Stabilizing protein fitness predictors via the PCS framework

Omer Ronen **Alex Zhao** **Ron Boger** **Chengzhong Ye**

Bin Yu

University of California, Berkeley
{omer_ronen, alex_zhao, ronb, czye, binyu}@berkeley.edu

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 *SP* 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 *SP* 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 *SP* 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 far-reaching implications for protein design, disease understanding, and beyond. In protein engineering, ML has shown promise in reducing the experimental cost associated with discovering high-fitness protein sequences—by prioritizing which sequences to test in costly wet-lab experiments [Yang et al., 2025].

However, successfully utilizing ML methods to guide wet lab experiments poses a few unique challenges. First, as the goal is to identify new protein sequences (typically mutations of given reference sequence) with desired functional properties, these methods must be able to provide useful predictions for out-of-distribution (OOD) sequences. Second, uncertainty quantification (UQ) is essential for assessing the reliability of predictions, as no existing model reliably generalizes across the vast, combinatorial space of protein sequences.

Existing efforts to improve OOD generalization [Tagasovska et al., 2024] and provide reliable UQ for protein engineering [Greenman et al., 2025] primarily focus on the modeling stage of the data-science life cycle (DSLC Yu and Kumbier [2020]). For UQ, Bayesian methods offer uncertainty estimates via the posterior distribution [Greenman et al., 2025, Notin et al., 2023b], while ensembling neural networks with random initializations captures variability due to randomness of the training procedure

First Exploration in AI Today Workshop at ICML (EXAIT at ICML 2025).

[Gruver et al., 2021]. Additionally, frequentist approaches estimate predictive variance through probabilistic modelling [Nix and Weigend, 1994, Greenman et al., 2025]. Conformal prediction techniques, which quantify uncertainty in the form of prediction intervals, have been used for model selection for protein design and function prediction [Fannjiang et al., 2022, Boger et al., 2025, Fannjiang and Park, 2025]. While the use of pre-trained protein language models (PLMs) and ensembling has shown promising results for OOD generalization in antibody design [Tagasovska et al., 2024], there remains a critical need to further strengthen these capabilities to ensure their utility across a broader range of protein engineering tasks [Yang et al., 2024a].

The goal of this work is to highlight the importance of stability under *reasonable* (for the purpose of prediction) perturbations to data processing choices, with a particular focus on protein engineering. Following the Predictability-Computability-Stability (PCS) framework for veridical data science [Yu and Kumbier, 2020, Yu and Barter, 2024], we introduce 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 derived from different pre-trained PLMs, or zero-shot evolutionary scores obtained by different methods. Our key contributions are:

- We identify a previously underappreciated but critical source of instability in protein fitness prediction: the choice of data representation (i.e., embeddings). Our findings reveal that the performance of state-of-the-art models can vary substantially depending on these choices (Section 4 and Appendix .4).
- We propose a simple and intuitive two-step procedure (Figure 1 and Section 3) to leverage multiple reasonable data processing choices, guided by the PCS framework. Our procedure has two steps, (1) **Pred-Check** step to select and weight representations which are predictive for the fitness of interest, and (2) an **ensembling** step that fits a given base method on each representation that passes the pred-check, yielding a Stable and Pred-Checked (**SP**) fitness predictor.
- On the supervised ProteinGym substitution benchmark, SP predictors consistently outperform their base models—including Gaussian Processes (GPs), linear models, and CNNs—in both predictive accuracy and uncertainty estimation (Section 4.1). Among them, SP Kermut improves over the 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 largest improvements observed under distribution shifts. For uncertainty, SP Kermut also achieves up to 70% stronger correlation between predicted uncertainty, and the true absolute errors compared to standard Kermut, which relies on a single representation.
- Finally, we show that our SP versions of Bayesian Ridge and Kermut methods improve performance in in-silico iterative protein design, leading to 6% improvement in recovering the highest fitness sequences over the prior best performing base method Kermut (Section 4.2).

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. We define a *mutated sequence* relative to p as a sequence of the same length that differs from p at one or more positions, and is 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 $\hat{f}^{(p)} : \mathbb{M}_p \to \mathbb{R}$ that maps a mutated sequence to its corresponding fitness value y—a scalar quantity that may reflect properties such as thermostability or binding affinity to a given ligand. 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 $m_i^{(p)}$.

Representations of protein sequences To apply ML methods for fitness prediction, protein sequences must first be converted into numerical representations, or *embeddings*, which serve as inputs to models trained to predict observed fitness values. The simplest and still commonly used approach is the one-hot encoding, where a sequence of length l is embedded as a vector of dimension $20 \times l$ by concatenating l one-hot vectors corresponding to the 20 canonical amino acids [Hsu et al., 2022a].

More recently, embeddings derived from pre-trained PLMs [Rives et al., 2021, Rao et al., 2021, Lin et al., 2023, Yang et al., 2024b] have demonstrated superior performance in some prediction tasks

