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

Alex Hawkins-Hooker [1 2 *] **Jakub Kmec** [2] **Oliver Bent** [2] **Paul Duckworth** [2]

Abstract

Although various schemes have been proposed for exploiting the distributional knowledge captured by protein language models (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. 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

In practical protein 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, however experimental constraints mean that it might only be feasible to generate measurements for tens or hundreds of proteins at a time (Biswas et al., 2021). It is therefore of considerable interest to ask how the zero-shot prediction capacities of PLMs can be combined with small labelled datasets to achieve improved predictive performance.

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 (Alley et al., 2019; Biswas et al., 2021; Rao et al., 2019; Dallago et al., 2021; Notin et al., 2023b). 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 (Ouyang et al., 2022; Rafailov et al., 2023), thereby fully exploiting the distributional knowledge contained in the original pretrained model. Although related fine-tuning strategies have been considered in the context of fitness prediction with unconditional autoregressive PLMs (Krause et al., 2021), previous work has not addressed their effectiveness across different classes of PLM, nor considered 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 approaches to alternative PLM-based fitness prediction strategies, including recent innovations in architectures for operating over frozen PLM embeddings, (Notin et al., 2023b), 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 (Notin et al., 2023b), 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. Likelihood-based fine-tuning

2.1. Background

Two recent works have advocated the use of ranking-based loss functions (Krause et al., 2021; Brookes et al., 2023) for training supervised fitness predictors. In particular, they suggest parameterising a Bradley-Terry model (Bradley & Terry, 1952) 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

[*]Work completed during an internship at InstaDeep. [1]AI Centre, University College London [2]InstaDeep Ltd.. Correspondence to: Alex Hawkins-Hooker <ucabawk@ucl.ac.uk>, Paul Duckworth <p.duckworth@instadeep.com>.

Proceedings of the ICML 2024 Workshop on Accessible and Efficient Foundation Models for Biological Discovery, Vienna, Austria.

by parameterising a binary classifier via the difference in values of 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, converting the regression problem with N labels into a binary classification problem with $N \times N$ labels. To fine-tune an autoregressive protein language model, Krause et al. (2021) propose using an unconditional sequence log-likelihood as the scoring function:

$$s_\theta(x) = \sum_{i=1}^{L} \log p(x_i | x_{<i}) \,. \tag{2}$$

More recently (Lee et al., 2023) proposed adapting direct preference optimization (DPO) (Rafailov et al., 2023), parameterising the scoring function via a difference in log-likelihood ratios between the fine-tuned model and a reference model. In our own experiments, we did not see improvements from introducing a reference model, so instead focussed on adapting the parameterisation used by (Krause et al., 2021) to other PLMs.

2.2. Extension to other classes of protein language model

Unconditional autoregressive models often underperform other classes of model including conditional autoregressive models and masked language models in fitness prediction settings (Notin et al., 2023a). We therefore extend fine-tuning via 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 $s_\theta(x, c)$, detailed below.

2.2.1. Masked protein language models

Masked language models do not define a sequence-level likelihood that can directly be used as a scoring function. Instead we build on the zero-shot scoring strategies proposed by Meier et al. (2021) to allow these models to be fine-tuned with ranking-based losses, similar to other concurrent work (Zhao et al., 2024). Concretely, we utilize the 'wild-type marginals' scoring function from Meier et al. (2021). 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:

$$s_\theta(x, x^{\text{wt}}) = \sum_{i : x_i^{\text{wt}} \neq x_i} \log p(x_i | x^{\text{wt}}) - \log p(x_i^{\text{wt}} | x^{\text{wt}}) \,. \tag{3}$$

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.

2.2.2. Family-based protein language models

Family-based protein language models represent the conditional distribution over family members given a subset of other family members (Rao et al., 2021; Hawkins-Hooker et al., 2021; Ram & Bepler, 2022; Truong Jr & Bepler, 2023). 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 (Truong Jr & Bepler, 2023), 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 2 on the sequences in the MSA:

$$s_\theta(x, M) = \sum_{i=1}^{L} \log p(x_i | x_{<i}, M) \,. \tag{4}$$

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 (Truong Jr & Bepler, 2023). 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.

