Likelihood-based fine-tuning of protein language models for few-shot fitness prediction and design

Alex Hawkins-Hooker*
AI Centre
University College London

Jakub Kmec
InstaDeep Ltd.

Oliver Bent
InstaDeep Ltd.

Paul Duckworth
InstaDeep Ltd.

Abstract

In order to correctly predict amino acid identities within natural proteins, protein language models (PLMs) must implicitly learn distributional constraints on protein sequences upheld over the course of evolution. As a consequence, the sequence and mutation-level likelihoods of such models form effective zero-shot predictors of mutations. Although various schemes have been proposed for exploiting the distributional knowledge captured by PLMs to enhance supervised fitness prediction and design, lack of head-to-head comparison across different prediction strategies and different classes of PLM has made it challenging to identify the best-performing methods, and to understand the factors contributing to performance. Here, we extend previously proposed ranking-based loss functions to adapt the likelihoods of family-based and masked protein language models, and demonstrate that the best configurations outperform state-of-the-art approaches based on frozen embeddings in the low-data setting. Furthermore, we propose ensembling strategies that exploit the strong dependence of the mutational distributions learned by PLMs on sequence context, showing that they can be used to guide efficient optimisation strategies over fitness landscapes.

1 Introduction

Natural protein sequences are the result of evolution via natural selection. Protein language models (PLMs) fit to the distribution of natural sequences therefore learn to implicitly model functional and structural constraints relevant to protein fitness, with their likelihoods forming effective zero-shot predictors of the fitness effects of mutations [26, 28]. These distribution learning capabilities are also highly relevant for protein design. Mutations assigned high likelihood by PLMs can be iteratively incorporated into a sequence to improve fitness-related properties [14], or entire sequences can be sampled directly from generative PLMs, enabling the diversification of natural protein families [25].

In practical multi-round design scenarios, it is often possible to use experimental techniques to generate labelled datasets associating sets of sequences with quantitative measurements of biological properties of interest. Although in some cases the relevant properties are amenable to high-throughput experimental screening methodologies, and the resulting datasets can be large [38, 43], in general, experimental constraints mean that it might only be feasible to generate measurements for tens or hundreds of proteins in each round [2]. It is therefore of considerable interest to ask how the zero-shot

*Work completed during an internship at InstaDeep Ltd.

Preprint. Under review.

prediction capacities of PLMs can be combined with small labelled datasets to achieve improved predictive performance in fitness prediction and design settings.

One popular paradigm for exploiting the information in pretrained PLMs involves extracting sequence representations and feeding these as inputs into task-specific downstream predictive models [1, 2, 34, 5, 19, 30]. However, recent trends in natural language processing have shown the benefits of directly adapting the distributions of models using task-specific labelled or preference data [31, 32], thereby fully exploiting the distributional knowledge contained in the original pretrained model. While related fine-tuning strategies have previously been considered in the context of fitness prediction with PLMs [22], they remain under-explored in comparison to approaches based on frozen or fine-tuned PLM embeddings. In particular, while previous work has demonstrated application of ranking losses to adapt the likelihoods of unconditional autoregressive PLMs [22, 23], it has not addressed whether similar strategies can effectively be applied across different classes of PLM, which often outperform unconditional autoregressive models for fitness prediction [29], nor how to exploit fine-tuning to improve performance in uncertainty-guided design tasks. Moreover, there has been relatively limited direct comparison of these fine-tuning strategies to alternative PLM-based fitness prediction strategies, including recent innovations in architectures for operating over frozen PLM embeddings [30], making it difficult to assess their utility in practice.

Seeking to address this gap, in this paper we (i) show that ranking losses can be extended to adapt the likelihoods of leading zero-shot fitness predictors trained with both masked and family-based autoregressive language modelling objectives, (ii) provide direct comparison with state-of-the-art approaches based on frozen protein language model embeddings [30], as well as fine-tuning with added regression heads, thereby offering compelling empirical evidence for the effectiveness of the proposed fine-tuning schemes, and (iii) develop ensembling strategies compatible with these fine-tuning schemes, demonstrating their effectiveness in both supervised and multi-round design settings.

2 Related work

Zero-shot protein fitness prediction: The most successful models for zero-shot prediction of protein fitness effects attempt to predict the likelihood of particular sets of mutations occurring within a natural protein given its evolutionary context. Traditional methods within this category involve statistical models trained directly on multiple sequence alignments (MSAs) for each protein of interest, such as profile models [15], Potts models [6, 15] and VAEs [7, 36]. More recent generalisations of such methods involve pretraining large PLMs across all natural proteins. For example, ESM-1v [26] is trained using a masked language modelling objective, allowing point mutations to be scored by the ratio of probabilities of mutant and wild-type amino acids. Alternatively, autoregressive models can directly compute the likelihood of entire protein sequences, making them more appropriate for scoring sequences containing multiple mutations [28, 27, 25]. However, unconditional PLMs suffer from a lack of context, often requiring fine-tuning to specialise their distributions towards a particular protein family of interest [25]. As a result, the leading autoregressive models exploit the information in MSAs to improve predictions, either by biasing language model likelihoods with statistics from the MSA, in the case of Tranception [28], or by explicitly conditioning on the MSA [12, 33, 42].

Supervised protein fitness prediction: Fitness prediction has also been studied as a supervised learning task in many prior works [34, 16, 22]. Several works have sought to exploit the pretrained representations of PLMs to improve performance, using either fine-tuned [34] or frozen embeddings [5, 30]. Nonetheless, approaches based on embeddings risk discarding useful distributional information captured in the models' output layers [22]. The importance of fully leveraging distribution modelling for fitness prediction is highlighted by the success of 'augmented density' predictors [16], which combine zero-shot fitness predictions with either one-hot encoded [16], or embedded [30] representations of input sequences. The state-of-the-art supervised fitness prediction method Protein-NPT [30] combines these strategies, training a custom non-parametric Transformer [21] to reason over both zero-shot predictions and associated sequence embeddings to produce fitness predictions.

Methods seeking to adapt the distributions learned by PLMs directly have been less well studied. Rives et al. [37] proposed to use the log-likelihood ratio between mutant and wild-type amino acids as a regression function, fine-tuning the full model. Krause et al. [22] suggest using a ranking-based loss function to fine-tune autoregressive PLMs, showing improvements over augmented density

baselines on a small set of fitness landscapes. A similar ranking-based loss function was proposed for training non-pretrained CNN architectures on fitness datasets in Brookes et al. [4]. Most recently, Lee et al. [23] apply ranking-based loss functions derived from the literature on large language model alignment [32] to fine-tune unconditional autoregressive PLMs. The application of ranking-based loss functions to masked PLMs is also considered in concurrent work [44].

Model-guided protein design: Several works have proposed variants of Bayesian optimization (BO) for designing biological sequences, including proteins [11, 18, 19, 40, 13]. The majority of these BO approaches are evaluated in an unconstrained setting, in which sequences are proposed by the optimiser and evaluated with a black-box oracle designed to mimic a biological property of interest. An alternative *in silico* evaluation strategy avoids the challenge of defining a meaningful oracle function by adopting a pool-based optimisation problem formulation over experimentally determined fitness landscapes [30]. Another line of works has sought to provide direct experimental validation of approaches combining uncertainty estimates with PLMs, in settings ranging from zero [14] or few-shot design [2] to single-round design given large training sets of sequence-fitness pairs [24]. In this paper, we focus on evaluating different PLM-based fitness prediction strategies in *in silico* settings designed to mimic applied design scenarios. We study both a supervised setting and a model-guided design setting, which extends the pool-based optimisation setting proposed in Notin et al. [30], to a set of the most diverse multi-mutant fitness landscapes in ProteinGym.

3 Background

3.1 Ranking-based loss functions

While mean squared error has been widely used as a loss function in training sequence-based predictive models of fitness landscapes, two recent works have advocated the use of ranking-based loss functions [22, 4]. In particular, they suggest parameterising a Bradley-Terry model [3] with a learned function of the sequence. The Bradley-Terry model represents the probability that a given sequence x_i has higher fitness y than another sequence x_j by parameterising a binary classifier via the difference in scores of each sequence under a learned scoring function $s_\theta(x)$:

$$p(y(x_i) > y(x_j)) = \sigma(s_\theta(x_i) - s_\theta(x_j)) \,, \tag{1}$$

where σ is the logistic sigmoid function. The model can be fit to data by maximising the likelihood of the complete set of pairwise comparisons between the fitness values of sequences with respect to the parameters θ of the scoring function. Concretely, in our implementation, given a batch of B sequences $x_1, ..., x_B$, the resulting loss is:

$$\mathcal{L} = \sum_{i=1}^{B} \sum_{j=1}^{B} -\mathbb{I}(y(x_i) > y(x_j)) \log\sigma(s_\theta(x_i) - s_\theta(x_j)) \,, \tag{2}$$

where $\mathbb{I}$ is an indicator function. In this way, fitness prediction for a dataset of size N is converted from a regression problem with N labels into a binary classification problem with $N \times N$ labels.

