Guided Generative Protein Design using Regularized Transformers

Egbert Castro  
Comp. Biol. and Bioinf. Program  
Yale University  
New Haven, CT 06511  
egbert.castro@yale.edu

Abhinav Godavarthi  
Dept. of Applied Mathematics  
Yale University  
New Haven, CT 06511  
abhinav.godavarthi@yale.edu

Julian Rubinfeld  
Dept. of Mathematics  
Yale University  
New Haven, CT 06511  
 julian.rubinfeld@yale.edu

Kevin B. Givechian  
Dept. of Genetics  
Yale University  
New Haven, CT 06511  
kevin.givechian@yale.edu

Dhananjay Bhaskar  
Dept. of Genetics  
Yale University  
New Haven, CT 06511  
dhananjay.bhaskar@yale.edu

Smita Krishnaswamy  
Depts. of Genetics and Computer Science  
Program in Comp. Biol. and Bioinf.  
Yale University  
New Haven, CT 06520  
smita.krishnaswamy@yale.edu

Abstract

The development of powerful natural language models have increased the ability to learn meaningful representations of protein sequences. In addition, advances in high-throughput mutagenesis, directed evolution, and next-generation sequencing have allowed for the accumulation of large amounts of labeled fitness data. Leveraging these two trends, we introduce Regularized Latent Space Optimization (ReLSO), a deep transformer-based autoencoder which is trained to jointly generate sequences as well as predict fitness. Using ReLSO, we explicitly model the underlying sequence-function landscape of large labeled datasets and optimize within latent space using gradient-based methods. Through regularized prediction heads, ReLSO introduces a powerful protein sequence encoder and novel approach for efficient fitness landscape traversal.

1 Introduction

The ability to efficiently navigate enormous combinatorial spaces in biology holds great potential, especially in the field of protein design where a core challenge in designing novel or improved proteins is overcoming the combinatorial explosion in the number of possible sequences. This challenge is further exacerbated in protein design by the effect of epistasis – higher-order interactions between positions of a sequence – which makes it difficult predict how small changes in sequence will affect fitness. These challenges motivate a need for approaches that can better leverage sequence-function relationships, which are often described as fitness landscapes, to more efficiently reach highly functional protein sequences. Such methods would have significant implications in fields of therapeutic discovery and synthetic biology.

Preprint. Under review.
The combination of high-throughput data examining biological function and deep learning allows for an opportunity to gain a broader understanding of the synergy between sequence/structure and function, and in turn improves our ability to design proteins. Current methods for applying deep learning to protein design mainly use a trained model for in-silico screening of candidates, primarily by regressing a fitness score. Though this provides value in reducing the experimental screening burden, the challenge of navigating through sequence space remains.

An alternative to working directly in sequence space is to leverage the ability of deep learning models to learn a lower dimensional manifold of semantically-rich internal representations of data points, which in sum are referred to as a latent space. It is then possible to optimize a candidate sequences using this representation in a procedure called latent space optimization. However the latent space of models such as the autoencoder are often fraught with their own challenges. One of which is the emergence of "holes" in the latent space by which decoding from a regions of latent space results in poor reconstructions. Furthermore approaches to deep representation learning have primarily focused on unsupervised learning where the model is tasked to recover the original sequence after a compression or corruption transformation. However this approach does not address many challenges of sequence-function relationships of proteins, leaving them to carry over to the latent space of these deep models. As a result the latent space reflects the "needle in a haystack" problem where high-fitness sequences are encoded within dense low-fitness regions of latent space. Lastly, the latent space of deep learning models often lack well-defined boundaries that relate to the models ability to successfully return latent points back to realistic samples. This presents an obstacle to latent space optimization as latent points produced from an optimization run may fall outside the models ability perform well.

While there are challenges to successfully conducting latent space optimization, solutions can introduce enormous gains in optimization efficiency. Still largely under studied, latent space optimization provides an avenue for data-driven optimization which can ameliorate domain-specific challenges through the flexible nature of deep learning architectures and training strategies.

Here we propose ReLSO, a deep transformer-based approach to protein design. ReLSO combines the powerful encoding ability of a transformer model with an information bottleneck such that the resulting latent representation of protein sequences is information-rich while still being low-dimensional. Using a multi-task learning training procedure, the model is trained to jointly generate protein sequences as well as predict fitness. In this way, we transfer the original protein sequence design problem from a high-dimensional, discrete space to a much more amenable low dimensional, continuous space. This also addresses the imbalance between lowly-functional and highly-functional sequences by creating a clear organization by both sequence and fitness, simplifying to otherwise challenging task of finding sequences with higher fitness.

One of the benefits of ReLSO is produced using an interpolation regularization that enforces smoothness with respect to sequence, wherein small walks in latent space correspond to minor changes in sequence. This is particularly relevant to the field of protein design as it allows for more dense sampling of the latent space for diverse protein sequences generation while still retaining properties of interest. We further leverage the fitness prediction head to perform a gradient-based optimization procedure to move efficiently through latent space towards regions of high-fitness. As the fitness function learned by this network is important to successful optimization results, a norm-based negative sampling penalty is used to reshape the latent fitness landscape to be pseudo-convex. This has the dual benefit of further easing the optimization challenge as well as creating an implicit trust boundary.

Finally, the trained ReLSO model can readily be examined for sequence-function relationships. Since the encoder of the model is tasked with learning a latent representation informative for both sequence generation and fitness prediction, the resulting attention weights in the transformer can reflect salient sequence-function relationships. This opens a path towards mechanism of function insights when structural or evolutionary information is not available. ReLSO is therefore trained to reflect several desiderata in this domain and in turn, we are able to more efficiently perform latent space optimization.

1.1 Contributions

• We propose ReLSO, a model which is able to produce a smooth fitness landscape for protein design and optimization. The produced fitness organization in latent space ameliorates the fitness distribution skew pervasive in this domain.
• We show that through a norm-based negative sampling technique, the learned fitness function can be reshaped to induce a natural boundary and stopping criterion for latent space optimization by gradient ascent. This serves to introduce a level of pessimism that keeps latent optimization from drifting far beyond the training data distribution.

• We propose a regularization which enforces gradual changes in decoded sequence space when traversing though latent space. This allows for more dense sampling of the underlying sequence data manifold on which the training data lies.

• Through the use of a transformer-based encoder, we also provide a route for interpretability when assessing sequence-fitness relationships learned by the model. These relationships can be further studied to guide residue-level function attribution.

