Evolution-Inspired Loss Functions for Protein Representation Learning

Chengyue Gong [1] **Adam Klivans** [1] **James Madigan Loy** [2] **Tianlong Chen** [1] **Qiang Liu** [1] **Daniel Diaz** [1 2]

Abstract

AI-based frameworks for protein engineering use self-supervised learning (SSL) to obtain representations for downstream mutation effect predictions. The most common training objective for these methods is wildtype accuracy: given a sequence or structure where a wildtype residue has been masked, predict the missing amino acid. Wildtype accuracy, however, does not align with the primary goal of protein engineering, which is to suggest a *mutation* rather than to identify what already appears in nature. Here we present Evolutionary Ranking (EvoRank), a training objective that incorporates evolutionary information derived from multiple sequence alignments (MSAs) to learn more diverse protein representations. EvoRank corresponds to ranking amino-acid likelihoods in the probability distribution induced by an MSA. This objective forces models to learn the underlying evolutionary dynamics of a protein. Across a variety of phenotypes and datasets, we demonstrate that EvoRank leads to dramatic improvements in zero-shot performance and can compete with models fine-tuned on experimental data. This is particularly important in protein engineering, where it is expensive to obtain data for fine-tuning.

1. Introduction

The success of AlphaFold (Jumper et al., 2021) has inspired a new era of deep-learning frameworks for protein design and engineering. Large protein language models (e.g., ESM (Rives et al., 2019a; Meier et al., 2021a)), structure generative models (e.g., RFDiffusion (Watson et al., 2023), NeuralPLexer (Qiao et al., 2023)) and structure-based self-supervised models (Sumida et al., 2024; Diaz et al., 2023; Lu et al., 2022) can accelerate the development of biotechnology with applications in identifying disease-causing variants (Braunisch et al., 2021; Kouba et al., 2023; Scherer et al., 2021) and enzyme engineering for biomanufacturing. Due to the prohibitive cost of generating experimental data, self-supervised learning (SSL) has become the primary technique used by the community to generate protein representations (*e.g.*, Riesselman et al., 2018a; Rives et al., 2019a; Meier et al., 2021a; Dauparas et al., 2022; Bepler & Berger, 2019; d'Oelsnitz et al., 2023; Notin et al., 2022; Hsu et al., 2022). These methods rely on masking followed by predicting the wildtype (WT) amino acids in extant proteins as the SSL training objective. For example, given as input a protein sequence and masked residue, models can be trained to predict what amino acid has been masked. The loss in WT-mask SSL is typically defined to be the cross entropy between a model's prediction and the one-hot encoding of the masked wildtype amino acid(s). This *wildtype accuracy* metric, also known as recovery ratio, is then reported as a proxy for the quality of the learned representations.

For machine learning-guided protein engineering (MLPE), practitioners desire models that suggest mutations to a protein *away* from wildtype, as opposed to models that merely predict wildtype. To address this disparity, several approaches have been proposed. Structure-based methods often adjust the temperature of the logits (Ingraham et al., 2019; Dauparas et al., 2022; Sumida et al., 2024) to bias away from wildtype. Sequence-based methods require large protein databases and incorporate MSAs as additional inputs to mitigate the existence of unique wildtype signatures (Rao et al., 2021a; Notin et al., 2022).

A more serious and often overlooked issue, however, is that improved wildtype accuracy may *not* correlate with downstream mutation effect performance. We sharply illustrate this phenomenon in Table 1 where we train a structure-based model to increasing levels of wildtype accuracy and show that its downstream performance on thermodynamic stability prediction begins to *decrease* beyond a wildtype accuracy threshold.

Additionally, current frameworks using either sequence or structure modalities can achieve greater than 90% wildtype accuracy (*e.g.*, Rives et al., 2019a; Meier et al., 2021a; Lin et al., 2023; Diaz et al., 2023), forcing the practitioner to

[*]Equal contribution [1]University of Texas at Austin [2]Intelligent Proteins, LLC.
Correspondence to: Daniel Diaz <dannyjdiaz305@gmail.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

make ad-hoc decisions about the optimal choice of wild-type accuracy for downstream applications. Developing a self-supervised learning objective that acts as an effective proxy for mutation effect prediction remains a critical open problem.

Loss	Metrics	MutComputeXGT						ProteinMPNN	ESM2
WT-mask	WT Acc	17%	29%	43%	68%	79%	92%	48%	94%
	Pearson	0.14	0.21	0.30	0.34	0.30	0.24	0.31	0.25
EvoRank	EvoRank	0.24	0.21	0.17	0.15	0.13	0.12	-	-
	Pearson	0.30	0.36	0.45	0.48	0.51	0.50	0.31	0.25

Table 1. **Case study for the relation between WT accuracy and correlation with experimental data.** We demonstrate that further improvements in WT accuracy do not necessarily translate to improvements in Pearson correlation on FireProtDB (Stourac et al., 2021). Models trained with the WT-mask loss are unable to that achieve Pearson correlation above 0.4, with performance peaking and then beginning to *decrease* after $\sim$ 68% WT accuracy. We further validate this observation with Protein MPNN and ESM2. In contrast, The EvoRank loss avoids overfitting to WT amino acids and enables learning SSL representation more conducive for guiding protein engineering.

Our main contribution is a new self-supervised training objective, EvoRank, that incorporates evolutionary information from multiple sequence alignments (MSAs) in order to address the limitations of WT-mask SSL. To emulate the mutation setting, EvoRank uses a ranking objective to force a model to learn fine-grained information about the MSA-induced distribution of amino acids at a particular location. We show that after initializing a model's wild-type predictions with an approximate MSA distribution, EvoRank results in dramatic empirical improvements for zero-shot performance across a variety of commonly studied benchmarks. Additionally, since MSAs are incorporated into the loss, they are only needed during training and not inference time, in contrast to models that require an MSA as an additional input (Notin et al., 2022). Further, empirical improvements on the EvoRank loss are correlated with improvements in downstream mutation effect prediction (see Table 1), leading to a reliable benchmark for protein representation learning.

2. Related Works

Multiple Sequence Alignments (MSAs) A multiple sequence alignment (MSA) is an established tool used to identify the evolutionary relationship between genes and can be generated for DNA, RNA, and protein sequences. For a particular protein, an MSA represents the genetic variation observed in extant homologous sequences present in a database, such as UniProt (Consortium, 2015), and capture evolutionary and structural constraints for a particular protein family (Thompson et al., 1994; 1997). This makes MSAs a rich source of biological information for computational biologist and recently for training machine learning models. For example, Alphafold2 demonstrates that the information within a protein's MSA is sufficient to predict its 3D structure with near experimental accuracy. Additionally, AlphaFold-Multimer demonstrates that using paired-MSA information improves protein-protein interaction predictions, resulting in significant improvements for predicting of protein complexes (Evans et al., 2021).

Sequence-based machine learning frameworks have used MSA information to predict mutational effects and protein fitness. Representative methods, *i.e.*, EVmutation (Hopf et al., 2017), DeepSequence (Riesselman et al., 2018b), MSA Transformer (Rao et al., 2021b), use MSA information to model the evolutionary sequence density with potts models, variational auto-encoders, and transformer, respectively. Biswas et al. (2021); Rives et al. (2021); Barrat-Charlaix et al. (2016) consider a semi-supervised manner which adopts a joint training on MSAs and labeled data for the prediction of protein's fitness.In this paper, instead of using MSA information to construct model inputs or for reconstruction, we incorporate MSA information into the training loss in order to learn protein representations with improved understanding of the mutational landscape. In practice, we achieve this by formulating the training loss to prioritize learning the rank order of the position specific amino acid distribution. Additionally, this paradigm shift on the application of MSA information has the benefit of only requiring MSA information at train time and not at inference time.

