Tuned Fitness Landscapes for Benchmarking Model-Guided Protein Design

Neil Thomas[1*†], Atish Agarwala[2*], David Belanger[2], Yun S. Song[1,3‡], and Lucy J. Colwell[2,4‡]

[1] Computer Science Division, University of California, Berkeley
[2] Google Research, Brain Team
[3] Department of Statistics, University of California, Berkeley
[4] Department of Chemistry, University of Cambridge

Abstract. Advancements in DNA synthesis and sequencing technologies have enabled a novel paradigm of protein design where machine learning (ML) models trained on experimental data are used to guide exploration of a protein fitness landscape. ML-guided directed evolution (MLDE) builds on the success of traditional directed evolution and unlocks strategies which make more efficient use of experimental data. Building an MLDE pipeline involves many design choices across the design-build-test-learn loop ranging from data collection strategies to modeling, each of which has a large impact on the success of designed sequences. The cost of collecting experimental data makes benchmarking every component of these pipelines on real data prohibitively difficult, necessitating the development of *synthetic* landscapes where MLDE strategies can be tested. In this work, we develop a framework called SLIP ("Synthetic Landscape Inference for Proteins") for constructing biologically-motivated synthetic landscapes with tunable difficulty based on Potts models. This framework can be extended to any protein family for which there is a sequence alignment. We show that without tuning, Potts models are easy to optimize. In contrast, our tuning framework provides landscapes sufficiently challenging to benchmark MLDE pipelines. SLIP is open-source and is available at `https://github.com/google-research/slip`.

Keywords: protein engineering · machine learning · directed evolution · benchmark

[*] These authors contributed equally to this work.
[†] Work partially done during an internship at Google Research.
[‡] To whom correspondence should be addressed: `yss@berkeley.edu`, `lcolwell@google.com`

1 Introduction

Directed evolution (DE) [1–3] has revolutionized bioengineering, enabling the development of proteins with novel function across industries including food, chemicals, and therapeutics [2, 4, 5]. Part of the power of directed evolution is its simplicity: synthesize a set of variants, screen them for the desired function, induce mutagenesis in the top variants that passed the screen and repeat. This selective pressure pushes the variants towards the desired activity without requiring any prior knowledge of how mutations affect function. We can view DE as a genetic algorithm exploring a high-dimensional *landscape*, where higher points correspond to greater levels of desired activities. DE works under the assumption that functional landscapes are relatively smooth [3, 6], that is, combining high-performing mutations results in further improved variants.

With sufficient experimental throughput, DE can successfully climb smooth fitness landscapes. However, testing a set of variants for activity is a costly and time-consuming process, requiring specialized laboratory expertise. For example, experimental throughput to test the turnover rate of an enzyme can be on the order of tens of variants per round over a 10-round campaign [7]. Each inactive design comes at a great cost, not only because of the cost of the single negative result, but also comes at an opportunity cost as the inactive variant cannot be leveraged for designs in future rounds of DE. Thus, a practitioner wants to make efficient use of their experimental budget by carefully curating the designed variants. Guiding the designs can be done with *rational design*, which uses an understanding of the protein's structural and functional properties to avoid poor designs. However, these properties are often unknown: an arbitrary wildtype starting sequence may not have any associated structure [8], or any characterization besides its homology to other sequences. To gain the understanding that would enable rational design would require experimental characterization which could be even more costly than the rest of the protein design campaign.

Effectively exploring rough (i.e., not smooth) landscapes with limited experimental throughput is a challenge for DE [9–11]. Thankfully, the confluence of advancements in DNA synthesis (bespoke oligonucleotide sequences), DNA sequencing (high throughput screens), and machine learning have enabled a new paradigm of machine learning guided directed evolution (MLDE) [12–15] which can address the challenge of exploring rough landscapes with limited data. In each round of MLDE, a set of (variant, activity) pairs are collected, which a practitioner can use to train a genotype-phenotype model to predict the effect of variants. In the next round, that model can be used to propose candidates (see Figure 1 in [12]). MLDE has been successfully applied to designing proteins, such as AAV capsids with preferential delivery to specific organs in the body [16, 17], or increasing the fluorescence of GFP [18].

Despite early successes, there is no consensus on best practices for MLDE. There are many decisions involved in the design of an MLDE pipeline. What data should be collected? Which model class should be used? What optimization objective should be used for model training? How should models be selected? How should proposals be generated using a trained model? How does one trade off between "exploiting" model confidence and "exploring" regions of model uncertainty when synthesizing the next round of designs [19,20]? Each choice interacts nonlinearly with every other choice, complicating the meta-optimization problem.

Running experiments is expensive, so it is essential to test the pipeline before using it in a new design campaign. One approach is to use *empirical landscapes* from publicly available experimental datasets. However, these datasets are limited by the prohibitive cost of producing enough high-quality data to benchmark an MLDE pipeline across multiple rounds. Datasets like [13, 17, 21–23] curate the highest quality protein function datasets from Deep Mutational Scanning [24], but focus on one- or two-mutation regions around a wildtype sequence [25]. Other landscapes contain all possible multi-mutants, but only cover a small portion of the overall protein (4 positions of the binding domain (B1) of protein G [26]), or test a limited allele vocabulary [27].

In parallel with deeply characterized empirical landscapes, *synthetic landscapes* are being explored as testing grounds for MLDE [20, 28]. Synthetic landscapes are defined by a software function that can be queried for any sequence of interest [10, 19, 28–32]. In this paper, we propose a specific synthetic landscape with two key properties:

- Properties **grounded** in the statistics of real protein families.
- **Tunable, interpretable** difficulty to match a range of plausible optimization landscapes.

To obtain these properties, we introduce and validate an MLDE benchmarking framework called SLIP: "Synthetic Landscape Inference for Proteins." SLIP is a set of synthetic fitness landscapes based on Potts

models [33–37], combined with utilities for tuning the landscape difficulty. SLIP is open-source and is available at `https://github.com/google-research/slip`.

2 Background and Related Work

We define a *fitness landscape* $\mathcal{F}$ as a scalar-valued function over sequences $\mathbf{x}$ of length L with A alleles at each position. In practice, "fitness" is used to refer to a molecular phenotype (e.g., fluorescence) or organismal fitness (e.g., reproductive rate). Empirical landscapes measure the underlying fitness through a noisy observation process. Our synthetic fitness landscapes refer to the underlying, noiseless quantity.

We will consider a sequence design process which starts at "wildtype" $\mathbf{x}_0$, which has allele a_i at site i. The goal of the design process is to *maximize* the fitness (as opposed to the minimization of the loss function that occurs in traditional supervised learning). Therefore, we focus on the fitness *gain* $\mathcal{F}(\mathbf{x}) - \mathcal{F}(\mathbf{x}_0)$. A successfully designed sequence will, at a minimum, display $\mathcal{F}(\mathbf{x}) - \mathcal{F}(\mathbf{x}_0) > 0$.

2.1 Epistasis and Landscape Ruggedness

In order to test an MLDE pipeline, we need landscapes that cannot be effectively navigated without guidance from an ML model. Thus it is useful to tune the nonlinearity (or "difficulty") of a landscape. Two basic quantities we can use to understand the difficulty of task are 1) the *single-mutant effects* and 2) *pairwise epistasis*.

Single-mutant effects. Letting $\mathbf{x}_{i\beta}$ denote the sequence obtained by mutating the wildtype allele at site i to allele β, the single mutant effect $s_{i\beta}$ is defined as:

$$s_{i\beta} = \mathcal{F}(\mathbf{x}_{i\beta}) - \mathcal{F}(\mathbf{x}_0).$$

Landscapes with many *adaptive* single mutations (site-allele pairs where $s_{i\beta} > 0$) tend to be easier to optimize.

Pairwise epistasis. If the fitness of multi-mutants was given by the sum of single-mutant effects, then the landscape would be linear and easy to optimize – simply combining adaptive single mutants would lead to good sequences $\mathbf{x}$ with high values of $\mathcal{F}(\mathbf{x})$. Nonlinear interactions between mutations make optimization more difficult. These nonlinear effects are known as *epistasis*, a pervasive property in empirical landscapes [9,38–42]. For two mutations $a_i \to \beta$ at site i and $a_j \to \gamma$ at site j, we can define the *pairwise epistasis* $\epsilon_{i\beta,j\gamma}$ by

$$\epsilon_{i\beta,j\gamma} = \mathcal{F}(\mathbf{x}_{i\beta,j\gamma}) - \mathcal{F}(\mathbf{x}_0) - (s_{i\beta} + s_{j\gamma})\,.$$

In other words, pairwise epistasis is the part of the fitness difference between $\mathbf{x}_{i\beta,j\gamma}$ and $\mathbf{x}_0$ which cannot be explained by the single mutants. Deleterious epistatic interactions between adaptive single mutants (known as *reciprocal sign epistasis*) make it more difficult to combine good individual mutations to obtain good multi-mutants, confounding a linear model of the landscape.

Note that both pairwise epistasis and the single-mutant fitness differences must be defined relative to a reference sequence - in our case, the wildtype $\mathbf{x}_0$.

2.2 Synthetic Landscapes

Synthetic landscapes are designed so that $\mathcal{F}$ can be evaluated quickly for arbitrary sequences, even if computing all L^A values is prohibitive (in memory or time). Many of these landscapes were originally designed to study evolutionary processes [10, 32, 43].

More recently, synthetic landscapes have been applied to benchmark sequence design algorithms [20, 28, 31]. We can divide synthetic landscapes into four broad, overlapping classes: supervised neural models, biophysical models, random models, and graphical models, briefly described below.

Supervised neural models are trained on experimental data to predict a regression output [44–47]. Of particular note is the structural score given by AlphaFold2, which can be used as an optimization objective to find sequences likely to fold into a desired structure [46,48]. If the model is sufficiently good, model outputs can be used as a synthetic replacement for the experimental target. Neural models are fast to evaluate with a single forward pass. However, they can exhibit pathological behavior when used as optimization objectives, giving high scores to unrealistic sequence [49, 50] or giving outsize influence to irrelevant parts of the sequence [51]. While trained neural models can exhibit high levels of ruggedness [52], it is not straightforward to tune the optimization difficulty of a neural landscape.

Biophysical models explicitly model the energetic interactions in the protein. Prominent examples of biophysical models are ViennaRNA [44] and Rosetta [53], which provide a score for an input sequence representing the free energy of the folded structure at equilibrium. These models return globally characterized landscapes without unexpected pathologies. However, they require a computationally expensive optimization procedure to report the free energy minimum for each query sequence. Since physical assumptions like physical constants and potential energy functions are baked into the model, there is no principled tuning procedure to make the landscapes more difficult to optimize.

Random models generate a landscape by drawing a random function on the configuration space $\{1, \ldots, A\}^L$ with a particular distribution. The NK-model explicitly models order-K interactions for a sequence of length N to provide tunably rugged fitness landscapes which exhibit high-order correlations [29, 30]. The generalized NK-model, which explicitly models sparse blocks of interacting positions, has been shown to reflect the sparsity of empirical fitness function when conditioned on real structures [54]. Distance-dependent models [11] have interactions at all orders, and are defined by fixing a functional form for the covariance between sequences as a function of genetic distance. All of these models have been applied to study evolutionary dynamics, and are typically not fit to data.

Graphical models explicitly represent the interactions between positions in the sequence as edges in a graph. Profile HMMs do not model epistasis, and only model first-order interactions (i.e., amino acid distributions at aligned positions). Despite this, they serve as powerful protein family classifiers [55], and HMM likelihoods have been used as synthetic optimization objectives [28]. Profile HMMs can also flexibly handle insertions and deletions, and do not require aligned sequences as input. We describe in detail below a graphical model known as a Potts model.

2.3 Potts Models of Protein Families

Definition. For a family of proteins of length L with A possible alleles at each position, a Potts model defines a probability distribution over sequences in the family as

$$p(\mathbf{x}) = \frac{1}{Z} \exp\left(\mathcal{F}(\mathbf{x})\right),$$

where $\mathcal{F}$ is a negated statistical energy, and the partition function Z is a normalization constant such that the $p(\mathbf{x})$ sum to 1. In a Potts model, for an input one-hot encoded sequence $\mathbf{x} \in \mathbb{R}^{L \times A}$, $\mathcal{F}$ is given by the sum over the marginal effects and pairwise interactions:

$$\mathcal{F}(\mathbf{x}) = \sum_{i=1}^{L} \sum_{\alpha=1}^{A} h_{i\alpha} x_{i\alpha} + \frac{1}{2} \sum_{i,j=1}^{L} \sum_{\alpha,\beta=1}^{A} H_{i\alpha,j\beta} x_{i\alpha} x_{j\beta}$$

where $\mathbf{h} = (h_{i\alpha})$ is a tensor of dimension $L \times A$ representing marginal terms and $\mathbf{H} = (H_{i\alpha,j\beta})$ is a symmetric tensor of dimension $L \times L \times A \times A$ representing pairwise coupling terms. The parameters of a Potts model can be fit using a set of aligned sequences; see Supplementary Material A.2 for details.

Modeling coevolution. There has been extensive work establishing that Potts models learn statistics grounded in protein structure and function. Potts models are useful as unsupervised structure predictors [35–37, 56], and are competitive with neural unsupervised structure predictors [57] on families with large, diverse alignments. The statistical energy of protein variants scored by a Potts model has been shown to correlate well with empirical fitness [58]. When the statistical energy is included as an additional feature to a regression model, it has been shown to improve predictive performance on empirical landscapes [25]. Used as generative models, Potts models have been shown to propose functional variants of a given protein target [59]. Synthetic sequences evolved *in silico* on a Potts landscape have been shown to correlate with summary statistics with *in vitro* evolved sequences [60]. The parameters of the Potts model can also be used as input featurizations that improve performance on downstream tasks [61, 62].