2.2.3. Evolutionary context ensembles

The amino acid output distributions learned by protein language models depend heavily on sequence context. We propose to exploit this property to build ensembles of fine-tuned PLMs, in which each ensemble member sees a different, but approximately biologically equivalent, context. 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. To score sequences, we use an ensembled scoring function:

$$s_{\theta_1,\ldots,\theta_K}(x, M) = \frac{1}{K} \sum_{k=1}^{K} s_{\theta_k}(x, M_k). \tag{5}$$

To achieve a similar effect with ESM-1v, which does not use MSAs, we instead sample a set of K input masks, and

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
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

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.

3. Few-shot fitness prediction

We study the performance of fitness prediction strategies on mutational landscapes from ProteinGym (Notin et al., 2023a). We utilise two subsets of ProteinGym: the validation set of 8 representative single-mutant landscapes selected by Notin et al. (2023b), and a set of multi-mutant landscapes, chosen to constitute a non-redundant set of the most diverse landscapes available (Appendix A). 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. 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, we compute the Spearman correlation with respect to the scoring function $s_\theta(x, c)$.

We evaluate the performance of the likelihood-based fine-tuning strategies introduced in Section 2.2 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 (Meier et al., 2021), the unconditional autoregressive model ProGen2 (Nijkamp et al., 2023), and the family-based autoregressive model PoET (Truong Jr & Bepler, 2023). For ProGen2 we obtained slightly better results with the 'small' checkpoint model than the 'medium' one, so report the former. We compare to two sets of baselines, representative of widely used approaches that either (i) fine-tune PLMs by adding a regression head (Rao et al., 2019), or (ii) train new models on top of frozen language model embeddings (Notin et al., 2023b). In the first case, we add linear regression heads to both ESM-1v and PoET, and fine-tune all parameters. As the leading example of the second class of approaches, we compare against ProteinNPT (Notin et al., 2023b), a state-of-the-art model operating on top of frozen language model embeddings. As additional baselines, we include the embedding-based 'augmented density' strategies used as baselines by (Notin et al., 2023b). These are regression models taking as input the zero-shot predictions of a PLM as well as an embedding extracted from the same PLM (Notin et al., 2023b). Hyperparameters for fine-tuned models are selected based on performance on the single mutant set, consistent with the practice used for ProteinNPT and associated baselines.

3.1. Results

Ranking-based fine-tuning of PoET outperforms ProteinNPT across all settings (Table 1), 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. Ranking-based fine-tuning also performs better than fine-tuning PoET with a linear head, although this is a strong baseline which also outperforms ProteinNPT. 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. The wild-type marginals scoring function 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. No such limitation applies to the scoring functions used for fine-tuning autoregressive models, explaining the fact that ProGen2 outperforms ranking-based fine-tuning of

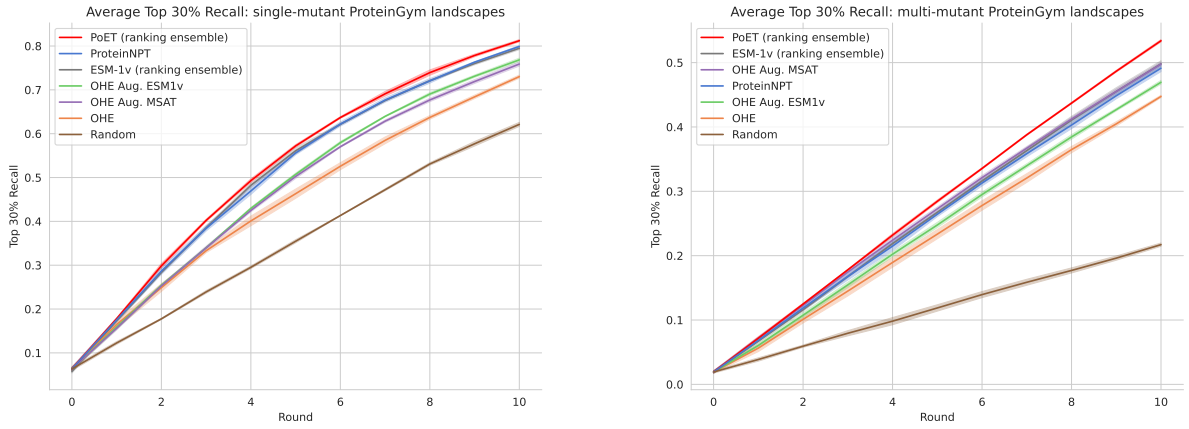

Figure 1. Top 30% recall averaged over: (left): 8 single-mutant landscapes and (right): 5 multi-mutant landscapes. The shading represents one standard deviation over 3 random seeds.

