Effective Surrogate Models for Protein Design with Bayesian Optimization

Nate Gruver[1] Samuel Stanton[1] Polina Kirichenko[1] Marc Finzi[1] Phillip Maffettone[2] Vivek Myers[2] Emily Delaney[2] Peyton Greenside[2] Andrew Gordon Wilson[1]

Abstract

Bayesian optimization, which uses a probabilistic surrogate for an expensive black-box function, provides a framework for protein design that requires a small amount of labeled data. In this paper, we compare three approaches to constructing surrogate models for protein design on synthetic benchmarks. We find that neural network ensembles trained directly on primary sequences outperform string kernel Gaussian processes and models built on pre-trained embeddings. We show that this superior performance is likely due to improved robustness on out-of-distribution data. Transferring these insights into practice, we apply our approach to optimizing the Stokes shift of green fluorescent protein, discovering and synthesizing novel variants with improved functional properties.

1. Introduction

Biological sequence design is a problem of clear significance and widespread application – for example, in engineering of vaccines (Yang et al., 2021; Malone et al., 2020). Recent advances in synthetic biology (Ran et al., 2013) have drastically accelerated the rate at which novel protein and DNA sequences can be produced and characterized, but the associated search spaces are still unfathomably large (e.g. 20^{300}, the number of proteins that are 300 residues long), necessitating learning-based approaches to the generation and prioritization of candidate sequences (Yang et al., 2019; Sinai and Kelsic, 2020; Wittmann et al., 2021).

Bayesian optimization (BO) is a general procedure for optimizing expensive black-box objectives (like the outcome of synthesizing a protein) by constructing a probabilistic surrogate of the objective. BO involves many design decisions, including the choice of surrogate model class and the hyperparameter selection that entails. Despite extensive work on the sequence design problem from both the biological (Saito et al., 2018; Yang et al., 2019; Biswas et al., 2020; Shin et al., 2021) and machine learning (Angermueller et al., 2020) communities, even the basic step of evaluating possible algorithmic design choices on tasks representative of the sequence design problem is surprisingly difficult. For example, evaluating potential surrogate models on their ability to predict some target feature on a static dataset will not capture how resilient the models are to distribution shift. Because of the expense of synthesizing sequences, it is vitally important that rigorous evaluation procedures are used to validate algorithmic design choices.

In this work we demonstrate how such evaluation procedures can be constructed by comparing several different potential surrogate models on a range of simulated design tasks. We also present promising results using our best surrogate to optimize a new target feature of green fluorescent protein (GFP) in a wet lab. This work can serve both as a blueprint for validating new models and algorithms, and as a case study in collaboration between ML and biomedical researchers.

2. Preliminaries

Design task We consider optimization of a fitness function $f : \mathcal{X} \mapsto \mathbb{R}$ where $\mathcal{X}$ is the set of all strings up to length m from the alphabet Σ. In the design task we begin with a dataset $\mathcal{D} = (X_{\text{train}}, \mathbf{y}_{\text{train}})$ of size n, where $X_{\text{train}} \subset \mathcal{X}$ and $y = f(\mathbf{x}) + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. In each phase of optimization we select k new points $X_{\text{query}} \subset \mathcal{X}$ at which to query f. After observing $\mathbf{y}_{\text{query}}$ we obtain a new dataset $\mathcal{D}' = \mathcal{D} \cup (X_{\text{query}}, \mathbf{y}_{\text{query}})$, and the process is repeated. The goal is to find $\mathbf{x}^* = \text{argmax}_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$.

Bayesian optimization BO uses a probabilistic surrogate of the objective to define an acquisition function $a : \mathcal{X} \mapsto \mathbb{R}$. The acquisition function is used to select the most promising candidates to query the objective function,

$$\mathbf{x}_{\text{query}} = \underset{\mathbf{x} \in \mathcal{X}}{\text{argmax}}\, a(\mathbf{x})$$

The acquisition function manages the explore-exploit tradeoff by preferring sequences with high predicted value that also provide the surrogate new information.