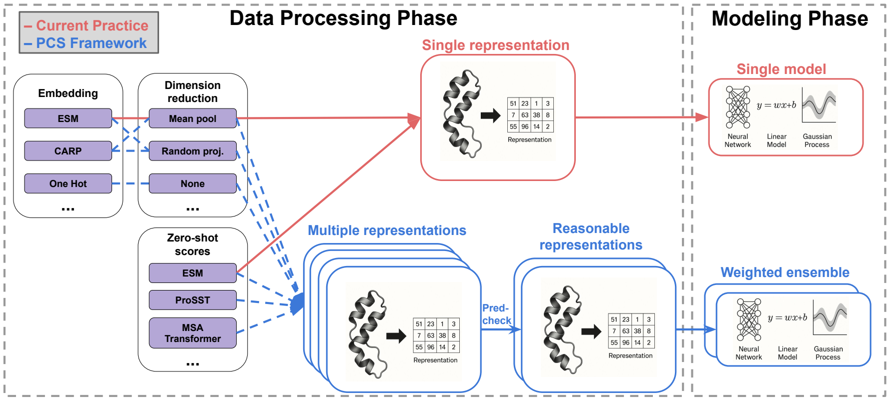

Figure 1: We stabilize protein fitness prediction by following the PCS framework. Our approach leverages variability at the representation level by considering multiple representations, filtering out those that are less predictive through a pred-check step, and ensembling across the remaining ones. This leads to substantial gains in both prediction accuracy and uncertainty quantification, highlighting the importance of considering multiple reasonable representations for a protein sequence.

[Li et al., 2024a] due to their ability to transfer information learned from large-scale unsupervised pre-training. These models typically embed a protein sequence as a matrix of size $l \times h$, where h is the hidden dimension of the PLM. This matrix is often reduced to a fixed-size vector, most commonly via mean pooling across the sequence dimension.

In addition to embeddings, PLMs can produce scalar representations of mutated sequences by approximating the log-likelihood ratio between a mutant and a reference sequence, using the sequence distribution learned during training. These likelihood-based scores—known as zero-shot scores—have been shown to correlate with various measures of fitness without requiring supervised training [Meier et al., 2021, Notin et al., 2023a].

Beyond PLMs, structure-based representations have also been explored for protein fitness prediction [Groth et al., 2024]. These include features derived from predicted 3D structures using methods like AlphaFold2 [Jumper et al., 2021] or from inverse folding models such as ESM-IF and ProteinMPNN [Hsu et al., 2022b, Dauparas et al., 2022].

It is important to emphasize that the space of *reasonable*[1] choices for representing a single protein sequence is prohibitively large. For example, Li et al. [2024a] show that different protein language models can produce embeddings that perform equally well in prediction tasks, and that increasing model size or pre-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 2 provides a non-exhaustive list of different and reasonable choices for embedding a single protein sequence.

Methods for fitness prediction The simplest approach for predicting protein fitness is to use a linear model trained on the one-hot encoding of sequences. Such models learn a distinct coefficient for each specific mutation, preventing them from generalizing to mutations unseen in training (whose coefficient value is zero). Additionally, as linear models, they are unable to capture non-linear, epistatic interactions between mutations. Alternatively, zero-shot scores—such as those from PLMs that estimate the density of their training data—typically naturally occurring sequences—have been empirically shown to correlate with fitness without additional training [Meier et al., 2021, Livesey and Marsh, 2023]. These scores generalize across mutation sites and can capture epistasis. But as

[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.

Table 1: A list of reasonable choices for the representation of a single protein. A single protein sequence can be 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., 2024b], 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]

scalar values, they cannot be improved using labeled data, and they perform poorly when natural sequence density does not align with the definition of fitness of interest.

The hybrid approach proposed by Hsu et al. [2022a] combines supervised learning with zero-shot scores. It does this by concatenating a one-hot sequence embedding with a chosen zero-shot score, then fitting a Ridge regression model. This simple method strikes a strong balance between bias and variance, and often beats more complex approaches like deep neural networks or fine-tuned PLMs. In some cases, using embeddings from pre-trained PLMs instead of one-hot encodings can further boost performance [Yang et al., 2018]. While ESM2 [Lin et al., 2023] is a popular default for such embeddings, other models—such as CARP [Yang et al., 2024b]—can match or even surpass its performance in specific settings [Li et al., 2024a].

ProteinNPT [Notin et al., 2023b] is a non-parametric transformer [Kossen et al., 2021] model that combines embeddings from the pre-trained MSA Transformer [Rao et al., 2021] with zero-shot scores, achieving substantial gains over the hybrid approach, at a high computational cost [Notin et al., 2023b, Groth et al., 2024]. Kermut [Groth et al., 2024] is a GP model for the distribution of the protein fitness conditioned on sequence. Its kernel is the sum of two main components (see Appendix .5 for details): (1) a sequence kernel, defined by an RBF kernel applied to mean-pooled ESM2 embeddings, and (2) a structure kernel, which is itself the product of three terms—one based on AlphaFold2-predicted distances between mutation sites, another based on the Hellinger distance between inverse folding probabilities computed by ProteinMPNN [Dauparas et al., 2022] and a third one based on the likelihood of the sequence under an inverse folding model. Kermut outperforms ProteinNPT while being more computationally efficient, and currently achieves state-of-the-art results on the supervised ProteinGym benchmark [Groth et al., 2024].

We emphasize that each one of these methods (one-hot, hybrid, ProteinNPT and Kermut) considers a single representation of the protein through some combination of the choices outlined in Table 2.

Uncertainty quantification for fitness prediction Existing approaches for UQ in fitness prediction can be broadly categorized into three groups. The first is probabilistic modeling, where assumptions are made about the conditional distribution of fitness given the sequence. Uncertainty estimates are derived from this distribution, whose parameters are estimated from data. These methods rely heavily on correct model specification (e.g., the choice of kernel in Gaussian Processes). A second category involves randomization based uncertainty, for example, ensembling neural networks with different initializations, with the uncertainty defined as the variance of the prediction across the ensemble. This approach has seen empirical success in some tasks [Gruver et al., 2021, Greenman et al., 2025], but may be suboptimal when the underlying neural network does not provide good predictions. Finally, conformal prediction provides uncertainty estimates in the form of prediction intervals with guaranteed marginal coverage. However, these intervals are not locally adaptive unless combined with an estimate of predictive variance—such as in studentized conformal prediction [Lei et al., 2018, Agarwal et al., 2025].