Protein Language and Structure Models. Protein representation learning borrows various insights from self-supervision research in the natural language processing community (Liu et al., 2019; Yang et al., 2019). The main goal of protein representation learning is to extract biological and functional knowledge of proteins from large unlabeled data to enable zero-shot generalization and/or rapid adaptation to various protein-related tasks. To learn amino acid-level representations from sequence, the community has used methods such as auto-encoding (Shuai et al., 2021), auto-regressive (Rives et al., 2019b; Meier et al., 2021b; Elnaggar et al., 2020; Riesselman et al., 2019), skip-gram language model (Kimothi et al., 2016), mask prediction (Vig et al., 2020; Brandes et al., 2022) or amino acid contrastive learning objectives (Lu et al., 2020), similarity metric learning (Bepler & Berger, 2019; Alley et al., 2019), *etc*. The most renown protein language models (pLMs) are the evolutionary-scale models (ESMs) (Rives et al., 2019a; Meier et al., 2021a) with ESM2 being the most recent and underpins ESMFold, a sequence-based structure prediction framework (Lin et al., 2023).

For protein structures, 3DCNNs (Townshend et al., 2020; Shroff et al., 2020), GNNs (Townshend et al., 2020; Dauparas et al., 2022), and graph-transformers (Diaz et al., 2023)

architectures have been developed to learn residue-level representations using the local chemical environment (microenvironment) or the protein backbone as input. These frameworks primarily use masking to obtain their representations but other pre-training task, such as structure contrastive learning (Moon et al., 2023), distance/angle prediction (Chen et al., 2023a), and denoising (Watson et al., 2023) have been proposed. Several structure-based frameworks have experimentally designed proteins. The microenvironment framework MutCompute (Shroff et al., 2020; d'Oelsnitz et al., 2023) has demonstrated the ability to guide the engineering of several functionally diverse enzymes (Lu et al., 2022; Paik et al., 2021; d'Oelsnitz et al., 2023). Inverse Folding frameworks, such as ESM-IF (Hsu et al., 2022) and ProteinMPNN (Dauparas et al., 2022), use the protein backbone to conditionally design novel sequences for de novo binder design (Watson et al., 2023) and enzyme engineering (Sumida et al., 2024). More works (Chen et al., 2023b; Gligorijević et al., 2021; Zheng et al., 2023; Zhang et al., 2023) focus on the effective knowledge integration between sequence and structure data. Due to the prohibitive cost of training a pLM and the added complexity of decoding an entire protein sequence during inverse folding, we focus on initially validating our EvoRank loss using the microenvironment modality.

3. Methods

This section introduces the main method. We start with introducing the widetype (WT) based mask prediction for self-supervised representation on proteins (Section 3.1). We then propose our two novel techniques: 1) a *MSA-based soft label* to introduce evolution information into the learning (Section 3.2); and 2) a *EvoRank* loss that allows us to fit evolution information more efficiently and robustly with a learning-to-rank idea (Section 3.3).

3.1. Self-Supervised Learning via WT-mask prediction

We are given a protein set $\mathcal{P} = \{P\}$, where the representation $P = (\mathcal{A}, \mathcal{V})$ of each protein consists of both its amino acid sequence $\mathcal{A}$ and atoms information $\mathcal{V}$. The sequence $\mathcal{A} = (aa_j, \cdots, aa_m)$ contains m amino acids, where aa_j is the one-hot representation of the 20 amino acid types. The $\mathcal{V} = \{v_j\}_{j=1}^n$ represents all the atoms contained in the protein, where each v_j contains the information of the j-th atom, including its 3D coordinates, atom type, partial charge and solvent accessible surface, etc.

WT-Mask Prediction In the WT-mask prediction task (Torng & Altman, 2017), we mask an amino acid aa_j, and learn a neural network to predict aa_j back based on the microenvironment surrounding aa_j. The learned network can then provide a useful representation of the protein for downstream tasks. Specifically, denote by $\mathrm{C}_\alpha(aa_j)$ the α-carbon atom of amino acid aa_j, and $\mathrm{Atom}(a_j)$ all the atoms contained in amino acid aa_j. We take the microenvironment of aa_j to be the atoms within $20\mathring{A}$ distance within $\mathrm{C}_\alpha(aa_j)$, excluding all atoms in $\mathrm{Atom}(aa_j)$, that is,

$$\mathcal{V}_j^{\text{mask}} = \{v\colon v \in \mathcal{V} \setminus \mathrm{Atom}(aa_j), \quad \mathrm{Dist}(C_\alpha^j, v) \le 20\mathring{A}\},$$

We train a neural network $y = f(x)$ that takes a microenvironment $x = \mathcal{V}_j^{\text{mask}}$ as input and outputs the logits for the 20 amino acid types. We want to train the model to make $f(\mathcal{V}_j^{\text{mask}}) \approx aa_j$:

$$\min_f \sum_{P\in\mathcal{P}} \sum_j \mathrm{D}(aa_j,\ \mathrm{Softmax}(f(\mathcal{V}_j^{\text{mask}}))),$$

where D denotes the loss function. A typical choice is the KL divergence, which corresponds to the cross entropy loss.

Zero-shot Mutation Effect Prediction Once f is trained, we use its representation to conduct zero-shot mutation effect prediction by taking the log ratio of the amino acid likelihoods.

3.2. Evolution Information via MSA-based Soft Labels

As described in the introduction, we desire a self-supervised learning procedure that (1) discourages low-entropy distributions skewed towards wildtype and (2) incorporates meaningful evolutionary and biochemistry from the input protein structure. Since Multiple sequence alignment (MSA) provides a powerful tool for capturing evolutionary relations between sequences, we propose to incoporate MSA information into the self-supervised learning with an *MSA soft-label loss* (equation 2), where the wildtype one-hot encoded label is replaced with a distribution computed from a protein's MSA, as shown in Figure 1.

Formally, instead of training network f to predict the one-hot vector of the wildtype amino acid, we predict the following soft label based on the following pdf derived from the MSA of the protein:

$$p_j^{\text{MSA}}(\ell) \propto \sum_{P'\in\mathrm{MSA}(P)} \delta(\ell = \mathrm{AA}(P', j)), \qquad (1)$$

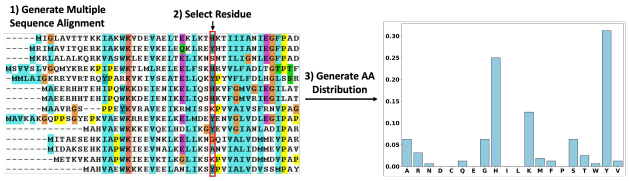


Figure 1. **Generation of position specific empirical amino acid distribution.** From this, we generate the MSA-based soft and rank labels.

where AA is the function that outputs the amino acid type at position j, ℓ is one of the 20 amino acids, δ is the delta function, $\text{MSA}(P)$ denotes the set of sequences that are best aligned with P via multiple sequence alignment on UniRef50 (Consortium, 2015). We refer to this distribution as the empirical amino acid distribution.

We define the MSA soft-label training loss as follows:

$$\min_f \sum_{P \in \mathcal{P}} \sum_j \text{D}(p_j^{\text{MSA}},\ \text{Softmax}(f(\mathcal{V}_j^{\text{mask}}))). \quad (2)$$

Although KL divergence has been the canonical choice, it is known to suffer from mode collapse. We experimented with taking $\text{D}(\cdot;\cdot)$ within a richer family of α-divergences. By applying different α values, we can adjust the sensitivity to multimodal distributions present in MSAs and find a better trade-off between over/under estimates of the top ranked amino acid (which is often wildtype). When we apply reverse KL divergence or $\alpha = 0.5$ divergence (Table 2), we observe marginally improved rank order but overall lower coefficients for the top-5 amino acids. This suggests the need for designing better loss functions.

Divergence	Label	Top-5	Top-10	20
KL Div	WT	0.54	0.38	0.28
KL Div	MSA	0.60	0.52	0.34
Reverse KL Div	MSA	0.54	0.56	0.40
Alpha Div ($\alpha = 0.5$)	MSA	0.57	0.53	0.40

Table 2. **Spearman correlation coefficient for amino acids at the same local chemical environment.** We deploy this on the test dataset for the mask prediction task. Here, 'Top-5' indicates the amino acids with the top-5 probability score based on the empirical amino acid distribution.