3 Methods: Tuned Quadratic Landscapes

We can use the statistical energy of the Potts model as the fitness function $\mathcal{F}$ to define a synthetic landscape. In [28], for example, the authors introduced a "PDB-Ising" synthetic fitness landscape, which combined the contact map of a protein with standard pair potentials for amino acid substitution to create a simple *quadratic landscape* with pairwise interactions. We will instead derive model parameters from alignment data, which has the advantage that the resulting synthetic landscape exhibits correlations grounded in the coevolutionary couplings in that family.

While useful in their own right, Potts models derived from alignment data are not challenging synthetic landscapes, as they can be optimized by combining top mutations one at a time (akin to an *in silico* DE algorithm). We quantify these shortcomings in Section 5. To benchmark the performance of MLDE pipelines on more difficult optimization scenarios, we require synthetic landscapes where strategies guided by nonlinear models can substantially outperform those guided by linear models. In what follows, we develop a framework for tuning quadratic landscapes to a desired level of difficulty.

3.1 Tuning Model Statistics

For a fitness function defined by a Potts model, the single-mutant and pairwise epistasis terms can be written explicitly in terms of the model parameters $\mathbf{h}$ and $\mathbf{H}$:

$$s_{i\beta} = \mathbf{h}_{i\beta} - \mathbf{h}_{ia_i} + \sum_{j=1}^{L} \left(\mathbf{H}_{i\beta,ja_j} - \mathbf{H}_{ia_i,ja_j} \right),$$
$$\epsilon_{i\beta,j\gamma} = \mathbf{H}_{i\beta,j\gamma} - \mathbf{H}_{i\beta,ja_j} - \mathbf{H}_{ia_i,j\gamma} + \mathbf{H}_{ia_i,ja_j},$$

where again a_i is the allele of the wildtype $\mathbf{x}_0$ at site i. Note that single-mutant effect $s_{i\beta}$ depends on both the linear and quadratic parameters of the Potts model. See Supplementary Material B for a detailed derivation.

Once we reparameterize the Potts model in terms of the single-mutant and pairwise epistasis terms, the fitness decomposes into the form

$$\mathcal{F}(\mathbf{x}) - \mathcal{F}(\mathbf{x}_0) = \sum_{(i,\beta)\in M} s_{i\beta} + \frac{1}{2} \sum_{(i,\beta),(j,\gamma)\in M:(i,\beta)\neq(j,\gamma)} \epsilon_{i\beta,j\gamma},$$

where M is the set of mutations in $\mathbf{x}$ encoded as site-allele pairs (i, β).

Introducing shift (μ_s, μ_ϵ) and scale $(\lambda_s, \lambda_\epsilon)$ parameters for single-mutant and epistatic terms, we can parameterize a family of tuned fitness functions $\tilde{F}$ with parameters $\tilde{s}$ and $\tilde{\epsilon}$ given by

$$\tilde{s}_{i\beta} = \lambda_s(s_{i\beta} + \mu_s) \quad \text{and} \quad \tilde{\epsilon}_{i\beta,j\gamma} = \lambda_\epsilon(\epsilon_{i\beta,j\gamma} + \mu_\epsilon).$$

The four parameters $(\mu_s, \mu_\epsilon, \lambda_s, \lambda_\epsilon)$ allow for the mean and variance of both $\tilde{s}$ and $\tilde{\epsilon}$, taken over position-allele pairs, to be independently tuned. This allows us the flexibility of changing the difficulty of the landscape (e.g., by making epistasis more negative on average) while maintaining much of the structure of the original problem (e.g., preserving the coevolutionary couplings).

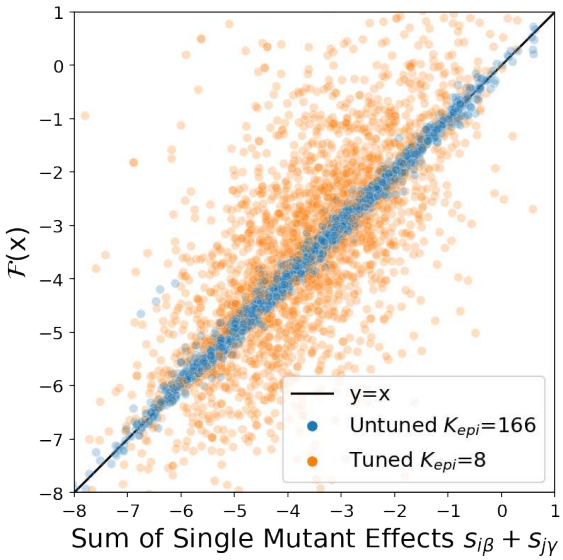


Fig. 1: Tuning the epistatic horizon increases the ruggedness of the resulting landscape. Fitness $\mathcal{F}(\mathbf{x})$ of 5000 variants with 2 mutations (centered so that $\mathcal{F}(\mathbf{x}_0) = 0$). The untuned fitness landscape with epistatic horizon $K_{\text{epi}} = 166$ (blue) is roughly linear, while the tuned fitness landscape with epistatic horizon $K_{\text{epi}} = 8$ (orange) exhibits more ruggedness. This landscape is derived from the alignment for PDB id 3er7. Note that the untuned epistatic horizon is greater than the length of the protein, $K_{\text{epi}} = 166 > 118 = L$.

3.2 Epistatic Horizon

For untuned landscapes, the single-mutant fitness effects approximate the double-mutant fitness effects well (Figure 1, blue), meaning the landscape is very linear. If combining random adaptive single mutants (mutations where $s_{i\beta} > 0$) is not a viable strategy, a naïve design strategy will struggle. For example, in Figure 1, points with $s_{i\beta} + s_{j\gamma} > 0$ but $\mathcal{F}(\mathbf{x}) - \mathcal{F}(\mathbf{x}_0) < 0$ (bottom right, in orange) would be candidate proposals by a naïve algorithm that would fail an experimental screen on a rugged landscape. One way to quantify the linearity (or non-linearity) of the landscape is to define an "epistatic horizon" K_{epi} – the number of mutations after which the linear approximation breaks down. With an eye towards tuning optimization difficulty, we define K_{epi} as follows.

Let $\bar{s}_+$ be the average of $s_{i\beta}$ over adaptive singles, and $\bar{\epsilon}_{+,+}$ the mean epistatic effect over random adaptive pairs. Then, the sequence design problem is difficult if the average interaction between individually good mutations is negative: $\bar{\epsilon}_{+,+} < 0$. Taking K random adaptive mutations, the average change in fitness for a K-mutant $\mathbf{x}_K$ is

$$\mathbb{E}_{\mathbf{x}_K}[\mathcal{F}(\mathbf{x}_K) - \mathcal{F}(\mathbf{x}_0)] = K\bar{s}_+ + \binom{K}{2}\bar{\epsilon}_{+,+}.$$

As K increases, the relative effect of epistasis grows relative to the single-mutant effects. We can compute a crossover value when $\mathbb{E}_{\mathbf{x}_K}[\mathcal{F}(\mathbf{x}_K) - \mathcal{F}(\mathbf{x}_0)] = 0$. Motivated by this example, we define the *epistatic horizon* K_{epi} as the non-zero solution to the equation:

$$K_{\text{epi}}\bar{s}_+ + \frac{K_{\text{epi}}(K_{\text{epi}} - 1)}{2}\bar{\epsilon}_{+,+} = 0. \tag{1}$$

This definition suggests two ways in which synthetic landscapes can fail to be difficult for MLDE:

- $K_{\text{epi}} > L$. In this case, the landscape is relatively linear at all relevant length scales. Combining adaptive single mutants leads to adaptive multi-mutants even for a large number of mutations.
- $K_{\text{epi}} < 0$. In this case, $\bar{\epsilon}_{+,+}$ is positive; that is, on average, adaptive single-mutants combine to be *better* than the sum of their parts. In this case, combining adaptive single-mutants also leads to adaptive multi-mutants, since typical pairs will interact positively with each other.

All the untuned landscapes we studied have $K_{\text{epi}} > L$, and many also have $K_{\text{epi}} < 0$; see Figure 1 and Supplementary Table S1. This means that the untuned landscapes are unsuitable for testing MLDE pipelines as is.

However, we can use tuning parameters to adjust K_{epi} and generate landscapes which are more non-linear and harder to optimize (Figure 1, orange). In the next section, we derive a tuning procedure which adjusts K_{epi} while leaving many other statistics of the fitness landscape fixed, thereby allowing us to benchmark MLDE pipelines.

3.3 Tuning Procedure

With four free parameters (two shift and two scale parameters), we can introduce additional constraints on the fitness landscape. First, we normalize the landscape such that the single-mutant effects have unit variance by setting $\lambda_s = \sigma(s)^{-1}$, where $\sigma(s)$ is the standard deviation of the $\{s_{i\beta}\}$. Second, we preserve the fraction of adaptive single mutants $\alpha_{s_+} \equiv \#\{s_{i\beta} > 0\}/L(A-1)$ by setting $\mu_s = 0$. Finally, to preserve the relative magnitudes (i.e., the ratios) of the pairwise epistatis terms, we set $\mu_\epsilon = 0$. This leaves λ_ϵ free. By fixing a target K_{epi}, we can use Equation (1) to solve for λ_ϵ.

To summarize: given a target epistatic horizon $K_{\text{epi}} > 0$, the tuning parameters are given by equations:

$$\mu_s = \mu_\epsilon = 0,$$
$$\lambda_s = \sigma(s)^{-1},$$

and

$$K_{\text{epi}}\lambda_s \bar{s}_+ + \frac{K_{\text{epi}}(K_{\text{epi}}-1)}{2}\lambda_\epsilon \bar{\epsilon}_{+,+} = 0,$$

where $\bar{s}_+$ and $\bar{\epsilon}_{+,+}$ are the values for the untuned landscape. Note that other tunings are possible for a given K_{epi}. This specific tuning scheme was chosen to preserve as much of the co-evolutionary structure of the Potts model as possible.

4 Methods: *In silico* Validation

To validate that our tuned synthetic landscapes are sufficiently difficult, we aim to create landscapes where nonlinear models outperform linear (naïve) models on sequence design tasks. To do so, we design an experimental framework which evaluates how effective linear and nonlinear models are at using training data to accurately rank design candidates.

We use the following procedure for each untuned landscape; see Figure 2 for a schematic:

1. Tune the epistatic horizon to $K_{\text{epi}} = 2^\ell$ for $\ell \in \{1, 2, \ldots, 10\}$ as in Section 3.3. Center at $\mathcal{F}(\mathbf{x}_0) = 0$.
2. Sample a dataset $D = \{(\mathbf{x}, y)\}$ of sequences $\mathbf{x}$ with their associated fitness y, where $|D| = 5000$. Each sample $(\mathbf{x}, y)$ is obtained by sampling a number of mutations from the wildtype $\mathbf{x}_0$ uniformly at random from $\{1, 2, 3\}$ and then sampling a variant uniformly at random at the selected distance.
3. Train Ridge and convolutional neural network (CNN) regression models across many different hyperparameter choices.
4. For the best performing model of each type, compute a paired performance metric on the evaluation set.

4.1 Untuned Landscapes

To select a suitable set of synthetic landscapes, we first initialize $\mathbf{h}$ and $\mathbf{H}$ by training Potts models on alignments of protein sequences. We choose protein targets to span a range of functions, structural folds, and primary sequence lengths, while ensuring the resulting Potts model has excellent contact accuracy on a high resolution structure. From the 748 Potts models trained in [63], the models corresponding to the 5 PDB IDs in Figure 3 were selected manually from the top performing models with respect to contact prediction accuracy. See Supplementary Figure S1 for predicted contact maps. We set the wildtype sequence $\mathbf{x}_0$ to be the alignment query sequence.

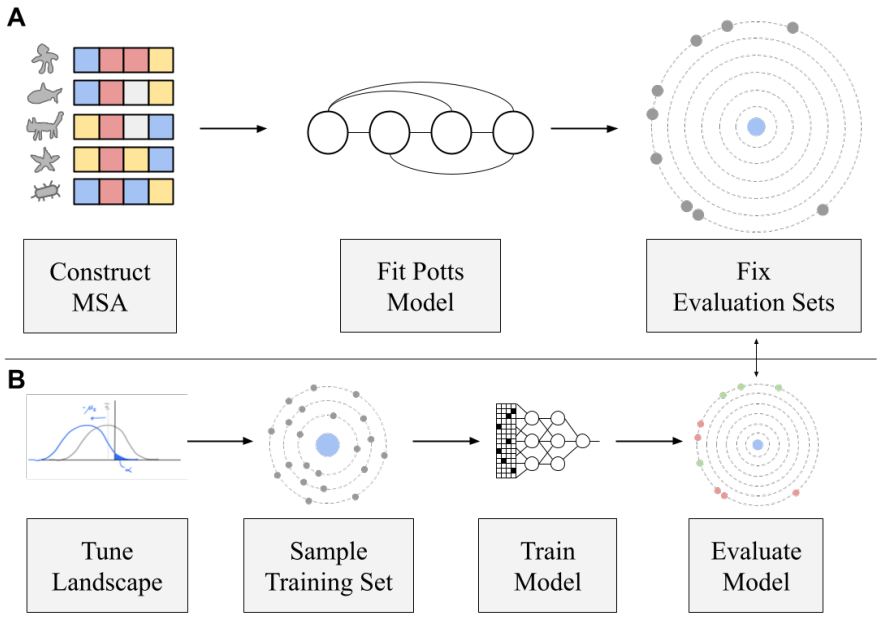


Fig. 2: *In silico* validation workflow. Panel A shows the tasks that are performed once for each PDB ID. After training a Potts model on an aligned set of sequences, we derive a set of evaluation sequences. Panel B shows the tasks that are performed for each replicate of the regression experiment: we tune the landscape, sample a set $(x, y) \in D$ of training sequences x with their associated synthetic fitnesses y, train a model on D, and then evaluate the model predictions on the evaluation sets. The evaluation sets are fixed for all landscapes derived from the same untuned Potts initialization.