ESM-1v on the multiple mutants datasets, but not the single mutants datasets. For both ESM-1v and PoET, the proposed ensembling strategies further improve performance, sometimes substantially, and show improved uncertainty calibration, as measured by the negative log likelihood of pairwise classifications from the test set (Table 2).

Random splits provide an estimate of performance on held-out 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 (Notin et al., 2023b). We report 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.

4. Multi-round design on fitness landscapes

We next ask whether the improvements in predictive performance translate to benefits in a multi-round design setting. We follow the evaluation protocol introduced by Notin et al. (2023b) in which 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. Models are seeded in round 0 with 100 sequences randomly sampled from the landscape. 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. We follow Notin et al. (2023b) 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), using the upper confidence bound acquisition function, and compare optimisation guided by ensembles of PoET and ESM-1v ranking models to ProteinNPT, as well as selected baselines 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 2).

5. Conclusion

Here we show that the distributions learned by PLMs can be rapidly adapted via feedback from relatively few experimental measurements. Even 128 sequences - of the order of a typical batch size in wet lab experiments - allow significant improvements over zero-shot performance. 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 across supervised and multi-round design settings. An intriguing possibility is that when generative PLMs are fine-tuned via likelihood-based loss functions, they may retain their generative capacity, and we believe studying this possibility by leveraging the connection to methods like DPO (Rafailov et al., 2023) to be a promising avenue for future work.

References

Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M., and Church, G. M. 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.

Biswas, S., Khimulya, G., Alley, E. C., Esvelt, K. M., and Church, G. M. 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.

Bradley, R. A. and Terry, M. E. 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.

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

Dallago, C., Mou, J., Johnston, K. E., Wittmann, B., Bhattacharya, N., Goldman, S., Madani, A., and Yang, K. K. FLIP: Benchmark tasks in fitness landscape inference for proteins. August 2021. URL https://openreview.net/forum?id=p2dMLEwL8tF.

Gonzalez Somermeyer, L., Fleiss, A., Mishin, A. S., Bozhanova, N. G., Igolkina, A. A., Meiler, J., Alaball Pujol, M.-E., Putintseva, E. V., Sarkisyan, K. S., and Kondrashov, F. A. 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.

Hawkins-Hooker, A., Jones, D. T., and Paige, B. MSA-Conditioned Generative Protein Language Models for Fitness Landscape Modelling and Design. In *Machine Learning in Structural Biology Workshop at NeurIPS*, 2021.

Hsu, C., Nisonoff, H., Fannjiang, C., and Listgarten, J. Learning protein fitness models from evolutionary and assay-labeled data. *Nature Biotechnology*, 40(7):1114–1122, July 2022. ISSN 1546-1696. doi: 10.1038/s41587-021-01146-5.

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

Krause, B., Naik, N., Liu, W., and Madani, A. Don't throw away that linear head: Few-shot protein fitness prediction with generative models. October 2021. URL https://openreview.net/forum?id=hHmtmT58pSL.

Lee, M., Lee, K., and Shin, J. 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.

Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., and Rives, A. Language models enable zero-shot prediction of the effects of mutations on protein function. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29287–29303. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f51338d736f95dd42427296047067694-Abstract.html.

Nijkamp, E., Ruffolo, J. A., Weinstein, E. N., Naik, N., and Madani, A. 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.

Notin, P., Kollasch, A. W., Ritter, D., Niekerk, L. V., Paul, S., Spinner, H., Rollins, N. J., Shaw, A., Weitzman, R., Frazer, J., Dias, M., Franceschi, D., Orenbuch, R., Gal, Y., and Marks, D. S. ProteinGym: Large-Scale Benchmarks for Protein Fitness Prediction and Design. November 2023a. URL https://openreview.net/forum?id=URoZHqAohf&noteId=LmiOZsZZAh.

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

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Gray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and

Lowe, R. Training language models to follow instructions with human feedback. May 2022. URL https://openreview.net/forum?id=TG8KACxEON.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. 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).

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

Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, P., Canny, J., Abbeel, P., and Song, Y. 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.