1.2 Related Work

Several deep learning approaches in recent years have been proposed to address the challenge of sequence-based protein design. These methods have mainly fallen into two categories:

Model-based design: Here a model is trained on a dataset to guide the search to better sequences either by allowing for the in-silico screening of candidates or by directly optimizing sequences. The former group has predominately involved the use of a supervised model $\hat{\phi}$ trained to predict fitness from sequence, $x$, by learning an approximate mapping $\hat{y} = \hat{\phi}(x)$. Using this trained model as a way to evaluate subsequent candidate sequences in silico, it is then possible to optimize sequences through search and screen refinement process as in Yang et al. [2019], Biswas et al. [2021]. The latter group of methods in model-based design operates directly on sequences, where a model seeks to optimize the sequence itself to maximize a target property Linder et al. [2020]. Model-based design methods have shown promising results in several applications Linder et al. [2020], Linder and Seelig [2020], Liu et al. [2020], Valeri et al. [2020], Angenent-Mari et al. [2020]. Recent efforts have sought to standardize this approach by introducing a benchmark environment Sinai et al. [2020].

Latent-space optimization: Another class of deep learning methods for protein design uses generative models to iteratively sample new candidates and perform a selection step in order to create pools of high-fitness sequences. These methods include DBas Brookes and Listgarten [2018], which performs a batch-wise sampling approach which performs alternating sampling and selection steps to arrive at a group of proposed high-fitness sequences. This method was further developed to include weighting candidates by the uncertainty of generative model and the predictive model used for fitness estimation Brookes et al. [2019]. Most related to ours is that of Tripp et al. [2020] where they also make the observation that the latent space of a model is often used post-hoc for optimization and may benefit for intentional training or regularization for the task of latent space optimization. In their work they do this by retraining a generative model using weighted samples, placing higher weights on samples that have a trait that is being optimized for. We are also inspired by the work of Gómez-Bombarelli et al. [2018] jointly train on an auxiliary task and learn a smooth latent space for optimization in the domain of small molecules. We build on their work and that of Castro et al. [2020], which used the joint-training set-up to create thermodynamic landscapes of RNA folding, to further generate a smooth latent space amenable to optimization using novel regularizations. In contrast to Tripp et al. [2020] and Brookes et al. [2019], our model only requires training once and learns a latent space where high-fitness samples gradated away from low-fitness samples. By only requiring one round of training, ReLSO is better able to leverage gains produced by large, transformer-based models. Furthermore, we depart from the work of Gómez-Bombarelli et al. [2018] as we extend their training approach to the protein sequence optimization domain and use a transformer-based encoder. Furthermore, we make use of the gradients provided by the fitness prediction network and highlight it’s efficiency in reaching high-fitness sequences.

2 Background

2.1 Protein Design

A quickly growing application area of deep learning has been in the field of protein design, in particular sequence-based protein design. In this field, the prevailing challenge is to design a protein via it’s sequence that is optimized for a function such as binding affinity ro stability. While physics-based methods are currently dominate, a data-driven approach to protein design can leverage the
We formulate our method by first starting at the traditional perspective of sequence-based protein design. With the emergence of powerful deep learning architectures, several promising approaches to protein design have been developed. However, these approaches are often limited to low-dimensional search spaces and are susceptible to overfitting. Recent work has shown that protein function is governed by higher-order interactions between residues, such as with residues comprising structural motifs or functional domains. These long-range dependencies can be challenging to model at the sequence level, as they necessitate some notion of "coupling" between positions far apart. Second, couplings themselves are likely insufficient: past work has shown that protein function is governed by higher-order interactions between residues (Russ et al. 2020) (i.e., epistasis) such that the collective effects of mutations can be non-additive (and as a result, difficult to predict) Pokusaeva et al. (2019).

To navigate protein sequence space, researchers have often utilized an iterative search procedure called directed evolution Arnold (1998). Here libraries of randomized sequences are generated and screened for a function or property of interest. The best sequence or sequences are then carried over to the next round of library generation and selection. Effectively, this searches sequence space using a hill climbing approach and as a consequence, is susceptible to local maxima. Additional approaches to protein design have included structure-based design Rohl et al. (2004), Norn et al. (2021) wherein ideal structures are chosen a priori and the task is to fit a sequence to the design. With the emergence of powerful deep learning architectures, several promising approaches to protein design have incorporated models into the design process, such as with residues comprising structural motifs or functional domains. These long-range dependencies can be challenging to model at the sequence level, as they necessitate some notion of "coupling" between positions far apart. Second, couplings themselves are likely insufficient: past work has shown that protein function is governed by higher-order interactions between residues (Russ et al. 2020) (i.e., epistasis) such that the collective effects of mutations can be non-additive (and as a result, difficult to predict) Pokusaeva et al. (2019).

2.2 Protein Representation Learning

We formulate our method by first starting at the traditional perspective of sequence-based protein design. While directed evolution has yielded various successes in a range of domains over the years, it is susceptible to the underlying topology of the fitness landscape wherein may lead to only locally optimal sequences. Recent work has sought to overcome the screening burden of directed evolution by performing in-silico evaluations of a candidate sequence’s fitness. This approach is comprised of training a model \( \hat{\phi}_x \) to approximate the “ground-truth” fitness landscape \( \phi_x \) by minimizing an objective function of the form \( \mathcal{L} = ||\hat{y} - y|| \), where \( \hat{y} = \hat{\phi}_x(x) \) and \( y = \phi_x(x) \). Here we refer to the ground-truth mapping between fitness and sequence as \( \phi_x \).

Once the model has converged, it is then used to evaluate sequence candidates \( \hat{x} \) in either using either an iterative modification or sampling approach. In either case, the local sequence space around the original sequence is explored using minor changes to \( x, \Delta_x \). However, the difficult to predict...
A second regularization further aids in generating from latent space by smoothing the latent space. As a result, the optimization process will spend much of its time traveling through dense regions of sequence space. A more recent approach to sequence-based protein design is to train a deep learning model to learn a representation of the sequence by pre-training the model on a large corpus of protein sequence data with an unsupervised task $||g(f(x)) - x||$ where $f_{\theta}$ is an encoder and $g_{\theta}$ is a decoder. The end result of this pre-training is a trained encoder that has learned a function $z = f_{\theta}(x)$, where $z$ is understood to contain abstract and useful information about protein sequence composition. For the unsupervised model, it can be further stated that learned latent code approximates the manifold on which the training data lies, where higher density is placed on realistic sequences.