4.2 Evaluation Sets

We choose an evaluation set relevant for the objective of sequence design. Combining top single mutations is a common strategy for proposing variants [5], and properly ranking these proposals is directly relevant to the objective of MLDE. For each (untuned) landscape, we construct evaluation sets at mutation distance 6 from the wildtype by taking the top 20 single mutants, combining them to construct variants at the desired distance, and then taking a random subset of the desired size (200). Because the set of top 20 single mutants does not change in response to tuning (i.e., single-mutant rankings are preserved by tuning), the evaluation set for a given PDB ID is fixed to the same set of 200 sequences (note that their fitness $\tilde{\mathcal{F}}(\mathbf{x})$ changes with tuning). See Supplementary Material A.3 for a discussion of other evaluation sets.

4.3 Models

Ridge regression. Our baseline linear model uses the `sklearn` implementation of Ridge regression, which has a single hyperparameter representing the L_2 penalty. The grid of hyperparameters used during model selection is reported in Supplementary Table S3. This is a strong baseline especially for landscapes where the level of epistasis is low. In addition, Hsu *et al.* [25] showed that the ridge penalty induces a powerful inductive bias that generalizes to unseen mutations by setting the effect of unseen mutations to the average effect seen at the same position. We remove the intercept term, since centering ensures $\tilde{\mathcal{F}}(\mathbf{x}_0) = 0$.

Convolutional Neural Network. Convolutional neural networks (CNNs) have been used to great success in protein sequence modeling [17, 64, 65], so we select them as our nonlinear model class. The CNN model architecture is 3 layers of 1D convolutions, followed by a dense layer. On 3er7 ($L = 118$), a CNN architecture with 32 filters, kernel size 5, and hidden size 64 results in a model with 255,329 parameters. The CNN is trained using an Adam optimizer [66] to minimize MSE loss. We tune the learning rate, number of filters, batch size, and number of training epochs. See Supplementary Table S4 for tuned hyperparameters and Supplementary Table S5 for fixed hyperparameters.

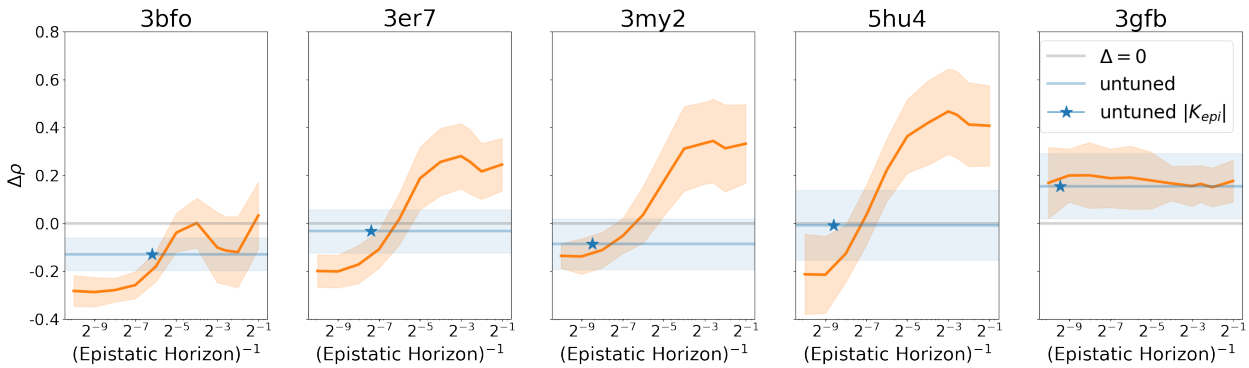


Fig. 3: Tuning the epistatic horizon interpolates between linear and non-linear model performance on ranking combinations of 6 adaptive singles after training on 5000 examples. The grey line shows $\Delta\rho = 0$ for ease of visualization of the threshold for one model outperforming the other. The orange line shows the difference between maximum CNN Spearman's ρ and maximum Ridge Spearman's ρ, with error bands showing ± 1 standard deviation across 20 random training set replicates. The x-axis corresponds to inverse epistatic horizon, so that more linear landscapes are to the left and more non-linear landscapes are to the right. The blue band shows model performance on the untuned landscape. The blue star shows the position on the x-axis which corresponds to the magnitude of the untuned epistatic horizon $|K_{\text{epi}}|$.

4.4 Paired Performance Metrics

An important driver of variability in evaluation performance are the training and evaluation sets. By keeping these fixed while allowing the model class to vary, we can isolate performance differences due to modeling capacity. For each sampled training set, we select the best Ridge model and the best CNN model in terms of ranking the given evaluation set. We then compare the computed performance metric on the given evaluation set and take the difference. "Differential Spearman ρ" (or $\Delta\rho$) refers to the difference, given a fixed training set, between the maximum CNN Spearman ρ and the maximum Ridge Spearman ρ on the evaluation set.

5 Results

5.1 Untuned Landscapes are Linear

In Figure 1, we plot the variant fitness as a function of the sum of constituent single-mutant effects. The untuned landscape fitness (in blue) follows a linear trend, where the fitness of a double-mutant can be well predicted by the sum of constituent single-mutant effects. Compared to the tuned landscape, the untuned landscape exhibits much less ruggedness. In Figure 3, the blue line corresponds to the differential performance of the CNN model compare to the Ridge model on the untuned landscape. Across all untuned landscapes except 3gfb, the CNN models have a mean performance change $\Delta\rho < 0$, i.e., that the CNN does not significantly boost performance on ranking combinations of adaptive singles compared to the Ridge model. This shows that untuned Potts models do not provide landscapes useful for benchmarking sequence design guided by nonlinear models, motivating the development of our tuning framework.

5.2 Epistatic Horizon Tunes the Nonlinearity of the Landscape

We aim to validate that our tuning framework can create fitness landscapes difficult enough to require nonlinear models. In Figure 3, for four of the five PDB IDs, as the epistatic horizon increases (to the left in the figure), evaluation set performance skews in favor of the Ridge model, confirming that as the landscape is tuned to be more linear, linear models are preferred to nonlinear models. Conversely, as the epistatic horizon decreases (to the right in the figure) and the landscape becomes dominated by nonlinear effects, evaluation set performance shifts to favor the CNN model. The intermediate-length proteins (3er7, 3my2,

5hu4 – the middle three panels of Figure 3), show consistent behavior, indicating that the epistatic horizon is a generalizable metric of landscape difficulty.

On the intermediate-length proteins 3er7, 3my2, and 5hu4, Spearman's ρ improves for the nonlinear model by between 0.2 and 0.4. For example, 3er7 shows an improvement in evaluation set performance from 0.1 to 0.3 (Supplementary Figure S2). For comparison, the authors in [12] found that zero-shot predictors on the 4-position GB1 landscape with a Spearman's ρ of 0.2 are sufficient to substantially improve sequence design. Across all landscapes, all models get worse with increased tuning, indicating that decreasing the epistatic horizon increases the landscape difficulty for nonlinear as well as linear models (see Supplementary Figure S2 for unpaired model performance). For horizons $K_{\text{epi}} \gg L$, the Ridge model achieves near perfect ranking accuracy $\rho \approx 1$ (see Supplementary Figure S2).

On 3gfb (far right in Figure 3), differential model performance remains roughly constant around $\Delta\rho = 0.2$ across all tested tunings. On 3bfo (the first panel of Figure 3), the shortest protein, differential model performance favors linear models for increased epistatic horizons, but does not achieve a regime where nonlinear models significantly outperform linear models. This may be due to CNN models overfitting on the short protein.

6 Discussion and Future Work

Using nonlinear models to guide sequence design has made a large impact in practice [13,14,17,67], but many questions still remain in regard to how to use these models as part of an MLDE pipeline. Our experimental results validate that our quadratic landscape tuning framework can generate synthetic landscapes which require nonlinear models for effective optimization. By deriving our landscapes from Potts models trained on real alignments, we ground the properties of our synthetic landscape in the structural and biochemical features of real proteins. These two properties enable tuned quadratic landscapes to be used to benchmark machine learning-guided protein design.

Our landscape tuning procedure relied on an optimization-motivated definition of the epistatic horizon K_{epi}. There are other scenarios where a more general definition for K_{epi} may be more relevant; for example, in a regression setting, the relevant crossover may be the point at which the *variance* in nonlinear fitness components is more than the variance of the linear components. Additionally, we focused on a tuning which increased the difficulty of combining adaptive mutants; there are other forms of nonlinearity which make ML-aided design more useful. One such situation is finding individually non-adaptive single mutants which combine to make adaptive mutants. Our landscape tuning framework is flexible enough to allow tuning of these types of properties.

Another frontier of ML for biological sequence design is making efficient use of labeled data with protein-specific priors provided, for example, by large language models [45,68,69]. There is room for model development to incorporate priors that allow sequence models such as CNNs to learn more nonlinear landscapes. By providing realistic datasets for model training, tuned quadratic landscapes are a useful sandbox environment for proposing modeling advancements that can take advantage of small datasets.

Optimizing MLDE pipelines involves more than tuning a neural network architecture. In MLDE, design choices ranging from training set curation to sequence proposal distributions can have a huge impact on the overall effectiveness of the pipeline. Benchmarking these choices against tuned quadratic landscapes would allow practitioners of MLDE to understand how to optimize their pipeline before having to collect expensive experimental data. Often, a new protein design campaign will have very specific constraints, such as assay-specific noise, or limitations on experimental throughput. Our tuned quadratic landscapes lend themselves easily to multiple extensions that allow an MLDE practitioner to impose these specific constraints and see how their pipeline performs. For a new design campaign with a sequence alignment, a bespoke synthetic landscape can be derived directly to match the structural constraints of the target. In addition, MLDE design choices can be correlated with landscape difficulty by testing across a range of tuning parameters. We hope that the benchmarks enabled by SLIP will further support the development of robust and efficient methods for biological sequence design.

Acknowledgments

This research is supported in part by an NIH grant R35-GM134922.