3.3. EvoRank: A New Rank-based Learning Objective

To further improve the performance of the self-supervised model for protein engineering tasks, we reformulate the training objective into a rank loss in order to learn representations that better model how evolution drives protein mutations. Rather than predicting the wildtype amino acid type aa_j or soft label p_j^{MSA}, we modified the architecture to use a mutation as an additional input. This mutation is represented as a "from" (aa^-) and "to" (aa^+) amino acid type embedding pair, and we predict their relative likelihood within the empirical amino acid distribution as proxy for evolutionary fitness. More precisely, we define the rank label of aa_j w.r.t. (aa^+, aa^-) as the following

$$r_i(aa^+, aa^-) = \frac{p_j^{\text{MSA}}(aa^+)}{p_j^{\text{MSA}}(aa^+) + p_j^{\text{MSA}}(aa^-)} - \frac{1}{2}, \quad (3)$$

where $p_j^{\text{MSA}}(a)$ denotes the probability assigned on aa according to p_j^{MSA}, and $\frac{1}{2}$ to ensure neutral predictions are made when $p_j^{\text{MSA}}(aa^+) = p_j^{\text{MSA}}(aa^-)$. The rank label, also referred to as the *EvoScore*, represents the relative likelihood of two amino acids being evolutionarily observed in a particular microenvironment, as demonstrated in Figure 2. We train a model $f(\mathcal{V}_j^{\text{mask}},\ aa^+, aa^-)$ to predict the rank label $r_j(aa^+, aa^-)$ via the following loss:

$$\min_f \sum_{P \in \mathcal{P}} \sum_j \sum_{aa^+, aa^-} \text{D}(r_j(aa^+, aa^-),\ f(\mathcal{V}_j^{\text{mask}}, aa^+, aa^-)), \quad (4)$$

where the aa^+, aa^- are summed on all the amino acid types and $\text{D}(x, y) = ||x - y||$. We refer to the loss in (4) as the EvoRank loss or EvoRank training objective.

In practice, we first initialize the parameters by training using the MSA soft-label loss (equation (2)) and then apply the EvoRank loss to further improve performance. Similar ideas are used in the recommendation system literature (*e.g.*, Cao et al., 2007; Aggarwal et al., 2016; Liu et al., 2009), where parameters are initially trained with a standard loss and then further refined using a ranking loss.

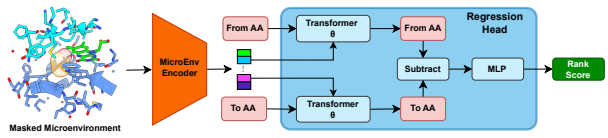


Figure 2. **Overview of the MutRank architecture.** In the regression head, a Siamese network is used to contextualize the "from" and "to" amino acid embeddings with the hidden representation of the microenvironment and produce a rank representation, which is decoded into a rank score.

Model Architecture The microenvironment-based model used here is based on previous work by Diaz et al. (2023). Briefly, the model uses a graph transformer backbone to process an input microenvironment, in which $\mathcal{V}_i^{\text{mask}}$ for amino acid a_i is the input and each atom in this set is represented by its 3D coordinates, atom type, partial charge and solvent accessible surface area. The atomic inputs are embedded and processed using graph transformer blocks, where the attention bias is based on the atom-wise Euclidean distance. We refer the readers to (Diaz et al., 2023) for more details on the graph transformer backbone architecture.

The regression head accepts two structural amino acid embedding vectors and the hidden representation of the microenvironment from the graph-transformer backbone as input. As shown in Figure 2, we use Siamese network architecture to contextualize each amino acid type to the masked microenvironment, and a MLP to decode a ranking prediction between the two contextualized amino acid embeddings. We refer the readers to (Diaz et al., 2023) for more details on the regression head architecture.

-	T2837		S669		S-Sym		Myolobin		FireProtDB		Gβ1		T2837 Reverse	
# Mutations	2837		669		342		134		1764		935		2837	
Metric	$\rho\uparrow$	AUC $\uparrow$	$\rho\uparrow$	AUC $\uparrow$	$\rho\uparrow$	AUC $\uparrow$	$\rho\uparrow$	AUC $\uparrow$	$\rho\uparrow$	AUC $\uparrow$	$\rho\uparrow$	AUC $\uparrow$	$\rho\uparrow$	AUC $\uparrow$
RaSP* (Blaabjerg et al., 2023)	0.58	0.61	0.39	0.69	0.64	0.73	**0.68**	0.75	0.56	0.71	**0.72**	0.66	0.23	0.59
ThermoMPNN* (Dieckhaus et al., 2023)	0.55	0.78	0.39	0.68	0.66	0.82	0.58	0.77	0.57	0.75	0.65	0.78	0.43	0.71
Prostata-IFML (Diaz et al., 2023)	0.53	0.75	0.49	0.76	0.55	0.75	0.54	0.67	-	-	0.66	**0.82**	0.52	0.75
Stability Oracle (Diaz et al., 2023)	**0.59**	**0.81**	**0.52**	**0.74**	**0.72**	**0.87**	**0.68**	**0.81**	**0.61**	**0.79**	**0.71**	**0.82**	**0.59**	**0.81**
ESM2* (Lin et al., 2023)	0.28	0.60	0.04	0.50	0.26	0.56	0.15	0.57	0.25	0.57	0.25	0.63	0.28	0.60
ProteinMPNN* (Dauparas et al., 2022)	0.36	0.70	0.25	0.59	0.32	0.64	0.35	0.66	0.31	0.70	0.35	0.67	0.36	0.70
MutComputeXGT (Diaz et al., 2023)	0.34	0.68	0.27	0.57	0.38	0.72	0.37	0.72	0.30	0.69	0.34	0.66	0.34	0.68
MutComputeXGT w/ MSA soft-label (Ours)	0.37	0.70	0.30	0.59	0.48	0.75	0.45	0.75	0.36	0.71	0.41	0.69	0.37	0.70
MutRank (Ours)	**0.51**	**0.78**	**0.40**	**0.67**	**0.62**	**0.84**	**0.68**	**0.84**	**0.51**	**0.77**	**0.62**	**0.77**	**0.51**	**0.78**
SSL Improvement $\uparrow$	42%	11%	48%	14%	63%	17%	83%	17%	65%	10%	77%	15%	42%	11%
Supervised Fine-Tuning Gap $\downarrow$	14%	4%	23%	9%	14%	3%	0%	-4%	16%	3%	13%	3%	14%	4%

Table 3. **Thermodynamic stability ($\Delta\Delta G$) zero-shot comparisons with self-supervised and fine-tuned frameworks.** ρ equals the Pearson correlation coefficient and AUC is the area under the receiver operating characteristic. The first block reports the performance of frameworks fine-tuned using experimental $\Delta\Delta G$ datasets. The second block reports the performance of self-supervised models common in the literature. The third block reports the performance of two models trained in this work. The first is trained only using the MSA soft-label loss and the second is MutRank, trained with both the MSA soft-label loss and the EvoRank loss (see Section 4.2). 'SSL Improvement' compares MutRank with respect to the best zero-shot model in the second block. 'Supervised Fine-Tuning Gap' compares MutRank with respect to the best supervised $\Delta\Delta G$ model in the first block. T2837 Reverse flips the "From" and "To" amino acids in T2837 to evaluate reversibility robustness. * denotes that we compute the metrics using the official checkpoint.