It is important to note that all of these approaches quantify uncertainty only at the **modeling** stage of the DSLC. They do not account for uncertainty introduced in **earlier** stages—such as problem formulation, data cleaning, or data processing—which is the primary focus of this work.

3 Stable fitness predictors via the PCS framework

Setup We study the supervised fitness prediction task described in Section 2, where the goal is to learn a predictor $\hat{f}^{(p)}$ that maps a mutated protein sequence $m^{(p)}$ to a scalar fitness value y, such as thermostability or binding affinity to a given ligand. We assume access to a dataset of n labeled examples $(m_1^{(p)}, y_1), \ldots, (m_n^{(p)}, y_n)$, where y_i denotes the experimentally measured fitness of $m_i^{(p)}$.

We consider base models that output both point predictions and uncertainty estimates. In this work, we study Bayesian Ridge regression, Kermut (a GP) and an ensemble of CNNs, 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 .5.

We define the representation of the protein as the following triplet: (1) an embedding from a pre-trained PLM, (2) a dimensionality reduction method, and (3) a zero-shot score. To keep the study tractable, we focus on a representative subset. For embeddings, we use ESM1v [Meier et al., 2021], ESM2 [Lin et al., 2023], and CARP [Yang et al., 2024b]. For dimensionality reduction, we apply standard mean pooling [Dallago et al., 2021] and three randomized variants. Each variant perturbs the mean-pooling weights: for a sequence of length l, we project the (l by d) embedding matrix using inner product with $\mathbf{m} = (1/l + \epsilon_1, \ldots, 1/l + \epsilon_l)$, where $\epsilon_i \sim \mathcal{N}(0, 1/l^2)$ independently, resulting in a h-dimensional vector. For zero-shot scores, we use MSA Transformer [Rao et al., 2021], ProSST [Li et al., 2024c], ESM2 [Lin et al., 2023] and TranceptEVE [Notin et al., 2022b]. We provide a detailed description of the zero-shot and embedding methods in Appendices .2.1 and .2.2 respectively.

Stabilizing and Pred-Checking fitness predictors via the PCS framework We describe our training procedure for exploring the space of reasonable protein representations while also quantifying the uncertainty associated with these choices. Our approach consists of two main steps, mirroring the first two steps of the procedure described in [Agarwal et al., 2025]. First, we apply a **Pred-Check step** to select a subset of informative zero-shot scores, which vary substantially in predictive power across datasets. To identify the most useful scores, we evaluate their fit to the training data. Of the methods we tested (Appendix .3), the most effective selected the top three zero-shot features, ranked by their Mean Decrease in Impurity (MDI) scores from a random forest trained on the fitness labels. We then divided each of the three MDI scores by their total sum to obtain non-negative weights that sum to one. These weights were assigned to all models associated with each of the selected scores, and used to compute a weighted average of their predictions (Appendix .4). We emphasize that the Pred-Check step is partial, as it focuses solely on the zero-shot score. Developing analogous checks for the embedding and dimensionality reduction steps is an important direction for future work. In the second step (**the ensembling step**), we consider the Cartesian product of the selected zero-shot scores (3 in total), the embeddings (3 in total), and the dimensionality reduction methods (4 in total) as our space of reasonable representations ($3 \times 3 \times 4 = 36$ representations in total). On each representation within this set, we train a single base model on the training dataset, resulting in an ensemble of 36 models. Final predictions are computed as a weighted (by the normalized MDI values) average of the ensemble outputs. Uncertainty is estimated by combining two sources: (1) the average uncertainty reported by the base models, and (2) the empirical standard deviation of the predictions across different representations.

This procedure builds on the PCS-UQ framework introduced by Agarwal et al. [2025], but shifts the focus from algorithmic variability to sensitivity with respect to data representation choices. Therefore, our approach includes two key modifications. First, the Pred-Check step filters representations (i.e., zero-shot scores) rather than algorithms. Second, instead of using bootstrap resampling, we perturb reasonable data representations. These changes reflect the central goal of this work: to highlight the often-overlooked impact of representation-level sensitivity on model performance.

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 .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 .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 SP 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 SP 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 SP version following the procedure described in Section 3. These SP versions are ensembles of the base models fitted on the same training data with different representations of the protein sequences. Each SP version consists of 36 models.

Results The results, presented in Figure 2 (a) and in Appendix .7, show that SP predictors consistently outperform their base estimators across all models and data splits. Among them, SP Kermut achieves the best performance, followed by SP 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, SP CNN increases Spearman correlation by 22%, SP Bayesian Ridge by 30%, and SP Kermut by 9% relative to their respective base models. Remarkably, SP 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, SP 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 .4 presents a detailed ablation of the proposed method. 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 SP variants.

Results SP variants of all base methods—Kermut, Bayesian Ridge, and CNN—consistently outperform their non-SP 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, SP Kermut and SP Bayesian Ridge recover the top sequence in more assays than their base versions. By step five, SP Kermut shows a 6% absolute improvement over base Kermut.

Appendix .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, SP 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.