Next, a prediction model $h_{\theta}$ is trained on the latent representation $z$ to learn a fitness landscape $\hat{y} = \hat{h}_{z} = h_{\theta}(z)$ that will be used to evaluate new candidates $x$. While this approach moves the optimization problem to a continuous setting, it suffers from a issue inherent in datasets from this domain. Often the datasets gathered from screening contain a small percentage of high-fitness sequences while the vast majority of sequences provide little to no measurable functionality. This leads to high-fitness latent encodings that are hidden within a dense point cloud of low-fitness latent encodings, leading to inefficiencies when searching latent space through small perturbations to $z$, $\Delta z$.

As a result, the optimization process will spend much of its time traveling through dense regions of low-fitness candidates. Again, a strong link between changes in sequence information, now encoded in $\Delta z$ and changes it’s associated fitness $\Delta y$ has yet to be established.

### 2.3 Problem setup

From a preliminary experimental screen of variants, a set of protein sequences $X$, where each sequence is a ordered set of amino acids $x = (\sigma_{1}, \sigma_{2}, ..., \sigma_{N-1}, \sigma_{N})$ composed from a finite alphabet of amino acids such that $\sigma_{i} \in \mathcal{V}$, $i \in [N]$ and their corresponding fitness values $y, y \in \mathbb{R}$ is produced. The final dataset $D$ is then comprised of pairs $(x_{i}, y_{i})$ of sequences and their observed fitness. From this data, we wish to find sequences $x^{*} \in \Phi$ that possess a high degree of fitness, as measured by some threshold $\{y^{*} \geq y_{\text{thresh}} \mid y^{*} = \phi(x^{*}), x^{*} \in \Phi\}$. We also desire solutions in $\Phi$ to be diverse and novel.

### 3 Our Approach

Here we propose to strongly connect two vital factors in protein design, sequence and fitness information, through the use of an jointly trained autoencoder model. The second task supplements the original autoencoder architecture, comprised of an encoder $f_{\theta}$ and decoder $g_{\theta}$, with a network $h_{\theta}$ which is tasked with predicting fitness from the latent representation $z$. The resulting objective function of this set-up takes the form

$$\mathcal{L} = ||g_{\theta}(f_{\theta}(x)) - x|| + ||h_{\theta}(f_{\theta}(x)) - y||$$

One can observe that in each update step, the encoder receives gradients from both the reconstruction loss and fitness prediction loss and is therefore directed to encode information about sequence and fitness in $z$. Indeed, when the dimensionality of $z$ is set to some low value $d << N$ the latent encoded is forced to include only the most salient information about sequence and fitness and induces a greater connection between the two in $z$. Through the use of this training strategy, we strengthen the connection between $\Delta z$ and $\Delta y$ for downstream applications.

For $f_{\theta}$, we leverage a transformer-based encoder which is able to take advantage of higher order couplings and epistasis in the sequence. This powerful encoder then passes it’s learned representation through an information bottleneck where the output produces an informative and low-dimensional representation of the sequence $z$ that can be used for latent space optimization.

To avoid latent space pathologies described in Section[1] we add several regularizations to the latent space of the model which induced a smooth, continuous, and pseudo-concave space in which to perform latent space optimization. The first uses a norm-based data augmentation strategy which reshapes the fitness function learned by $h_{\theta}$ to be pseudo-concave. This creates both a natural boundary for regions of latent space feasible for latent space optimization as well as a global fitness maxima. A second regularization further aids in generating from latent space by smoothing the latent space
Figure 1: Overall flow of information in ReLSO. (Top box) Foundational to ReLSO is the idea that protein sequences can be mapped to a latent fitness landscape, which is maintains better properties than the traditional sequence-level fitness landscape. In ReLSO, we use a transformer-based encoder to map sequences to points in a well-organized latent space produced after training. (Bottom box) To encourage latent fitness landscapes amenable to latent space optimization, ReLSO incorporates several regularizations. Critical among these regularizations are negative sampling at the fitness prediction head $h_\theta$ and interpolative sampling at the sequence generation head $g_\theta$.

with respect to sequence identity, creating a latent space which can be more densely sampled from. These regularizations are added to the objective function shown in Figure [1].

### 3.1 Encoding Sequences

In this study, we use a transformer-based encoder to learn the mapping from sequence space $x$ to latent space $z$. While other encoding methods that rely on convolutional and recurrent architectures have demonstrated success in this domain [Liu et al., 2020, Alley et al., 2019], we choose to use a transformer for several key reasons. One such reason is that the inductive bias of the transformer architecture matches the prevailing understanding that protein function is a consequence of pairwise interactions between residues (e.g., catalytic traits of proteases). This encoding mechanism has demonstrated great success when applied to proteins in recent years [Jumper et al., 2021, Rives et al., 2021, Rao et al., 2020]. In addition, the transformer’s attention-based encoding scheme is readily amenable to interpretability analysis. Finally, transformers show promise in their ability to represent long sequences due to their ability to view the entire sequence during the forward pass, avoiding limitations inherent to recurrent neural network-based encoders [Hochreiter, 1998].
3.2 Joint-Training of Models

A common characteristic shared across datasets in this domain is that high-fitness sequences occupy a small fraction of total sequences in the dataset. This is reflective of the nature of protein sequences in general, where few variations in sequence confer the necessary fold required for functionality.

To address this imbalance, we apply a multi-task or joint-training approach to learning a representation of sequences, combining the usual two-step process described in Section 2.2 into one. In relation to the multi-task learning literature, we follow a hard parameter sharing setup, where the encoder parameters are shared between prediction head. As we are interested in the model’s ability to reconstruct sequences as well as predict the input sequence’s fitness, we train using two tasks.

\[ L_{\text{task}} = \gamma L_{\text{recon}} + \alpha L_{\text{reg}} \]

where \( L_{\text{recon}} \) represents the reconstruction task and \( L_{\text{reg}} \) represents the fitness prediction task. Additionally, \( \gamma \) and \( \alpha \) are scalar hyperparameters which weight their respective tasks. We refer to this model as a JT-AE.

An important consequence of the joint training set-up is that the latent code is updated during each train step with gradient signals from both sequence and fitness information. The resulting \( z \) encoding is thereby induced to resolve the two pieces of information. After model convergence, the latent space is endowed with a strong sequence-fitness association which we leverage here for latent space optimization.