Dataset	Phenotype	# Mut	MutComputeXGT			MutRank			ESM2			Stability Oracle		
			Pearson	Spearman	AUC	Pearson	Spearman	AUC	Pearson	Spearman	AUC	Pearson	Spearman	AUC
levoglucosan kinase	ΔSolubility	7195	0.26	0.30	0.61	0.29	**0.34**	**0.64**	0.27	0.32	0.62	**0.32**	**0.34**	0.63
TEM1-β-Lactamase	ΔSolubility	4345	0.16	0.21	0.60	**0.22**	**0.26**	**0.64**	0.08	0.18	0.61	0.10	0.16	0.60
AcrIIA4	Activity	1653	0.36	0.34	0.65	**0.59**	**0.53**	**0.75**	0.06	0.06	0.56	0.48	0.40	0.69
Amidase	Activity	6227	0.38	0.39	0.66	**0.64**	**0.64**	**0.83**	0.56	0.56	0.78	0.48	0.46	0.75
Deaminase	Activity	5689	0.26	0.26	0.63	**0.41**	**0.42**	**0.73**	0.38	0.39	0.70	0.24	0.24	0.63
SKEMPI-V2	Protein-Protein $\Delta\Delta G_{\text{bind}}$	4102	0.28	0.26	0.62	**0.42**	**0.42**	**0.69**	0.23	0.19	0.57	0.39	0.39	0.67
S487	Protein-Protein $\Delta\Delta G_{\text{bind}}$	487	0.24	0.25	0.58	**0.38**	**0.38**	0.67	0.01	0.01	0.48	**0.38**	**0.38**	**0.70**
PlatinumDB	Protein-Ligand $\Delta\Delta G_{\text{bind}}$	925	0.05	0.01	0.48	**0.28**	**0.28**	**0.64**	0.03	0.06	0.51	0.26	0.26	**0.64**
ABBind	Antibody-Antigen $\Delta\Delta G_{\text{bind}}$	309	0.36	0.42	0.73	**0.41**	**0.46**	**0.74**	-0.07	-0.05	0.60	0.38	0.42	0.72

Table 4. **Zero-shot evaluations on solubility and binding free energy datasets.** In comparison with both sequence and structure-based models trained using wildtype accuracy, training a structure-based model with EvoRank leads to greatly improved zero-shot performance. Stability Oracle is initialized with MutComputeXGT weights and fine-tune for $\Delta\Delta$G prediction (Diaz et al., 2023)

4. Experimental Results

For our experiments, we retrained a SOTA structure model (Diaz et al., 2023) using both the MSA soft-label loss and the EvoRank loss as described in Section 4.2. We name the MutComputXGT structure model trained with EvoRank loss as MutRank. We did not retrain ProteinMPNN, another SOTA structure model, as it is unclear how to incorporate the loss with its encoder/decoder architecture. We lack the computational resources to retrain ESM2, a billion-parameter SOTA sequence model.

We empirically evaluate MutRank on the most commonly studied point mutation phenotype prediction benchmarks in protein engineering: thermodynamic stability ($\Delta\Delta G$) and binding free energy. Additionally, we evaluate two solubility and three activity DMS datasets.

For zero-shot evaluation, we directly regress the model's prediction for the EvoScore (equation 3) against the experimental label. For classification, we predict whether the change was positive or negative and report AUC. We used the cDNA117K dataset (Diaz et al., 2023) to evaluate the supervised fine-tuning of EvoRank representations. Since EvoRank is used for self-supervised pre-training, the supervised fine-tuning method is identical to the one described in (Diaz et al., 2023).

We compare the predictions of MutRank to self-supervised models trained with different modalities, *e.g.*, structure-based (MutComputeXGT), sequence-based (ESM2), and inverse folding (ProteinMPNN) models. Our results show that EvoRank leads to large improvements across the board for zero-shot prediction. In fact, our zero-shot predictions are competitive even with models that have been fine-tuned on phenotype-specific datasets. For thermodynamic stability, supervised fine-tuning MutRank gives a more modest improvement compared to SOTA fine-tuned models.

4.1. Datasets

For the self-supervised training, we use the same procedure as MutComputeX (d'Oelsnitz et al., 2023). Briefly, this dataset consists of a 90:10 split of 2,569,256 microenvironments sampled from 22,759 protein sequences clustered at 50% sequence similarity and having a structure resolution of at least 3Å from the RCSB (November 2021). Our test data for the folding free energy changes and binding free energy changes are proposed in Diaz et al. (2023); Gong

et al. (2023) and we refer the readers to these works for details. These datasets are curated from literature datasets and incorporate additional policies (*e.g.*, below 30% sequence similarity between training and test sets) for better quality.

For mutation effect prediction tasks, we use the experimental structure files from RCSB and AlphaFold structures if the protein lacks an experimental structure. Due to the prohibitive cost of generating experimental data, no phenotype has sufficient experimental data to properly benchmark ML frameworks and evaluate generalization. Thus, we explore datasets for several phenotypes. To date, the most characterized mutation effect phenotype is thermodynamic stability of folding ($\Delta\Delta G$) with several established datasets reserved for evaluation of computational tools: S-Sym, S669, T2837, Gβ1, Myoglobin, and P53. Recently, a cDNA-display proteolysis technique enabled the multiplex characterization of single-domain mini-proteins to provide the first exhaustive, systematically generated training set for machine learning (Tsuboyama et al., 2023). However, this dataset used a proteolytic stability as proxy for thermodynamic stability and the technique does not generalize to full-length functional proteins. For evaluating against the binding free energy changes of point mutations, we used SKEMPIv2 (Jankauskaitė et al., 2019) and AB-Bind (Sirin et al., 2016) for protein-protein interface and PlatinumDB for protein-ligand interface (Pires et al., 2015). For the activity, we used an anti-CRISPR protein (A0A247D711) (Stadelmann et al., 2021) and an amidase (Wrenbeck et al., 2017). These datasets are curated from the literature, thus, different techniques–with different biases–were used for data collection. Thus, we filtered mutational data for the techniques known to provide higher-quality measurements: SPR, ITC, FL, IASP, SFFL. To evaluate a non-thermodynamic phenotype, we evaluate against the solubility change deep mutational scanning (DMS) datasets of levoglucosan kinase and TEM1-β-lactamase (Klesmith et al., 2017). To obtain these solubility change measurements, a yeast surface display readout was used not of their wildtype sequences but rather for chimeric variants with a N-terminus Aga2p domain and a C-terminus epitope tag. Thus, solubility change results should be interpreted with caution since the input sequence and structure used to generate predictions are for the native proteins.

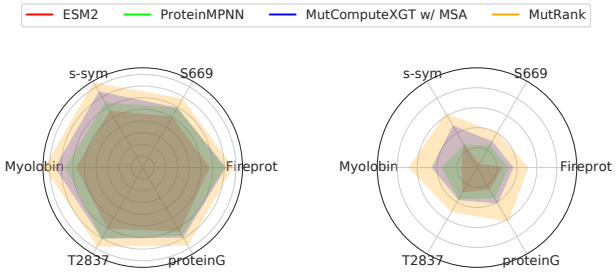
4.2. Training

Self-supervised training was done with the AdamW optimizer, 512 batch size, 5×10^{-5} learning rate, 10^{-5} weight decay. We first train using the soft-label loss in equation (2) for 100K iterations, and then refine with the *EvoRank* loss defined in equation (4), for an additional 100K iterations. Training the model typically requires approximately two GPU days on one A100. We generate MSAs with JackHMMer (Remmert et al., 2012) against UniRef90, using the default configuration of AlphaFold2. For the supervised fine-tuning, we apply the AdamW optimizer, with backbone learning rate 10^{-5} and regression head learning rate 5×10^{-5}. We tune it with 500 iterations on the curated cDNA dataset generated by Diaz et al. (2023).

Evaluation Metrics and Baselines We assess the model's performance using a comprehensive set of evaluation metrics encompassing both regression and classification aspects. The regression metrics include the Spearman correlation coefficient, Pearson correlation coefficient, and Root Mean Squared Error (RMSE). For classification evaluation, we employ AUROC (Area Under the Receiver Operating Characteristic curve). This dual approach ensures a thorough and nuanced evaluation of the model's capabilities across different dimensions of prediction tasks. To compare with results in the literature, we report the Spearman correlation on different DMS datasets. To establish baselines, we incorporate a range of self-supervised and supervised methods. As a representative self-supervised method, we employ the extensively used ESM2 models. The default baseline is set with the 650M-parameter ESM2 model, and we provide results for other scales of ESM2 models and alternative protein language models. We first evaluate different model performance first on different $\Delta\Delta$G datasets, since these datasets have high-quality labels. Then, we further compare models on more phenotype datasets, to examine whether our model can generalize to different settings.