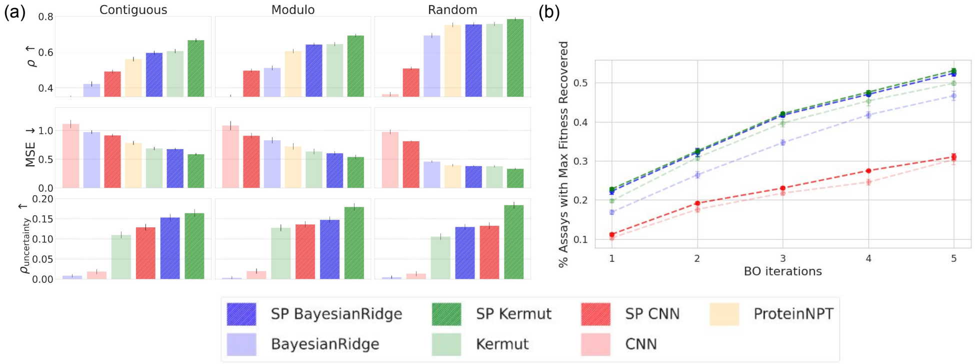

Figure 2: (a) ProteinGym benchmark. We report the average Spearman correlation (ρ), mean squared error (MSE), and the Spearman correlation between the uncertainty scores and the absolute errors of the predictions ($\rho_{\text{uncertainty}}$). Each column corresponds to a different ProteinGym split (b)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

5 Discussion

This work highlights the often-overlooked impact of instability from representation choices in protein fitness prediction, and shows that leveraging this variability—guided by the PCS framework—can yield substantial performance gains, particularly on out-of-distribution (OOD) data such as covariate shift. While we examined a subset of reasonable representations, such as zero-shot scores and embeddings from pre-trained models, many parts of the representation space remain unexplored—for example, the choice of network layer. Stabilizing and Pred-Checking predictions with respect to these choices presents a promising direction for future improvement. Although SP predictors yield improved performance, they come with increased computational cost: in our experiments, we used an ensemble of 36 models, which is relatively modest in size. Incorporating additional Pred-Check steps to filter embeddings and dimensionality reduction methods could reduce this cost.

References

Abhineet Agarwal, Michael Xiao, Rebecca Barter, Omer Ronen, Boyu Fan, and Bin Yu. Pcs-uq: Uncertainty quantification via the predictability-computability-stability framework. *arXiv preprint arXiv:2505.08784*, 2025.

Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1): 235–242, 2000.

Nicholas Bhattacharya, Neil Thomas, Roshan Rao, Justas Dauparas, Peter K Koo, David Baker, Yun S Song, and Sergey Ovchinnikov. Single layers of attention suffice to predict protein contacts. *Biorxiv*, pages 2020–12, 2020.

Ron S Boger, Seyone Chithrananda, Anastasios N Angelopoulos, Peter H Yoon, Michael I Jordan, and Jennifer A Doudna. Functional protein mining with conformal guarantees. *Nature Communications*, 16(1):85, 2025.

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

J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, and D. Baker. 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`.

Clara Fannjiang and Ji Won Park. Reliable algorithm selection for machine learning-guided design. *arXiv preprint arXiv:2503.20767*, 2025.

Clara Fannjiang, Stephen Bates, Anastasios N Angelopoulos, Jennifer Listgarten, and Michael I Jordan. Conformal prediction under feedback covariate shift for biomolecular design. *Proceedings of the National Academy of Sciences*, 119(43):e2204569119, 2022.

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

Peter Mørch Groth, Mads 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.

Nate Gruver, Samuel Stanton, Polina Kirichenko, Marc Finzi, Phillip Maffettone, Vivek Myers, Emily Delaney, Peyton Greenside, and Andrew Gordon Wilson. Effective surrogate models for protein design with bayesian optimization. In *ICML Workshop on Computational Biology*, volume 198, 2021.

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

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

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.

Jannik Kossen, Neil Band, Clare Lyle, Aidan N Gomez, Thomas Rainforth, and Yarin Gal. Self-attention between datapoints: Going beyond individual input-output pairs in deep learning. *Advances in Neural Information Processing Systems*, 34:28742–28756, 2021.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523): 1094–1111, 2018.

Francesca-Zhoufan Li, Ava P Amini, Yisong Yue, Kevin K Yang, and Alex X Lu. Feature reuse and scaling: Understanding transfer learning with protein language models. *bioRxiv*, pages 2024–02, 2024a.

Francesca-Zhoufan Li, Jason Yang, Kadina E Johnston, Emre Gürsoy, Yisong Yue, and Frances H Arnold. Evaluation of machine learning-assisted directed evolution across diverse combinatorial landscapes. *bioRxiv*, pages 2024–10, 2024b.

Mingchen Li, Yang Tan, Xinzhu Ma, Bozitao Zhong, Huiqun Yu, Ziyi Zhou, Wanli Ouyang, Bingxin Zhou, Liang Hong, and Pan Tan. Prosst: Protein language modeling with quantized structure and disentangled attention. biorxiv. 2024c.

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

Benjamin J Livesey and Joseph A Marsh. Updated benchmarking of variant effect predictors using deep mutational scanning. *Molecular systems biology*, 19(8):e11474, 2023.

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

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

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

Pascal Notin, Lood Van Niekerk, Aaron W Kollasch, Daniel Ritter, Yarin Gal, and Debora S Marks. Trancepteve: Combining family-specific and family-agnostic models of protein sequences for improved fitness prediction. *bioRxiv*, pages 2022–12, 2022b.

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.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

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

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

Ian Sillitoe, Nicola Bordin, Natalie Dawson, Vaishali P Waman, Paul Ashford, Harry M Scholes, Camilla SM Pang, Laurel Woodridge, Clemens Rauer, Neeladri Sen, et al. Cath: increased structural coverage of functional space. *Nucleic acids research*, 49(D1):D266–D273, 2021.