Figure 2: An inherent weakness of a joint-training set-up is that a monotonic function is learned, which lacks any stopping criterion when used for latent space optimization. To address this, we reshape the fitness function such that the global maxima lies in/near the training data. Here we train the ReLSO architecture with a 2-dimensional latent encoding on the GIFFORD dataset and then visualize the learned fitness prediction function by overlaying predicted fitness values as well as model-derived fitness gradients. Shown are the effects of the latent encoding magnitude weighting \( \eta \) and the type of regressor network used on the overall shape of the fitness function.
\[ z^{(t+1)} \leftarrow z^{(t)} - \eta \cdot (\nabla_z L_{\text{recon}} + \nabla_z L_{\text{reg}}) \]

### 3.3 Negative Sampling

A core challenge of optimizing within the latent space of deep learning models is moving far from the training data into regions where the model’s predictive performance deteriorates or is otherwise untrustworthy. Recent work has proposed techniques to define boundaries for model-based optimization, such as through a sequence mutation radius \[ \text{Biswa et al. [2021]} \] or by relying on model-derived likelihood values \[ \text{Brookes et al. [2019]} \]. While mutation radii are straightforward to implement, the diversity of fitness levels in even the immediate mutational neighborhood of a protein sequence can make such cutoffs less than ideal.

While the supervised head of a jointly-trained autoencoder provides directional information for latent space optimization, it does not readily provide a stopping criterion nor any strong notion of a boundary. Furthermore, simply adding a auxiliary task often results in a unidirectional organization by that attribute in latent space as shown by \[ \text{Gómez-Bombarelli et al. [2018]} \] and Figure 2 where following the gradient will quickly leave the training manifold.

In order to fully leverage the gradient signal provided by the fitness prediction head \( h_\theta \) of our model during latent space optimization, we introduce a bias in learned fitness function \( \hat{\phi}_z \) towards regions in latent space near the training data. This is done using a data augmentation technique called norm-based negative sampling where during training, each real set of latent points \( z \) from the current batch is complemented with a set of negative, artificial samples \( z_n \). These artificial samples \( z_n \) are produced by sampling high-norm regions of latent space surrounding real latent points, as shown in Figure 2. By assigning these artificial points \( z_n \) low fitness and including them in the fitness prediction loss, the learned fitness function \( \hat{\phi}_z \) is reshaped in such a way where there is a single fitness maxima located in or near the training data manifold. From this regularization function, an implicit trust region forms as well as a natural stopping criterion when used for latent space optimization. We will refer to the JT-AE model augmented with this regularization as ReLSO (neg).

Figure 3: Visualization of optimization paths in GIFFORD (A) Optimization trajectories of all 30 low-fitness seed sequences from the GIFFORD dataset from gradient-free methods (B) Optimization paths taken by gradient ascent converge to a high-fitness region of latent space. This is a result of the underlying regularized fitness function learned by ReLSO (C) For the optimization of a single seed, paths taken by gradient-free methods show a less-efficient progression to the data-driven global fitness maxima.
3.4 Interpolative Sampling

To further improve the latent space of our jointly-trained for protein sequence optimization, we also introduce a penalty which enforces smoother interpolation in latent space with respect to sequence modifications. This is appealing for sequence-based protein design as we would like to be able more densely sample latent space for both analysis of latent space optimization trajectories as well as enrichment of the areas of sequence space around high fitness sequences.

We enforce gradual changes in the decoded sequence space during latent space traversal by the addition of an interpolation regularization term. For this term, we take a fraction of batch of latent points and compute a KNN graph using pairwise euclidean distances. A set of new latent points $z_p$ are then generated by interpolating between nearest neighbors. This new set of points $z_p$ are passed through the decoder network $g_{\theta}$ to produce a set of decoded sequences $\hat{x}_p$. We then penalize by the distance between two sequences in $\hat{x}$ and their interpolant $\hat{x}_p$. Formally, this penalty calculated element-wise by:

$$L_{\text{interp}} = \max(0, \frac{||\hat{x}_1 - \hat{x}_i|| + ||\hat{x}_2 - \hat{x}_i|| - ||\hat{x}_1 - \hat{x}_2||}{2})$$

where $\hat{x}_1$ and $\hat{x}_2$ are nearest neighbors in latent space and $\hat{x}_i$ is the decoded sequence of the interpolated latent point. We will refer to the JT-AE model augmented with only this regularization as ReLSO (interp). Finally the full model, with both negative sampling and interpolative sampling regularization, is referred to as ReLSO.

3.5 Latent Space Optimization

To examine the effect of such a latent space in protein sequence optimization, we use a set of gradient-free and gradient-based optimization algorithms in our work. Additionally, we compare against two algorithms which instead operate in sequence space.

**Sequence-Based** We compare against two dominant forms of protein sequence in the literature. The first of which is in-silico directed evolution (DE) approach taken from Yang et al. [2019], where a subset of positions are chosen to be iteratively optimized. In this approach, the optimal amino acid for each of the chosen positions is determined and then held constant while the next position is optimized. We note that the evaluation "budget" is larger for this approach since it requires exhaustive evaluation of each position, incuring a cost of $20^N$, where $N$ is the number of positions chosen. The second sequence-space algorithm we include in our work is a MCMC approach from Biswas et al. [2021] where the initial sequence is randomly mutated and mutations are accepted with some probability. We refer to this approach as MCMC Seq.

**Gradient-Free** We test the latent space fitness landscapes using several gradient-free optimization routines. With this set of optimization algorithms we effectively treat our prediction head as a black-box regressor, using it to only label latent points we encounter. First, we pursue a simple hill-climbing algorithm (HC) which takes a greedy search through latent space. We also test a stochastic variant of hill climbing, which should better avoid local minima. Here the algorithm samples $z_{t+1}$ uniformly from $\{z \mid h_0(z + \epsilon) > h_0(z_t), \epsilon \sim \mathcal{N}(\mu, k)\}$, where $k$ is a parameter. We refer to this variation of hill climbing as stochastic hill climbing (SHC). We also use a MCMC scheme similar to that of Biswas et al. [2021], however here we perform this optimization in latent space (MCMC), where a small perturbation in the latent encoding is added $z_{t+1} = z_t + \epsilon$, where $t$ is the step index and $z_{t+1}$ is accepted with a probability equal to $\min(1, e^{\frac{h_0(z_{t+1}) - h_0(z_t)}})$. 

**Gradient-based** We also examine the performance of a gradient ascent optimization approach. Here we leverage the ability to extract gradient directions provided by the fitness prediction head of the model $h_\theta$ to travel to areas of latent space associated with higher fitness sequences. Through the aforementioned negative sampling regularization, the latent fitness landscape is shaped to be concave such that gradient ascent should arrive at a global maximum.

$$z^{(t+1)} \leftarrow z^{(t)} + \eta \cdot \nabla_z h_\theta$$
Figure 4: A. Here we compare the three major representation methods for proteins sequences in this work, namely the amino acid sequence (green edge), latent encoding from an unsupervised autoencoder (blue edge), and a jointly-trained autoencoder (orange edge). B. Grids of the representations of sequences from four datasets (left to right: GIFFORD, GB1, GFP, TAPE) visualized by PCA and colored by their respective fitness values. C. Here we visualize the same represenations now using PHATE [Moon et al., 2017], which visualizes multi-scale organization of the data. D. We use a smoothness metric which measure neighborhood-level variation of fitness values in latent space and compare the smoothness across the various approaches to protein representations in Table 2.