3.2 Ranking-based fine-tuning of autoregressive protein language models

To use the Bradley-Terry model to fine-tune an autoregressive protein language model, Krause et al. [22] propose using an unconditional sequence log-likelihood as the scoring function:

$$s_\theta(x) = \sum_{i=1}^{L} \log p(x_i | x_{<i}) \,. \tag{3}$$

Since the log-likelihoods of autoregressive protein language models are strong zero-shot predictors of the fitness effects of mutations [28], the difference in log-likelihoods used to parameterise the Bradley-Terry model of Equation 1 can already produce an effective pairwise classifier at initialisation, which may help maximise the effectiveness of fine-tuning on small datasets.

4 Likelihood-based fine-tuning of masked and family-based language models

We describe below how ranking-based losses can be extended to fine-tune the likelihoods of other widely used classes of protein language models, including masked language models and autoregressive

family-based models. These extensions are derived from the principle that fine-tuning strategies should exploit as far as possible the properties of models that lead to strong zero-shot performance [22]. We enforce this principle by suggesting an appropriate choice of the scoring function s_θ used to parameterise the Bradley-Terry model in each case.

4.1 Conditional scoring functions

Previous applications of the Bradley-Terry model to fine-tune PLMs have focussed on unconditional autoregressive models. However, these models often underperform other classes of model including conditional autoregressive models and masked language models in fitness prediction settings [29]. We therefore propose to extend existing applications of the Bradley-Terry model to accommodate these more performant PLMs. To do so, we incorporate the additional conditioning information c exploited by these models into conditional scoring functions $s_\theta(x, c)$:

$$p\big((y(x_i) > y(x_j))|c\big) = \sigma(s_\theta(x_i, c) - s_\theta(x_j, c)). \tag{4}$$

Below, we will consider cases where c represents either a wild-type sequence or a multiple sequence alignment (MSA), since conditioning on evolutionary context is especially effective in fitness prediction [42], but note that the same approach could be applied to models which condition on protein structure [17].

4.2 Scoring functions for masked protein language models

Masked language models do not define a sequence-level likelihood, meaning that it is not immediately obvious how to define a scoring function for the Bradley-Terry model. Meier et al. [26] proposed a variety of strategies for zero-shot scoring of mutants using the likelihoods assigned to sets of mutations by masked language models. We propose to use these zero-shot scoring-functions to parameterise the Bradley-Terry model in Equation 4, allowing the models to be fine-tuned with ranking-based losses, similar to other concurrent work [44]. Concretely, we utilize the 'wild-type marginals' scoring function from Meier et al. [26]. Under this strategy the score for a mutated sequence is given by the summation of the log-likelihood ratios between mutated and wild-type amino acids across mutated positions, given the unmasked wild-type sequence as input:

$$s_\theta(x, x^{\text{wt}}) = \sum_{i:x_i^{\text{wt}} \neq x_i} \log p(x_i|x^{\text{wt}}) - \log p(x_i^{\text{wt}}|x^{\text{wt}}) \,. \tag{5}$$

We choose the wild-type marginals strategy for its combination of computational efficiency and strong performance as a zero-shot scoring function. Since all sequences are scored under the residue distributions obtained by feeding the wild-type sequence through the model, a set of mutated sequences of arbitrary size can be scored using a single forward pass, making both fine-tuning and prediction extremely efficient. We apply this fine-tuning strategy to ESM-1v, but expect it to be equally applicable to other models for which similar scoring rules are used.

4.3 Scoring functions for family-based protein language models

Family-based protein language models represent the conditional distribution over family members given a subset of other family members [35, 12, 33, 42]. These models have proved especially effective as zero-shot fitness predictors, due to their ability to explicitly condition on evolutionary context to predict the effects of mutations.

In this paper we work with PoET [42], which models entire protein families autoregressively. To produce zero-shot predictions given a mutant sequence x and an MSA $M = \{m^{(1)}, ..., m^{(N)}\}$ of homologues of a wild-type sequence x^{wt}, PoET computes the likelihood of the mutant x given the MSA. To exploit this capacity to condition on family members during fine-tuning, we condition the autoregressive scoring function in Equation 3 on the sequences in the MSA:

$$s_\theta(x, M) = \sum_{i=1}^{L} \log p(x_i|x_{<i}, M) \,. \tag{6}$$

Since PoET operates natively on unaligned sequences and is sensitive to alignment depth, we subsample a small set of sequences from the MSA and discard gaps before feeding them into the model, following [42] (details in Appendix A.2).

To increase the efficiency of fine-tuning PoET, in practice we cache a single set of hidden layer representations obtained by passing the subsampled MSA M through the model, and fine-tune only the mapping between these frozen representations and the sequence likelihoods (Appendix A.3), decoupling the encoding of prior context from the decoding of future amino acids given this context.

4.4 Uncertainty quantification with evolutionary context ensembles

The amino acid output distributions learned by protein language models depend heavily on sequence context, which language models must use to infer the (structural or functional) constraints determining native amino acid identities. We propose to exploit this property to build ensembles of fine-tuned PLMs, in which each ensemble member sees a different, but functionally equivalent, sequence context. Concretely, to fine-tune an ensemble of PoET models, for each fitness dataset we sub-sample a set of K input MSAs M_k from the full MSA associated with the wild-type sequence. We then fine-tune a separate set of parameters to minimise the ranking loss conditioned on each MSA, producing K sets of parameters, each specialised to a single input MSA (further details are provided in Appendix A.2). To score sequences, we use an ensembled scoring function:

$$s_{\theta_1,\ldots,\theta_K}(x, M) = \frac{1}{K}\sum_{k=1}^{K} s_{\theta_k}(x, M_k). \tag{7}$$

This procedure extends the practice of MSA ensembling used to improve the zero-shot predictions of MSA-based PLMs [26, 42], to the supervised setting. To achieve a similar effect with ESM-1v, which has no MSA representation, we instead sample a set of K input masks, and fine-tune a separate set of parameters for each input mask, exploiting the intuition that differently masked sequences are functionally equivalent, but may nonetheless produce different outputs when passed through the model (Appendix A.4).

4.5 Relationship to preference learning strategies for LLMs

Direct preference optimisation (DPO) [32] is a recently proposed method for aligning large language models (LLMs) using datasets of human preference data. DPO also uses scoring functions from pretrained models to parameterise a Bradley-Terry model. Instead of parameterising a classifier directly via differences in log likelihoods, DPO uses the difference in scaled log likelihood *ratios* between the fine-tuned model and a frozen reference model. Thus the probabillity that a completion x_1 is preferred to a completion x_2 given an input prompt c is modelled as:

$$p_\theta(x_1 \succ x_2|c) = \sigma\left(\beta\log\frac{p_\theta(x_1|c)}{p_{\text{ref}}(x_1|c)} - \beta\log\frac{p_\theta(x_2|c)}{p_{\text{ref}}(x_2|c)}\right). \tag{8}$$

In our notation, the DPO preference model therefore amounts to a particular choice of scoring function $s_\theta(x, c) = \beta\log\frac{p(x|c)}{p_{\text{ref}}(x|c)}$. Assuming an autoregressive decomposition of $p(x|c)$, this scoring function is equivalent to the conditional autoregressive scoring function in Equation 6 if the reference model is chosen to be constant and $\beta = 1$.

The non-constant reference model in DPO imposes a KL penalty on the deviation between the fine-tuned p_θ and the reference model, which helps prevent collapse in the fine-tuned distribution [32]. Although some recent work has reported success in adapting DPO to the protein fitness prediction setting [23], in our own experiments we did not find this regularisation necessary to achieve good performance, possibly owing to the fact that we require neither generations from the model nor generalisation to different 'conditions' at test time, unlike typical applications of DPO.