[1] New York University [2] BigHat Biosciences. Correspondence to: Samuel Stanton <ss13641@nyu.edu>.

The 2021 ICML Workshop on Computational Biology. Copyright 2021 by the author(s).

Gaussian processes Gaussian process (GP) regression models are often favored as surrogate models for BO because they precisely quantify uncertainty. The behavior of a GP is determined by its kernel, which encodes some measure of distance between inputs (e.g. ℓ_2 distance for inputs in a vector space). There are two basic approaches to define kernels over string inputs. The first is to define specialized kernels that specifically operate on strings, e.g. the substring kernel (Moss et al., 2020, SSK). The second is to use some embedding procedure to map the strings to features in a vector space, after which a standard kernel (e.g. RBF) can be used.

Pre-trained language model embeddings Representations from pre-trained language models are a particularly interesting embedding procedure. In the last few years there has been rapid progress in pre-trained language models for proteins (Rao et al., 2019; Rives et al., 2019; Rao et al., 2021). Biswas et al. (2020) and Hie et al. (2020) explore using pre-trained language models for sequence design, demonstrating good performance even with limited task-specific data. In our experiments we focus on embeddings derived from the BERT language model (Devlin et al., 2018).

Deep ensembles One limitation of GPs, in particular those equipped with string kernels, is superlinear scaling in the number of datapoints and length of strings. *Deep ensembles* of independently trained neural networks (Lakshminarayanan et al., 2016), are a popular alternative, and have been shown to be robust to distribution shift and have well-calibrated predictive uncertainty (Izmailov et al., 2021).

3. Evaluating Surrogate Models

In this section we discuss our procedure for evaluating different potential surrogate models before progressing to wet-lab experiments.

Model types We focus on methods that *scale to large search spaces* (e.g. $m = 300$, $|\Sigma| = 20$ for the GFP protein families) and are *effective under data-scarcity*.

RNN Ensemble A deep ensemble of recurrent neural networks (Cho et al., 2014).

CNN Ensemble A deep ensemble of convolutional neural networks (LeCun et al., 1998).

Embedding Ensemble A deep ensemble of multi-layer perceptrons (MLPs) trained on fine-tuned BERT embeddings

GP SSK GP regression with the SSK kernel (Moss et al., 2020).

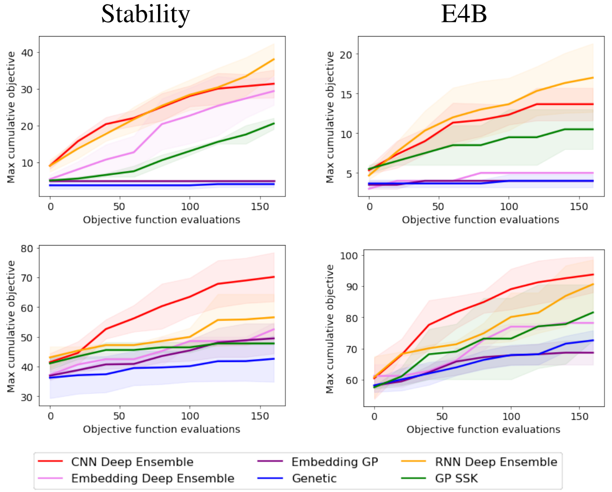

Figure 1: Cumulative max value of the objective function given a fixed number of function evaluations. (**Top**): Substring (**Bottom**): RNAFold. All Bayesian optimization methods are baselined against the genetic optimizer. Ensemble methods clearly outperform competing approaches both in terms of max objective value obtained and area under the curve. Shaded regions show two standard deviations taken over 3 different seeds. Other objectives shown in Appendix D

Embedding GP GP regression with a Matérn-$\frac{5}{2}$ kernel over fine-tuned BERT embeddings.

Protein datasets We use three datasets as the basis for our surrogate evaluation. Two of these datasets are standard protein benchmarks made available in Rao et al. (2019), while the third is obtained from Esposito et al. (2019).