4 Datasets

Quantitative readouts of fitness landscapes have remained elusive until novel breakthroughs in high-throughput molecular biology, such as directed evolution and deep mutational scanning. Broadly speaking, these methods aim to introduce mutations in the sequence (or a set of interesting positions) in a systematic (saturation mutagenesis) or random (directed evolution) manner. The fitness metrics vary depending on the exact experiment; the specifics of the data which we benchmark on here are described further below with characteristics of these datasets summarized in Table 1.

- **Gifford Dataset**: Enrichment data from directed evolution; in this experiment, Liu et al. [2020] pitted a vast library (10^10 unique mutants) of an antibody against a single target. This library was then pitted through three consecutive rounds of selection, washing, and amplification. Next-gen sequencing was used between rounds to identify which sequences...
were enriched Here, the fitness is the log ratio of sequence enrichment between rounds of selection (i.e., how well the sequence performed relative to other members of the library).

- **GB1 Dataset** [Wu et al. 2016] carried out a saturation mutagenesis study targeted four sites and generated all 20^4 possible mutants to explore the local fitness landscape of GB1, an immunoglobulin-binding protein. This particular site is known to be an epistatic cluster. Fitness was measured by testing stability and binding affinity.

- **GFP Dataset** [Sarkisyan et al. 2016b] carried out random mutagenesis on a fluorescent protein (avGFP) to generate 51,175 unique protein coding sequences, with an averages of 3.7 mutations. Fitness was determined by measuring fluorescence of mutated constructs via a fluorescence-activated cell sorting (FACS) assay.

- **TAPE Dataset** In addition to the datasets we pulled from prior work, we used the TAPE benchmark datasets for fluorescence from [Rao et al. 2020]. Note that we also kept the train/test/validation splits consistent so as to establish a fair comparison. The data here is the same as [Sarkisyan et al. 2016b] but is simply split by sequence distance.

## 5 Results

### 5.1 Assessing Representation Smoothness

![Figure 5: Smoothness along 100 sampled walks between pairs of distant points in latent space. Each step $x_i$ is along a nearest neighbor search to a final latent point $x_f$. ∆Fit denotes L1 distance between each step’s fitness and the final point’s fitness. Similarly, ∆Seq denotes Hamming distance of each step’s sequence to the final step’s sequence. Ablation of negative sampling or interpolative sampling affects the smoothness of latent space traversal.](image)

\[
\Delta \text{Fit} = |y_i - y_f|
\]

\[
\Delta \text{Seq} = \text{Hamming}(x_i - x_f)
\]

As smoothness within the latent space encodings of our model play a major role in our approach, we measure smoothness with a metric derived from a graph-based approach. We construct a symmetric

### Table 1: Dataset Statistics. Diversity is measured using the average pairwise Hamming distances within a dataset

| Dataset                  | Sequence Length | Number of Datapoints | Diversity | Test Size |
|--------------------------|-----------------|----------------------|-----------|-----------|
| GB1 [Wu et al. 2016]     | 56              | 149361               | 0.068     | 0.25      |
| Gifford [Lau et al. 2020]| 20              | 90459                | 0.695     | 0.25      |
| GFP [Sarkisyan et al. 2016a] | 237             | 54024                | 0.032     | 0.30      |
| TAPE [Rao et al. 2019]   | 237             | 54024                | 0.032     | 0.56      |
KNN graph from the latent codes $Z = z_i, z_j, \ldots$ from a set of sequences such that $z_i$ is $z_j$ are connected by an edge if either $z_i$ is within the K-nearest neighbors of $z_j$ or conversely, if $z_j$ is within the K-nearest neighbors of $z_i$. By constructing our graphs in this way, we ensure our metric is scale-invariant. The KNN graph $A$ is then used to construct the combinatorial graph laplacian operator $L = D - A$ from which we calculate our smoothness metric as

$$\lambda_y = \frac{1}{N} y^T L y$$

where $y$ is our signal of interest and $N$ corresponds to the number of datapoints used to construct the graph. The quadratic form of the graph Laplacian operator can be interpreted as taking the sum of squared differences along edges in the underlying graph such that the resulting sum is lower if the signal is smooth i.e. small differences between neighboring points. We present these values in Table 2.

We first examine representation smoothness with respect to fitness and sequence using amino acid sequence directly, labeled as 'Sequence' in Table 2. As expected, this representation possesses the highest smoothness with respect to sequence however it’s smoothness with respect to fitness is less than that of other approaches. This likely due to the often tangled relationship between mutational distance and fitness described in Section 2.2. Furthermore, models which have trained to predict fitness produce a smoother representation with respect to fitness than models trained solely on reconstruction. The smoothing effect of the fitness prediction task can be readily observed in Figure 4, where latent encodings from four models are visualized using principle component analysis (PCA) and PHATE [Moon et al., 2017]. With the ablations performed in ReLSO, it is observed that removal of the interpolation sampling regularization (ReLSO (neg)) reduces the smoothness with respect to sequence. This effect is also observed in Figure 5 where walks in latent space possess greater variation in sequence change when the interpolation sampling regularization is removed. Lastly, we also compare to the protein representations learned by the pre-trained TAPE transformer model from Rao et al. [2019] which was trained on Pfam [El-Gebali et al., 2019].