Baris E Suzek, Hongzhan Huang, Peter McGarvey, Raja Mazumder, and Cathy H Wu. Uniref: comprehensive and non-redundant uniprot reference clusters. *Bioinformatics*, 23(10):1282–1288, 2007.

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

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

Mihaly Varadi, Damian Bertoni, Paulyna Magana, Urmila Paramval, Ivanna Pidruchna, Malarvizhi Radhakrishnan, Maxim Tsenkov, Sreenath Nair, Milot Mirdita, Jingi Yeo, 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.

Jason Yang, Francesca-Zhoufan Li, and Frances H Arnold. Opportunities and challenges for machine learning-assisted enzyme engineering. *ACS Central Science*, 10(2):226–241, 2024a.

Jason Yang, Ravi G Lal, James C Bowden, Raul Astudillo, Mikhail A Hameedi, Sukhvinder Kaur, Matthew Hill, Yisong Yue, and Frances H Arnold. Active learning-assisted directed evolution. *Nature Communications*, 16(1):714, 2025.

Kevin K Yang, Zachary Wu, Claire N Bedbrook, and Frances H Arnold. Learned protein embeddings for machine learning. *Bioinformatics*, 34(15):2642–2648, 2018.

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

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

Bin Yu and Karl Kumbier. 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.

.1 Reasonable choices for embedding a protein sequence

.2 List of embeddings and zero shot score

.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.

Table 2: A list of reasonable choices for the representation of a single protein. A single protein sequence can be 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., 2024b], 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]

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.

.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. [2024b]) 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.

.3 Pred-check procedure

Following the PCS framework, we apply a **prediction check (Pred-Check)** step to filter and weight 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 fitness labels and $\{z_i^{(j)}\}_{i=1}^n$ for $j = 1, \dots, h$ 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 $k \leq h$ 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.

.4 Ablation studies

We perform an ablation study to quantify the contribution of each component in our SP 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., 2024b]), 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 .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.

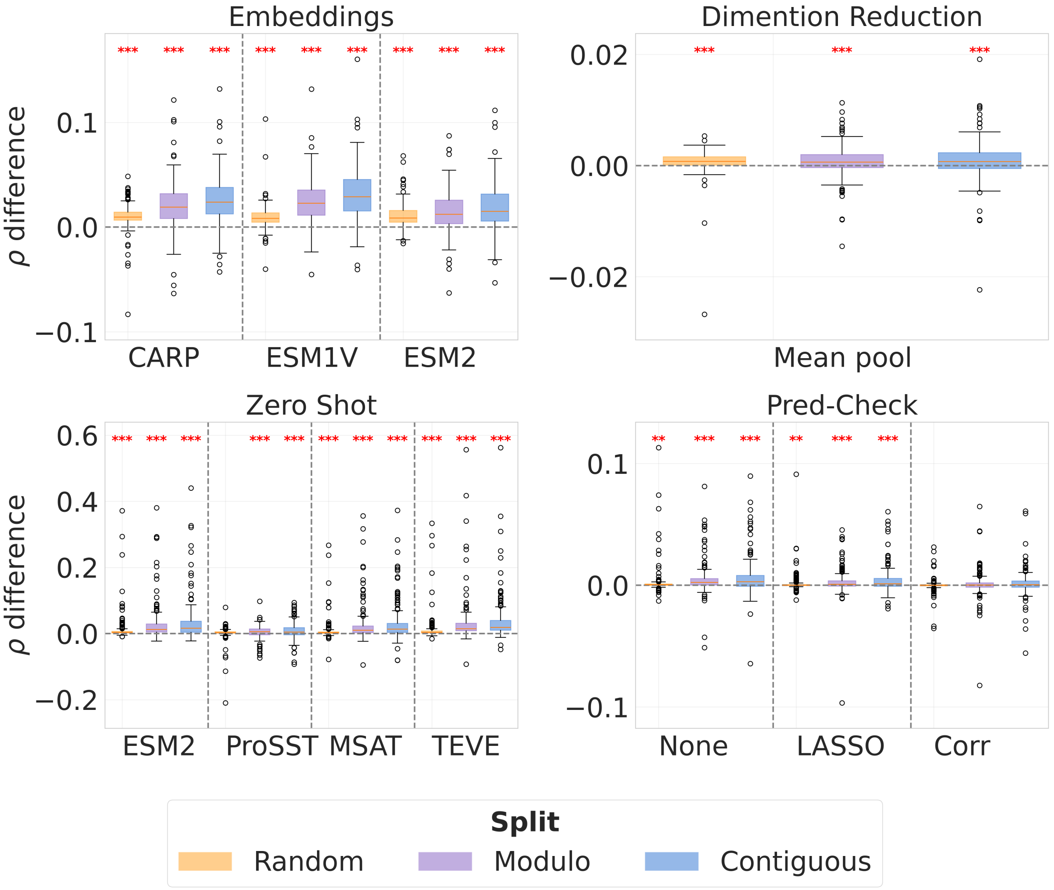

Figure 3: Each component of SP Kermut improves prediction accuracy. Box plots show the change in test set Spearman correlation when ablations are applied to individual components. Asterisks indicate significance levels from one-sided two-sample t-tests (*<0.05, **<0.01, ***<0.001), and colors denote different data splits. The top left panel compares SP 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).

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 SP Kermut, SP Bayesian Ridge, and SP 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 SP Kermut and SP Bayesian Ridge. For SP CNN—the weakest overall model—embedding ensembling performs similarly to using ESM2 alone, but not worse.