5 Few-shot fitness prediction

5.1 Fitness landscapes

We study the performance of fitness prediction strategies on mutational landscapes from ProteinGym [29]. Each landscape contains a set of protein sequences together with experimentally determined fitness values. Typically the protein sequences within a landscape contain a small number

of mutations relative to a 'wild-type' protein, and the fitness values are quantitative measurements of a functional property associated with the wild-type. We utilise two subsets of ProteinGym: the first subset is the validation set of 8 representative single-mutant landscapes selected by Notin et al. [30]. The second is a set of 5 landscapes containing multiple substitutions, chosen to constitute a non-redundant set of the most diverse landscapes available in ProteinGym (Appendix A.1).

In contrast to some prior work (e.g. Notin et al. [30]), we focus explicitly on the low-data setting. For each landscape, we train all methods on $n = 128$ or $n = 512$ sequences randomly sampled from the landscape and evaluate on either 2000 (for single-mutant landscapes) or 5000 (for multiple-mutant landscapes) randomly sampled held-out sequences. An additional set of 128 randomly sampled sequences is used as a validation set to perform early stopping. For each landscape, and each n, we generate three sets of random splits, and report test set Spearman correlation averaged across the three splits. For models trained with ranking losses, the Spearman correlation is computed between the scoring function $s_\theta(x, c)$ and the ground truth fitness values.

Prior work has also considered non-random splits [5, 30], including splits that test generalisation to mutations at unseen positions. We note that such generalisation is automatically required in order to achieve good performance in the low-data setting, as not all positions will be represented in the small training sets. We therefore present our main results on randomly generated splits, similar to [16, 22]. However, we also assess generalisation by computing metrics on subsets of the test sets containing mutations at positions for which no mutations were present in the training set sequences.

5.2 Fitness prediction strategies

We evaluate the performance of the likelihood-based fine-tuning strategies introduced in Section 4 on the selected landscapes. To attain an understanding of the effectiveness of these strategies across different classes of PLM, we apply them to the masked language model ESM-1v [26], the unconditional autoregressive model ProGen2 [27], and the family-based autoregressive model PoET [42]. In each case, the model is fine-tuned by parameterising the Bradley-Terry model of Equation 1 via the corresponding scoring functions in Sections 3 and 4, and minimising the loss in Equation 2.

We compare to two sets of baselines, representative of widely used approaches that either (i) fine-tune PLMs by adding a regression head [34], or (ii) train new models on top of frozen language model embeddings [30]. In the first case, we add linear regression heads to both ESM-1v and PoET, and fine-tune all parameters (Appendix A.2.1). As the leading example of the second class of approaches, we compare against ProteinNPT [30], a state-of-the-art model operating on top of frozen language model embeddings. As additional baselines, we include the 'augmented density' strategies used as baselines by [30]. These models are regression models taking as input the zero-shot predictions of a PLM as well as either a one-hot representation of the mutated sequence [16], or an embedding extracted from a PLM [30] (further details in Appendix A.6). We refer to these distinct choices of augmented density representation as 'OHE augmented' (OHE aug.) and 'Embedding augmented' (Emb. aug.) respectively, following Notin et al. [30]. Hyperparameters are selected based on performance on the single mutant set, consistent with the practice used for ProteinNPT and associated baselines. We report metrics obtained when using these hyperparameters on both single-mutant and multiple mutant landscapes for each method.

5.3 Results

Ranking-based fine-tuning outperforms regression-based fine-tuning We first focus on the comparison between ranking-based fine-tuning and regression-based fine-tuning, *using the same models*. For PoET, ranking-based fine-tuning performs best across both dataset sizes for single and multi-mutant landscapes (Table 1). Regression-based fine-tuning is nonetheless a strong baseline, performing slightly better than the best ProteinNPT configuration. For ESM-1v, ranking-based fine-tuning performs much better than regression-based fine-tuning on the single mutant landscapes, but worse on the $n = 512$ multi-mutant landscapes. Unlike regression-based fine-tuning via a linear head, the wild-type marginals scoring rule used in ranking-based fine-tuning of ESM-1v is unable to capture the interactions between multiple mutations, since it assumes that mutation effects are additive. In contrast, the scoring functions used for autoregressive models are able to capture interactions between mutations and therefore perform well on the multiple mutants datasets: indeed

Table 1: Spearman correlation on 8 single mutant landscapes and 5 multiple mutant landscapes from ProteinGym. Results for $n = 0$ are computed on the $n = 128$ test splits. Where methods use a frozen base model to produce embeddings and zero-shot predictions, the base model type is provided in parentheses, and zero-shot performance is that of the base model.

		Singles			**Multiples**		
Model name (base model)	**Loss type**	$n = 0$	$n = 128$	$n = 512$	$n = 0$	$n = 128$	$n = 512$
ESM-1v	ranking	0.384	0.552	0.637	0.425	0.653	0.736
ESM-1v + linear head	regression	-	0.425	0.583	-	0.649	0.780
PoET	ranking	0.417	**0.589**	**0.668**	0.592	**0.738**	**0.806**
PoET + linear head	regression	-	0.554	0.649	-	0.711	0.784
ProGen2 small	ranking	0.385	0.521	0.623	0.358	0.670	0.768
ProGen2 medium	ranking	0.418	0.497	0.629	0.405	0.666	0.765
ProteinNPT (MSAT)	regression	0.399	0.545	0.635	0.534	0.689	0.782
ProteinNPT (ESM-1v)	regression	**0.437**	0.497	0.602	0.392	0.646	0.775
Emb. aug. (MSAT)	regression	0.399	0.541	0.627	0.534	0.707	0.783
Emb. aug. (ESM1v)	regression	**0.437**	0.532	0.609	0.392	0.638	0.765
OHE aug. (MSAT)	regression	0.399	0.465	0.500	0.534	0.643	0.745
OHE aug. (ESM1v)	regression	**0.437**	0.496	0.531	0.392	0.556	0.709
OHE	regression	-	0.303	0.494	-	0.467	0.671

the unconditional autoregressive ProGen models outperform the ranking-based version of ESM-1v on the multiples datasets, but not on the single mutant datasets, while the family-based autoregressive model PoET achieves strong performance across the board. We note that these observations do not necessarily point to a fundamental limitation of fine-tuning masked PLMs with ranking-based losses; more expressive masked PLM scoring functions involving multiple forward passes may be able to achieve better performance at the price of increased computational expense.

Ranking-based fine-tuning outperforms models trained on frozen embeddings We next focus on the comparison between the best-performing ranking-based fine-tuning schemes and baselines based on frozen embeddings. Ranking-based fine-tuning of PoET outperforms ProteinNPT across all settings (Table 3), with the gap largest in the $n = 128$ regime, suggesting that directly adapting the likelihoods of the pretrained model is especially helpful for maximising performance given very limited data. Notably this is not simply by virtue of PoET producing better zero-shot predictions: on the single mutant datasets, the zero-shot ESM-1v predictions used by ProteinNPT (ESM-1v) outperform those produced by PoET. ESM-1v fine-tuned with a ranking loss is also slightly better than ProteinNPT on the single-mutant datasets, but performs worse on the multi-mutant datasets, likely due to the limited expressivity of the scoring rule, as discussed above. For both ESM-1v and PoET, the proposed ensembling strategies further improve performance (Table 2), sometimes substantially, and show improved uncertainty calibration, as measured by the negative log likelihood of pairwise classifications from the test set. The fine-tuning strategies are also much more efficient than ProteinNPT, which requires expensive cross-sequence attention operations (Appendix A.9).

Ranking-based fine-tuning generalises to unseen positions Random splits provide an estimate of performance on heldout data. However, similar mutations can occur in both train and test sets (e.g. related amino acid substitutions at the same position), meaning that measuring performance on predicting the effects of these mutations does not necessarily test a model's capacity for generalisation [30]. This issue is somewhat mitigated in our setup by the choice to use relatively small training sets, and the emphasis on multi-mutant datasets, where the degree of overlap between train and test sets is typically lower. We assess the capacity of models to generalise to mutations at unseen positions in the single mutant datasets by reporting the performance of all models for mutations in the $n = 128$ test sets occurring at positions at which no mutations were present in the training set sequences (Table 3). While there is a clear drop in performance at these unseen positions, PoET fine-tuned with a ranking loss still performs the best, indicating that it is able to generalise across positions better than other methods.