|                      | GIFFORD | GB1      | GFP      |
|----------------------|---------|----------|----------|
|                      | $\lambda_f$ | $\lambda_s$ | $\lambda$ | $\lambda_f$ | $\lambda_s$ | $\lambda$ |
| Sequence             | 1.47    | 1.49     | 1.48     | 0.99       | 0.09       | 0.54     | 8.42      | 0.07       | 4.25       |
| AE                   | 1.70    | 1.96     | 1.83     | 1.00       | 0.09       | 0.54     | 7.52      | 0.10       | 3.81       |
| TAPE [Rao et al., 2019] | 1.91    | 2.08     | 2.00     | 0.86       | 1.98       | 1.42     | 6.09      | 0.10       | 3.09       |
| JT-AE                | 1.38    | 2.03     | 1.70     | 0.05       | 0.11       | 0.08     | 0.88      | 0.12       | 0.50       |
| TAPE + finetune      | 1.53    | 2.96     | 2.24     | 1.64       | 0.33       | 0.99     | 11.17     | 0.16       | 5.67       |
| ReLSO (interp)       | 1.36    | 2.03     | 1.70     | 0.04       | 0.11       | 0.08     | 7.20      | 0.11       | 3.65       |
| ReLSO (neg)          | 1.39    | 2.06     | 1.72     | 0.05       | 0.11       | 0.08     | 1.80      | 0.15       | 0.97       |
| ReLSO $\alpha = 0.1$ | 1.83    | 1.96     | 1.89     | 0.40       | 0.10       | 0.25     | 1.15      | 0.11       | 0.63       |
| ReLSO $\alpha = 0.5$ | 1.33    | 2.00     | 1.67     | 0.07       | 0.11       | 0.09     | 0.96      | 0.12       | 0.54       |
| ReLSO                | 1.36    | 2.05     | 1.70     | 0.05       | 0.11       | 0.08     | 3.68      | 0.16       | 1.92       |

Table 2: Quantification of latent space ruggedness, described in Section 5.1 in Table 2. Ruggedness values with respect to fitness ($\lambda_f$) and sequence ($\lambda_s$). The average of those two values $\bar{\lambda} = (\lambda_f + \lambda_s)/2$ is also reported.

5.2 Latent Space Optimization Results

Once the model has been trained, we then use the model’s latent representation $z$ for fitness optimization where we find high-fitness sequences $\Phi = \{x_i|y_i > \gamma\}$, where $\gamma$ is our "high-fitness" threshold. These sequences are considered for further evaluation either through other in-silico approaches (docking, etc.) or through experimental means. In pursuit of these high-fitness sequences, we expect the latent space of the trained model to provide a more amenable search space. To evaluate this, we use a collection of gradient-free and gradient-based optimization algorithms described in Section 3.5.

We present the results of these latent space optimization approaches in Table 3, where we report both the max fitness found and the cardinality of $\Phi$ over 30 seeds. We begin with a set of 30 seed
Table 3: Optimization of protein fitness. Here we evaluate the performance of three categories of optimization methods. The first three entries can be understood as optimization within latent space using local search approach. Latent space Metropolis Hastings MCMC searches pockets of latent space whereas hill climbing (HC) and stochastic hill climbing (SHC) use the locations of nearby points for directions to move in. The second category made up of Metropolis Hastings MCMC (MCMC Seq) and directed evolution (DE) are optimization approaches which operate on the sequence directly. The final category is a latent space optimization approach which uses a gradient-guided search process, gradient ascent (GA). We observe generally greater performance with this third class of method in the proposed evaluation criteria.

sequences drawn from the bottom 25 percentile of each dataset’s test split. This approach reflects a scenario where a researcher wishes to optimize the fitness of a novel sequence. We then allow each optimization method a \textit{in-silico} evaluation budget of 60. In other words, each optimization method is allowed to label, \textit{in-silico}, a maximum of 60 sequences over the course of the optimization run. We also report the diversity of the sequences in $\Phi$, which we measure as the average pairwise Hamming distance. Lastly, we report a measure of novelty which is the fraction of sequences in $\Phi$ not present in the training data.

To visualize the path taken by each optimization method, we co-embed the sequences encountered by each method into the latent space of the ReLSO model. We observe that gradient ascent is able to efficiently move through the latent space to a high-fitness region and as a result of the reshaped model fitness landscape, optimization trajectories converge near or at peak fitness, as shown in Figure 3B. This behavior keeps optimization trajectories from drifting into regions of latent space that the model has little information about and thus would likely make false predictions about. This is in contrast

| Method   | MCMC | GB1 | GFP |
|----------|------|-----|-----|
| Max Fit. | 0.19 | 0.09| 5.81|
| Novelty  | 3    | 10  | 20  |
| Diversity| 0.01 | 0.09| 0.85|
| HC       | -0.30| NA  | NA  |
| SHC      | 0.15 | 0.15| 0.15|
| MCMC Seq | 0.75 | 1.51| 5.81|
| DE       | 0.30 | 0.13| 0.15|
| GA       | 1.20 | 2.74| 5.81|

Figure 6: Optimization efficiency. (A) Here the final step fitness values of all 30 seeds are plotted, separated by dataset. We observe that gradient ascent is able to generate a greater number of high-fitness protein sequences compared to other methods. In the case of GFP, we observe that all optimized candidates of gradient ascent reside within the high-flourescence mode of the dataset. (B) The evolution of fitness values across GIFFORD seed sequences show a wide spread of efficiency. As some methods label multiple sequences during a single optimization step (e.g. HC, directed evolution) some lines do not each across the entire plot. Gradient ascent’s quick convergence to high-fitness is highlighted by the inclusion of a dotted vertical line at optimization step 15.
with other optimization methods which show less efficient paths towards high-fitness sequences, as shown in Figure 6A. We note that with the GFP dataset, it is likely that there is a easily reachable maximally fit sequence and as a result, all optimization methods converge on such a sequence. We find that ReLSO is able to produce a larger set of high-fitness sequences across the datasets with fewer optimization steps. This is observed in Figure 6B where gradient ascent (GA) is able to nearly converge to a region of high-fitness sequences with a fraction of the evaluation budget expended.

5.3 High Fitness Latent Space Sampling

To subsequently explore the ability of the model to generalize to similar but nonidentical sequences in the latent space, we generated 11,061 sequences using the top sequences within the Gifford dataset (enrichment binding affinity > 1.5). This filtered to 43 high fitness ‘seed’ antibody sequences. Given the variability in physicochemical properties with respect to the potential of a single amino acid, we generated variants from each of these 43 sequences at one mutational amino acid distance away from these 43 starting sequences, which were of identical length and index position within the CDR3 region for consistency. Each of 20 possible amino acids was substituted at each position, without mutating more than one position for a given generated variant. This allowed exploration of the local neighborhood of the high fitness latent space while also localizing to a particular window of the CDR3 domain.