Rao, R. M., Liu, J., Verkuil, R., Meier, J., Canny, J., Abbeel, P., Sercu, T., and Rives, A. Msa transformer. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8844–8856. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/rao21a.html.

Truong Jr, T. F. and Bepler, T. PoET: A generative model of protein families as sequences-of-sequences. November 2023. URL https://openreview.net/forum?id=1CJ8D7P8RZ.

Zhao, J., Zhang, C., and Luo, Y. 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. Fitness landscapes

We use the set of 8 single-mutant landscapes selected for ablations and hyperparameter selection by (Notin et al., 2023b). 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 (Gonzalez Somermeyer et al., 2022) are landscapes of close homologues of the GFP protein whose landscape was reported by Sarkisyan et al. (2016). 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

B. Batch level optimisation of Bradley-Terry model

Given a batch of B sequences $x_1, ..., x_B$, and a scoring function $s_\theta(x, c)$, maximising the likelihood of the Bradley-Terry model over all pairwise comparisons between sequences in the batch is equivalent to minimising the loss:

$$\mathcal{L} = \sum_{i=1}^{B} \sum_{j=1}^{B} -\mathbb{I}(y(x_i) > y(x_j)) \log \sigma(s_\theta(x_i) - s_\theta(x_j)) \,, \tag{6}$$

C. 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:

$$p(M) = \prod_i p(m^{(i)} | m^{(<i)}) = \prod_{ij} p(m_j^{(i)} | m_{<j}^{(i)}, m^{(<i)}) \,. \tag{7}$$

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:

$$H_{<i} = f_\theta(m^{(<i)}) \,. \tag{8}$$

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:

$$\text{logit}_{ij} = g_\theta(m^{(i)}_{<j}, H_{<i}) \,. \tag{9}$$

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 6 on the labelled data. Concretely, the scoring function used to parameterise the Bradley-Terry model becomes:

$$s_\theta(x, M) \equiv s_\theta(x, H) = \sum_i \log p_\theta(x_i | x_{<i}, H) \tag{10}$$

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

D. 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 (Kingma & Ba, 2015) 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.

D.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.

D.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.

E. 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

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**

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:

$$s(x, x^{\text{wt}}) = \frac{1}{K} \sum_k s(x, \tilde{x}_k^{\text{wt}}) \tag{11}$$

F. 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, requiring a minimum coverage of 75% and a maximum sequence identity of 90%. Subsequently, we use weighted sampling 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 C. When ensembling, a separate MSA is sampled for each ensemble member, and held fixed during the fine-tuning of that ensemble member.

G. Baseline models

ProteinNPT and the embeddings augmented (Emb. aug.) baselines were run using the code released by (Notin et al., 2023b). The one-hot and embedding augmented models both use the strategy from (Hsu et al., 2022) 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 (Notin et al., 2023b) 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 (Notin et al., 2023b). For design experiments, we used the Monte Carlo dropout uncertainty quantification strategy proposed by (Notin et al., 2023b) for both ProteinNPT and baselines. Notin et al. (2023b) report best results with a 'hybrid' uncertainty quantification strategy, however this strategy is not implemented in the publicly available code.

H. Evaluating ensemble performance

We report performance of ensembles on the multiple mutant datasets in Table 2. In addition to the Spearman correlation, we report the negative log likelihood for pairwise classifications under the Bradley-Terry model, evaluated across pairs of sequences in the test set. This test set log-likelihood provides a measure of the calibration of the ensemble.