References

1. Frances H Arnold. Design by directed evolution. *Acc. Chem. Res.*, 31(3):125–131, March 1998.
2. Philip A Romero and Frances H Arnold. Exploring protein fitness landscapes by directed evolution. *Nat. Rev. Mol. Cell Biol.*, 10(12):866–876, December 2009.
3. Frances H Arnold. Nobel lecture: Innovation by evolution: Bringing new chemistry to life. `https://www.nobelprize.org/prizes/chemistry/2018/arnold/lecture/`, December 2018. Accessed: 2022-10-8.
4. S B Jennifer Kan, Russell D Lewis, Kai Chen, and Frances H Arnold. Directed evolution of cytochrome c for carbon-silicon bond formation: Bringing silicon to life. *Science*, 354(6315):1048–1051, November 2016.
5. John A McIntosh, Tamas Benkovics, Steven M Silverman, Mark A Huffman, Jongrock Kong, Peter E Maligres, Tetsuji Itoh, Hao Yang, Deeptak Verma, Weilan Pan, Hsing-I Ho, Jonathan Vroom, Anders M Knight, Jessica A Hurtak, Artis Klapars, Anna Fryszkowska, William J Morris, Neil A Strotman, Grant S Murphy, Kevin M Maloney, and Patrick S Fier. Engineered Ribosyl-1-Kinase enables concise synthesis of molnupiravir, an antiviral for COVID-19. *ACS Cent. Sci.*, October 2021.
6. Cara A Tracewell and Frances H Arnold. Directed enzyme evolution: climbing fitness peaks one amino acid at a time. *Curr. Opin. Chem. Biol.*, 13(1):3–9, February 2009.
7. Jonathan C Greenhalgh, Sarah A Fahlberg, Brian F Pfleger, and Philip A Romero. Machine learning-guided acyl-ACP reductase engineering for improved in vivo fatty alcohol production. *Nat. Commun.*, 12(1):5825, October 2021.
8. Nelson Perdigão, Julian Heinrich, Christian Stolte, Kenneth S Sabir, Michael J Buckley, Bruce Tabor, Beth Signal, Brian S Gloss, Christopher J Hammang, Burkhard Rost, Andrea Schafferhans, and Seán I O'Donoghue. Unexpected features of the dark proteome. *Proc. Natl. Acad. Sci. U. S. A.*, 112(52):15898–15903, December 2015.
9. Daniel M. Weinreich, Nigel F. Delaney, Mark A. Depristo, and Daniel L. Hartl. Darwinian evolution can follow only very few mutational paths to fitter proteins. *Science (New York, N.Y.)*, 312(5770):111–114, April 2006.
10. Johannes Neidhart, Ivan G. Szendro, and Joachim Krug. Adaptation in Tunably Rugged Fitness Landscapes: The Rough Mount Fuji Model. *Genetics*, 198(2):699–721, October 2014.
11. Atish Agarwala and Daniel S. Fisher. Adaptive walks on high-dimensional fitness landscapes and seascapes with distance-dependent statistics. *bioRxiv*, page 435669, February 2019.
12. Bruce J Wittmann, Yisong Yue, and Frances H Arnold. Informed training set design enables efficient machine learning-assisted directed protein evolution. *Cell Syst*, August 2021.
13. Kevin K Yang, Zachary Wu, and Frances H Arnold. Machine learning-guided directed evolution for protein engineering. Technical report, 2019.
14. Philip A Romero, Andreas Krause, and Frances H Arnold. Navigating the protein fitness landscape with gaussian processes. *Proc. Natl. Acad. Sci. U. S. A.*, 110(3):E193–201, January 2013.
15. Chase R Freschlin, Sarah A Fahlberg, and Philip A Romero. Machine learning to navigate fitness landscapes for protein engineering. *Curr. Opin. Biotechnol.*, 75:102713, April 2022.
16. Danqing Zhu, David H Brookes, Akosua Busia, Ana Carneiro, Clara Fannjiang, Galina Popova, David Shin, Edward F Chang, Tomasz J Nowakowski, Jennifer Listgarten, and David V Schaffer. Machine learning-based library design improves packaging and diversity of adeno-associated virus (AAV) libraries. November 2021.
17. Drew H Bryant, Ali Bashir, Sam Sinai, Nina K Jain, Pierce J Ogden, Patrick F Riley, George M Church, Lucy J Colwell, and Eric D Kelsic. Deep diversification of an AAV capsid protein by machine learning. *Nat. Biotechnol.*, February 2021.
18. Surojit Biswas, Grigory Khimulya, Ethan C Alley, Kevin M Esvelt, and George M Church. Low-N protein engineering with data-efficient deep learning. *Nat. Methods*, 18(4):389–396, April 2021.
19. Sam Sinai, Richard Wang, Alexander Whatley, Stewart Slocum, Elina Locane, and Eric D Kelsic. AdaLead: A simple and robust adaptive greedy search algorithm for sequence design. October 2020.
20. Sam Sinai and Eric D Kelsic. A primer on model-guided exploration of fitness landscapes for biological sequence design. October 2020.
21. Yuxiang Jiang, Tal Ronnen Oron, Wyatt T Clark, Asma R Bankapur, Daniel D'Andrea, Rosalba Lepore, Christopher S Funk, Indika Kahanda, Karin M Verspoor, Asa Ben-Hur, Da Chen Emily Koo, Duncan Penfold-Brown, Dennis Shasha, Noah Youngs, Richard Bonneau, Alexandra Lin, Sayed M E Sahraeian, Pier Luigi Martelli, Giuseppe Profiti, Rita Casadio, Renzhi Cao, Zhaolong Zhong, Jianlin Cheng, Adrian Altenhoff, Nives Skunca, Christophe Dessimoz, Tunca Dogan, Kai Hakala, Suwisa Kaewphan, Farrokh Mehryary, Tapio Salakoski, Filip Ginter, Hai Fang, Ben Smithers, Matt Oates, Julian Gough, Petri Törönen, Patrik Koskinen, Liisa Holm, Ching-Tai Chen, Wen-Lian Hsu, Kevin Bryson, Domenico Cozzetto, Federico Minneci, David T Jones, Samuel Chapman, Dukka Bkc, Ishita K Khan, Daisuke Kihara, Dan Ofer, Nadav Rappoport, Amos Stern, Elena Cibrian-Uhalte, Paul Denny, Rebecca E Foulger, Reija Hieta, Duncan Legge, Ruth C Lovering, Michele Magrane, Anna N Melidoni, Prudence Mutowo-Meullenet, Klemens Pichler, Aleksandra Shypitsyna, Biao Li, Pooya Zakeri, Sarah ElShal, Léon-Charles Tranchevent, Sayoni Das, Natalie L Dawson, David Lee, Jonathan G Lees, Ian Sillitoe, Prajwal

Bhat, Tamás Nepusz, Alfonso E Romero, Rajkumar Sasidharan, Haixuan Yang, Alberto Paccanaro, Jesse Gillis, Adriana E Sedeño-Cortés, Paul Pavlidis, Shou Feng, Juan M Cejuela, Tatyana Goldberg, Tobias Hamp, Lothar Richter, Asaf Salamov, Toni Gabaldon, Marina Marcet-Houben, Fran Supek, Qingtian Gong, Wei Ning, Yuanpeng Zhou, Weidong Tian, Marco Falda, Paolo Fontana, Enrico Lavezzo, Stefano Toppo, Carlo Ferrari, Manuel Giollo, Damiano Piovesan, Silvio C E Tosatto, Angela Del Pozo, José M Fernández, Paolo Maietta, Alfonso Valencia, Michael L Tress, Alfredo Benso, Stefano Di Carlo, Gianfranco Politano, Alessandro Savino, Hafeez Ur Rehman, Matteo Re, Marco Mesiti, Giorgio Valentini, Joachim W Bargsten, Aalt D J van Dijk, Branislava Gemovic, Sanja Glisic, Vladmir Perovic, Veljko Veljkovic, Nevena Veljkovic, Danillo C Almeida-E-Silva, Ricardo Z N Vencio, Malvika Sharan, Jörg Vogel, Lakesh Kansakar, Shanshan Zhang, Slobodan Vucetic, Zheng Wang, Michael J E Sternberg, Mark N Wass, Rachael P Huntley, Maria J Martin, Claire O'Donovan, Peter N Robinson, Yves Moreau, Anna Tramontano, Patricia C Babbitt, Steven E Brenner, Michal Linial, Christine A Orengo, Burkhard Rost, Casey S Greene, Sean D Mooney, Iddo Friedberg, and Predrag Radivojac. An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biol.*, 17(1):184, September 2016.

22. Vanessa E Gray, Ronald J Hause, Jens Luebeck, Jay Shendure, and Douglas M Fowler. Quantitative missense variant effect prediction using Large-Scale mutagenesis data. *Cell Syst*, 6(1):116–124.e3, January 2018.
23. Christian Dallago, Jody Mou, Kadina Elizabeth Johnston, Bruce Wittmann, Nick Bhattacharya, Samuel Goldman, Ali Madani, and Kevin K Yang. FLIP: Benchmark tasks in fitness landscape inference for proteins. August 2021.
24. Douglas M Fowler and Stanley Fields. Deep mutational scanning: a new style of protein science. *Nat. Methods*, 11(8):801–807, August 2014.
25. Chloe Hsu, Hunter Nisonoff, Clara Fannjiang, and Jennifer Listgarten. Learning protein fitness models from evolutionary and assay-labeled data. *Nat. Biotechnol.*, January 2022.
26. Nicholas C Wu, Lei Dai, C Anders Olson, James O Lloyd-Smith, and Ren Sun. Adaptation in protein fitness landscapes is facilitated by indirect paths. *Elife*, 5, July 2016.
27. Oksana M Subach, Vladimir N Malashkevich, Wendy D Zencheck, Kateryna S Morozova, Kiryl D Piatkevich, Steven C Almo, and Vladislav V Verkhusha. Structural characterization of acylimine-containing blue and red chromophores in mTagBFP and TagRFP fluorescent proteins. *Chem. Biol.*, 17(4):333–341, April 2010.
28. Christof Angermueller, David Dohan, David Belanger, Ramya Deshpande, Kevin Murphy, and Lucy Colwell. Model-based reinforcement learning for biological sequence design. September 2019.
29. E D Weinberger. Local properties of kauffman's n-k model: A tunably rugged energy landscape. *Phys. Rev. A*, 44(10):6399–6413, November 1991.
30. Sungmin Hwang, Benjamin Schmiegelt, Luca Ferretti, and Joachim Krug. Universality classes of interaction structures for NK fitness landscapes. *J. Stat. Phys.*, 172(1):226–278, July 2018.
31. 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. June 2020.
32. Atish Agarwala and Daniel S Fisher. Adaptive walks on high-dimensional fitness landscapes and seascapes with distance-dependent statistics. *Theor. Popul. Biol.*, 130:13–49, December 2019.
33. R B Potts. Some generalized order-disorder transformations. *Math. Proc. Cambridge Philos. Soc.*, 48(1):106–109, January 1952.
34. A S Lapedes, B G Giraud, L C Liu, and G D Stormo. Correlated mutations in protein sequences: Phylogenetic and structural effects. Technical report, December 1998.
35. Debora S Marks, Lucy J Colwell, Robert Sheridan, Thomas A Hopf, Andrea Pagnani, Riccardo Zecchina, and Chris Sander. Protein 3D structure computed from evolutionary sequence variation. *PLoS One*, 6(12):e28766, December 2011.
36. Hetunandan Kamisetty, Sergey Ovchinnikov, and David Baker. Assessing the utility of coevolution-based residue-residue contact predictions in a sequence- and structure-rich era. *Proc. Natl. Acad. Sci. U. S. A.*, 110(39):15674–15679, September 2013.
37. Sergey Ovchinnikov, Lisa Kinch, Hahnbeom Park, Yuxing Liao, Jimin Pei, David E Kim, Hetunandan Kamisetty, Nick V Grishin, and David Baker. Large-scale determination of previously unsolved protein structures using evolutionary information. *Elife*, 4:e09248, September 2015.
38. H Kacser and J A Burns. The molecular basis of dominance. *Genetics*, 97(3-4):639–666, March 1981.
39. Lizhi Ian Gong, Marc A Suchard, and Jesse D Bloom. Stability-mediated epistasis constrains the evolution of an influenza protein. *Elife*, 2:e00631, May 2013.
40. Chuan Li, Wenfeng Qian, Calum J Maclean, and Jianzhi Zhang. The fitness landscape of a tRNA gene. *Science*, 352(6287):837–840, May 2016.
41. Jakub Otwinowski, David M McCandlish, and Joshua B Plotkin. Inferring the shape of global epistasis. *Proc. Natl. Acad. Sci. U. S. A.*, 115(32):E7550–E7558, August 2018.
42. Richard A. Neher and Boris I. Shraiman. Competition between recombination and epistasis can cause a transition from allele to genotype selection. *Proceedings of the National Academy of Sciences*, 106(16):6866–6871, April 2009.

43. Stuart A. Kauffman and Edward D. Weinberger. The NK model of rugged fitness landscapes and its application to maturation of the immune response. *Journal of Theoretical Biology*, 141(2):211–245, November 1989.
44. 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 Mol. Biol.*, 6:26, November 2011.
45. Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S Song. Evaluating protein transfer learning with TAPE. *Adv. Neural Inf. Process. Syst.*, 32:9689–9701, December 2019.
46. John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin **v**Zídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, pages 1–11, July 2021.
47. Christoffer Norn, Basile I M Wicky, David Juergens, Sirui Liu, David Kim, Brian Koepnick, Ivan Anishchenko, Foldit Players, David Baker, and Sergey Ovchinnikov. Protein sequence design by explicit energy landscape optimization. July 2020.
48. Ziyue Yang, Katarina A Milas, and Andrew D White. Now what sequence? pre-trained ensembles for bayesian optimization of protein sequences. August 2022.
49. Nathan Killoran, Leo J Lee, Andrew Delong, David Duvenaud, and Brendan J Frey. Generating and designing DNA with deep generative models. December 2017.
50. Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks. `https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html`, June 2015. Accessed: 2022-10-8.
51. Brandon Carter, Maxwell Bileschi, Jamie Smith, Theo Sanderson, Drew Bryant, David Belanger, and Lucy J Colwell. Critiquing protein family classification models using sufficient input subsets. *J. Comput. Biol.*, December 2019.
52. Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. June 2017.
53. Rebecca F Alford, Andrew Leaver-Fay, Jeliazko R Jeliazkov, Matthew J O'Meara, Frank P DiMaio, Hahnbeom Park, Maxim V Shapovalov, P Douglas Renfrew, Vikram K Mulligan, Kalli Kappel, Jason W Labonte, Michael S Pacella, Richard Bonneau, Philip Bradley, Roland L Dunbrack, Jr, Rhiju Das, David Baker, Brian Kuhlman, Tanja Kortemme, and Jeffrey J Gray. The rosetta All-Atom energy function for macromolecular modeling and design. *J. Chem. Theory Comput.*, 13(6):3031–3048, June 2017.
54. David H Brookes, Amirali Aghazadeh, and Jennifer Listgarten. On the sparsity of fitness functions and implications for learning. *Proc. Natl. Acad. Sci. U. S. A.*, 119(1), January 2022.
55. Sean R Eddy. Accelerated profile HMM searches. *PLoS Comput. Biol.*, 7(10):e1002195, October 2011.
56. Sivaraman Balakrishnan, Hetunandan Kamisetty, Jaime G Carbonell, Su-In Lee, and Christopher James Langmead. Learning generative models for protein fold families. *Proteins*, 79(4):1061–1078, April 2011.
57. Roshan Rao, Joshua Meier, Tom Sercu, Sergey Ovchinnikov, and Alexander Rives. Transformer protein language models are unsupervised structure learners. December 2020.
58. 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. *Nat. Biotechnol.*, 35(2):128–135, February 2017.
59. William P Russ, Matteo Figliuzzi, Christian Stocker, Pierre Barrat-Charlaix, Michael Socolich, Peter Kast, Donald Hilvert, Remi Monasson, Simona Cocco, Martin Weigt, and Rama Ranganathan. Evolution-based design of chorismate mutase enzymes. April 2020.
60. Matteo Bisardi, Juan Rodriguez-Rivas, Francesco Zamponi, and Martin Weigt. Modeling sequence-space exploration and emergence of epistatic signals in protein evolution. June 2021.
61. Jianzhu Ma, Sheng Wang, Zhiyong Wang, and Jinbo Xu. Protein contact prediction by integrating joint evolutionary coupling analysis and supervised learning. *Bioinformatics*, 31(21):3506–3513, November 2015.
62. Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin **v**Zídek, Alexander W R Nelson, Alex Bridgland, Hugo Penedones, Stig Petersen, Karen Simonyan, Steve Crossan, Pushmeet Kohli, David T Jones, David Silver, Koray Kavukcuoglu, and Demis Hassabis. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, January 2020.
63. Nicholas Bhattacharya, Neil Thomas, Roshan Rao, Justas Dauparas, Peter K Koo, David Baker, Yun S Song, and Sergey Ovchinnikov. Interpreting potts and transformer protein models through the lens of simplified attention. *Pac. Symp. Biocomput.*, 27:34–45, 2022.

