Bayesian Ridge	0.422 ± 0.014	0.973 ± 0.031	0.008 ± 0.004
StaPred Bayesian Ridge	0.597 ± 0.011	0.677 ± 0.022	0.153 ± 0.008
Kermut	0.606 ± 0.012	0.688 ± 0.028	0.110 ± 0.008
StaPred Kermut	0.667 ± 0.010	0.587 ± 0.020	0.164 ± 0.010
ProteinNPT	0.561 ± 0.013	0.784 ± 0.034	-

Table 4. Benchmark results on ProteinGym single-substitution assays for the random train / test split. ProteinNPT results are taken from the ProteinGym website, which does not provide uncertainty estimates.

Model	$\rho \uparrow$	MSE $\downarrow$	$\rho_{uncertainty} \uparrow$
CNN	0.365 ± 0.012	0.975 ± 0.040	0.013 ± 0.006
StaPred CNN	0.509 ± 0.010	0.815 ± 0.011	0.133 ± 0.008
Bayesian Ridge	0.693 ± 0.013	0.460 ± 0.020	0.005 ± 0.004
StaPred Bayesian Ridge	0.755 ± 0.011	0.382 ± 0.016	0.130 ± 0.007
Kermut	0.758 ± 0.012	0.377 ± 0.019	0.106 ± 0.008
StaPred Kermut	0.785 ± 0.010	0.334 ± 0.016	0.184 ± 0.008
ProteinNPT	0.753 ± 0.013	0.397 ± 0.022	-

Table 5. Benchmark results on ProteinGym multiple-substitution assays for the random train / test split.

Model	$\rho \uparrow$	MSE $\downarrow$	$\rho_{uncertainty} \uparrow$
CNN	0.423 ± 0.018	1.027 ± 0.037	0.083 ± 0.019
StaPred CNN	0.588 ± 0.020	0.757 ± 0.020	0.003 ± 0.018
Bayesian Ridge	0.927 ± 0.004	0.140 ± 0.008	0.023 ± 0.004
StaPred Bayesian Ridge	0.939 ± 0.004	0.118 ± 0.008	0.214 ± 0.013
Kermut	0.935 ± 0.005	0.129 ± 0.014	0.203 ± 0.015
StaPred Kermut	0.945 ± 0.004	0.110 ± 0.008	0.264 ± 0.014

Table 6. Benchmark results on ProteinGym multiple-substitution assays for the one vs. two train / test split.

Model	$\rho \uparrow$	MSE $\downarrow$	$\rho_{uncertainty} \uparrow$
CNN	0.324 ± 0.030	1.417 ± 0.203	0.021 ± 0.031
StaPred CNN	0.384 ± 0.030	1.463 ± 0.181	0.015 ± 0.032
Bayesian Ridge	0.678 ± 0.028	1.189 ± 0.157	0.057 ± 0.023
StaPred Bayesian Ridge	0.715 ± 0.024	0.991 ± 0.133	0.182 ± 0.024
Kermut	0.666 ± 0.022	0.666 ± 0.117	0.124 ± 0.024
StaPred Kermut	0.669 ± 0.026	0.689 ± 0.121	0.143 ± 0.027