Table 2: Spearman correlation and calibration of pairwise predictions for single models versus ensembles on the 5 multi-mutant datasets.

		Spearman		NLL	
Model name	**Loss type**	$n = 128$	$n = 512$	$n = 128$	$n = 512$
ESM1v	ranking	0.653	0.736	1.42	0.768
ESM1v ensemble	ranking	0.677	0.753	0.841	0.584
PoET	ranking	0.738	0.806	0.987	0.620
PoET ensemble	ranking	**0.752**	**0.818**	**0.750**	**0.507**

Table 3: Single-mutant Spearman correlations for test set mutations at seen and unseen positions (n=128). Test set mutants are assigned to the unseen set if they contain mutations in sequence positions at which none of the training set sequences have mutations.

		Spearman	
Model name	**Loss type**	Seen	Unseen
ESM1v	ranking	0.587	0.474
ESM1v + linear head	regression	0.484	0.303
PoET	ranking	**0.617**	**0.531**
PoET + linear head	regression	0.573	0.515
ProteinNPT (MSAT)	regression	0.570	0.486

6 Multi-round design on fitness landscapes

6.1 Experiment details

We next ask whether the improvements in predictive performance translate to benefits in a multi-round design setting. To do so, we follow the evaluation protocol introduced by Notin et al. [30], with minor modifications. Design is formulated as a pool-based optimisation task over the sequences in an empirical fitness landscape. For a given landscape, the goal is to retrieve as many high-scoring sequences as possible over the course of 10 optimisation rounds. In each round, the model's predictions are used to guide the selection of a batch of 100 sequences to acquire from a pool of candidate sequences. The pool of candidate sequences is either the complete landscape, or, in the case of the multiple mutant landscapes, a randomly selected subset of 5000 sequences. Before the first round, models are seeded in round 0 with 100 sequences randomly sampled from the landscape.

We follow Notin et al. [30] in using ensembling strategies to derive uncertainty estimates which can be used to guide the selection of candidates from the pool within the framework of Bayesian optimisation (BO). To make the comparison between the modelling strategies as direct as possible, we choose to follow Notin et al. [30]'s use of the upper confidence bound (UCB) acquisition function, although we note that our use of a ranking loss means that our ensembles should properly be considered as preferential surrogates within the framework of preferential BO [10], and alternative preferential acquisition strategies might better exploit the model's pairwise classification uncertainty. At each round, we rank all remaining sequences in the pool by their acquisition values, and select the top 100 to add to the training set. For ProteinNPT and embedding-augmented baselines, we use Monte Carlo dropout [8] to produce uncertainty estimates.

6.2 Results

We compare optimisation guided by ensembles of PoET and ESM-1v ranking models to ProteinNPT, as well as selected baselines. Results for both single and multi-mutant landscapes are shown in Figure 1. Across both sets of landscapes, the PoET ranking ensemble outperforms all other methods. In general, the design curves show similar trends to the supervised results. Ranking-based fine-tuning outperforms regression-based fine-tuning, and using ensembles leads to the best performance, though a single model also performs very well (Figure 4). While recall of high-fitness sequences saturates for the single mutant landscapes, it improves steadily for the multiple mutant landscapes, since the starting pools are larger, and it is not possible to reach perfect recall within the fixed budget of

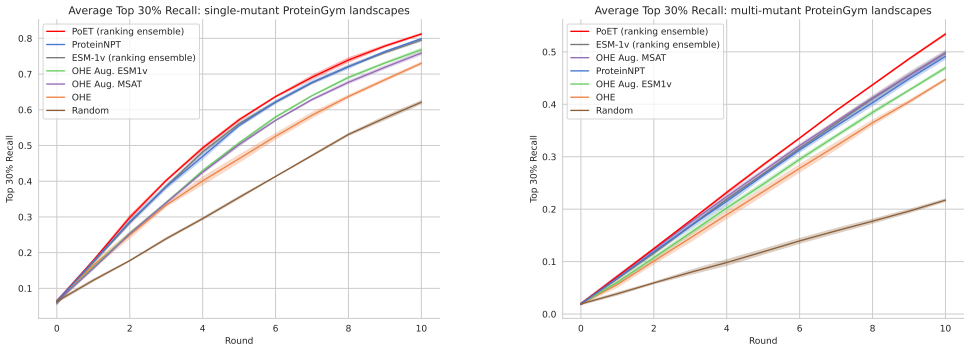

Figure 1: Top 30% recall averaged over: (left): 8 single-mutant landscapes and (right): 5 multi-mutant landscapes. The shading represents one standard deviation over 3 random seeds.

acquisitions. Curves for individual landscapes are displayed in Figures 5 and 6. The relative ordering of the methods is reasonably stable across individual landscapes, although there are some cases where simple baselines perform comparably to the best-performing methods, suggesting these landscapes may contain noisy or otherwise difficult-to-predict fitness labels [30].

7 Conclusion

The ability of language models to learn distributional constraints governing natural protein sequences makes them powerful zero-shot predictors of the effects of mutations on protein function. Here we show that the learned distributions can also be rapidly adapted via feedback from relatively few experimental measurements. Even 128 sequences - of the order of a typical batch size in biological experiments - are sufficient to significantly improve on the zero-shot performance of these models, and incremental feedback from batches of this size leads to strong performance in simulated multi-round design settings. While previous works have also suggested the effectiveness of directly fine-tuning likelihoods, we extend this strategy to the classes of PLM whose distributions best reflect fitness, and find that this is crucial to obtaining performance surpassing leading approaches based on frozen embeddings [30]. An intriguing possibility is that when autoregressive models are fine-tuned using labelled data in this way, they may retain their generative capacity. We believe studying this possibility by leveraging the connection to methods used for fine-tuning language models from human preference data [32] to be a promising avenue for future work.

References

[1] Ethan C. Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M. Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature Methods*, 16(12):1315–1322, December 2019. ISSN 1548-7105. doi: 10.1038/s41592-019-0598-1. URL `https://www.nature.com/articles/s41592-019-0598-1`. Number: 12 Publisher: Nature Publishing Group.

[2] Surojit Biswas, Grigory Khimulya, Ethan C. Alley, Kevin M. Esvelt, and George M. Church. Low-N protein engineering with data-efficient deep learning. *Nature Methods*, 18(4):389–396, April 2021. ISSN 1548-7105. doi: 10.1038/s41592-021-01100-y. URL `https://www.nature.com/articles/s41592-021-01100-y`. Number: 4 Publisher: Nature Publishing Group.

[3] Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: the mehod of paired comparisons. *Biometrika*, 39(3-4):324–345, December 1952. ISSN 0006-3444. doi: 10.1093/biomet/39.3-4.324. URL `https://doi.org/10.1093/biomet/39.3-4.324`.

[4] David H. Brookes, Jakub Otwinowski, and Sam Sinai. Contrastive losses as generalized models of global epistasis, May 2023. URL `http://arxiv.org/abs/2305.03136`. arXiv:2305.03136 [cs, q-bio].

[5] Christian Dallago, Jody Mou, Kadina E. Johnston, Bruce Wittmann, Nick Bhattacharya, Samuel Goldman, Ali Madani, and Kevin K. Yang. FLIP: Benchmark tasks in fitness landscape inference for proteins. August 2021. URL https://openreview.net/forum?id=p2dMLEwL8tF.

[6] Matteo Figliuzzi, Hervé Jacquier, Alexander Schug, Oliver Tenaillon, and Martin Weigt. Coevolutionary Landscape Inference and the Context-Dependence of Mutations in Beta-Lactamase TEM-1. *Molecular Biology and Evolution*, 33(1):268–280, January 2016. ISSN 0737-4038. doi: 10.1093/molbev/msv211. URL https://doi.org/10.1093/molbev/msv211.

[7] Jonathan Frazer, Pascal Notin, Mafalda Dias, Aidan Gomez, Joseph K. Min, Kelly Brock, Yarin Gal, and Debora S. Marks. Disease variant prediction with deep generative models of evolutionary data. *Nature*, 599(7883):91–95, November 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-04043-8.

[8] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/gal16.html.

[9] Louisa Gonzalez Somermeyer, Aubin Fleiss, Alexander S Mishin, Nina G Bozhanova, Anna A Igolkina, Jens Meiler, Maria-Elisenda Alaball Pujol, Ekaterina V Putintseva, Karen S Sarkisyan, and Fyodor A Kondrashov. Heterogeneity of the GFP fitness landscape and data-driven protein design. *eLife*, 11:e75842, May 2022. ISSN 2050-084X. doi: 10.7554/eLife.75842. URL https://doi.org/10.7554/eLife.75842. Publisher: eLife Sciences Publications, Ltd.