Upon interpolation of the high fitness sequences, each of 11,061 sequences was independently run through the trained model for both binding affinity and embedding predictions. The embedding of each sequence was obtained and plotted onto the latent space in the context of the entire training dataset. Interestingly, the generated sequences localized to a single high fitness region of the latent space. This indicated a denser sampling and fitness prediction of the local space using generated sequences that were not part of the initial dataset. Additionally, the generated sequence abundance appeared to be associated with increasing fitness as shown in Figure 7, which was in line with the learned latent space embedding from the initial training data. These generative results suggest

Figure 7: Latent space visualization of high-fitness generated variants from the Gifford dataset. (A) Principal component analysis of the original training dataset model predicted embedding for each sequence. (B) Principal component addition of the 11,061 generated sequences onto the original latent space (yellow = high binding affinity/fitness, blue = low binding affinity/fitness). (C) Histogram distribution of fitness values for the original training dataset. (D) Histogram distribution of fitness values of the predicted fitness values from the generated sequences overlapped onto the original dataset distribution for visualization (black = original, red = generated)
the potential implication of low mutational distances from regions of interest within the latent space of encoded sequences. Moreover, in the context of biotherapeutic design, single amino acid manipulations may demonstrate significant differences in clinical efficacy. To this end, latent space optimization followed by a more dense, local generative sampling of high fitness variants may suggest an interesting avenue for sequence space exploration and biotherapeutic optimization.

5.4 Prediction Head Performance

In our model set-up, we train on two tasks - a reconstruction task and a fitness prediction task. The first of requires that the model be able to generate sequences from the low-dimensional latent encoding. This allows the model to be generative, which is crucial in the domain of biomolecules as complete enumeration of sequence space is infeasible. We train the model using a Cross-Entropy loss across the possible amino acid identities of each position in the sequence. Here we report accuracy as well as perplexity to assess model performance in this task in Table 4.

|          | Task 1 | Task 2 | Task 1 | Task 2 | Task 1 | Task 2 |
|----------|--------|--------|--------|--------|--------|--------|
| Perplexity | 1.03   | 0.90   | 0.88   | -0.15  | 1.00   | 1.00   |
| Accuracy  | 1.00   | 1.00   | 1.00   | 0.17   | 0.00   | 1.00   |
| MSE       | 0.99   | 0.18   | 0.85   |        | 0.00   | 0.00   |

Table 4: Task performance across the two tasks. Task 1 has the model predict sequence from latent space whereas Task 2 requires the model to predict fitness.

The second prediction head of this model is tasked with predicting the fitness value of input sequence. With this task, the model enforces the smoothness desideratum as well as learns a mapping from sequence to fitness. Due to the bottleneck introduced by the low-dimensional latent encoding $z$, the model produces a latent encoding there the coordinates of $z_i$ carry information about the fitness value of sequence $x_i$. We attempt to keep this mapping as simple as possible, and thus smooth, without losing fitness prediction performance by using a two-layer fully-connected network. We report these performance values in Table 4.

While the focus of this work is not to produce a new state-of-the-art fitness prediction model, it is useful to view differential performance of ReLSO and it’s ablations. It is observed that ReLSO’s ability to predict fitness is not impacted by the inclusion of it’s negative sampling based penalty. This is important as ReLSO’s fitness prediction head guides the search process towards an optimally fit sequence. Moreover, the interpolative sampling regularization has a similar minor effect on reconstruction ability. We note that due the low diversity of GB1’s protein sequences, it is difficult to meaningfully assess reconstruction ability.

5.5 Interpretability

Encouraged by the success of other works of Tamburini [2020] and Rao et al. [2020], we study the attention heads of our model for sequence-fitness attribution.

We are able to observe that the variable residues in the GB1 dataset (39-41, 54) are highly-weighted in the attention maps of the ReLSO model, as shown in Figure 8. As these 4 positions are the only changing residues across sequences in the GB1 dataset, we confirm their importance for the prediction tasks with their attention maps. Furthermore, we observe a clear difference between the attention maps of the ReLSO model and the AE model in the GFP model. While we do not identify these learned relationships in depth, this difference serves as evidence that differential attention relationships are a potential route towards interpretability. In other words, the attention maps of ReLSO have learned different residue-residue relationships than AE and these differences, likely a direct consequence of the additional fitness prediction task, present information the model deemed informative for prediction.
Figure 8: (Top) GB1 protein structure with overlaid attention. Here we visualize the pairs of residues with the highest interactions as determined by the attention maps in our model. Here we see a preference for residue 43. (Bottom) GFP protein structure with overlaid attention. Differential attention between ReLSO and the purely unsupervised AE model may indicates relations important for function.

6 Conclusion

The ability to find better representations is vital to extracting insights from noisy, high-dimensional data within the fields of protein biology. Defined by their biochemical interactions, evolutionary selection pressures, and function-stability tradeoffs, biomolecules are an increasingly important domain for the application of deep learning. More specifically, the field of biotherapeutic development has recently shown significant benefits from the application of both linear and non-linear models. Some of the very impactful models in this space have been largely supervised, but more recent work has proven the usefulness of leveraging unsupervised learning to pre-train predictive models to identify protein sequences with an enhanced property of interest.

Here we took an alternative path to combining these two types of learning objectives by instead taking a multi-task learning approach. Through simultaneously optimizing for protein sequence generation and fitness level prediction, we explicitly enforce a latent space rich in information about both sequence and fitness information. Importantly, this fitness information may encompass a variety of different properties such as binding affinity and fluorescence, which is smoothly embedded in the latent space of our trained model. We further add regularizations which reflect principles of protein engineering, reshaping the latent space in the process. Leveraging these regularizations and the architecture of the model, we show how gradient ascent optimization can deliver improvements in protein optimization when searching over protein sequence space.

The departure of this approach from other methods demonstrates a novel and promising avenue for improving our ability to design and optimize proteins. Furthermore, the reliance of this method solely on sequence information paired to a fitness value suggests ReLSO is likely translatable to other biomolecules such as DNA and RNA. The application to nucleic acids could have an interesting path towards reducing off-target effects in gene editing modalities such as CRISPR-Cas9. As such, with the growing prominence of biological therapeutics, this research direction has potential to deliver improvements in the development of improved therapeutics.
The intersection of machine learning and medicine as a whole has grown in recent years to suggest promise both in research as well as clinical practice. Diagnostic image data-rich fields such as pathology and radiology continue to pioneer towards optimized clinical management. Genomic data, and more recently, single-cell RNA sequencing data, have also pushed boundaries towards deciphering the complexity of genome regulation and target identification in hopes of better managing patient pathophysiology and disease. In the present study, we leveraged public protein fitness datasets to explore the boundaries of latent space optimized protein design. Moreover, while we explore several applications of our methodology and show robust generalized performance independent of a specific property, we envision an exciting application in biotherapeutic optimization. Specifically, we see an interesting avenue in tuning protein protein binding affinity to increase selectivity of a certain target or isoform, but against others. Therapeutically modifying characterized diseased cell pathways using this methodology may have great potential in fields such as oncology, neurology, and infectious disease amongst others. Ultimately, the growing production of biotherapeutic fitness data, latent space optimization of therapeutic proteins may eventually improve the therapeutic management of patient disease
References

Kevin K Yang, Zachary Wu, and Frances H Arnold. Machine-learning-guided directed evolution for protein engineering. *Nature Methods*, 16(8):1 – 8, 2019. doi: 10.1038/s41592-019-0496-6.