4.3. Results

Zero-shot thermodynamic stability evaluations In Table 3 and Figure 3 report the zero-shot Pearson correlation coefficient (ρ) and area under the ROC curve (AUC) performance of various machine learning frameworks across multiple $\Delta\Delta$G datasets: T2837 (Forward and Reverse) (Diaz et al., 2023), S-Sym (Li et al., 2020), S669 (Pancotti et al., 2022), FireProtDB (Stourac et al., 2021), Gβ1 (Nisthal et al., 2019), and Myoglobin (Li et al., 2020). Our results affirm the significance of prioritizing rank order during self-



AUC Pearson Correlation

Figure 3. **Comparisons between the self-supervised frameworks on stability datasets.** We refer readers to Table 3 for exact numbers and comparisons with supervised models.

# Proteins	# Mut (K)	Fine-Tune MutComputeXGT w/ WT-Mask				Fine-Tuned MutComputeXGT w/ EvoRank			
		Pearson	Spearman	AUC	RMSE	Pearson	Spearman	AUC	RMSE
10	11K	0.50	0.52	0.73	1.92	0.58	0.60	0.78	1.73
50	54K	0.55	0.58	0.77	1.78	0.59	0.61	0.80	1.66
116	117K	0.59	0.62	0.81	1.64	0.61	0.63	0.81	1.62

Table 5. **Evaluation of supervised fine-tuning SSL representations.** We fine-tune both WT-mask and EvoRank representations on various subsets of the cDNA dataset (Diaz et al., 2023) and test on the T2837 dataset set. '#Mut' denotes number of mutations in the training set. The learning rate and number of iterations are tuned for each SSL pretraining task in order to maximize performance.

supervised training for zero-shot $\Delta\Delta G$ predictions.

Training with the MSA-based soft labels was sufficient to surpass literature self-supervised baselines in both Pearson correlation and AUC. However, by reformulating the training objective with EvoRank, MutRank achieve a substantial improvement over the previous best literature zero-shot model. Moreover, MutRank demonstrates robustness to thermodynamic reversibility augmentation (T2837 Reverse). Direct comparison with its WT-masked predecessor, MutComputeXGT, MutRank achieves a 66% and 16% improvement in Pearson correlation and AUC, respectively. Notably, compared to the well-known self-supervised methods ESM2 and ProteinMPNN, MutRank achieves an average Pearson correlation improvements of $\sim$288% and $\sim$72% across the six $\Delta\Delta G$ datasets, respectively. These results demonstrate the effectiveness of EvoRank representations for $\Delta\Delta G$ mutation effect prediction.

Next, we compared against supervised fine-tuned frameworks trained on the large scale cDNA $\Delta\Delta G$ dataset. While the MSA-based soft label framework fell short in competitiveness, MutRank demonstrated competitive performance. Compared to the SOTA-supervised framework, Stability Oracle, MutRank zero-shot Pearson correlation and AUC are only $\sim$13% and $\sim$3% lower on average across the six datasets. Overall, our results demonstrate how pretraining with the EvoRank loss significantly narrows the gap between supervised fine-tuned and self-supervised frameworks for $\Delta\Delta G$ mutation effect prediction.

Zero-shot evaluation on multiple phenotypes To further characterize the generalization capability of MutRank , we evaluate performance on binding free energy change datasets and four DMS datasets: two for solubility and two for activity (Table 4). Unlike folding stability, which has seen significant increases in available public data (Tsuboyama et al., 2023), binding free energy change datasets are scarce, filled with mutation type and label biases, and suffer from noisy labels. These challenges make developing supervised frameworks challenging for these phenotypes and underline the importance of improving the zero-shot performance of self-supervised frameworks.

For the binding free energy datasets, we use the protein-protein interface binding $\Delta\Delta G$ datasets SKEMPIv2 (Jankauskaitė et al., 2019), AB-Bind (Sirin et al., 2016), S487 (Geng et al., 2019) and the protein-ligand interface binding $\Delta\Delta G$ dataset PlatinumDB (Pires et al., 2015). For the solubility and activity datasets, we used Deep Mutational Scanning (DMS) datasets, which leverage a high throughput screen or next-generation sequencing as a proxy for function. For solubility, we use the DMS datasets for for levoglucosan kinase (uniprot id:B3VI55) and TEM1-β-Lactamase (uniprot id: P62593) from Klesmith et al. (2017). For activity evaluation, we use the DMS datasets for the aliphatic hydrolase (uniprot id: P11436), the Anti-CRISPR protein AcrIIA4 (uniprot id: A0A247D711), and Porphobilinogen deaminase (uniprot id: P08397). We compare against two WT-mask SSL frameworks, MutComputeXGT and ESM2, and one supervised fine-tune framework, Stability Oracle. Comparison between just the literature methods on the binding $\Delta\Delta G$ datasets demonstrate that ESM2 did the worst and Stability Oracle did the best across all metrics (Pearson and Spearman correlation and AUC). These results are expected since binding free energy (interactions between proteins) is fundamentally related to folding free energy (interactions within a protein). ESM2 is unable to "see" the binding partner (protein or ligand) and must rely purely on the single sequence representation. This most likely explains ESM2 low performance on AcrIIA4 since it illicits its anti-CRISPR activity through binding inhibition.

Remarkably, MutRank outperforms MutcomputeXGT across all datasets for all metrics. This demonstrates that the EvoRank loss improves zero-shot generalization across multiple phenotypes compared to its WT-masked predecessor. Additionally, MutRank outperforms ESM2 on all datasets for all metrics even though it is a much smaller model trained on only $\sim$23K proteins compared to UniRef50. Surprisingly, MutRank's zero-shot performance surpasses or ties Stability Oracle performance on nearly all metrics for binding $\Delta\Delta G$ datasets (except S487 AUC). Furthermore, it significantly outperforms Stability Oracle on the TEM1-β-Lactamase solubility dataset and the three activity datasets. Stability Oracle performance on the TEM1-β-Lactamase dataset is lower than it's pre-trained representation, MutComputeXGT. This finding highlights the superior phenotype generalization of EvoRank loss and demonstrates how supervised fine-tuning can improve the performance on one phenotype at the expense of others. Finally, we highlight

MutRank's substantial improvement on the protein-ligand interface binding $\Delta\Delta G$ dataset, PlatinumDB: compared to MutComputeXGT: $\mathrm{MutRank}$ improves the Pearson correlation and AUC from 0.05 and 0.48 to 0.28 and 0.64, respectively. We conclude that for the activity, solubility and binding free energy phenotypes, $\mathrm{MutRank}$ representations significantly improves the zero-shot generalization over the WT-mask representations of MutComputeXGT. However, more thorough evaluations of $\mathrm{MutRank}$ are needed to better understand its generalization across phenotypes.

Impact on supervised fine-tuning Representation learning is essential for enabling transfer learning to domains with limited labeled data. Therefore, to assess the effectiveness of EvoRank representations compared to the WT-mask representations, we conduct a comparative analysis (Table 5) of supervised fine-tuning for thermodynamic stability using the Stability Oracle framework: train on cDNA and evaluate on T2837. The evaluation metrics include Pearson correlation, Spearman correlation, AUC, and RMSE. To achieve optimal performance, WT-mask and EvoRank representations require 3000 (same as Stability Oracle) and 500 iterations of fine-tuning, respectively.

Our results demonstrate that both models reach approximately the same performance on T2837 from training on the cDNA dataset, with EvoRank representation having a marginal improvement. Interestingly, EvoRank loss impact is most apparent when there is significantly less fine-tuning data available. When fine-tuned with $\sim$9% of the proteins (10 proteins and 11K mutations) in the cDNA dataset, EvoRank representations outperforms WT-mask representations by 16%, 15%, 7% for Pearson and Spearman correlation and AUC, respectively, and required 6$\times$ fewer training iterations (500 vs 3000). Furthermore, EvoRank representation's Pearson and Spearman correlation and AUC metrics are only 2%, 3%, and 4% lower than Stability Oracle, respectively. The corresponding WT-mask representation's Pearson and Spearman correlation and AUC metrics are 15%, 16%, and 10% lower than Stability Oracle, respectively. These gaps are less drastic when 43% of the proteins (50 proteins and 54K mutations) are used for supervised fine-tuning. We posit that this is because EvoRank representations have nearly saturated the capacity of the cDNA dataset. We conclude that the supervised fine-tuning of the EvoRank representations can substantially enhance the model's ability to generalize to phenotypes with limited mutational data, while also accelerating training time.