LocalFL GFP variants with log fluorescence labels (Sarkisyan et al., 2016).

Stability Short proteins with stability labels (Rocklin et al., 2017) .

E4B Variants of the Ubiquitination Factor E4B gene, which we translate into proteins. The labels for this dataset are the rate of ubiquitination of the target protein by E3 ubiquitin (Starita et al., 2013).

Black-box optimization tasks Our simulated black-box objectives each map a protein sequence to a continuous score.

Substring The number of occurrences of the most common bigram in the string dataset. Similar objectives appear in (Moss et al., 2020).

NN-Oracle The ground truth label for each dataset approximated by a neural network oracle. Specifically, we fit a CNN oracle to the ground-truth labels of the entire

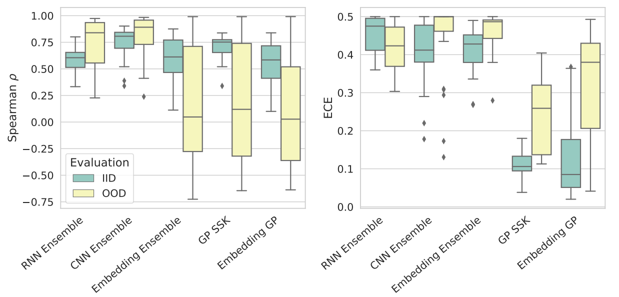

Figure 2: Spearman's ρ (**Left**) and ECE (**Right**) evaluated on the surrogate models for in-distribution (IID) and out-of-distribution (OOD). For the OOD evaluation, each surrogate model is trained on the first 10 iterations of genetic optimization and evaluated on the last 10 iterations (500 data points each) which tend to have higher values of the objective. With the distribution shift the RNN and CNN ensembles generalize well, while the performance of the embedding models and GPs degrade substantially.

dataset and treat its predictions as the true objective. Similar objectives are proposed in (Kumar and Levine, 2019; Angermueller et al., 2020).

RNAFOLD The free energy of the protein's RNA sequence calculated by a statistical simulation. We reverse-translated each protein deterministically using a dictionary lookup and then use the publicly available ViennaRNA package (Lorenz et al., 2011) to calculate the free energy.

Optimization procedure Every surrogate was required to propose 20 rounds of 100 query points (the size of a typical wet-lab batch) for each dataset-objective pair. We maximized the upper confidence bound acquisition function (Auer, 2002, UCB) with a tournament-style genetic optimizer (Moss et al., 2020) to determine each batch, training the surrogates on the initial data and their previous queries. We compare the best seen objective value at each proposal round for each surrogate, compared to that obtained by the genetic optimizer with access to the true objective, restricted to the same number of total objective function queries (Figure 1). Successful methods should both find the candidates with the highest value of the objective and find them with the fewest evaluations. We see that CNN and RNN ensemble surrogate models consistently perform better than competing approach across datasets and objective functions, finding the best sequences faster.

Out-of-distribution robustness We now further investigate what factors contribute to the success of ensemble methods, since it is important to understand the underlying factors of success before deploying the method to costly wet-lab experiments. Prior work has predominantly focused on evaluating methods end-to-end or creating hand-crafted out-of-distribution (OOD) splits using heuristics like edit distance to evaluate surrogates offline. We hope to test surrogate models on OOD generalization by comparing them in regression on holdout sets that reflect the types of distribution shifts we might actually see in black-box optimization. In order to construct realistic shifts, we generate batches of query points with increasing fitness by maximizing our black-box objectives with a genetic optimizer. The distribution shift between batches is the same kind of shift we would expect the surrogates to see candidates are gradually improving. We can evaluate our surrogates offline by training on early batches and evaluating on later ones.