64. Maxwell L Bileschi, David Belanger, Drew H Bryant, Theo Sanderson, Brandon Carter, D Sculley, Alex Bateman, Mark A DePristo, and Lucy J Colwell. Using deep learning to annotate the protein universe. *Nat. Biotechnol.*, 40(6):932–937, June 2022.
65. Kevin K Yang, Alex X Lu, and Nicolo Fusi. Convolutions are competitive with transformers for protein sequence pretraining. May 2022.
66. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. December 2014.
67. Danqing Zhu, David H Brookes, Akosua Busia, Ana Carneiro, Clara Fannjiang, Galina Popova, David Shin, Kevin C Donohue, Edward F Chang, Tomasz J Nowakowski, Jennifer Listgarten, and David V Schaffer. Optimal trade-off control in machine learning-based library design, with application to adeno-associated virus (AAV) for gene therapy. September 2022.
68. Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rihawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Debsindhu Bhowmik, and Burkhard Rost. ProtTrans: Towards cracking the language of life's code through Self-Supervised deep learning and high performance computing. July 2020.
69. 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. *Proc. Natl. Acad. Sci. U. S. A.*, 118(15), April 2021.
70. Joerg Schaarschmidt, Bohdan Monastyrskyy, Andriy Kryshtafovych, and Alexandre M J J Bonvin. Assessment of contact predictions in CASP12: Co-evolution and deep learning coming of age. *Proteins*, 86 Suppl 1:51–66, March 2018.
71. Rojan Shrestha, Eduardo Fajardo, Nelson Gil, Krzysztof Fidelis, Andriy Kryshtafovych, Bohdan Monastyrskyy, and Andras Fiser. Assessing the accuracy of contact predictions in CASP13. *Proteins*, 87(12):1058–1068, December 2019.
72. Sergey Ovchinnikov, Hetunandan Kamisetty, and David Baker. Robust and accurate prediction of residue-residue interactions across protein interfaces using evolutionary information. *Elife*, 3:e02030, May 2014.
73. Justas Dauparas, Haobo Wang, Avi Swartz, Peter Koo, Mor Nitzan, and Sergey Ovchinnikov. Unified framework for modeling multivariate distributions in biological sequences. June 2019.
74. S D Dunn, L M Wahl, and G B Gloor. Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. *Bioinformatics*, 24(3):333–340, February 2008.

Supplementary Material

A Potts Models and Evaluation Sets

A.1 Landscape Details

PDB	Length	# Seqs	Precision @ L	Epistatic Horizon	α_{s_+}	p_ϵ	$\bar{\epsilon}_{+,+}$	$\sigma(s)$
3bfo	85	12153	0.78	-72.3	0.0099	0.42	0.013	2.22
3er7	118	33672	0.71	166.6	0.036	0.51	-0.0087	2.15
3my2	126	5497	0.82	-358.4	0.011	0.48	0.0049	3.01
5hu4	145	6440	0.81	-389.3	0.016	0.36	0.0052	2.87
3gfb	347	13554	0.76	-689.7	0.041	0.46	0.0027	2.91

Table S1: Untuned landscape details. Contact precision is computed in the standard way: predicting the top L entries (>6 apart in the primary sequence) in $\mathbf{H}$ to be contacts, and computing precision [70,71]. α_{s_+} refers to the fraction of adaptive singles with effect $s_{i\beta} > 0$. p_ϵ refers to the fraction of reciprocal sign epistasis for pairs of adaptive singles. $p_\epsilon = \frac{\#\{\epsilon_{+,+} < 0\}}{\#\{\epsilon_{+,+}\}}$

PDB	Functional Keywords
3bfo	Immunoglobulin-like beta sandwich
3er7	Nuclear transport factor
3my2	Transmembrane protein
5hu4	Prokaryotic Sortase
3gfb	L-threonine Dehydrogenase

Table S2: Functional keywords associated with the selected PDBs.

A.2 Fitting Potts Models

The initial training of the Potts model involves sampling batches from an alignment X. We train the model to maximize the regularized pseudolikelihood objective

$$Loss(\mathbf{h}, \mathbf{H}; X) = \mathcal{L}(\mathbf{h}, \mathbf{H}; X) + R(\mathbf{h}, \mathbf{H}),$$

where the regularization term is given by

$$R(\mathbf{h}, \mathbf{H}) = \frac{1}{2}\lambda AL||\mathbf{H}||_F^2 + \lambda||\mathbf{h}||_F^2,$$

following the scaling procedure in [72]. The Potts model is trained using a modified version of Adam [66] presented in Dauparas, et al. [73], modified to improve performance of Adam to match that of L-BFGS, using batches X_b from the overall alignment X. Before computing any forward passes, we symmetrize $\mathbf{H}$ and mask the diagonal. All models were trained using $\lambda = 0.01$, Adam learning rate 0.5, and batch size 128. Training was done for 200 steps on a NVIDIA GeForceRTX 2080 Ti GPU. The training script can be found at `https://github.com/songlab-cal/factored-attention`. Models are trained using the "use-bias" flag to explicitly include $\mathbf{h}$. From the 748 potts models trained in Bhattacharya et al. [63], the 5 PDB ids listed

Ridge Hyperparameter	Grid
L2 penalty (α)	10^x for $x \in \{-3, -2.5, -2, ..., 1.0, 1.5, 2.0\}$

Table S3: Tuned hyperparameters for the Ridge model. (Grid size: 11)

CNN Hyperparameter	Grid
Learning Rate (Adam)	10^x for $x \in \{-3, -2.9, -2.8, ..., 2.0\}$
Batch Size	$\{64, 128\}$
Num Training Epochs	$\{100, 500, 1000\}$
Num filters	$\{16, 32, 64\}$

Table S4: Tuned hyperparameters for the CNN model. (Grid size: 198)

in Table S1 were selected manually from the top performing models with respect to contact precision @ L. Note that the deeper the alignment, the more robust the Potts model training is to hyperparameter choices.

CNN Hyperparameter	Fixed Value
Kernel Size	5
Hidden Size	64
Activation Function	ReLU
Dropout probability	25%

Table S5: Fixed hyperparameters for the CNN model.

A.3 Epistasis-enriched evaluation sets

In Section 4.2 we describe an evaluation set based on combining multiple adaptive singles into multi-mutants. In this section we describe two additional evaluation sets for each landscape based on enrichment for epistasis. The motivation for selecting variants with large magnitude epistatic effects is to confound the linear model, and test the nonlinear model's ability to learn epistasis from a training set of multi-mutants. We build two evaluation sets: Adaptive Epistasis and Deleterious Epistasis. For both sets the procedure for constructing them is the same, but with the sign of the epistatic terms inverted.

We construct the full $L \times L \times A \times A$ tensor of epistatic terms $\epsilon_{i\beta,j\gamma}$. Ranking these terms by their value, we pick out the highest 1000 terms (for Deleterious Epistasis, we pick out the lowest 1000). From this set of strong epistatic interactions, we construct a pool of site-allele pairs: $M = \{(i\beta_1, j\gamma_1), \ldots, (i\beta_{1000}, j\gamma_{1000})\}$. From this pool of pairs M we construct variants at the desired distance from the wildtype. For an evaluation set at distance 6, we choose 3 pairs at random and combine them. We continue to sample combinations of epistatic pairs until we have an evaluation set of the desired size $n = 200$. Note that some site-allele pairs conflict with one another by mutating the same position. We discard combinations that do not reach the desired distance.

B Quadratic Landscape Theory

In the following, we develop the basic theory for quadratic landscapes in detail. We will derive a tuning scheme which allows us to separately control the distribution of single mutant fitness effects double mutant fitness effects. This will allow us to *tune* landscapes in order to explore different regimes of the overall optimization space.

We will be interested in understanding fitness landscapes defined on sequence space. We will consider sequences of length L on A characters, encoded by vectors $\mathbf{x}$ of length LA, which are one-hot every A

elements. (In computational settings, the sequence is often represented as an $L \times A$ matrix, one-hot in the second index.)

B.1 Quadratic fitness function

Consider a fitness function $\mathcal{F}$ given by

$$\mathcal{F}(\mathbf{x}) = \mathbf{h}^{\mathrm{T}}\mathbf{x} + \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{H}\mathbf{x}.$$

Here $\mathbf{H}$ is an $LA \times LA$ symmetric coupling matrix, and $\mathbf{h}$ is a length LA vector. Note that in computational settings, $\mathbf{H}$ is often implemented as a tensor of dimension $L \times L \times A \times A$, and $\mathbf{h}$ as a tensor of dimension $L \times A$.

We note that in biological applications, we generally care about fitness differences; fitness functions which differ by a constant value are considered to be the same.

Given the L-hot structure of $\mathbf{x}$, we see that the L distinct $A \times A$-dimensional subblocks of $\mathbf{H}$ corresponding to within-site interactions are special. In particular, only the diagonal terms contribute to $\mathcal{F}$, since $\mathbf{x}$ is one-hot within a block. Due to this one-hot structure, without loss of generality we can absorb the within-site interactions into the linear term by setting $\hat{\mathbf{h}}_{i\alpha} = \mathbf{h}_{i\alpha} + \frac{1}{2}\mathbf{H}_{i\alpha,i\alpha}$, and $\hat{\mathbf{H}}_{i\alpha,i\beta} = 0$ for each site i and characters α and β, and $\hat{\mathbf{H}} = \mathbf{H}$ otherwise. The functional form of our fitness is unchanged:

$$\mathcal{F}(\mathbf{x}) = \hat{\mathbf{h}}^{\mathrm{T}}\mathbf{x} + \frac{1}{2}\mathbf{x}^{\mathrm{T}}\hat{\mathbf{H}}\mathbf{x}.$$

Thus, for the remainder of the notes we guarantee, without loss of generality, that in addition to being symmetric, $\mathbf{H}$ has 0 diagonal, i.e. $\mathbf{H}_{i\alpha,i\beta} = 0$.

B.2 Local statistics

We are interested in the statistics near a particular sequence $\mathbf{x}_0$, which we call the "wildtype" sequence. For example, in enzyme design, we often start with a wildtype sequence which can be used in a reaction of interest, and the goal of the optimization is to arrive at a designed sequence which carries out the reaction more efficiently.

Without loss of generality, for the remainder of the notes we will refer to the wildtype sequence with a generic character $\mathbf{x}_0(i) = a$ at all positions i. We often consider the relative fitness $\mathcal{F}(\mathbf{x}) - \mathcal{F}(\mathbf{x}_0)$ rather than the absolute fitness $\mathcal{F}(\mathbf{x})$. The quadratic model is defined by two quantities: the single mutant fitness effects s and the *pairwise epistasis* ϵ. The single mutant fitness effects are defined by $s(\mathbf{x}_1) = \mathcal{F}(\mathbf{x}_1) - \mathcal{F}(\mathbf{x}_0)$, where $\mathbf{x}_1$ is a single mutant which differs from $\mathbf{x}_0$ in exactly one of the L positions. We can write out the effect explicitly. Let $\mathbf{x}_0(i)$ denote the character at position i in the wildtype sequence $\mathbf{x}_0$. Consider a mutation $s_{i\beta}$ at site i, which takes character $\mathbf{x}_0(i) = a$ to character β. We have:

$$s_{i\beta} = \mathbf{h}_{i\beta} - \mathbf{h}_{ia} + \sum_{j=1}^{L} \left(\mathbf{H}_{i\beta,ja} - \mathbf{H}_{ia,ja}\right).$$

In many cases, these linear effects are enough to begin to design sequences and optimize over the fitness landscape. Note that this linear structure depends on both $\mathbf{h}$ and $\mathbf{H}$.

The higher order interactions can be quantified using *pairwise epistasis*. In general, the term epistasis is used by geneticists to refer to interaction between the effects of multiple mutations. There are many ways to quantify these interactions. We focus on a definition of pairwise epistasis which measures the deviation from linearity of a landscape.