Generalizing beyond the MSA distribution While $\mathrm{MutRank}$ is trained to recapitulate MSA priors via the EvoRank loss, the MSA itself can also directly serve as a predictor for mutation effects. In the literature, MSAs are often used to create a sequence profile (Lüthy et al., 1994) or position-specific scoring matrix (PSSM) (Jones, 1999), which can be used to predict the impact of a mutation by assessing the deviation from the expected amino acid at a specific position.

Method	Pearson	Spearman	AUROC	RMSE
EvoScore	0.15	0.13	0.62	2.17
ESM2	0.37	0.37	0.65	5.48
MutComputeXGT	0.38	0.38	0.64	1.89
$\mathrm{MutRank}$	0.45	0.46	0.71	1.09

Table 6. **Zero-shot evaluations on the cDNA117K dataset.** We demonstrate that MutRank has superior generalization capability over the EvoScore used for training.

We evaluate if $\mathrm{MutRank}$ can outperform the MSA priors it is trained to predict using the large cDNA dataset ($\sim$ 117K mutations from 116 single domain proteins) provided in Diaz et al. (2023). For these 116 proteins, the average and std of their MSA depth is 3.9K$\pm$0.6K sequences. To calculate naive predictions from a protein's MSA, we use the log-odds of the empirical amino acid distribution at a position (Figure 1): $\log(p_{\mathrm{to}}/p_{\mathrm{from}})$. Furthermore, we provide MutComputeXGT and ESM2 as additional baselines for comparison. As demonstrated in Table 6, our method not only outperforms ESM2 but also significantly improves upon the MSA priors it is trained to predict: for Pearson correlation, our method achieves 0.45, surpassing MSA's 0.15, ESM2's 0.37, and MutComputeXGT's 0.38. These results demonstrate that $\mathrm{MutRank}$ capture residue specific variability beyond what is encoded in MSAs.

5. Conclusion

We propose the EvoRank self-supervised training objective to enhance protein representation learning for zero-shot mutation effect prediction tasks. EvoRank reformulates the learning task to capture how evolution drives protein mutations by replacing the 20-class classification head with a regression head that is trained to recapitulate the rank of amino acids at a specific position observed in MSA. To evaluate EvoRank, we trained a structure-based graph transformer and observed performance improvements in all downstream single-point mutation effect prediction tasks compared to WT-mask pretrained predecessor, MutCopmputeXGT. Furthermore, when compared to the most renowned sequence-based (ESM2) and structure-based (ProteinMPNN) self-supervised frameworks, EvoRank also demonstrates superior zero-shot performance across all evaluated benchmarks. Notably, the EvoRank loss enables training the first self-supervised framework that is competitive with $\Delta\Delta G$ supervised fine-tuned frameworks. From our results, we conclude that the EvoRank training objective learns representations with an enriched understanding of the complex mutational landscape of proteins and improve generalization when fine-tuning to downstream phenotypes with limited data.

Impact Statement

In this paper, we present the EvoRank training loss and demonstrate its ability to learn representations that better model a protein's mutational landscape. Models trained with EvoRank are better suited for machine learning-guided protein engineering and will drastically accelerate the development of protein-based biotechnologies. Protein-based biotechnologies will have a profound impact on the sustainable procurement of agricultural and chemical commodities, pharmaceuticals, and food ingredients. We would like to highlight that the mutational understanding present in the representations learned with the EvoRank loss can help with the identification of synthetic sequences for pathogen components. This issue will become more prevalent with the rise of machine learning guided de novo protein design.

Acknowledgement

This work is supported by the NSF AI Institute for Foundations of Machine Learning (IFML). We would like to thank the reviewers for taking the time and effort necessary to review the manuscript. We sincerely appreciate all valuable comments and suggestions, which helped us to improve the quality of the manuscript.

References

Aggarwal, C. C. et al. *Recommender systems*, volume 1. Springer, 2016.

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

Barrat-Charlaix, P., Figliuzzi, M., and Weigt, M. Improving landscape inference by integrating heterogeneous data in the inverse ising problem. *Scientific Reports*, 6(1):37812, 2016.

Bepler, T. and Berger, B. Learning protein sequence embeddings using information from structure. In *International Conference on Learning Representations*, 2019.

Biswas, S., Khimulya, G., Alley, E. C., Esvelt, K. M., and Church, G. M. Low-n protein engineering with data-efficient deep learning. *Nature methods*, 18(4):389–396, 2021.

Blaabjerg, L. M., Kassem, M. M., Good, L. L., Jonsson, N., Cagiada, M., Johansson, K. E., Boomsma, W., Stein, A., and Lindorff-Larsen, K. Rapid protein stability prediction using deep learning representations. *eLife*, 12:e82593, 2023.

Brandes, N., Ofer, D., Peleg, Y., Rappoport, N., and Linial, M. Proteinbert: A universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110, 2022.

Braunisch, M. C., Riedhammer, K. M., Herr, P.-M., Draut, S., Günthner, R., Wagner, M., Weidenbusch, M., Lungu, A., Alhaddad, B., Renders, L., et al. Identification of disease-causing variants by comprehensive genetic testing with exome sequencing in adults with suspicion of hereditary fsgs. *European Journal of Human Genetics*, 29(2):262–270, 2021.

Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., and Li, H. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pp. 129–136, 2007.

Chen, C., Zhou, J., Wang, F., Liu, X., and Dou, D. Structure-aware protein self-supervised learning. *Bioinformatics*, 39(4):btad189, 2023a.

Chen, T., Gong, C., Diaz, D. J., Chen, X., Wells, J. T., qiang liu, Wang, Z., Ellington, A., Dimakis, A., and Klivans, A. Hotprotein: A novel framework for protein thermostability prediction and editing. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=YDJRFWBMNby.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

Consortium, U. Uniprot: a hub for protein information. *Nucleic acids research*, 43(D1):D204–D212, 2015.

Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R. J., Milles, L. F., Wicky, B. I., Courbet, A., de Haas, R. J., Bethel, N., et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378 (6615):49–56, 2022.

Dehghani, M., Djolonga, J., Mustafa, B., Padlewski, P., Heek, J., Gilmer, J., Steiner, A. P., Caron, M., Geirhos, R., Alabdulmohsin, I., et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pp. 7480–7512. PMLR, 2023.

Diaz, D. J., Gong, C., Ouyang-Zhang, J., Loy, J. M., Wells, J., Yang, D., Ellington, A. D., Dimakis, A., and Klivans, A. R. Stability oracle: A structure-based graph-transformer for identifying stabilizing mutations. *bioRxiv*, pp. 2023–05, 2023.

Dieckhaus, H., Brocidiacono, M., Randolph, N., and Kuhlman, B. Transfer learning to leverage larger datasets for improved prediction of protein stability changes. *bioRxiv*, 2023.

d'Oelsnitz, S., Diaz, D. J., Acosta, D. J., Schechter, M. W., Minus, M. B., Howard, J. R., Do, H., Loy, J., Alper, H., and Ellington, A. D. Synthetic microbial sensing and biosynthesis of amaryllidaceae alkaloids. *bioRxiv*, pp. 2023–04, 2023.

Elnaggar, A., Heinzinger, M., Dallago, C., Rihawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., et al. Prottrans: towards cracking the language of life's code through self-supervised deep learning and high performance computing. *arXiv preprint arXiv:2007.06225*, 2020.

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2021.

Evans, R., O'Neill, M., Pritzel, A., Antropova, N., Senior, A., Green, T., Žídek, A., Bates, R., Blackwell, S., Yim, J., et al. Protein complex prediction with alphafold-multimer. *biorxiv*, pp. 2021–10, 2021.

Geng, C., Vangone, A., Folkers, G. E., Xue, L. C., and Bonvin, A. M. isee: Interface structure, evolution, and energy-based machine learning predictor of binding affinity changes upon mutations. *Proteins: Structure, Function, and Bioinformatics*, 87(2):110–119, 2019.