.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

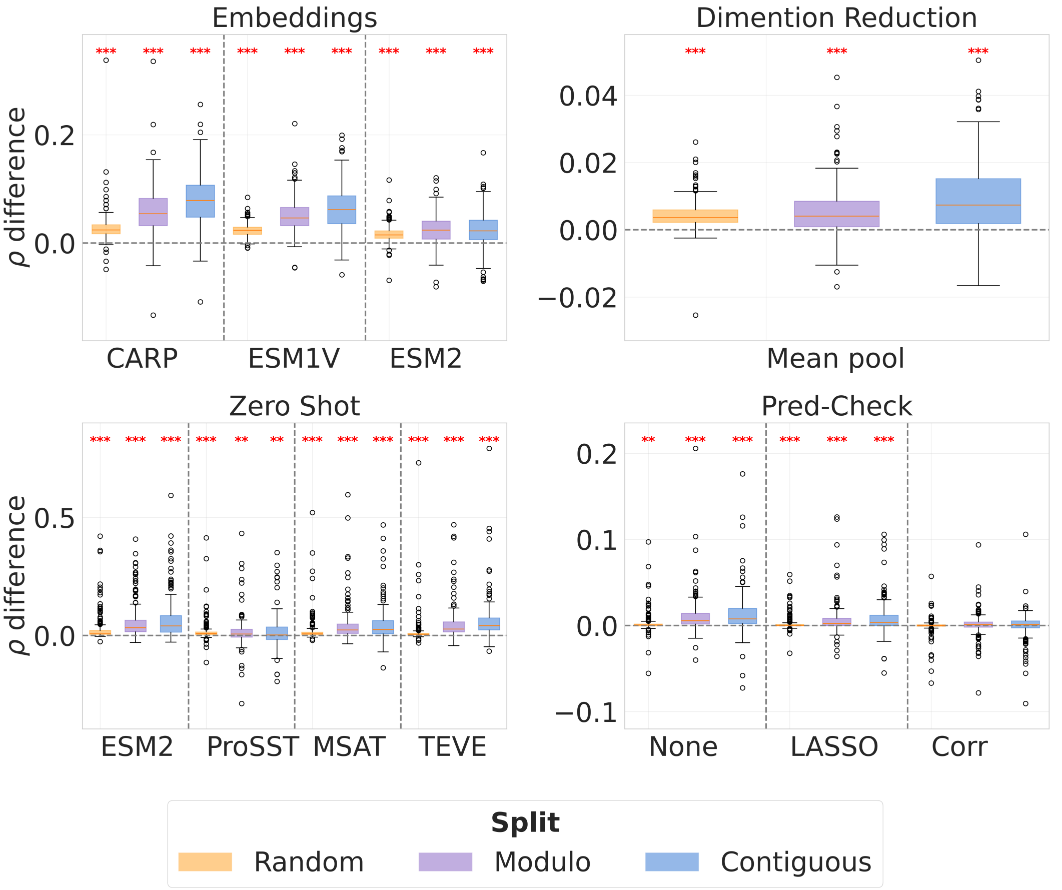

Figure 4: Each component of SP Bayesian Ridge improves prediction accuracy. Box plots show the change in test set Spearman correlation when ablations are applied to individual components. Asterisks indicate significance levels from one sided two-sample t-tests (*<0.05, **<0.01, ***<0.001), and colors denote different data splits. The top left panel compares SP 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).