Figure 2 shows an evaluation of the surrogate models on static splits from the optimizer averaged across the three datasets and reward functions (9 distinct tasks). We use Spearman's ρ to gauge regression performance on the objective and expected calibration error (ECE) to gauge the quality of the model's uncertainties. Although the embedding models and GPs approaches are well calibrated on in-distribution data, the predictive performance and uncertainty calibration of these models degrade sharply on out-of-distribution data. In Figure 3, we visualize these errors along with the predictive variances in a two dimensional ESM embedding (Rives et al., 2021).

It has recently been suggested by Shanehsazzadeh et al. (2020) that despite conventional wisdom, methods using unsupervised pre-training do not outperform simple CNN ensembles on benchmark regression tasks. We show that the situation is even more extreme in the presence of distribution shift. Over many rounds of optimization, these OOD errors compound, leading to impoverished queries of the objective function and worse optimization outcomes. Even when given the amount of data, the superior generalization abilities of deep ensembles enable rapid progress towards promising regions of the search space.

4. Designing GFP Variants

With the previous evaluation complete, we are now sufficiently confident in the reliability of CNN deep ensembles to deploy them as surrogates to optimize a real black-box protein optimization problem — maximizing the Stokes shift of GFP variants. As its name suggests, GFP is a fluorescent protein that is useful for a variety of biomedical research applications. When *excited* by light, GFP will respond by *emitting* fluorescent light at varying wavelengths. Briefly put, the wavelength that causes the protein to emit the most light is the peak excitation wavelength, and the wavelength emitted by the protein with the highest intensity is the peak emission wavelength. The Stokes shift is the difference between the peak excitation and peak emission wavelengths. In order to avoid interference between the light used to excite the protein and the light to be observed emitting from the protein, it is desirable that the Stokes shift

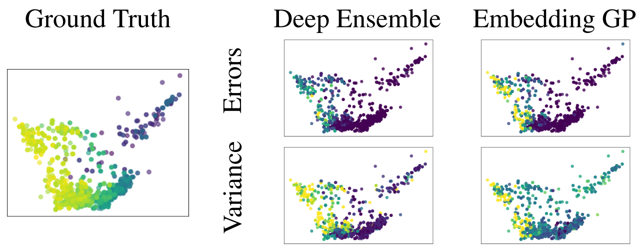

Figure 3: An illustration of model error and uncertainty under distribution shift. Individual points represent proteins generated by 10 rounds of genetic optimization on the RNAFold objective. Locations are given by the first two principle components of the ESM (Rives et al., 2021) embeddings. (**left**) The ground truth scores of the proteins assigned by the RNAFold objective function showing the distribution shift created by the optimizer. (**right top**) Errors in the GP model become large more quickly as the distribution shifts. (**right bottom**) Uncertainties in the GP model are more well-calibrated in the GP model because posterior variance in higher in all regions far from the training data.

Acq. Fn.	Seq. Type	Stokes	Δ Stokes	MaxRFU
Control	Base	3.65e+01	-	1.99e+07
	Opt.	7.32e+01	3.72e+01	8.54e+06
UCB-0.1	Base	3.05e+01	-	2.41e+07
	Opt.	8.91e+01	5.85e+01	3.34e+06

Table 1: **Wet-lab results optimizing GFP sequences.** Δ Stokes indicates the change in average Stokes shift from the base sequences to the optimized sequences. MaxRFU is a measure of brightness. Both the control and CNN-ensemble UCB increased Stokes shift and decreased brightness.

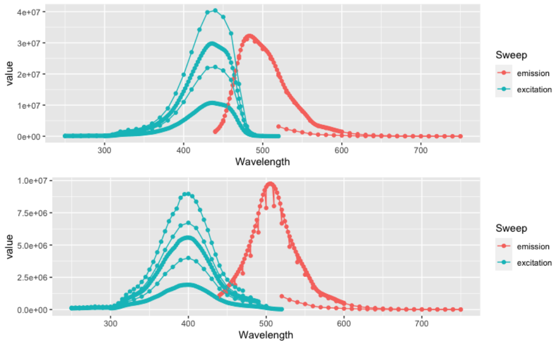

Figure 4: Emission and excitation spectra for a parent GFP variant (top) and our proposed descendent (bottom). Distinct lines denote separate sweeps of measurement over wavelength. In the bottom plot the distance between the two peaks has increased significantly, as we intended.

be large. The Stokes shift of many commercially available fluorescent proteins is less than 30nm.