Gligorijević, V., Renfrew, P. D., Kosciolek, T., Leman, J. K., Berenberg, D., Vatanen, T., Chandler, C., Taylor, B. C., Fisk, I. M., Vlamakis, H., et al. Structure-based protein function prediction using graph convolutional networks. *Nature communications*, 12(1):3168, 2021.

Gong, C., Klivans, A., Wells, J., Loy, J., Dimakis, A., Diaz, D., et al. Binding oracle: Fine-tuning from stability to binding free energy. In *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*, 2023.

Hopf, T. A., Ingraham, J. B., Poelwijk, F. J., Schärfe, C. P., Springer, M., Sander, C., and Marks, D. S. Mutation effects predicted from sequence co-variation. *Nature biotechnology*, 35(2):128–135, 2017.

Hsu, C., Verkuil, R., Liu, J., Lin, Z., Hie, B., Sercu, T., Lerer, A., and Rives, A. Learning inverse folding from millions of predicted structures. *bioRxiv*, 2022. doi: 10.1101/2022.04.10.487779. URL https://www.biorxiv.org/content/early/2022/04/10/2022.04.10.487779.

Ingraham, J., Garg, V., Barzilay, R., and Jaakkola, T. Generative Models for Graph-Based Protein Design. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/f3a4ff4839c56a5f460c88cce3666a2b-Paper.pdf.

Jankauskaitė, J., Jiménez-García, B., Dapkūnas, J., Fernández-Recio, J., and Moal, I. H. Skempi 2.0: an updated benchmark of changes in protein–protein binding energy, kinetics and thermodynamics upon mutation. *Bioinformatics*, 35(3):462–469, 2019.

Jones, D. T. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of molecular biology*, 292(2):195–202, 1999.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Kimothi, D., Soni, A., Biyani, P., and Hogan, J. M. Distributed representations for biological sequence analysis. *arXiv preprint arXiv:1608.05949*, 2016.

Klesmith, J. R., Bacik, J.-P., Wrenbeck, E. E., Michalczyk, R., and Whitehead, T. A. Trade-offs between enzyme fitness and solubility illuminated by deep mutational scanning. *Proceedings of the National Academy of Sciences*, 114(9):2265–2270, 2017.

Kouba, P., Kohout, P., Haddadi, F., Bushuiev, A., Samusevich, R., Sedlar, J., Damborsky, J., Pluskal, T., Sivic, J., and Mazurenko, S. Machine learning-guided protein engineering. *ACS catalysis*, 13(21):13863–13895, 2023.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Li, B., Yang, Y. T., Capra, J. A., and Gerstein, M. B. Predicting changes in protein thermodynamic stability upon point mutation with deep 3d convolutional neural networks. *PLoS computational biology*, 16(11):e1008291, 2020.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., and Rives, A. Evolutionary-scale prediction of atomic-level protein structure with a language

model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574.

Liu, T.-Y. et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3): 225–331, 2009.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Lu, A. X., Zhang, H., Ghassemi, M., and Moses, A. Self-supervised contrastive learning of protein representations by mutual information maximization. *BioRxiv*, 2020.

Lu, H., Diaz, D. J., Czarnecki, N. J., Zhu, C., Kim, W., Shroff, R., Acosta, D. J., Alexander, B. R., Cole, H. O., Zhang, Y., et al. Machine learning-aided engineering of hydrolases for pet depolymerization. *Nature*, 604(7907): 662–667, 2022.

Lüthy, R., Xenarios, I., and Bucher, P. Improving the sensitivity of the sequence profile method. *Protein Science*, 3 (1):139–146, 1994.

Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., and Rives, A. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in Neural Information Processing Systems*, 34:29287–29303, 2021a.

Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., and Rives, A. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in Neural Information Processing Systems*, 34:29287–29303, 2021b.

Moon, K., Im, H.-J., and Kwon, S. 3d graph contrastive learning for molecular property prediction. *Bioinformatics*, 39(6):btad371, 2023.

Nisthal, A., Wang, C. Y., Ary, M. L., and Mayo, S. L. Protein stability engineering insights revealed by domain-wide comprehensive mutagenesis. *Proceedings of the National Academy of Sciences*, 116(33):16367–16377, 2019.

Notin, P., Dias, M., Frazer, J., Hurtado, J. M., Gomez, A. N., Marks, D., and Gal, Y. Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval. In *International Conference on Machine Learning*, pp. 16990–17017. PMLR, 2022.

Paik, I., Ngo, P. H., Shroff, R., Diaz, D. J., Maranhao, A. C., Walker, D. J., Bhadra, S., and Ellington, A. D. Improved bst dna polymerase variants derived via a machine learning approach. *Biochemistry*, 62(2):410–418, 2021.

Pancotti, C., Benevenuta, S., Birolo, G., Alberini, V., Repetto, V., Sanavia, T., Capriotti, E., and Fariselli, P. Predicting protein stability changes upon single-point mutation: a thorough comparison of the available tools on a new dataset. *Briefings in Bioinformatics*, 23(2): bbab555, 2022.

Pires, D. E., Blundell, T. L., and Ascher, D. B. Platinum: a database of experimentally measured effects of mutations on structurally defined protein–ligand complexes. *Nucleic acids research*, 43(D1):D387–D391, 2015.

Qiao, Z., Nie, W., Vahdat, A., Miller III, T. F., and Anandkumar, A. State-specific protein-ligand complex structure prediction with a multi-scale deep generative model. *Preprint at arXiv https://doi. org/10.48550/arXiv*, 2209, 2023.

Rao, R. M., Liu, J., Verkuil, R., Meier, J., Canny, J., Abbeel, P., Sercu, T., and Rives, A. Msa transformer. In *International Conference on Machine Learning*, pp. 8844–8856. PMLR, 2021a.

Rao, R. M., Liu, J., Verkuil, R., Meier, J., Canny, J., Abbeel, P., Sercu, T., and Rives, A. Msa transformer. In *International Conference on Machine Learning*, pp. 8844–8856. PMLR, 2021b.

Remmert, M., Biegert, A., Hauser, A., and Söding, J. Hhblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment. *Nature methods*, 9(2):173–175, 2012.

Riesselman, A., Shin, J.-E., Kollasch, A., McMahon, C., Simon, E., Sander, C., Manglik, A., Kruse, A., and Marks, D. Accelerating protein design using autoregressive generative models. *BioRxiv*, pp. 757252, 2019.

Riesselman, A. J., Ingraham, J. B., and Marks, D. S. Deep generative models of genetic variation capture the effects of mutations. *Nature methods*, 15(10):816–822, 2018a.

Riesselman, A. J., Ingraham, J. B., and Marks, D. S. Deep generative models of genetic variation capture the effects of mutations. *Nature methods*, 15(10):816–822, 2018b.

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., and Fergus, R. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *PNAS*, 2019a. doi: 10.1101/622803. URL https://www.biorxiv.org/content/10.1101/622803v4.

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., and Fergus, R. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *bioRxiv*, 2019b. doi: 10.

1101/622803. URL https://www.biorxiv.org/content/10.1101/622803v4.

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., 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):e2016239118, 2021.

Scherer, M., Fleishman, S. J., Jones, P. R., Dandekar, T., and Bencurova, E. Computational enzyme engineering pipelines for optimized production of renewable chemicals. *Frontiers in bioengineering and biotechnology*, 9: 673005, 2021.

Shroff, R., Cole, A. W., Diaz, D. J., Morrow, B. R., Donnell, I., Annapareddy, A., Gollihar, J., Ellington, A. D., and Thyer, R. Discovery of novel gain-of-function mutations guided by structure-based deep learning. *ACS synthetic biology*, 9(11):2927–2935, 2020.

Shuai, R. W., Ruffolo, J. A., and Gray, J. J. Generative language modeling for antibody design. *bioRxiv*, 2021.

Sirin, S., Apgar, J. R., Bennett, E. M., and Keating, A. E. Ab-bind: antibody binding mutational database for computational affinity predictions. *Protein Science*, 25(2): 393–409, 2016.

Stadelmann, T., Heid, D., Jendrusch, M., Mathony, J., Rosset, S., Correia, B. E., and Niopek, D. A deep mutational scanning platform to characterize the fitness landscape of anti-crispr proteins. *bioRxiv*, pp. 2021–08, 2021.