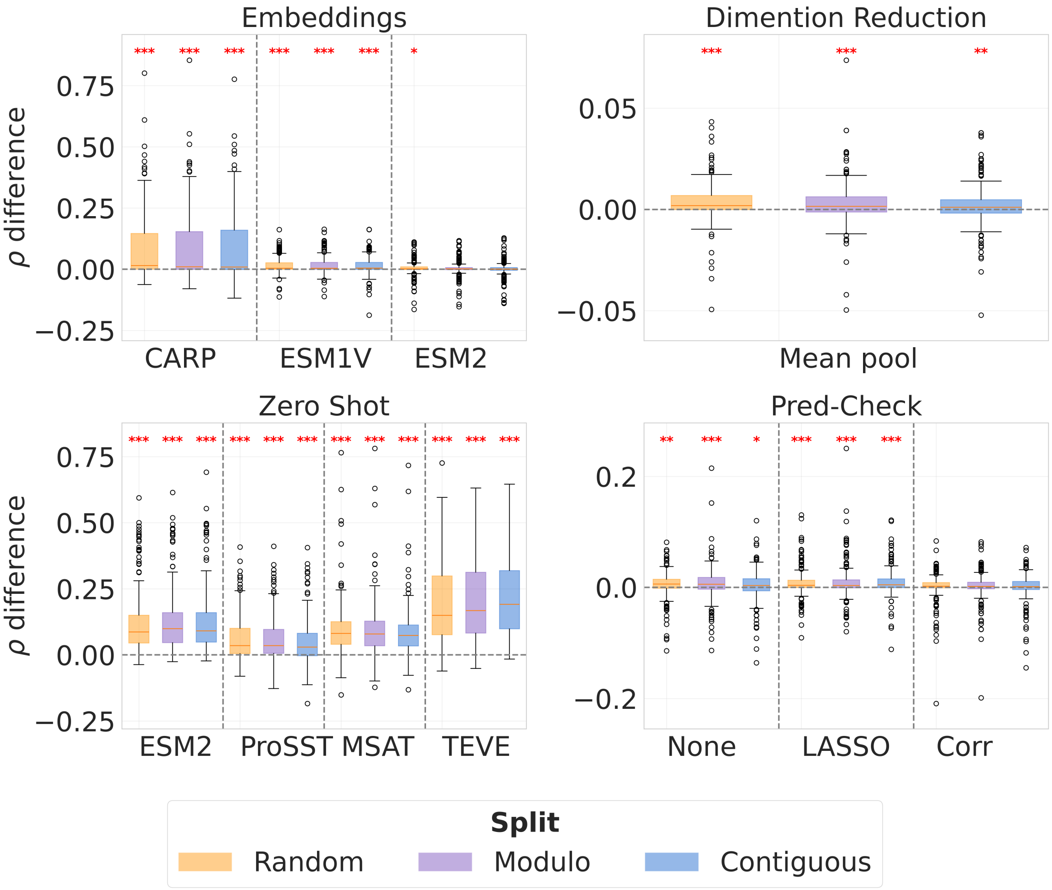

Figure 5: Each component of SP CNN improves or does not hurt prediction accuracy. Box plots show the change in test set Spearman correlation when ablations are applied to individual components. Asterisks indicate significance levels from one-sided two-sample t-tests (*<0.05, **<0.01, ***<0.001), and colors denote different data splits. The top left panel compares SP 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).

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.

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 $\tau = 1/\sigma^2$:

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

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:

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

where $K_{\textit{seq}}(x, x')$ represents a sequence kernel defined using RBF Kernel applied to mean pooled embeddings of a PLM. and $K_{\textit{struc}}(x, x')$ 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 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 ±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 assays in which at least one highest-fitness sequence has been recovered for $\lambda = 0.1$.

SP 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, SP Kermut and SP 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.

.7 Additional ProteinGym benchmark results

.7.1 Results on single mutants

We report the numerical results for the random split in Table 5, modulo split Table 4 and contiguous split in Table 3

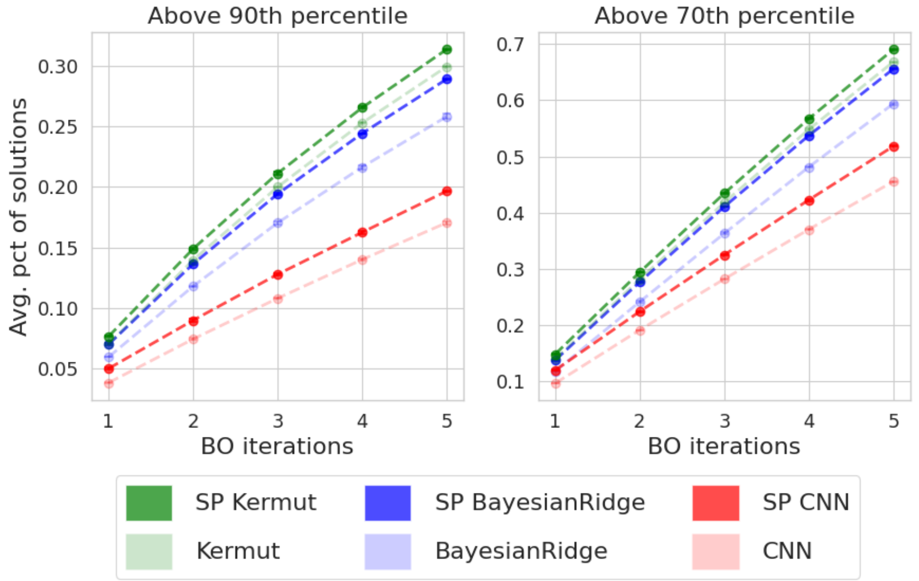

Figure 6: SP predictors recover high-fitness sequences more often than their base models when using BO with $\lambda = 0.1$. The y-axis shows the cumulative percentage of solutions that are above the 90th (left) and 70th (right) percentiles of the assay's fitness distribution. The x-axis shows the BO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

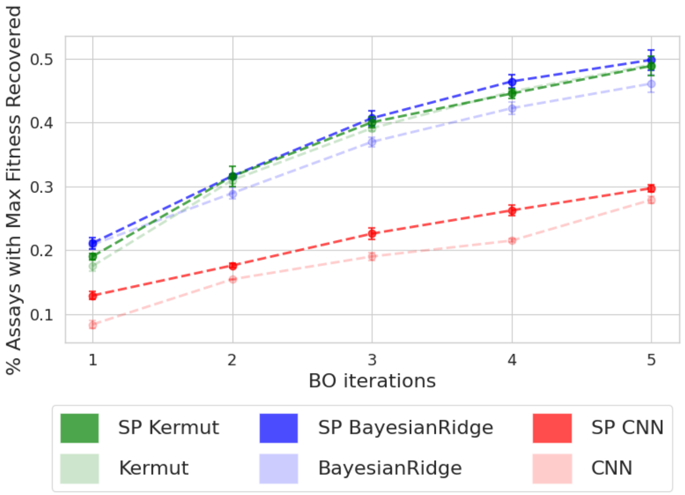

Figure 7: SP predictors recover the highest-fitness sequence more often—or at least as often—as their base models when using BO with $\lambda = 0.1$. The y-axis shows the percentage of experiments in which the top-fitness sequence is recovered; the x-axis shows the number of BO steps. Results are averaged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