[10] Javier González, Zhenwen Dai, Andreas Damianou, and Neil D. Lawrence. Preferential Bayesian Optimization. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1282–1291. PMLR, July 2017. URL https://proceedings.mlr.press/v70/gonzalez17a.html. ISSN: 2640-3498.

[11] 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. 2021.

[12] Alex Hawkins-Hooker, David T Jones, and Brooks Paige. MSA-Conditioned Generative Protein Language Models for Fitness Landscape Modelling and Design. In *Machine Learning in Structural Biology Workshop at NeurIPS*, 2021.

[13] Brian L. Hie and Kevin K. Yang. Adaptive machine learning for protein engineering. *Current Opinion in Structural Biology*, 72:145–152, February 2022. ISSN 0959-440X. doi: 10.1016/j.sbi.2021.11.002. URL https://www.sciencedirect.com/science/article/pii/S0959440X21001457.

[14] Brian L. Hie, Varun R. Shanker, Duo Xu, Theodora U. J. Bruun, Payton A. Weidenbacher, Shaogeng Tang, Wesley Wu, John E. Pak, and Peter S. Kim. Efficient evolution of human antibodies from general protein language models. *Nature Biotechnology*, pages 1–9, April 2023. ISSN 1546-1696. doi: 10.1038/s41587-023-01763-2. URL https://www.nature.com/articles/s41587-023-01763-2. Publisher: Nature Publishing Group.

[15] Thomas A. Hopf, John B. Ingraham, Frank J. Poelwijk, Charlotta P. I. Schärfe, Michael Springer, Chris Sander, and Debora S. Marks. Mutation effects predicted from sequence co-variation. *Nature Biotechnology*, 35(2):128–135, February 2017. ISSN 1546-1696. doi: 10.1038/nbt.3769.

[16] 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, July 2022. ISSN 1546-1696. doi: 10.1038/s41587-021-01146-5.

[17] 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 *Proceedings of the 39th International Conference on Machine Learning*, pages 8946–8970. PMLR, June 2022. URL https://proceedings.mlr.press/v162/hsu22a.html. ISSN: 2640-3498.

[18] Moksh Jain, Emmanuel Bengio, Alex Hernandez-Garcia, Jarrid Rector-Brooks, Bonaventure F. P. Dossou, Chanakya Ajit Ekbote, Jie Fu, Tianyu Zhang, Michael Kilgour, Dinghuai Zhang, Lena Simine, Payel Das, and Yoshua Bengio. Biological Sequence Design with GFlowNets. In *Proceedings of the 39th International Conference on Machine Learning*, pages 9786–9801. PMLR, June 2022. URL https://proceedings.mlr.press/v162/jain22a.html. ISSN: 2640-3498.

[19] Asif Khan, Alexander I. Cowen-Rivers, Antoine Grosnit, Derrick-Goh-Xin Deik, Philippe A. Robert, Victor Greiff, Eva Smorodina, Puneet Rawat, Rahmad Akbar, Kamil Dreczkowski, Rasul Tutunov, Dany Bou-Ammar, Jun Wang, Amos Storkey, and Haitham Bou-Ammar. Toward real-world automated antibody design with combinatorial Bayesian optimization. *Cell Reports Methods*, 3(1):100374, January 2023. ISSN 2667-2375. doi: 10.1016/j.crmeth.2022.100374.

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

[21] 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. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 28742–28756. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/f1507aba9fc82ffa7cc7373c58f8a613-Paper.pdf.

[22] Ben Krause, Nikhil Naik, Wenhao Liu, and Ali Madani. Don't throw away that linear head: Few-shot protein fitness prediction with generative models. October 2021. URL https://openreview.net/forum?id=hHmtmT58pSL.

[23] Minji Lee, Kyungmin Lee, and Jinwoo Shin. Fine-tuning protein language models by ranking protein fitness. In *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*, 2023. URL https://openreview.net/forum?id=DUjUJCqqA7.

[24] Lin Li, Esther Gupta, John Spaeth, Leslie Shing, Rafael Jaimes, Emily Engelhart, Randolph Lopez, Rajmonda S. Caceres, Tristan Bepler, and Matthew E. Walsh. Machine learning optimization of candidate antibody yields highly diverse sub-nanomolar affinity antibody libraries. *Nature Communications*, 14(1):3454, June 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-39022-2. URL https://www.nature.com/articles/s41467-023-39022-2. Number: 1 Publisher: Nature Publishing Group.

[25] Ali Madani, Ben Krause, Eric R. Greene, Subu Subramanian, Benjamin P. Mohr, James M. Holton, Jose Luis Olmos, Caiming Xiong, Zachary Z. Sun, Richard Socher, James S. Fraser, and Nikhil Naik. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, 41(8):1099–1106, August 2023. ISSN 1546-1696. doi: 10.1038/s41587-022-01618-2. URL https://www.nature.com/articles/s41587-022-01618-2. Number: 8 Publisher: Nature Publishing Group.

[26] 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. In *Advances in Neural Information Processing Systems*, volume 34, pages 29287–29303. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f51338d736f95dd42427296047067694-Abstract.html.

[27] Erik Nijkamp, Jeffrey A. Ruffolo, Eli N. Weinstein, Nikhil Naik, and Ali Madani. ProGen2: Exploring the boundaries of protein language models. *Cell Systems*, 14(11):968–978.e3, November 2023. ISSN 2405-4712, 2405-4720. doi: 10.1016/j.cels.2023.10.002. URL https://www.cell.com/cell-systems/abstract/S2405-4712(23)00272-7. Publisher: Elsevier.

[28] 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 *Proceedings of the 39th International Conference*

on Machine Learning, pages 16990–17017. PMLR, June 2022. URL https://proceedings.mlr.press/v162/notin22a.html. ISSN: 2640-3498.

[29] Pascal Notin, Aaron W. Kollasch, Daniel Ritter, Lood Van Niekerk, Steffan Paul, Han Spinner, Nathan J. Rollins, Ada Shaw, Ruben Weitzman, Jonathan Frazer, Mafalda Dias, Dinko Franceschi, Rose Orenbuch, Yarin Gal, and Debora Susan Marks. ProteinGym: Large-Scale Benchmarks for Protein Fitness Prediction and Design. November 2023. URL https://openreview.net/forum?id=URoZHqAohf&noteId=LmiOZsZZAh.

[30] Pascal Notin, Ruben Weitzman, Debora Susan Marks, and Yarin Gal. ProteinNPT: Improving Protein Property Prediction and Design with Non-Parametric Transformers. November 2023. URL https://openreview.net/forum?id=AwzbQVuDBk&referrer=%5Bthe%20profile%20of%20Yarin%20Gal%5D(%2Fprofile%3Fid%3D~Yarin_Gal1).

[31] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. May 2022. URL https://openreview.net/forum?id=TG8KACxEON.

[32] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. October 2023. URL https://openreview.net/forum?id=53HUHMvQLQ&referrer=%5Bthe%20profile%20of%20Chelsea%20Finn%5D(%2Fprofile%3Fid%3D~Chelsea_Finn1).

[33] Soumya Ram and Tristan Bepler. Few Shot Protein Generation, April 2022. URL http://arxiv.org/abs/2204.01168. arXiv:2204.01168 [cs, q-bio].

[34] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating Protein Transfer Learning with TAPE. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://papers.nips.cc/paper_files/paper/2019/hash/37f65c068b7723cd7809ee2d31d7861c-Abstract.html.

[35] Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8844–8856. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/rao21a.html.

[36] Adam J. Riesselman, John B. Ingraham, and Debora S. Marks. Deep generative models of genetic variation capture the effects of mutations. *Nature Methods*, 15(10):816–822, October 2018. ISSN 1548-7105. doi: 10.1038/s41592-018-0138-4. URL https://www.nature.com/articles/s41592-018-0138-4. Number: 10 Publisher: Nature Publishing Group.

[37] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, April 2021. doi: 10.1073/pnas.2016239118. URL https://www.pnas.org/doi/10.1073/pnas.2016239118. Publisher: Proceedings of the National Academy of Sciences.