FPBase (Lambert, 2019) is a database of fluorescent proteins with labels including Stokes shift and brightness. Since the dataset contains many sequences that are not GFP, we first filtered out any sequences that were more than 20 edits from the archetype avGFP sequence, denoting the remaining data as “Trunc-FPBase”.

We used a CNN ensemble surrogate, the UCB acquisition ($\lambda = 0.1$), and a genetic optimizer to propose a batch of 30 query sequences. The genetic optimizer was constrained to make a maximum of 3 string edits relative to the proposal's parent sequence in the start pool to reduce the likelihood of proposing proteins that would fail to fold. To ensure the batch was diverse we performed 30 rounds of optimization of the acquisition function, each time taking only the best ranked query point and removing its parent sequence from the genetic optimizer start pool for later rounds. We also restricted the optimizer start pool to sequences in the top 50% of Trunc-FPBase ranked by brightness in order to avoid the possibility of proposing sequences with no measurable fluorescence. For comparison, we also proposed 10 sequences with a control acquisition function that simply returned uniform random noise when evaluated. The procedural details for the wet-lab experiments can be found in Appendix A.

We present the results of our first batch of proposals in Table 1. We find that both the control acquisition and the CNN-ensemble UCB acquisition successfully increased the average Stokes shift of the base sequences. The increased Stokes shift is visible in the qualitative example we provide in Figure 4.

5. Discussion and Future Work

While our initial wet-lab results are encouraging, there are several avenues for improvement. First, it is clear that the task should be treated as a *constrained* black-box optimization problem, since there is a minimal level of brightness the GFP variants must have (as well as the requirement that it can be expressed in the lab). There are also acquisition functions explicitly meant for querying points in batches that would promote diversity without resorting to heuristics.

In this work we investigated the performance of deep ensembles, pre-trained embeddings, and Gaussian Processes for application to the high dimensional optimization problem of protein design. We introduced several synthetic benchmarks that can be used on real protein sequences and closely align with with typical target functions. We find that CNN and RNN ensembles outperform the other approaches in optimizing the target over multiple rounds of evaluation, owing to their better robustness to distribution shift. Using these insights, we aim these models at optimizing the Stokes shift of Green Fluorescent Protein with a wet-lab experiment. Synthesizing the novel GFP variants, we are able to improve the average Stokes shift with new variants after one round of optimization.

References

Christof Angermueller, David Belanger, Andreea Gane, Zelda Mariet, David Dohan, Kevin Murphy, Lucy Colwell, and D Sculley. Population-based black-box optimization for biological sequence design. In *International Conference on Machine Learning*, pages 324–334. PMLR, 2020.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

Surojit Biswas, Grigory Khimulya, Ethan C Alley, Kevin M Esvelt, and George M Church. Low-n protein engineering with data-efficient deep learning. *BioRxiv*, 2020.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. *Syntax, Semantics and Structure in Statistical Translation*, page 103, 2014.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Daniel Esposito, Jochen Weile, Jay Shendure, Lea M Starita, Anthony T Papenfuss, Frederick P Roth, Douglas M Fowler, and Alan F Rubin. Mavedb: an open-source platform to distribute and interpret data from multiplexed assays of variant effect. *Genome biology*, 20(1):1–11, 2019.

Brian Hie, Bryan D Bryson, and Bonnie Berger. Leveraging uncertainty in machine learning accelerates biological discovery and design. *Cell Systems*, 11(5):461–477, 2020.

Pavel Izmailov, Sharad Vikram, Matthew D. Hoffman, and Andrew Gordon Wilson. What are bayesian neural network posteriors really like? *International conference on machine learning*, 2021.