I. 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 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

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

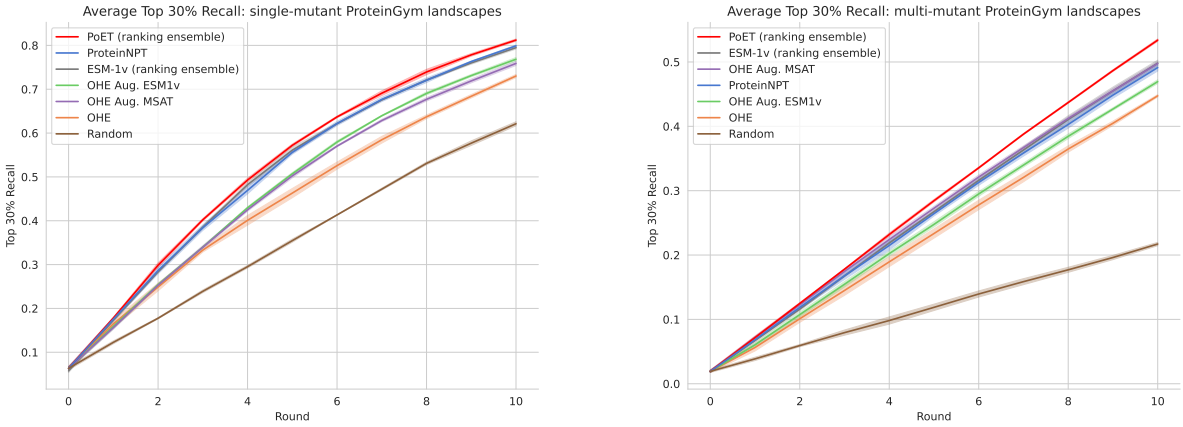

Figure 2. **Left**: Average top 30% recall for 8 single-mutant landscapes for alternative PoET configurations as well as selected baselines. **Right**: Average top 30% recall for 5 multi-mutant landscapes for alternative PoET configurations as well as selected baselines.

J. Additional design plots

We compare different PoET configurations for design on the multiple mutants landscapes in Figure 2. We provide per-landscape plots at the end of the Appendix.

K. 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.

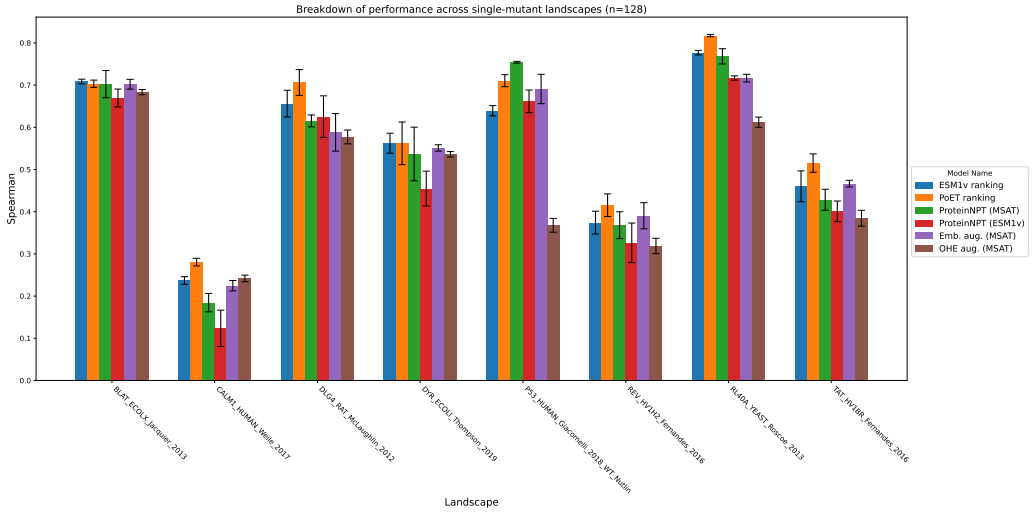

Figure 3. Per-landscape performance for singles datasets ($n = 128$). Error bars represent standard deviations across the three train/test splits. The shading represents one standard deviation over 3 random seeds.