[38] Gabriel J. Rocklin, Tamuka M. Chidyausiku, Inna Goreshnik, Alex Ford, Scott Houliston, Alexander Lemak, Lauren Carter, Rashmi Ravichandran, Vikram K. Mulligan, Aaron Chevalier, Cheryl H. Arrowsmith, and David Baker. Global analysis of protein folding using massively parallel design, synthesis and testing. *Science (New York, N.Y.)*, 357(6347):168–175, July 2017. ISSN 0036-8075. doi: 10.1126/science.aan0693. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5568797/.

[39] Karen S. Sarkisyan, Dmitry A. Bolotin, Margarita V. Meer, Dinara R. Usmanova, Alexander S. Mishin, George V. Sharonov, Dmitry N. Ivankov, Nina G. Bozhanova, Mikhail S. Baranov, Onuralp Soylemez, Natalya S. Bogatyreva, Peter K. Vlasov, Evgeny S. Egorov, Maria D. Logacheva,

Alexey S. Kondrashov, Dmitry M. Chudakov, Ekaterina V. Putintseva, Ilgar Z. Mamedov, Dan S. Tawfik, Konstantin A. Lukyanov, and Fyodor A. Kondrashov. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603):397–401, May 2016. ISSN 1476-4687. doi: 10.1038/nature17995. URL https://www.nature.com/articles/nature17995. Number: 7603 Publisher: Nature Publishing Group.

[40] Samuel Stanton, Wesley Maddox, Nate Gruver, Phillip Maffettone, Emily Delaney, Peyton Greenside, and Andrew Gordon Wilson. Accelerating Bayesian Optimization for Biological Sequence Design with Denoising Autoencoders. In *Proceedings of the 39th International Conference on Machine Learning*, pages 20459–20478. PMLR, June 2022. URL https://proceedings.mlr.press/v162/stanton22a.html. ISSN: 2640-3498.

[41] Martin Steinegger, Markus Meier, Milot Mirdita, Harald Vöhringer, Stephan J. Haunsberger, and Johannes Söding. HH-suite3 for fast remote homology detection and deep protein annotation. *BMC Bioinformatics*, 20(1):473, September 2019. ISSN 1471-2105. doi: 10.1186/s12859-019-3019-7. URL https://doi.org/10.1186/s12859-019-3019-7.

[42] Timothy Fei Truong Jr and Tristan Bepler. PoET: A generative model of protein families as sequences-of-sequences. November 2023. URL https://openreview.net/forum?id=1CJ8D7P8RZ.

[43] Kotaro Tsuboyama, Justas Dauparas, Jonathan Chen, Elodie Laine, Yasser Mohseni Behbahani, Jonathan J. Weinstein, Niall M. Mangan, Sergey Ovchinnikov, and Gabriel J. Rocklin. Megascale experimental analysis of protein folding stability in biology and design. *Nature*, 620 (7973):434–444, August 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06328-6. URL https://www.nature.com/articles/s41586-023-06328-6. Number: 7973 Publisher: Nature Publishing Group.

[44] Junming Zhao, Chao Zhang, and Yunan Luo. Contrastive Fitness Learning: Reprogramming Protein Language Models for Low-N Learning of Protein Fitness Landscape, February 2024. URL https://www.biorxiv.org/content/10.1101/2024.02.11.579859v1. Pages: 2024.02.11.579859 Section: New Results.

A Appendix

A.1 Fitness landscapes

We use the set of 8 single-mutant landscapes selected for ablations and hyperparameter selection by [30]. The names of these landscapes in ProteinGym are:

- BLAT_ECOLX_Jacquier_2013
- CALM1_HUMAN_Weile_2017
- DYR_ECOLI_Thompson_2019
- DLG4_RAT_McLaughlin_2012
- REV_HV1H2_Fernandes_2016
- TAT_HV1BR_Fernandes_2016
- RL40A_YEAST_Roscoe_2013
- P53_HUMAN_Giacomelli_WT_Nutlin

We additionally select a set of 5 of the most diverse multi-mutant landscapes in ProteinGym. To select these landscapes, we identified the landscapes with the largest number of mutations in ProteinGym, and discarded redundant landscapes (for example the GFP landscapes of [9] are landscapes of close homologues of the GFP protein whose landscape was reported by [39]. We therefore include only the latter.

The selected multi-mutant landscapes are:

- PABP_YEAST_Melamed_2013
- CAPSD_AAV2S_Sinai_2021
- GFP_AEQVI_Sarkisyan_2016
- GRB2_HUMAN_Faure_2021
- HIS7_YEAST_Pokusaeva_2019

A.2 Hyperparameter details

Hyperparameters for the fine-tuning methods are selected based on performance on the single mutant set, consistent with the practice used to select hyperparameters for the baselines from ProteinNPT. We report metrics obtained when using these hyperparameters on both single-mutant and multiple-mutant landscapes for each method.

ESM-1v, ProGen2 and PoET models were fine-tuned using the Adam optimizer [20] using gradient accumulation with an effective batch size of 32. We use the first of the five ESM-1v checkpoints. Learning rates for regression-based and ranking-based fine-tuning were selected separately in each case after after a sweep over the values $1e-4, 3e-5, 1e-5$ on the 8 single mutant landscapes. For ESM-1v, we computed the loss by scoring all sequences using the logits generated by passing the wild-type sequence through the model in a single forward pass. In the fitness prediction experiments, the models were trained for 50 epochs. During training on each landscape the Spearman correlation on a separate validation set of 128 sequences from the landscape was used to determine the epoch whose checkpoint should be used to produce predictions on the test set.

A.2.1 Regression heads

Linear regression heads were added to embeddings extracted from PoET and ESM-1v. In the former case, we used final token embeddings, and in the latter case we averaged embeddings across the sequence dimension before feeding them to the regression head.

A.2.2 Ensembles

Ensembles of size 5 were used for both ESM-1v and PoET. During design, the ensemble members were trained for a fixed number of epochs (15 for PoET; 20 for ESM-1v) each round. All ensemble members were reinitialised from the pretrained model each round.

A.3 Decoder-only fine-tuning of PoET

PoET parameterises a sequence of conditional distributions over the amino acids in a set of protein sequences in the same family. The model represents the joint likelihood of a set of sequences $M = \{m^{(1)}, ..., m^{(N)}\}$, via an autoregressive factorisation over sequences and over positions within each sequence:

$$p(M) = \prod_i p(m^{(i)}|m^{(<i)}) = \prod_{ij} p(m_j^{(i)}|m_{<j}^{(i)}, m^{(<i)}) \, . \tag{9}$$

To parameterise this distribution, PoET uses a causally masked Transformer architecture, which maps from previous amino acids to logits for the current amino acid. Conceptually, this function can be decomposed into two stages: first the entire history of previous sequences $m_{<i}$ is encoded into a sequence of embeddings $H_{<i} \in \mathbb{R}^{L_{<i} \times D \times E}$, where D is the number of layers and E is the embedding dimension, via a stack of causally masked layers:

$$H_{<i} = f_\theta(m^{(<i)}) \, . \tag{10}$$

The current sequence m_i is then decoded by a function which maps these prior sequence embeddings and previous amino acids in the current sequence to logits for each position j:

$$\text{logit}_{ij} = g_\theta(m_{<j}^{(i)}, H_{<i}) \, . \tag{11}$$

To fine-tune PoET from fitness data, we propose to fine-tune only the weights of the function g, representing the 'decoding' of the current sequence given its context. To achieve this, we first clone the PoET weights, producing a set of 'encoder' weights ϕ and a set of 'decoder' weights θ. We use the frozen encoder weights to produce an embedding $H \in \mathbb{R}^{L_M \times D \times E}$ of the input MSA sequences: $H = f_\phi(\{m^{(1)}, ..., m^{(N)}\})$, where L_M is the total length of all sequences in the input MSA. We then fine-tune the weights θ of the cloned 'decoder' to minimise the cross-entropy loss of Equation 2 on the labelled data. Concretely, the scoring function used to parameterise the Bradley-Terry model becomes:

$$s_\theta(x, M) \equiv s_\theta(x, H) = \sum_i \text{log} p_\theta(x_i|x_{<i}, H) \tag{12}$$

To maximise computational efficiency, the MSA embeddings H are pre-computed before the start of the fine-tuning process, and remain frozen throughout.