Stourac, J., Dubrava, J., Musil, M., Horackova, J., Damborsky, J., Mazurenko, S., and Bednar, D. Fireprotdb: database of manually curated protein stability data. *Nucleic acids research*, 49(D1):D319–D324, 2021.

Sumida, K. H., Núñez-Franco, R., Kalvet, I., Pellock, S. J., Wicky, B. I. M., Milles, L. F., Dauparas, J., Wang, J., Kipnis, Y., Jameson, N., Kang, A., De La Cruz, J., Sankaran, B., Bera, A. K., Jiménez-Osés, G., and Baker, D. Improving Protein Expression, Stability, and Function with ProteinMPNN. *Journal of the American Chemical Society*, 146(3):2054–2061, jan 2024. ISSN 0002-7863. doi: 10.1021/jacs.3c10941. URL https://doi.org/10.1021/jacs.3c10941.

Thompson, J. D., Higgins, D. G., and Gibson, T. J. Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research*, 22(22):4673–4680, 1994.

Thompson, J. D., Gibson, T. J., Plewniak, F., Jeanmougin, F., and Higgins, D. G. The clustal_x windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. *Nucleic acids research*, 25(24): 4876–4882, 1997.

Torng, W. and Altman, R. B. 3d deep convolutional neural networks for amino acid environment similarity analysis. *BMC bioinformatics*, 18:1–23, 2017.

Townshend, R. J., Vögele, M., Suriana, P., Derry, A., Powers, A., Laloudakis, Y., Balachandar, S., Jing, B., Anderson, B., Eismann, S., et al. Atom3d: Tasks on molecules in three dimensions. *arXiv preprint arXiv:2012.04035*, 2020.

Tsuboyama, K., Dauparas, J., Chen, J., Laine, E., Mohseni Behbahani, Y., Weinstein, J. J., Mangan, N. M., Ovchinnikov, S., and Rocklin, G. J. Mega-scale experimental analysis of protein folding stability in biology and design. *Nature*, 2023. doi: 10.1038/s41586-023-06328-6. URL https://doi.org/10.1038/s41586-023-06328-6.

Vig, J., Madani, A., Varshney, L. R., Xiong, C., Rajani, N., et al. Bertology meets biology: Interpreting attention in protein language models. In *International Conference on Learning Representations*, 2020.

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976): 1089–1100, 2023.

Wrenbeck, E. E., Azouz, L. R., and Whitehead, T. A. Single-mutation fitness landscapes for an enzyme on multiple substrates reveal specificity is globally encoded. *Nature communications*, 8(1):15695, 2017.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

Zhang, Z., Xu, M., Chenthamarakshan, V., Lozano, A., Das, P., and Tang, J. Enhancing protein language models with structure-based encoder and pre-training. *arXiv preprint arXiv:2303.06275*, 2023.

Zheng, Z., Deng, Y., Xue, D., Zhou, Y., Ye, F., and Gu, Q. Structure-informed language models are protein designers. *bioRxiv*, pp. 2023–02, 2023.

A. Additional Experiments

Loss	T2837	S487
$\frac{p_j^{\text{MSA}}(a^+)}{p_j^{\text{MSA}}(a^+)+p_j^{\text{MSA}}(a^-)} - 0.5$	0.51	0.38
$\text{CLMAP}\{\log\{p_j^{\text{MSA}}(a^+)/p_j^{\text{MSA}}(a^-)\}, \pm 5\}$	0.52	0.38
$\text{CLMAP}\{[p_j^{\text{MSA}}(a^+)/p_j^{\text{MSA}}(a^-)]^2, \pm 5\}$	0.50	0.37

Table 7. We demonstrate the model Pearson correlation coefficient with different rank score loss. The first block shows the loss as the default setting. The second block displays the loss with other formulations.

Dataset	Phenotype	EvoRank w/ Classification Head				EvoRank w/ Joint Heads				EvoRank w/ Regression Head			
		Pearson	Spearman	AUC	RMSE	Pearson	Spearman	AUC	RMSE	Pearson	Spearman	AUC	RMSE
T2837	$\Delta\Delta$G	0.47	0.49	0.76	1.78	**0.51**	**0.53**	0.77	1.76	**0.51**	**0.53**	**0.78**	**1.70**
levoglucosan kinase	Δ Solubility	0.28	0.34	0.62	1.40	**0.30**	**0.34**	**0.65**	**1.37**	0.29	**0.34**	0.64	1.39
S487	Protein-Protein $\Delta\Delta G_{\text{bind}}$	0.36	0..35	0.65	1.35	0.37	0.37	**0.67**	1.36	**0.38**	**0.38**	**0.67**	**1.26**
platinumDB	Protein-Ligand $\Delta\Delta G_{\text{bind}}$	0.25	0.24	0.61	1.58	0.27	0.27	**0.65**	1.58	**0.28**	**0.28**	0.64	**1.53**
ABBind	Antibody-Antigen$\Delta\Delta G_{\text{bind}}$	0.39	0.45	0.72	1.48	**0.41**	**0.46**	0.72	1.57	**0.41**	**0.46**	**0.74**	**1.42**

Table 8. We illustrate that MutRank without additional regression head can still generate good results on the test sets. The numbers reported are averaged over three trials.

Head architecture ablations In our approach, to train with the EvoRank loss, we replace the classification head with a regression head. This head contextualize the embedding vectors for the two amino acids with the hidden representation for a particular microenvironment in order to compute a residue specific rank score. Alternatively, we can use the EvoRank loss with the original classification head by calculating the rank score from the logits. In this ablation study, shown in Table 8, we observe that introducing the additional regression head generally results in a modest performance improvement ranging from 1% to 4% across 5 datasets. More importantly, these results demonstrate the superior zero-shot generalization of the EvoRank representations over the WT-mask baseline regardless of the head architecture.

Exploring different loss formulation Training with EvoRank loss is a two-stage procedure. Initially, we train the backbone using MSA-based soft labels with the α-divergence loss and subsequently fine-tune with the EvoRank loss. 1) We evaluate the impact of jointly training with α-divergence loss and EvoRank (Table 8, middle column). Our results indicate that the linear combination of the α-divergence and EvoRank losses with 0.4 and 0.6 coefficients, respectively, provides the best performance. However, these results match our previous performance. 2) We then evaluate different ways to compute the rank score for a residue from the MSA distribution, and benchmark on the T2837 and S487 datasets. As demonstrated in Table 7, all rank score formulations converge to similar performance on T2837 and S487. Thus, the exact formulation for computing the rank score has an insignificant impact on performance and further demonstrates the robustness of the EvoRank loss.

Dataset	#Mut	MutRank-2M	MutRank-8M	MutRank-24M	MutRank-48M
T2837	2837	0.48	0.51	0.51	0.51
levoglucosan kinase	9011	0.27	0.29	0.29	0.28
Gβ1	935	0.58	0.62	0.62	0.62
S487	487	0.36	0.38	0.40	0.40
PlatinumDB	925	0.25	0.28	0.28	0.26

Table 9. We demonstrate the model Pearson correlation coefficient with different model sizes. All the results are averaged over three trials.

Model size ablations The machine learning community has empirically demonstrated the benefits of increasing model size (Dehghani et al., 2023; Chowdhery et al., 2023). This too has been demonstrated by protein language models (Elnaggar et al., 2021; Rives et al., 2019a; Lin et al., 2023). However, to the best of our knowledge no study has explored the impact of model size for protein structure-based machine learning frameworks. We conducted a comprehensive analysis ranging the parameters from $\sim$2M to $\sim$48M. The results, presented in Table 9, demonstrate marginal to no improvements from scaling the model parameters. For example, the smallest model ($\sim$2M) exhibit diminished performance compared to the largest ($\sim$48M) model but the average performance improvement across 4 datasets is only $\sim$6%. But the same analysis between the

(∼8M) and (∼48M) models results in an average performance decrease of 1.25%. Further experiments, such as scaling the dataset beyond ∼20K proteins, are required to confirm if structure-based ML frameworks trained with EvoRank loss will benefit from model scaling. All experiments reported in this work are from the 8M parameter model.