Surojit Biswas, Grigory Khimulya, Ethan C Alley, Kevin M Esvelt, and George M Church. Low-n protein engineering with data-efficient deep learning. *Nature Methods*, 18(4):389–396, 2021.

Johannes Linder, Nicholas Bogard, Alexander B Rosenberg, and Georg Seelig. A Generative Neural Network for Maximizing Fitness and Diversity of Synthetic DNA and Protein Sequences. *Cell Systems*, 11(1):49 – 62.e16, 2020. doi: 10.1016/j.cels.2020.05.007.

Johannes Linder and Georg Seelig. Fast differentiable dna and protein sequence optimization for molecular design, 2020.

Ge Liu, Haoyang Zeng, Jonas Mueller, Brandon Carter, Ziheng Wang, Jonas Schilz, Geraldine Horny, Michael E Birnbaum, Stefan Ewert, and David K Gifford. Antibody complementarity determining region design using high-capacity machine learning. *Bioinformatics*, 36(7):2126–2133, 2020.

Jacqueline A Valeri, Katherine M Collins, Pradeep Ramesh, Miguel A Alcantar, Bianca A Lepe, Timothy K Lu, and Diogo M Camacho. Sequence-to-function deep learning frameworks for engineered riboregulators. *Nature Communications*, 11(1):1 – 14, 2020. doi: 10.1038/s41467-020-18676-2.

Nicolaas M Angenent-Mari, Alexander S Garruss, Luís R Soenksen, George Church, and James J Collins. A deep learning approach to programmable rna switches. *Nature Communications*, 11(1):1–12, 2020.

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. *arXiv preprint arXiv:2010.02141*, 2020.

David H Brookes and Jennifer Listgarten. Design by adaptive sampling. *arXiv preprint arXiv:1810.03714*, 2018.

David Brookes, Hahnbeom Park, and Jennifer Listgarten. Conditioning by adaptive sampling for robust design. In *International Conference on Machine Learning*, pages 773–782. PMLR, 2019.

Austin Tripp, Erik Daxberger, and José Miguel Hernández-Lobato. Sample-efficient optimization in the latent space of deep generative models via weighted retraining. *Advances in Neural Information Processing Systems*, 33, 2020.

Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamin Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

Egbert Castro, Andrew Benz, Alexander Tong, Guy Wolf, and Smita Krishnaswamy. Uncovering the folding landscape of rna secondary structure using deep graph embeddings. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 4519–4528. IEEE, 2020.

Kai Chen and Frances H Arnold. Engineering new catalytic activities in enzymes. *Nature Catalysis*, 3(3):203–213, 2020.

William P Russ, Matteo Figliuzzi, Christian Stocker, Pierre Barrat-Charlaix, Michael Socolich, Peter Kast, Donald Hilvert, Remi Monasson, Simona Cocco, Martin Weigt, et al. An evolution-based model for designing chorismate mutase enzymes. *Science*, 369(6502):440–445, 2020.

Victoria O Pokusaeva, Dinara R Usmanova, Ekaterina V Putintseva, Lorena Espinar, Karen S Sarkisyan, Alexander S Mishin, Natalya S Bogatyreva, Dmitry N Ivankov, Arseniy V Akopyan, Sergey Ya Avvakumov, et al. An experimental assay of the interactions of amino acids from orthologous sequences shaping a complex fitness landscape. *PLoS genetics*, 15(4):e1008079, 2019.

Frances H Arnold. Design by directed evolution. *Accounts of chemical research*, 31(3):125–131, 1998.
Carol A Rohl, Charlie EM Strauss, Kira MS Misura, and David Baker. Protein structure prediction using rosetta. *Methods in enzymology*, 383:66–93, 2004.

Christoffer Norn, Basile IM Wicky, David Juergens, Sirui Liu, David Kim, Doug Tischer, Brian Koepnick, Ivan Anishchenko, David Baker, and Sergey Ovchinnikov. Protein sequence design by conformational landscape optimization. *Proceedings of the National Academy of Sciences*, 118(11), 2021.

Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, 16(12):1315–1322, 2019.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

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.

Roshan Rao, Sergey Ovchinnikov, Joshua Meier, Alexander Rives, and Tom Sercu. Transformer protein language models are unsupervised structure learners. *bioRxiv*, 2020.

Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

Kevin R Moon, David van Dijk, Zheng Wang, William Chen, Matthew J Hirn, Ronald R Coifman, Natalia B Ivanova, Guy Wolf, and Smita Krishnaswamy. Phate: a dimensionality reduction method for visualizing trajectory structures in high-dimensional biological data. *BioRxiv*, page 120378, 2017.

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:e16965, 2016.

Karen S Sarkisyan, Dmitriy A Bolotin, Margarita V Meer, Dinara R Usmanova, Alexander S Mishin, George V Sharonov, Dmitriy 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, 2016a.

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating protein transfer learning with tape. In *Advances in Neural Information Processing Systems*, pages 9689–9701, 2019.

Karen S Sarkisyan, Dmitriy A Bolotin, Margarita V Meer, Dinara R Usmanova, Alexander S Mishin, George V Sharonov, Dmitriy N Ivankov, Nina G Bozhanova, Mikhail S Baranov, Onuralp Soylemez, Natalya S Bogatyreva, Peter K Vlasov, Evgeny S Egorov, Maria D Logacheva, Alexey S Kondrashov, Dmitriy M Chudakov, Ekaterina V Putintseva, Ilgar Z Mamedov, Dan S Tawfik, Constantin A Lukyanov, and Fyodor A Kondrashov. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603):397–401, 2016b. doi: 10.1038/nature17995.

Sara El-Gebali, Jaina Mistry, Alex Bateman, Sean R Eddy, Aurélien Luciani, Simon C Potter, Matloob Qureshi, Lorna J Richardson, Gustavo A Salazar, Alfredo Smart, et al. The pfam protein families database in 2019. *Nucleic acids research*, 47(D1):D427–D432, 2019.

Fabio Tamburini. How “bertology” changed the state-of-the-art also for italian nlp. In *Proceedings of the Seventh Italian Conference on Computational Linguistics (CLiC-it 2020)*, Bologna, Italy, 2020.