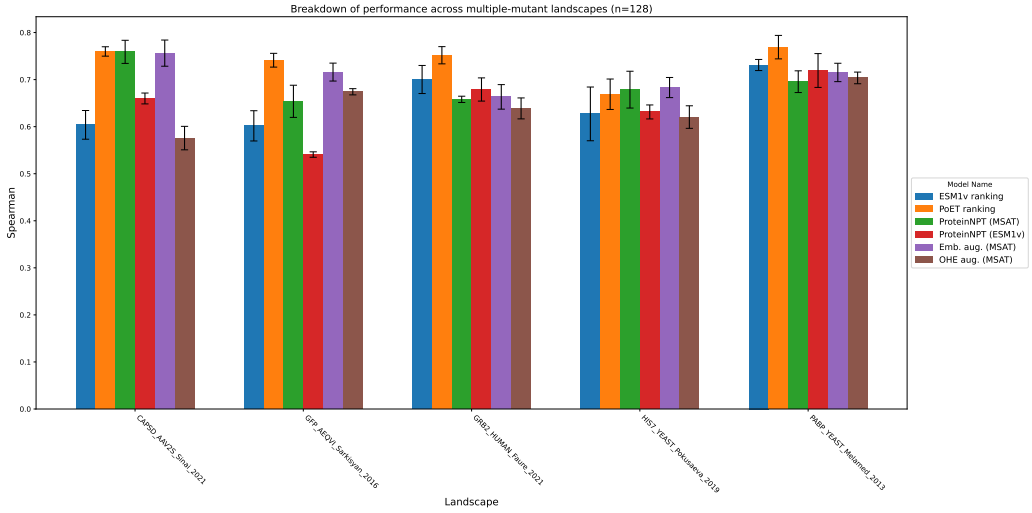

Figure 4. Per-landscape performance for multiples datasets ($n = 128$). Error bars represent standard deviations across the three train/test splits.

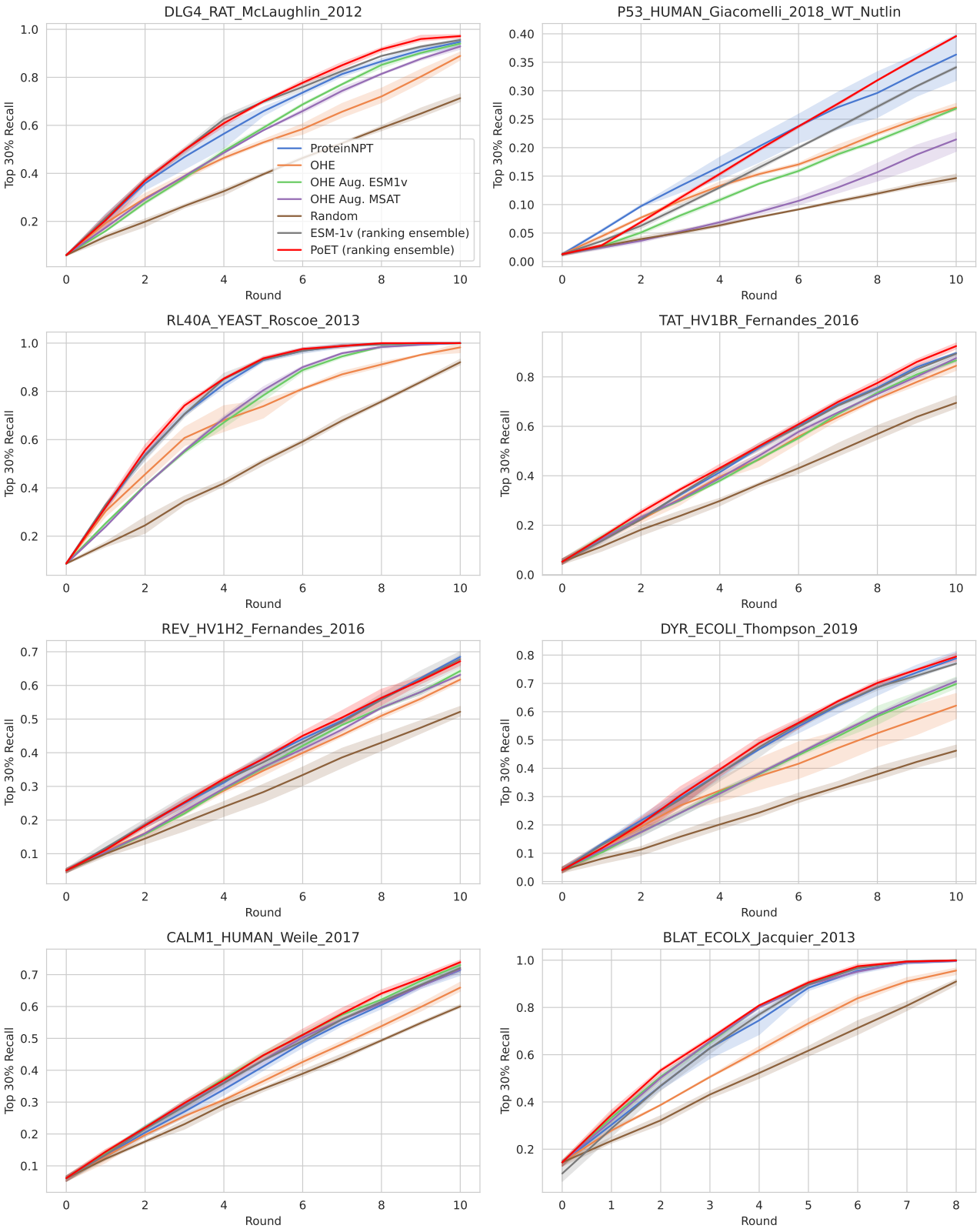

Figure 5. Design curves for individual single mutant landscapes.