Given a double mutant $\mathbf{x}_{12}$, with single mutant sequences given by $\mathbf{x}_1$ and $\mathbf{x}_2$, we define the *pairwise epistasis* $\epsilon(\mathbf{x}_{12})$ by

$$\epsilon(\mathbf{x}_{12}) = \mathcal{F}(\mathbf{x}_{12}) - \mathcal{F}(\mathbf{x}_1) - \mathcal{F}(\mathbf{x}_2) + \mathcal{F}(\mathbf{x}_0).$$

This definition can be motivated by re-writing as

$$\begin{aligned}\epsilon(\mathbf{x}_{12}) &= \mathcal{F}(\mathbf{x}_{12}) - \mathcal{F}(\mathbf{x}_0) - (\mathcal{F}(\mathbf{x}_1) - \mathcal{F}(\mathbf{x}_0) + \mathcal{F}(\mathbf{x}_2) - \mathcal{F}(\mathbf{x}_0)) \\ &= \mathcal{F}(\mathbf{x}_{12}) - \mathcal{F}(\mathbf{x}_0) - (s(\mathbf{x}_1) + s(\mathbf{x}_2))\,.\end{aligned}$$

In other words, it's the part of the fitness difference between $\mathbf{x}_{12}$ and $\mathbf{x}_0$ which can't be explained by the single mutants $\mathbf{x}_1$ and $\mathbf{x}_2$.

If the two mutations are $a \to \beta$ at site i and $a \to \gamma$ at site j, then the epistasis $\epsilon_{i\beta,j\gamma}$ is given by

$$\epsilon_{i\beta,j\gamma} = \mathbf{H}_{i\beta,j\gamma} - \mathbf{H}_{i\beta,ja} - \mathbf{H}_{ia,j\gamma} + \mathbf{H}_{ia,ja}.$$

B.3 Difference expansion

It is useful to explicitly write the fitness difference $\mathcal{F}(\mathbf{x}) - \mathcal{F}(\mathbf{x}_0)$ for some general $\mathbf{x}$, in order to understand and manipulate local statistics. Let $M = \{m_1, m_2, ...m_k\}$ be the sequence of k mutations from wildtype in $\mathbf{x}$, where $m_l = (i_l, \beta_l)$. No two mutations affect the same position, so $i_l \neq i_{l'}$ for $l \neq l'$. We have

$$\mathcal{F}(\mathbf{x}) - \mathcal{F}(\mathbf{x}_0) = \mathbf{h}^{\mathrm{T}}\mathbf{x} + \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{H}\mathbf{x} - \mathbf{h}^{\mathrm{T}}\mathbf{x}_0 - \frac{1}{2}\mathbf{x}_0^{\mathrm{T}}\mathbf{H}\mathbf{x}_0$$

We can rewrite this in terms of the sequence difference $\boldsymbol{\delta} \equiv \mathbf{x} - \mathbf{x}_0$. We have

$$\mathcal{F}(\mathbf{x}) - \mathcal{F}(\mathbf{x}_0) = (\mathbf{h}^{\mathrm{T}} + \mathbf{x}_0^{\mathrm{T}}\mathbf{H})\boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^{\mathrm{T}}\mathbf{H}\boldsymbol{\delta}.$$

The first term captures linear effects with respect to $\boldsymbol{\delta}$, and the second term captures quadratic effects. By comparing the first term to equation B.2, we can see that it in fact corresponds to the sum of the single mutant fitness effects:

$$(\mathbf{h}^{\mathrm{T}} + \mathbf{x}_0^{\mathrm{T}}\mathbf{H})\boldsymbol{\delta} = \sum_{(i,\beta)\in M} s_{i\beta}.$$

The second term is related to the epistatic effects, which we show explicitly. We have

$$\boldsymbol{\delta}^{\mathrm{T}}\mathbf{H}\boldsymbol{\delta} = \sum_{(i,\beta),(j,\gamma)\in M^2} \mathbf{H}_{i\beta,j\gamma} - \mathbf{H}_{i\beta,ja} - \mathbf{H}_{ia,j\gamma} + \mathbf{H}_{ia,ja},$$

where M^2 refers to ordered pairs of mutations drawn from M. For $i \neq j$,

$$\begin{aligned}\mathbf{H}_{i\beta,j\gamma} - \mathbf{H}_{i\beta,ja} - \mathbf{H}_{ia,j\gamma} + \mathbf{H}_{ia,ja} &= \epsilon_{i\beta,j\gamma} \\ \boldsymbol{\delta}^{\mathrm{T}}\mathbf{H}\boldsymbol{\delta} &= \sum_{(i,\beta)\neq(j,\gamma)\in M^2} \epsilon_{i\beta,j\gamma} + \sum_{(i,\beta)\in M} (\mathbf{H}_{i\beta,i\beta} - 2\mathbf{H}_{i\beta,ia} + \mathbf{H}_{ia,ia})\,.\end{aligned}$$

We showed that without loss of generality, we can reparameterize $\mathbf{H}$ and $\mathbf{h}$ so that $\mathbf{H}$ has no diagonal terms. Assume this is the case. Then, we can write

$$\frac{1}{2}\boldsymbol{\delta}^{\mathrm{T}}\mathbf{H}\boldsymbol{\delta} = \frac{1}{2}\sum_{(i,\beta)\neq(j,\gamma)\in M^2} \epsilon_{i\beta,j\gamma}.$$

So we have shown that the second term in Equation B.3 is the sum of the pairwise epistatis effects for all pairs of mutations in $\mathbf{x}$ relative to the wildtype $\mathbf{x}_0$.

The expansion in Equation B.3 is useful for two reasons. Theoretically, it shows that the single mutant effects and epistatic effects control fitness differences completely, and gives us an easy way to compute them:

$$\mathcal{F}(\mathbf{x}) - \mathcal{F}(\mathbf{x}_0) = \sum_{(i,\beta)\in M} s_{i\beta} + \frac{1}{2}\sum_{(i,\beta)\neq(j,\gamma)\in M^2} \epsilon_{i\beta,j\gamma}.$$

The practical consequence is that we can use the decomposition to separately manipulate the single mutant and epistasis properties of the landscape, as we will discuss in Section B.4.

B.4 Tuning landscapes

In order to benchmark and understand methods for exploring fitness landscapes, we want to test those methods on fitness landscapes with variable properties. In particular, given some fitness function $\mathcal{F}$ of interest, we are interested in modifications of $\mathcal{F}$ which make the problem "easier" or "harder" by some metric.

For a quadratic $\mathcal{F}$, a set of simple modifications is given by shifting and scaling the distribution of fitnesses of single and double mutants relative to the wildtype. In particular, we can independently shift (add a constant to) and scale (multipy by a constant) the single mutant statistics s and the epistasis statistics ϵ uniformly for all sequences.

As we will see, this is different from simply modifying $\mathbf{h}$ and $\mathbf{H}$. Modifying s and ϵ corresponds to modifying the landscape in terms of first and second order expansions around the wildtype $\mathbf{x}_0$. In many biological problems, we care about understanding behavior near the wildtype; in addition, inferred landscape (e.g. using DCA [35]) are likely correlated with the "true" fitness landscape in limited neighborhood of the wildtype.

The shifting and scaling approach we outline maintains the relative ordering of fitnesses within the single mutants and within the double mutants. If we start with a fitness landscape whose properties are relevant for optimization, the modified landscape is one which has some similar qualitative features (e.g. important interactions in the original landscape are important in the tuned landscape). The modified landscapes can also probe different questions, such as "What happens when epistasis is more important than single mutant effects?"

We note that in most applications, we only care about the *relative* values of $\mathcal{F}$ (e.g. $\mathcal{F}(\mathbf{x}) - \mathcal{F}(\mathbf{x}_0)$), rather than the absolute values. We will take that approach here. If the absolute value also matters, for example if $\mathcal{F}(\mathbf{x}_0)$ needs to be set to 0, then this can be accomplished by adding the appropriate constant to $\mathcal{F}$.

Shifting the single-mutant distribution. Suppose we wish to shift the distribution of single mutant fitness effects, relative to wildtype, by some constant μ_s, without modifying ϵ. This can be accomplished by modifying $\mathbf{h}$ such that $\tilde{\mathbf{h}} = \mathbf{h} + \mathbf{v}$. Given $\mathcal{F}$ with parameters $\mathbf{h}$ and $\mathbf{H}$, we define $\tilde{\mathcal{F}}$ as

$$\tilde{\mathcal{F}}(\mathbf{x}) = (\mathbf{h} + \mathbf{v})^{\mathrm{T}}\mathbf{x} + \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{H}\mathbf{x}$$

Using the expansion in Equation B.3, we have

$$\tilde{\mathcal{F}}(\mathbf{x}) - \tilde{\mathcal{F}}(\mathbf{x}_0) = (\mathbf{h} + \mathbf{v} + \mathbf{x}_0\mathbf{H})^{\mathrm{T}}\boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^{\mathrm{T}}\mathbf{H}\boldsymbol{\delta}$$

We know that the first term controls s and the second controls ϵ. Therefore, with the appropriate choice of $\mathbf{v}$, we can modify s without modifying ϵ.

Let $\mathbf{v} = -\mu_s\mathbf{x}_0$. We note that $(\mathbf{x}_0^{\mathrm{T}}\boldsymbol{\delta})_{i\beta} = -1$ when $\mathbf{x}$ has a mutation $a \to \beta$ at position i, and 0 otherwise. Then we have:

$$\begin{aligned}\tilde{s}_{i\beta} &= \mathbf{h}_{i\beta} - \mathbf{h}_{ia} + \mu_s + \sum_{j=1}^{L}(\mathbf{H}_{i\beta,ja} - \mathbf{H}_{ia,ja}) \\ &= s_{i\beta} + \mu_s\end{aligned}$$

which corresponds exactly to the desired shift.

We note that the choice of $\mathbf{v}$ is not unique, since the quadratic form of $\mathcal{F}$, coupled with the gauge symmetry induced by the structured L-hot nature of of $\mathbf{x}$ means that the constant function can be written in many different ways. For example, the shift $\tilde{\mathbf{h}} = \mathbf{h} + \frac{c}{L}\mathbf{1}$ is equivalent to adding a constant c to the fitness function. This lack of uniqueness is not a problem computationally because our chosen form for $\mathbf{v}$ achieves the desired s and ϵ distributions - which are all that's needed to define $\mathcal{F}(\mathbf{x}) - \mathcal{F}(\mathbf{x}_0)$.

Shifting the epistatic distribution. Shifting the epistatic distribution is more complicated. From Equation B.3, we see that modifying $\mathbf{H}$ affects both the epistasis distribution as well as the single-mutant distribution. Therefore, we will modify both $\mathbf{H}$ and $\mathbf{h}$ in order to modify ϵ without changing any of the s.

We shift $\tilde{\mathbf{H}} = \mathbf{H} + \mathbf{C}$ and $\tilde{\mathbf{h}} = \mathbf{h} + \mathbf{w}$. Since $\tilde{\mathbf{H}}$ and $\mathbf{H}$ are symmetric and have 0 diagonal, $\mathbf{C}$ must be symmetric and have 0 diagonal. Our desired modified fitness function $\tilde{\mathcal{F}}$ therefore, is:

$$\tilde{\mathcal{F}}(\mathbf{x}) - \tilde{\mathcal{F}}(\mathbf{x}_0) = (\mathbf{h} + \mathbf{w} + \mathbf{x}_0(\mathbf{H} + \mathbf{C}))^{\mathrm{T}} \boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^{\mathrm{T}}(\mathbf{H} + \mathbf{C})\boldsymbol{\delta}$$

which has the same s as $\mathcal{F}$, but all ϵ shifted by μ_ϵ. We proceed by deriving $\mathbf{C}$ to modify ϵ, and then compute $\mathbf{w}$ to ensure there is no change in s.

From Equation B.2, we see that one way to change the epistasis is to modify the (ia, ja) terms in $\mathbf{H}$, and no others. This suggests that $\mathbf{C}$ should be proportional to $\mathbf{x}_0\mathbf{x}_0^{\mathrm{T}}$, which is equal to 1 at (ia, ja) and 0 otherwise. We define

$$\mathbf{C}_{i\beta,j\gamma} = \begin{cases} \mu_\epsilon & \text{for } i \neq j,\ \beta = \gamma = a \\ 0 & \text{otherwise} \end{cases}$$

In other words,

$$\mathbf{C} = \mu_\epsilon \left(\mathbf{x}_0\mathbf{x}_0^{\mathrm{T}} - \mathrm{diag}\left(\mathbf{x}_0\mathbf{x}_0^{\mathrm{T}}\right)\right)$$

Computing the epistasis for the modified $\tilde{\mathbf{H}} = \mathbf{H} + \mathbf{C}$ using Equation B.2, we have:

$$\begin{aligned} \tilde{\epsilon}_{i\beta,j\gamma} &= \mathbf{H}_{i\beta,j\gamma} - \mathbf{H}_{i\beta,ja} - \mathbf{H}_{ia,j\gamma} + \mathbf{H}_{ia,ja} + \mu_\epsilon \\ &= \epsilon_{i\beta,j\gamma} + \mu_\epsilon \end{aligned}$$

which gives us the intended shift. To ensure that s is unchanged, we set

$$\mathbf{h} + \mathbf{w} + (\mathbf{H} + \mathbf{C})\mathbf{x}_0 = \mathbf{h} + \mathbf{H}\mathbf{x}_0$$

Which gives us $\mathbf{w} = -\mathbf{C}\mathbf{x}_0$. Note that $\mathbf{x}_0^{\mathrm{T}}\mathbf{x}_0 = L$ and $\mathrm{diag}(\mathbf{x}_0\mathbf{x}_0^{\mathrm{T}})\mathbf{x}_0 = \mathbf{x}_0$. Then,

$$\begin{aligned} \mathbf{w} &= -\mathbf{C}\mathbf{x}_0 \\ &= -\mu_\epsilon \mathbf{x}_0\mathbf{x}_0^{\mathrm{T}}\mathbf{x}_0 + \mu_\epsilon \mathrm{diag}\left(\mathbf{x}_0\mathbf{x}_0^{\mathrm{T}}\right)\mathbf{x}_0 \\ &= -L\mathbf{x}_0 + \mathbf{x}_0 \\ &= -\mu_\epsilon(L-1)\mathbf{x}_0 \end{aligned}$$

Setting $\mathbf{w} = -\mu_\epsilon(L-1)\mathbf{x}_0$, s is left unchanged as desired.