Aviral Kumar and Sergey Levine. Model inversion networks for model-based optimization. *arXiv preprint arXiv:1912.13464*, 2019.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*, 2016.

Talley J Lambert. Fpbase: a community-editable fluorescent protein database. *Nature methods*, 16(4):277–278, 2019.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Ronny Lorenz, Stephan H Bernhart, Christian Höner Zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F Stadler, and Ivo L Hofacker. Viennarna package 2.0. *Algorithms for molecular biology*, 6(1):1–14, 2011.

Brandon Malone, Boris Simovski, Clément Moliné, Jun Cheng, Marius Gheorghe, Hugues Fontenelle, Ioannis Vardaxis, Simen Tennøe, Jenny-Ann Malmberg, Richard Stratford, et al. Artificial intelligence predicts the immunogenic landscape of sars-cov-2 leading to universal blueprints for vaccine designs. *Scientific reports*, 10(1):1–14, 2020.

Henry B Moss, Daniel Beck, Javier González, David S Leslie, and Paul Rayson. Boss: Bayesian optimization over string spaces. *arXiv preprint arXiv:2010.00979*, 2020.

F Ann Ran, Patrick D Hsu, Jason Wright, Vineeta Agarwala, David A Scott, and Feng Zhang. Genome engineering using the crispr-cas9 system. *Nature protocols*, 8(11):2281–2308, 2013.

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S Song. Evaluating protein transfer learning with tape. *Advances in Neural Information Processing Systems*, 32:9689, 2019.

Roshan Rao, Jason Liu, Robert Verkuil, Joshua Meier, John F Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. *bioRxiv*, 2021.

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. *bioRxiv*, 2019. doi: 10.1101/622803. URL https://www.biorxiv.org/content/10.1101/622803v4.

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

Gabriel J Rocklin, Tamuka M Chidyausiku, Inna Goreshnik, Alex Ford, Scott Houliston, Alexander Lemak, Lauren Carter, Rashmi Ravichandran, Vikram K Mulligan, Aaron Chevalier, et al. Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science*, 357(6347):168–175, 2017.

Yutaka Saito, Misaki Oikawa, Hikaru Nakazawa, Teppei Niide, Tomoshi Kameda, Koji Tsuda, and Mitsuo Umetsu.

Machine-learning-guided mutagenesis for directed evolution of fluorescent proteins. *ACS synthetic biology*, 7(9): 2014–2022, 2018.

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, et al. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603):397–401, 2016.

Amir Shanehsazzadeh, David Belanger, and David Dohan. Is transfer learning necessary for protein landscape prediction? *arXiv preprint arXiv:2011.03443*, 2020.

Jung-Eun Shin, Adam J Riesselman, Aaron W Kollasch, Conor McMahon, Elana Simon, Chris Sander, Aashish Manglik, Andrew C Kruse, and Debora S Marks. Protein design and variant prediction using autoregressive generative models. *Nature Communications*, 12(1):1–11, 2021.

Sam Sinai and Eric Kelsic. A primer on model-guided exploration of fitness landscapes for biological sequence design. *arXiv preprint arXiv:2010.10614*, 2020.

Lea M Starita, Jonathan N Pruneda, Russell S Lo, Douglas M Fowler, Helen J Kim, Joseph B Hiatt, Jay Shendure, Peter S Brzovic, Stanley Fields, and Rachel E Klevit. Activity-enhancing mutations in an e3 ubiquitin ligase identified by high-throughput mutagenesis. *Proceedings of the National Academy of Sciences*, 110(14):E1263–E1272, 2013.

Bruce J Wittmann, Kadina E Johnston, Zachary Wu, and Frances H Arnold. Advances in machine learning for directed evolution. *Current Opinion in Structural Biology*, 69:11–18, 2021.

Kevin K Yang, Zachary Wu, and Frances H Arnold. Machine-learning-guided directed evolution for protein engineering. *Nature methods*, 16(8):687–694, 2019.