A.4 ESM-1v ensembling strategy

To fine-tune an ensemble of models given a single ESM-1v checkpoint, we randomly sampled a set of 5 masks. Within each mask, each sequence position had a 15% probability of being masked. We fine-tuned one model for each mask, by using the correspondingly masked wild-type sequence $\tilde{x}_k^{\text{wt}}$ as input to the model, instead of the unmasked wild-type sequence. The ensembled scoring function used to generate predictions was:

$$s(x, x^{\text{wt}}) = \frac{1}{K} \sum_k s(x, \tilde{x}_k^{\text{wt}}) \tag{13}$$

A.5 PoET MSA subsampling

For PoET, in both single-model and ensemble configurations, we sampled context sequences from the same filtered MSAs used to extract MSA Transformer embeddings for ProteinNPT. These MSAs are generated from the full MSAs provided with ProteinGym by running hhfilter [41], requiring a minimum coverage of 75% and a maximum sequence identity of 90%. Subsequently, we use weighted sampling as described in [42] to select sequences to pass as context to PoET, up to a maximum context length of 8192 tokens. The MSA is encoded using a frozen copy of the PoET model into a set of cached hidden representations, as described in Appendix A.3. When ensembling, a separate

MSA is sampled for each ensemble member, and held fixed during the fine-tuning of that ensemble member.

A.6 Baseline models

ProteinNPT, the embeddings augmented (Emb. aug.) baselines, and the one-hot encoding augmented (OHE aug.) baselines were all run using the code released by [30]. The one-hot and embedding augmented models both use the strategy from [16] of combining the zero-shot predictions from a pretrained model with sequence features in a regression framework. They differ in the way sequence features are extracted: in the former case, ridge regression is performed directly on the one-hot encoded sequences. In the latter case, PLM embeddings are used to featurise the sequences. We refer to [30] for further details.

For the fitness prediction experiments, separate ProteinNPT models were trained for 2000 and 10000 steps, and the results of the best-performing model were reported. The other baselines appeared to benefit more from longer training and were trained for 10000 steps, as in [30]. For design experiments, we used the Monte Carlo dropout uncertainty quantification strategy proposed by [30] for both ProteinNPT and baselines. Notin et al. [30] report best results with a 'hybrid' uncertainty quantification strategy, however this strategy is not implemented in the publicly available code.

A.7 Upper confidence bound acquisition function

For all ensembles, we use a parameter of $\lambda = 0.1$ within the upper confidence bound acquisition function $a(x, \lambda) = \mu(x) + \lambda\sigma(x)$, where $\mu(x)$ and $\sigma(x)$ respectively are the mean and standard deviation of the predictions of the model given an input sequence x.

A.8 Implementation of ranking loss

We implement a pairwise ranking loss by using binary cross-entropy to train the model on ranking tasks by comparing predicted scores for all pairs in a batch. It computes the difference in predictions for every pair, determines the ground-truth ranking based on actual values, and applies BCE loss to these pairwise comparisons, excluding self-comparisons via a diagonal mask. An example PyTorch implementation can be found below:

```
import torch
import torch.nn.functional as F

def full_ranking_bce(preds: torch.Tensor, targets: torch.Tensor) ->
    torch.Tensor:
    # Calculate pairwise differences between all predictions
    pairwise_diffs = preds[:, None] - preds[None, :]

    # Determine if each target is greater than others in a pairwise
        manner
    target_comparisons = targets[:, None] > targets[None, :]

    # Compute binary cross-entropy loss for pairwise comparisons
    ranking_loss = F.binary_cross_entropy_with_logits(pairwise_diffs,
        target_comparisons.float(), reduction='none')

    # Create a mask to exclude diagonal elements (self-comparisons)
    batch_size = preds.size(0)
    diag_mask = 1 - torch.eye(batch_size, device=preds.device)

    # Apply the mask and calculate the mean loss, excluding the
        diagonal
    masked_loss = 0.5 * ranking_loss * diag_mask
    return masked_loss.mean((-1, -2))
```

A.9 Compute requirements

All experiments were run on either V100 or A100 NVIDIA GPUs. Compute required for a single fine-tuning run varies based on the model, the length of the protein sequences, and the size of the dataset. We provide representative timings for the AAV dataset in Table 4. Design experiments involved 10 rounds of fine-tuning and therefore required roughly ten times the computation of a single fine-tuning run.

Table 4: Representative run times for fine-tuning on the AAV landscape ($n = 512$) on an A100 GPU, averaged across 3 seeds.

Model name	Time
ProteinNPT (MSAT)	4h 40 m
ESM1v regression	2h 27 m
ESM1v ranking	4 m
PoET ranking	41 m
PoET regression head	36 m

A.10 Performance by landscape for supervised experiments

We provide barplots summarising per-landscape performance for selected models on the $n = 128$ single and multi-mutant splits in Figures 2 and 3.

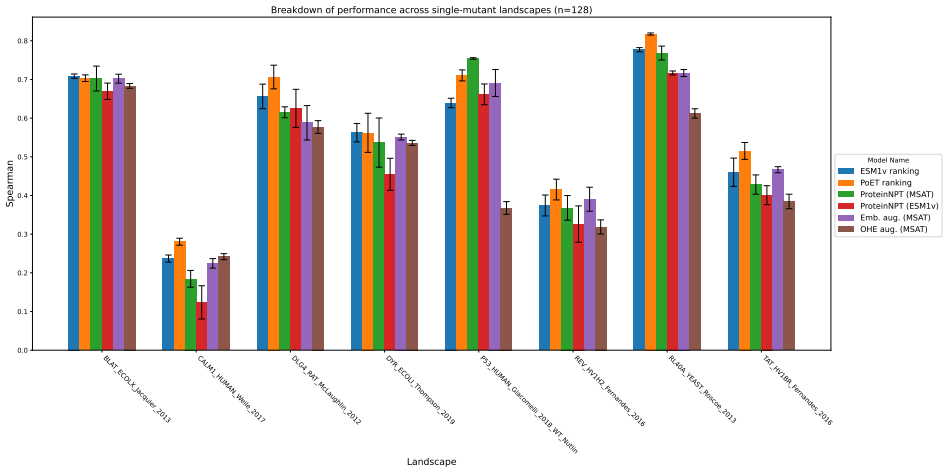

Figure 2: Per-landscape performance for singles datasets ($n = 128$). Error bars represent standard deviations across the three train/test splits.

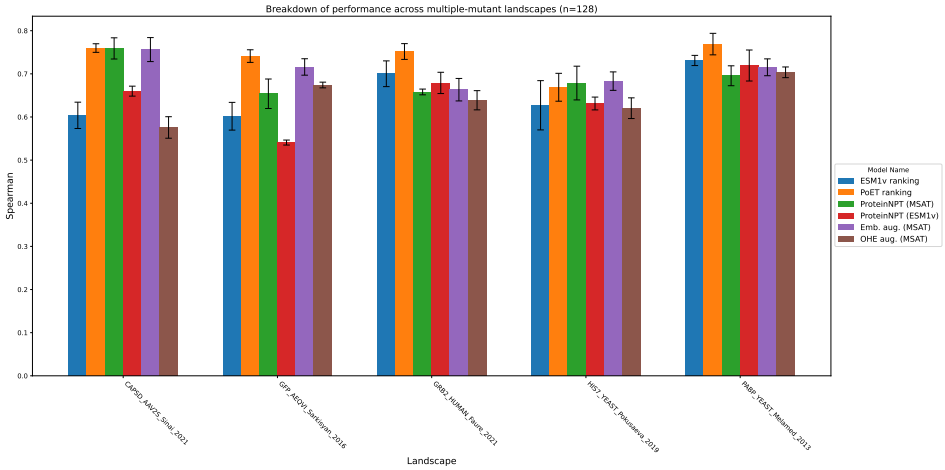

Figure 3: Per-landscape performance for multiples datasets ($n = 128$). Error bars represent standard deviations across the three train/test splits.