Scaling the distributions. Now we consider the problem of *scaling* the distributions. That is, we want to modify $\mathbf{h}$ and $\mathbf{H}$ such that s and ϵ are multiplied uniformly by constants λ_s and λ_ϵ respectively. Using the difference expansion in equation B.3, we can see that to accomplish this we need to choose constants A, B and vector $\mathbf{u}$ such that:

$$B\mathbf{H} = \lambda_\epsilon \mathbf{H}$$

and

$$A\mathbf{h} + \mathbf{u} + B\mathbf{H}\mathbf{x}_0 = \lambda_s(\mathbf{h} + \mathbf{H}\mathbf{x}_0).$$

We immediately see that $B = \lambda_\epsilon$. In order to obtain the correct scaling of s, we have:

$$A\mathbf{h} + \mathbf{u} + B\mathbf{H}\mathbf{x}_0 = \lambda_s(\mathbf{h} + \mathbf{H}\mathbf{x}_0).$$

With our extra degree of freedom, we choose to set $A = \lambda_s$, so we have:

$$\mathbf{u} = (\lambda_s - \lambda_\epsilon)\mathbf{H}\mathbf{x}_0.$$

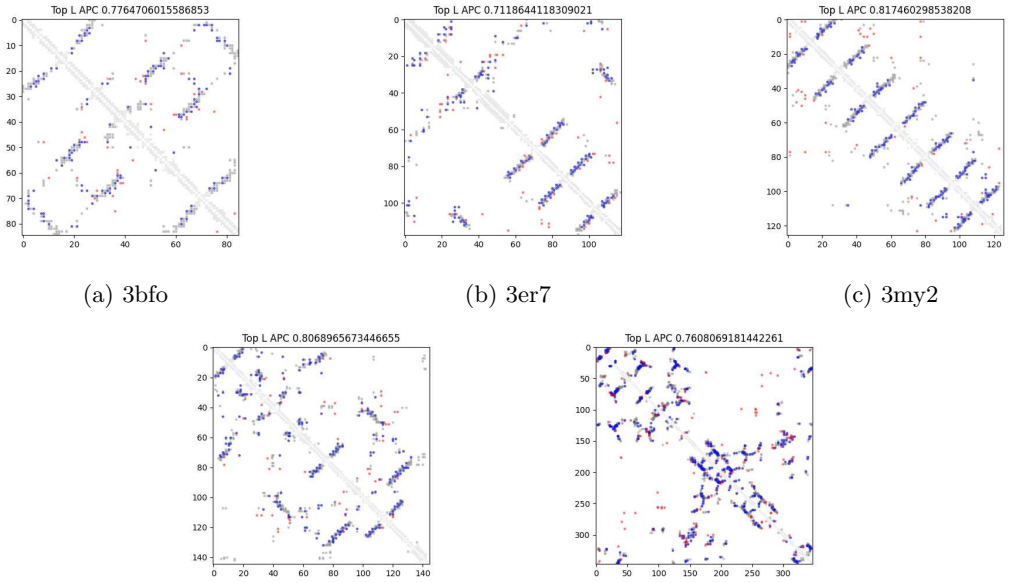
Our final fitness function is therefore

$$\tilde{\mathcal{F}}(\mathbf{x}) - \tilde{\mathcal{F}}(\mathbf{x}_0) = (\lambda_s \mathbf{h} + (\lambda_s - \lambda_\epsilon)\mathbf{H}\mathbf{x}_0)^{\mathrm{T}}\boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^{\mathrm{T}}(\lambda_\epsilon \mathbf{H})\boldsymbol{\delta}.$$

Each of the modifications outlined in Sections B.4, B.4 and B.4 can be composed:

$$\tilde{\mathcal{F}}(\mathbf{x}) - \tilde{\mathcal{F}}(\mathbf{x}_0) = \lambda_s \sum_{(i,\beta)\in M} [s_{i\beta} + \mu_s] + \frac{\lambda_\epsilon}{2} \sum_{(i,\beta)\neq(j,\gamma)\in M^2} [\epsilon_{i\beta,j\gamma} + \mu_\epsilon] .$$

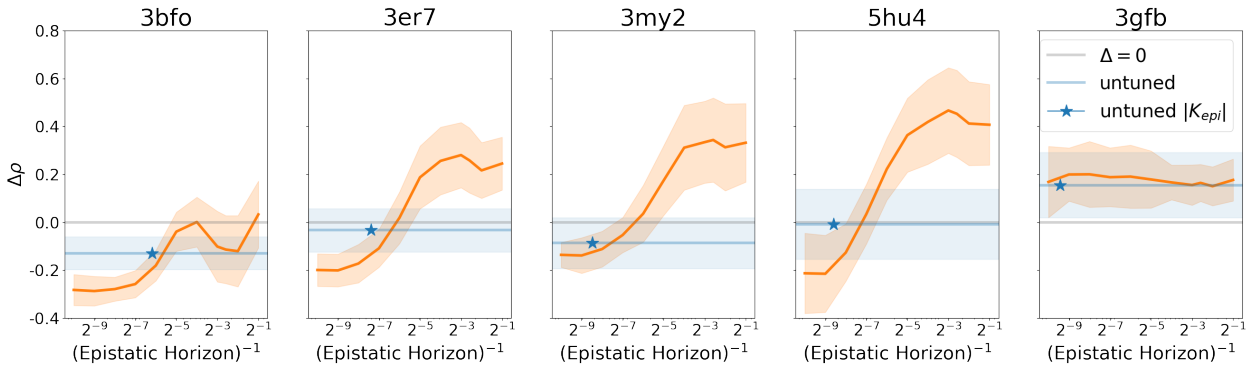
C Supplemental Figures



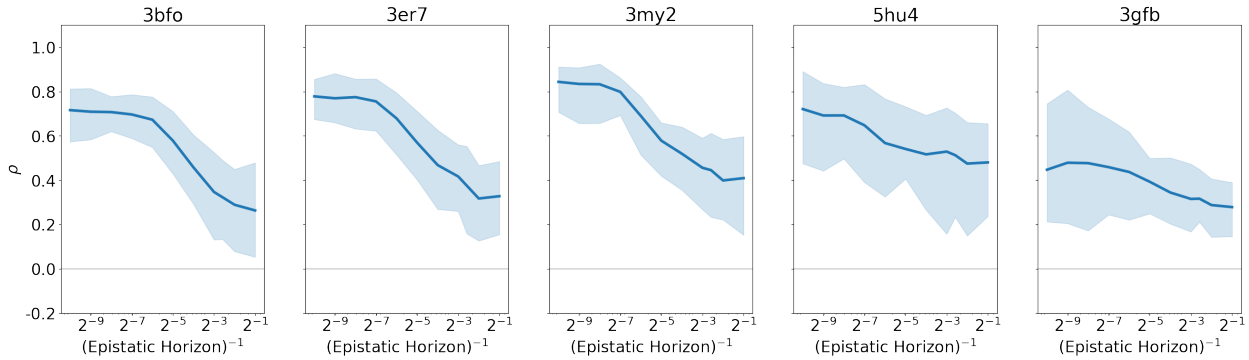
(a) 3bfo (b) 3er7 (c) 3my2

(d) 5hu4 (e) 3gfb

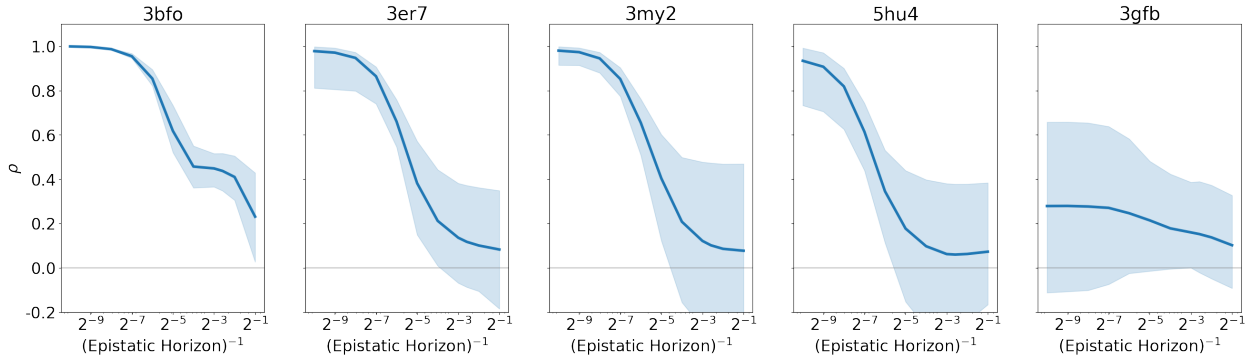
Fig. S1: Predicted contact maps derived from Potts models fit to the corresponding alignment. Shown are Top L predicted contacts after APC correction [74], with the precision shown in the title of each plot. Blue are true predictions, red are false predictions. In grey are all contacts.



(a) Copy of Figure 3. Differential performance ($\Delta\rho$) of the CNN versus the Ridge model on ranking combinations of adaptive singles.

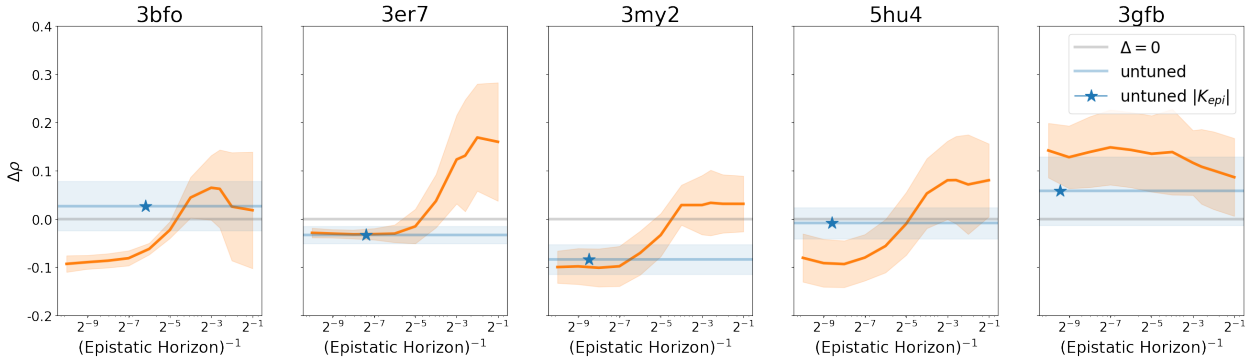


(b) Performance (ρ) of the CNN model on ranking combinations of adaptive singles.

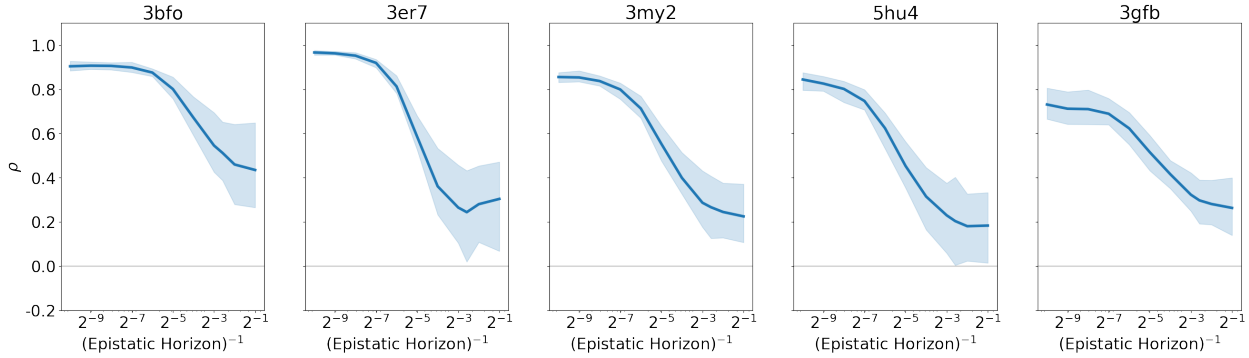


(c) Performance (ρ) of the Ridge model on ranking combinations of adaptive singles. $\rho \approx 1$ for linear landscapes with large horizons.