Zikun Yang, Paul Bogdan, and Shahin Nazarian. An in silico deep learning approach to multi-epitope vaccine design: a sars-cov-2 case study. *Scientific reports*, 11(1): 1–21, 2021.

Method	LocalFL	Stability	E4B
DE (CNN)	$\mathbf{0.75 \pm 0.04}$	$\mathbf{0.84 \pm 0.03}$	$\mathbf{0.76 \pm 0.04}$
DE (RNN)	0.54 ± 0.03	0.59 ± 0.01	0.52 ± 0.05
DE (Emb.)	0.71 ± 0.01	0.57 ± 0.17	0.29 ± 0.12
GP (Emb.)	0.58 ± 0.07	0.56 ± 0.12	0.13 ± 0.08
GP (SSK)	0.51 ± 0.05	0.46 ± 0.05	0.07 ± 0.04

Table 2: Spearman's ρ between predicted and ground truth labels on test set when training on relatively small splits ($n = 500$) of the training data. Errors represent one standard deviation over 3 independent splits.

A. Detailed Wet-Lab Procedure

We synthesized fluorescent proteins with NEB PURExpress in vitro protein synthesis kits in 30 uL reactions from linear DNA purchased from IDT as eBlock dsDNA gene fragments. We ran reactions at 30C overnight in black, half-area microplates (Corning #3993) with optically clear plate adhesive and measured excitation and emission through a series of sweeps (fixing the excitation wavelength and scanning emission every 2 nm, or vice versa). We determined peak excitation and peak emission as the wavelength that gave maximum fluorescence units and calculated the Stokes shift as the difference in between the peaks.

B. Additional Comparisons

We also present results for regression on each of the datasets we describe in Section 3, demonstrating that deep ensembles yet again out-performing the competing approaches.

C. Implementation Details

```
def ConvBNswish(in_ch, out_ch, ksize=5, stride=1):
    return nn.Sequential(
        nn.Conv1d(in_ch, out_ch, ksize, padding=ksize//2, stride=stride),
        nn.BatchNorm1d(out_channels),
        Swish()
    )

class CNN(nn.Module):
    def __init__(self, dict_size=20,k=128,p=0.5):
        super().__init__()
        self.net = nn.Sequential(
            nn.Embedding(dict_size,k),
            Expression(lambda x: x.permute(0,2,1)),
            ConvBNswish(k,k),
            ConvBNswish(k,2*k),
            nn.MaxPool1d(2),
            nn.Dropout2d(p),
            ConvBNswish(2*k,2*k),
            ConvBNswish(2*k,2*k),
            nn.MaxPool1d(2),
            nn.Dropout2d(p),
            ConvBNswish(2*k,2*k),
            ConvBNswish(2*k,2*k),
            nn.Dropout2d(p),
            Expression(lambda u:u.mean(-1)),
            nn.Linear(2*k,1)
        )
    def forward(self,x):
        return self.net(x)[...,0]

class RNN(nn.Module):
    def __init__(self,dict_size=20,k=256, nlayer=2, bi=True):
        super().__init__()
        self.nlayer = nlayer
        self.gru = nn.GRU(k,k,nlayer, bidirectional=bi)
        self.embedding = nn.Embedding(dict_size,k)
        self.linear = nn.Linear(k*(1+bi),1)
        self.bi = bi

    def forward(self,x):
        x = self.embedding(x).permute(1,0,2)
        n,bs,k  = x.shape
        h0 = torch.zeros(self.nlayer*(1+self.bi),bs,k)
        out,hf = self.gru(x,h0)
        return self.linear(out.mean(0)).reshape(-1)
```

D. Remaining Evaluation Plots

We show max cumulative objective plots for the remaining datasets and tasks in Figure 5.