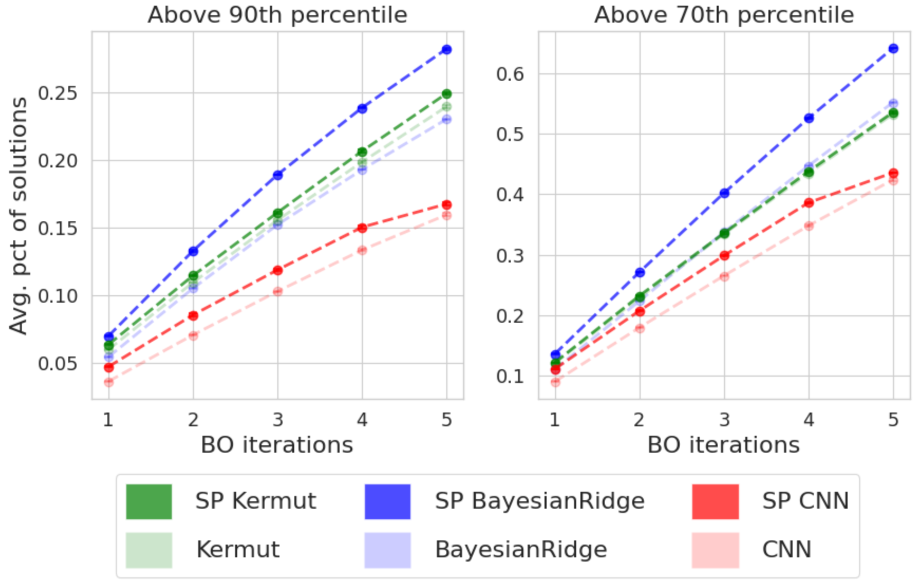

Figure 8: SP predictors recover high-fitness sequences more often—or at least as often—as their base models when using BO with $\lambda = 2$. The y-axis shows the cumulative percentage of solutions that are above the 90th (left) and 70th (right) percentiles of the assay's fitness distribution. The x-axis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

Table 3: 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	-

.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 storing 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 SP Kermut is zero).

Results The results are presented in Figure 9 and Tables 7 and 6 for one vs. two and random, respectively. On the challenging one vs. two split, SP 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

Table 4: 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
SP 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
SP 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
SP Kermut	0.667 ± 0.010	0.587 ± 0.020	0.164 ± 0.010
ProteinNPT	0.561 ± 0.013	0.784 ± 0.034	-

Table 5: 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
SP 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
SP 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
SP Kermut	0.785 ± 0.010	0.334 ± 0.016	0.184 ± 0.008
ProteinNPT	0.753 ± 0.013	0.397 ± 0.022	-

SP Kermut. For the random split SP Kermut achieves a spearman value of 0.95 surpassing Kermut with Spearman of 0.94. SP Bayesian Ridge is able to match Kermut's performance with Spearman of 0.94.

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

Table 6: 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
SP 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
SP 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
SP Kermut	0.945 ± 0.004	0.110 ± 0.008	0.264 ± 0.014

.8 Computational Resources

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

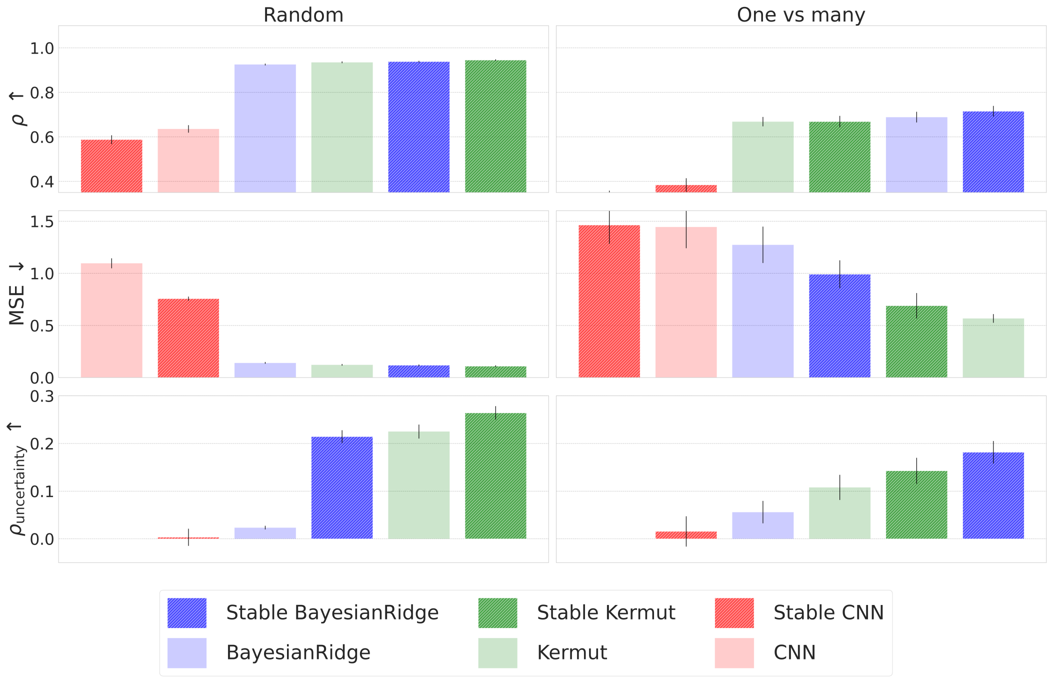

Figure 9: ProteinGym benchmark results on 51 multiples mutation datasets studied in Groth et al. [2024]. We report the average Spearman correlation (ρ), mean squared error (MSE), and the Spearman correlation between the uncertainty scores and the absolute errors of the predictions ($\rho_{\text{uncertainty}}$).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

Table 7: 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
SP 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
SP 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
SP Kermut	0.669 ± 0.026	0.689 ± 0.121	0.143 ± 0.027