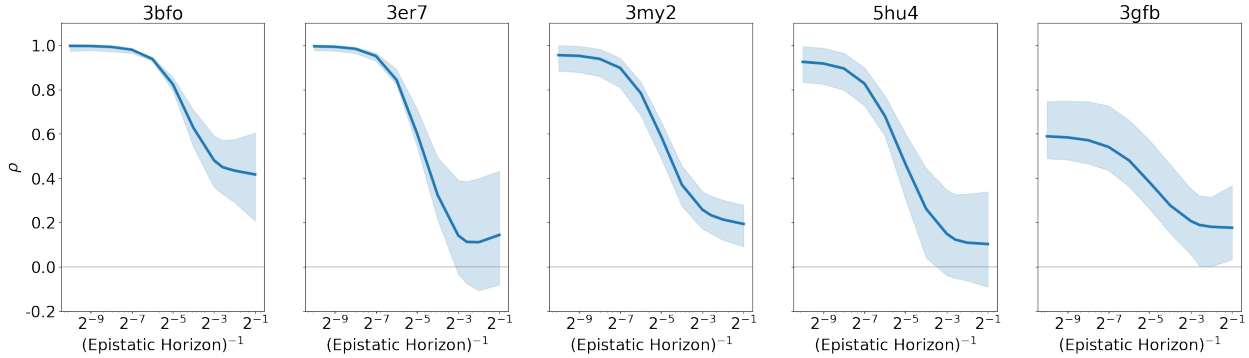
Fig. S2: Performance (ρ) on ranking sequences constructed from combinations of adaptive singles.



(a) Differential performance ($\Delta\rho$) of the CNN versus the Ridge model on ranking sequences enriched for deleterious epistasis

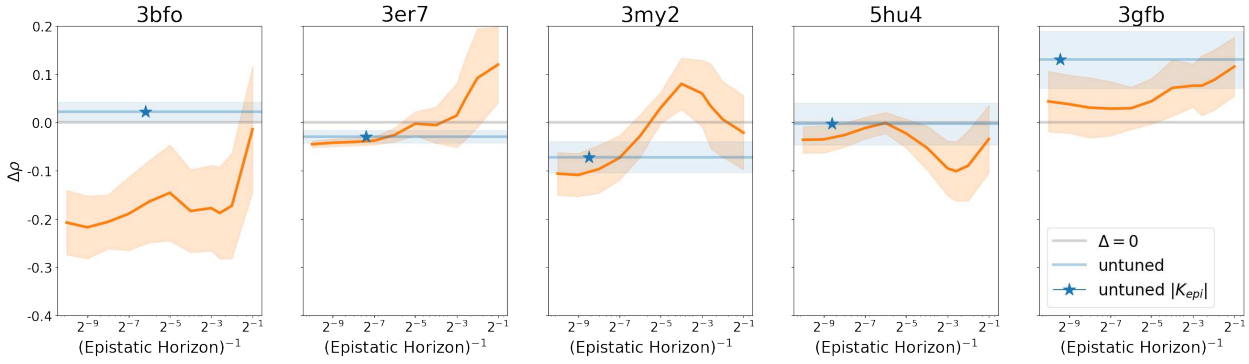


(b) Performance (ρ) of the CNN model on ranking sequences enriched for deleterious epistasis.

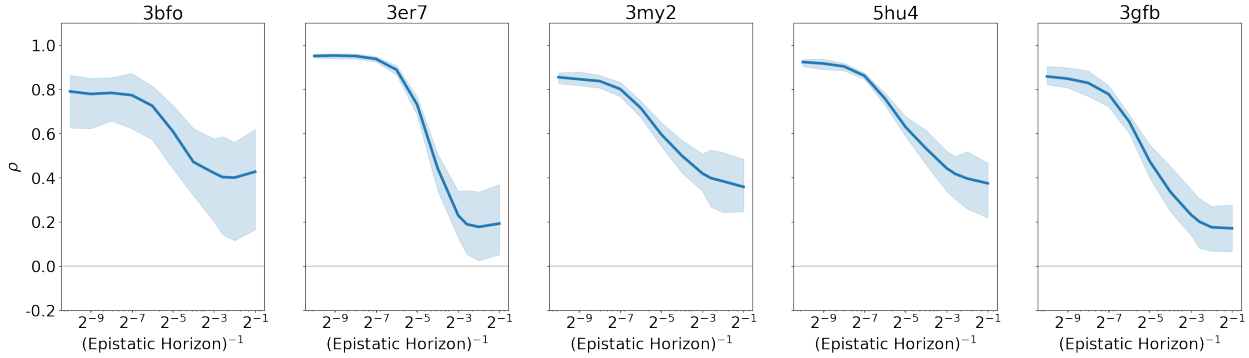


(c) Performance (ρ) of the Ridge model on ranking sequences enriched for deleterious epistasis.

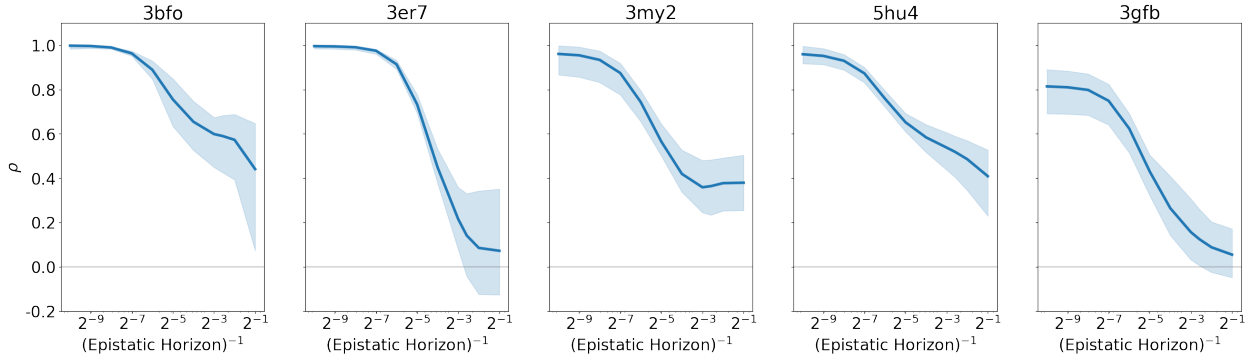
Fig. S3: Performance (ρ) on ranking sequences enriched for deleterious epistasis.



(a) Differential performance ($\Delta\rho$) of the CNN versus the Ridge model on ranking sequences enriched for adaptive epistasis.



(b) Performance (ρ) of the CNN model on ranking sequences enriched for adaptive epistasis.



(c) Performance (ρ) of the Ridge model on ranking sequences enriched for adaptive epistasis.

Fig. S4: Performance (ρ) on ranking sequences enriched for adaptive epistasis.

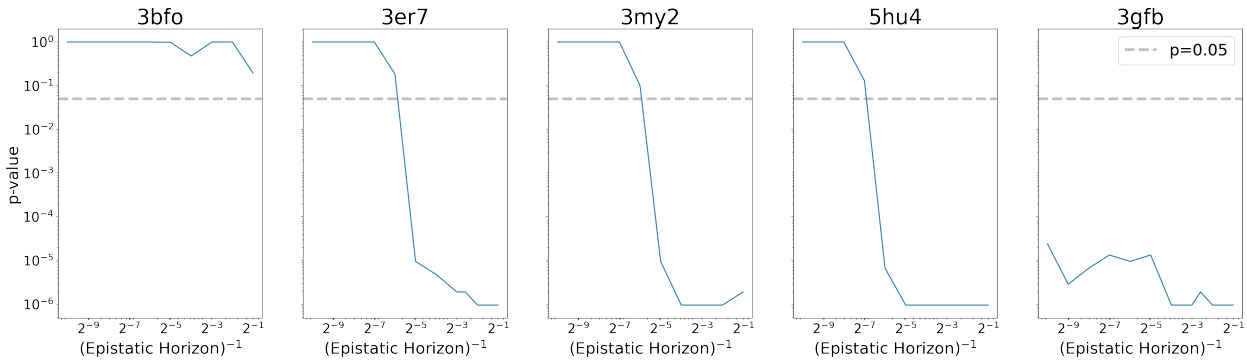


Fig. S5: Evaluation set: Adaptive Singles. One-sided Wilcoxon signed-rank test p-value on $\Delta\rho$ where samples are paired by training set replicates. This tests the null hypothesis that the two models have similar evaluation performance (i.e., the distribution of $\Delta\rho$ is symmetric). The grey line indicates the significance threshold p=0.05. 3er7, 3my2, 5hu4 all exhibit a clear transition point where the distribution of CNN model performance is significantly better than the distribution of Ridge model performance. The CNN significantly outperforms the Ridge model across all horizons for 3gfb.

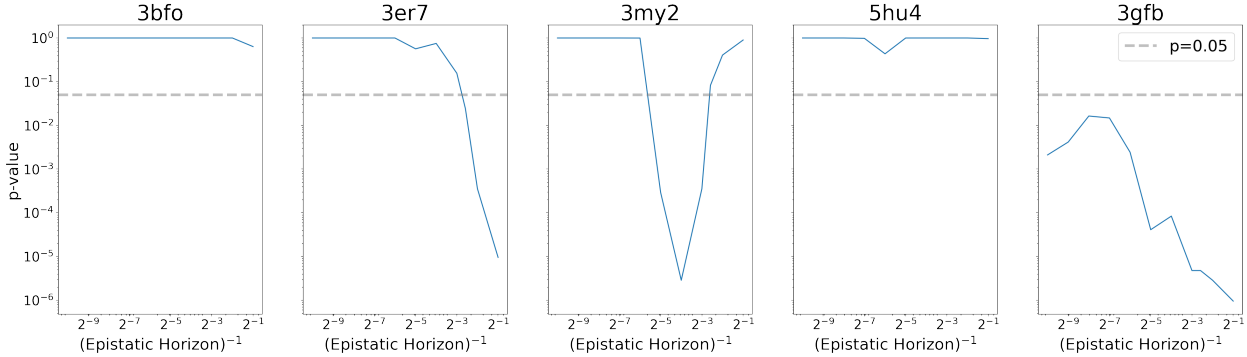


Fig. S6: Evaluation set: Adaptive Epistasis. One-sided Wilcoxon signed-rank test p-value on $\Delta\rho$ where samples are paired by training set replicates. This tests the null hypothesis that the two models have similar evaluation performance (i.e., the distribution of $\Delta\rho$ is symmetric). The grey line indicates the significance threshold p=0.05.

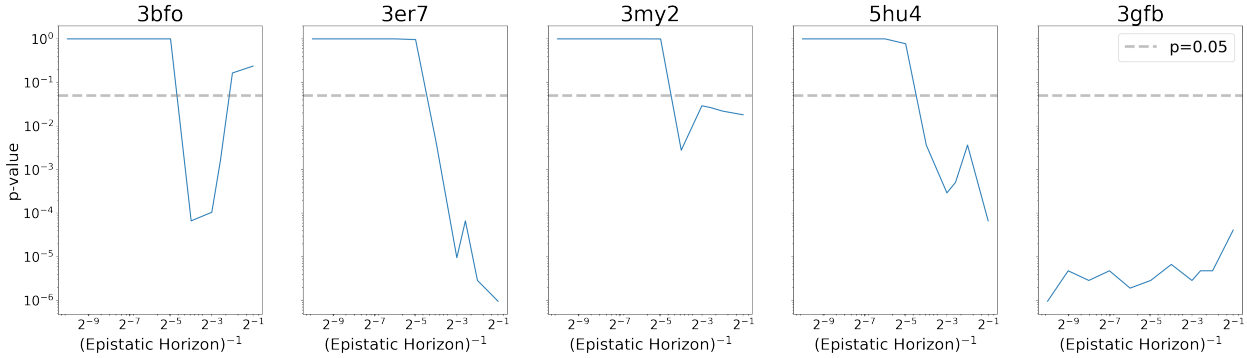


Fig. S7: Evaluation set: Deleterious Epistasis. One-sided Wilcoxon signed-rank test p-value on $\Delta\rho$ where samples are paired by training set replicates. This tests the null hypothesis that the two models have similar evaluation performance (i.e., the distribution of $\Delta\rho$ is symmetric). The grey line indicates the significance threshold p=0.05.

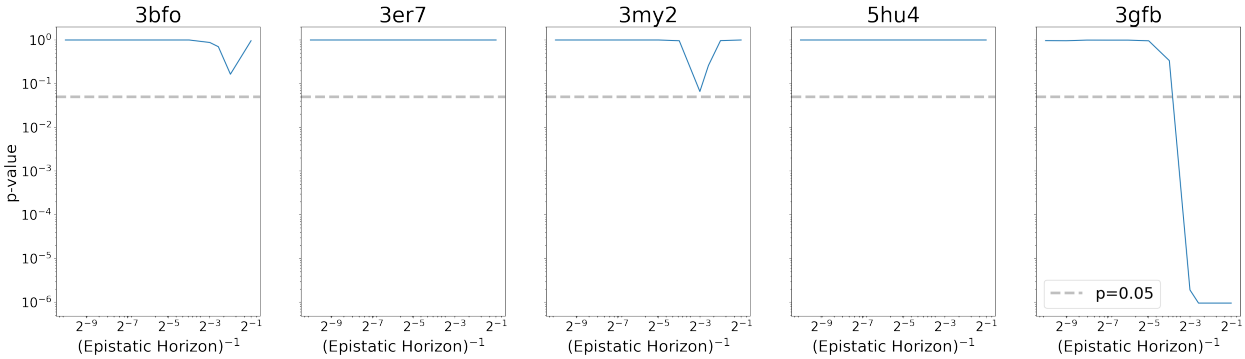


Fig. S8: Evaluation set: Adaptive Singles. One-sided Wilcoxon signed-rank test p-value on ΔMSE where samples are paired by training set replicates. This tests the null hypothesis that the two models have similar evaluation performance (i.e., the distribution of ΔMSE is symmetric). The grey line indicates the significance threshold p=0.05.