A.11 Additional design plots

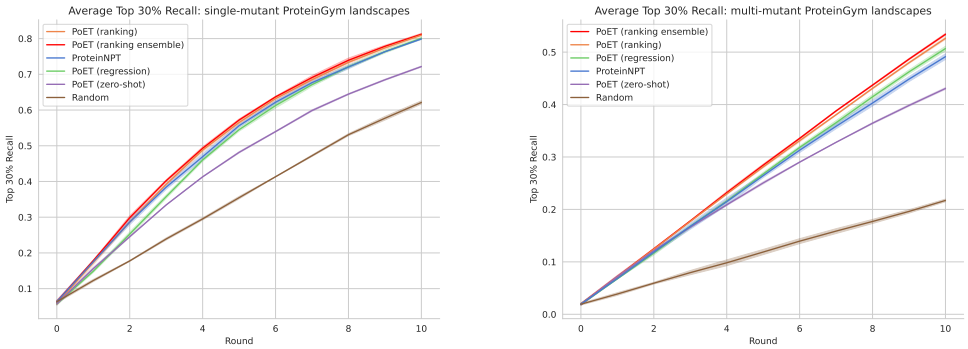

Figure 4: **Left**: Average top 30% recall for 8 single-mutant landscapes for alternative PoET configurations as well as selected baselines. **Right**: Average top 30% recall for 5 multi-mutant landscapes for alternative PoET configurations as well as selected baselines.

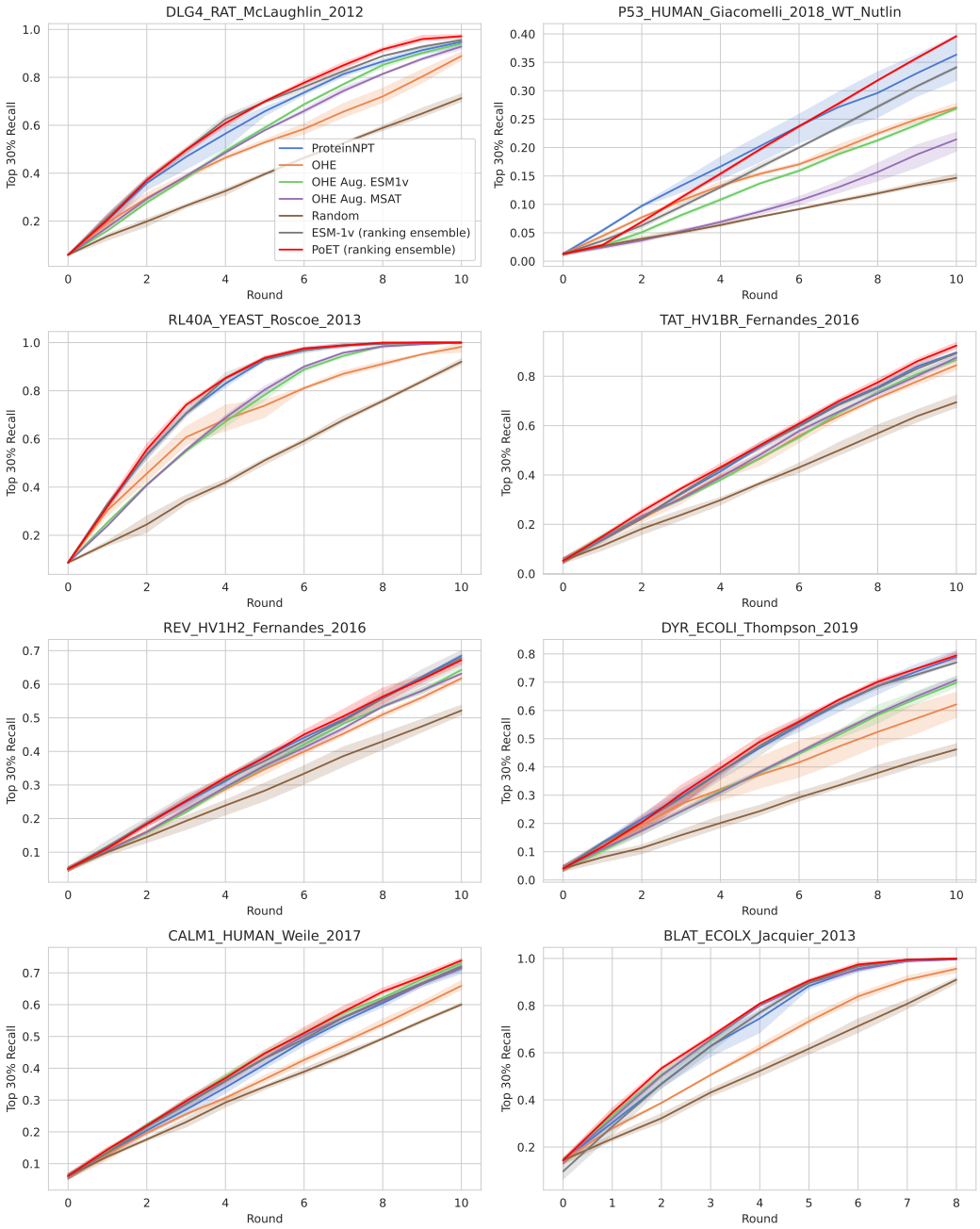

Figure 5: Design curves for individual single mutant landscapes.

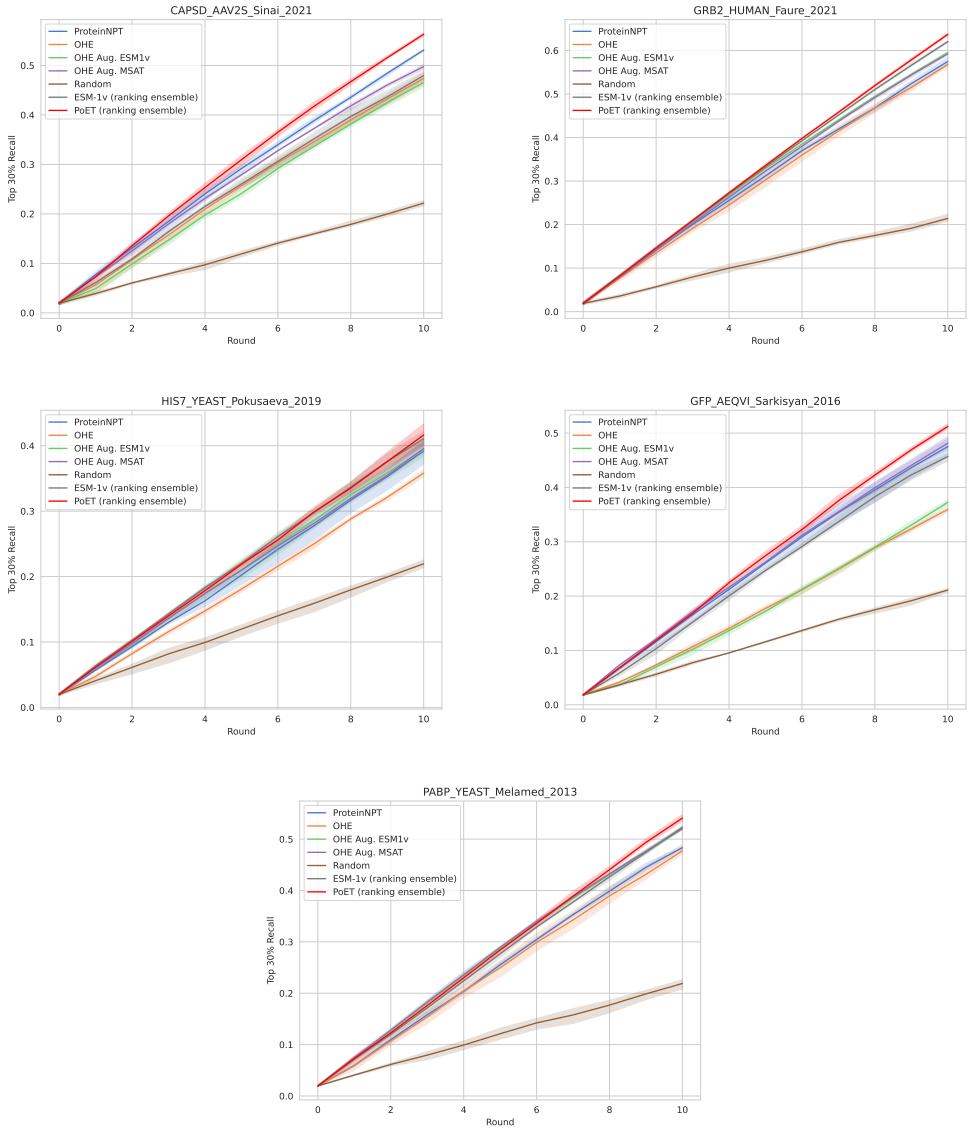

Figure 6: Design curves for individual multi-mutant landscapes