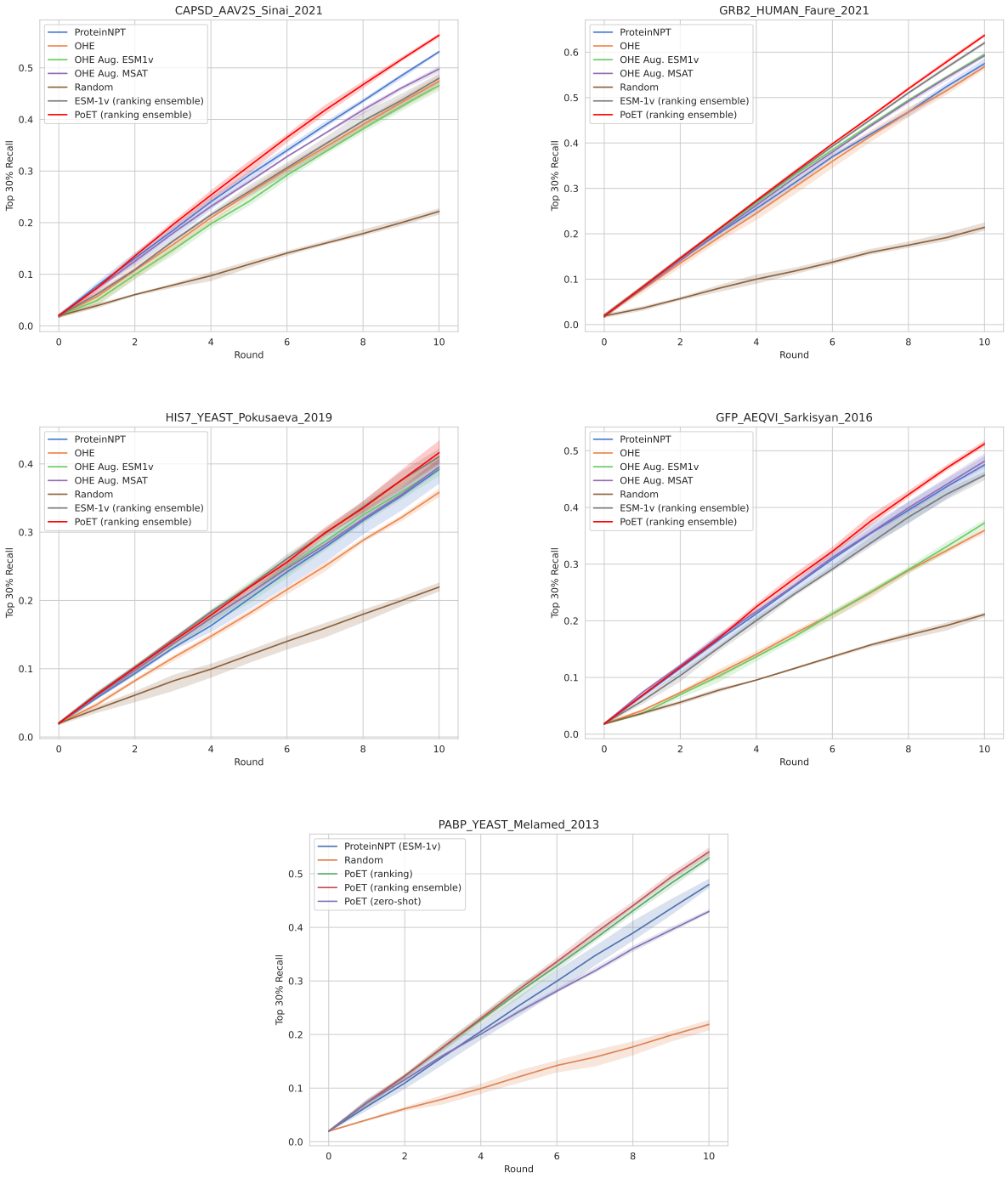

Figure 6. Design curves for individual multi-mutant landscapes