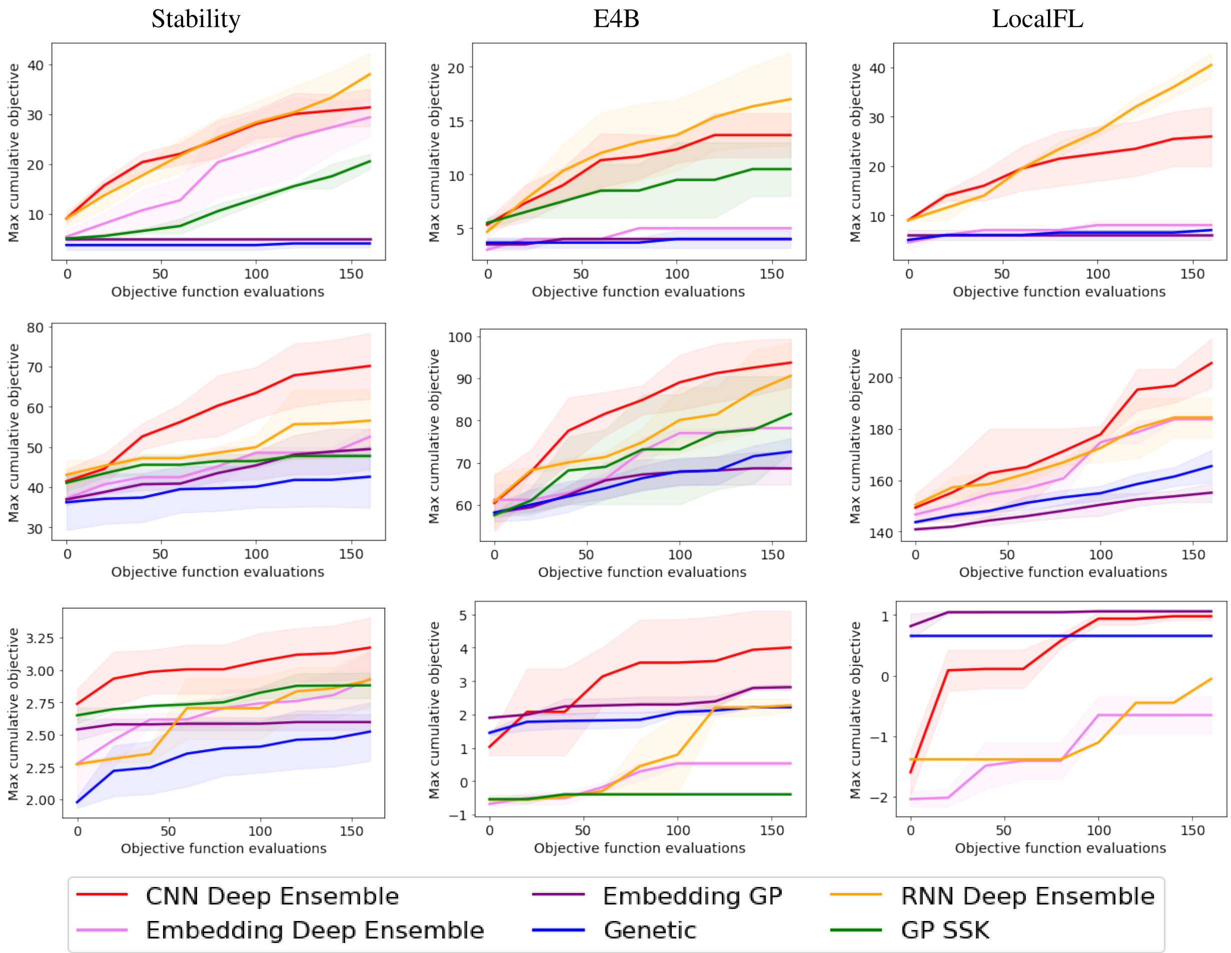

Figure 5: Cumulative max value of the objective function given a fixed number of function evaluations. (**Top**): Substring. (**Middle**): RNAFold. (**Bottom**): NNOracle. All Bayesian optimization methods are baselined against the genetic optimizer. Results for SSK GP on LocalFL are not shown because of computational overhead. Shaded regions show two standard deviations taken over 3 different seeds.