Binding Oracle: Fine-Tuning From Stability to Binding Free Energy

Chengyue Gong[1], **Daniel J. Diaz**[1, 2, 3], **Jordan Wells**[1], **James M. Loy**[3], **Qiang Liu**[1], **Alexandros G. Dimakis**[4], **Adam R. Klivans**[1]

[1] Computer Science, UT Austin [2] Chemistry, UT Austin
[3] Intelligent Proteins, LLC [4] Electrical and Computer Engineering, UT Austin

Abstract

The ability to predict changes in binding free energy ($\Delta\Delta G_{\rm bind}$) for mutations at protein-protein interfaces (PPIs) is critical for the understanding genetic diseases and engineering novel protein-based therapeutics. Here, we present Binding Oracle: a structure-based graph transformer for predicting $\Delta\Delta G_{\rm bind}$ at PPIs. Binding Oracle fine-tunes Stability Oracle with Selective LoRA: a technique that synergizes layer selection via gradient norms with LoRA. Selective LoRA enables the identification and fine-tuning of the layers most critical for the downstream task, thus, regularizing against overfitting. Additionally, we present new training-test splits of mutational data from the SKEMPI2.0, Ab-Bind, and NABE databases that use a strict 30% sequence similarity threshold to avoid data leakage during model evaluation. Binding Oracle, when trained with the Thermodynamic Permutations data augmentation technique , achieves SOTA on S487 without using any evolutionary auxiliary features. Our results empirically demonstrate how sparse fine-tuning techniques, such as Selective LoRA, can enable rapid domain adaptation in protein machine learning frameworks.

1 Introduction

Selective protein-protein interfaces (PPIs) are the fundamental interaction that underpins most biological processes and diseases. Numerous studies have highlighted how disease-related mutations are over-represented at PPIs [1, 2, 3, 4]. Engineered antibodies, such as monoclonal antibodies and CAR-T cells, highlight the importance of PPIs in modern medicine where they are designed to fight viral infections, as immunotherapies for cancer, prevent organ transplant rejection, and more other applications [5, 6, 7, 8]. The impact of a mutation on binding affinity between two proteins can be measured by the difference in binding free energy ($\Delta\Delta G_{\rm bind}$) before and after mutation. Currently, accurate experimental measurements of the $\Delta\Delta G_{\rm bind}$ of a mutation are laborious, expensive, and time-consuming. As such, computational methods capable of accurately predicting the $\Delta\Delta G_{\rm bind}$ of a mutation are critical for scientific advancements. An accurate $\Delta\Delta G_{\rm bind}$ computational method can accelerate important therapeutic activities such as identifying disease-causing missense mutations from genetic screens and engineering therapeutic proteins to treat diseases.

Machine learning is rapidly revolutionizing the computational landscape in biological and physical sciences, with Alphafold2 being the quintessential example [9]. For PPIs, several machine learning frameworks have already been proposed [10, 11, 12, 13, 14] and are primarily trained on the largest experimental database of mutational $\Delta\Delta G_{\rm bind}$ for structurally solved PPIs, SKEMPI 2.0 [15]. Unfortunately, SKEMPI2.0 has two limiting aspects: (1) it is majorly composed of mutations to alanine ($\sim$55%) and (2) only a small amount of mutations in the dataset actually increase the binding affinity of the complex (which is often the main goal in PPI engineering). These data issues, along with

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

the small size of SKEMPI2.0 has hindered the development of robust machine learning frameworks capable of generalization. However, such data issues are not unique to $\Delta\Delta G_{\text{bind}}$ but are observed in most biological domains where machine learning can be have the largest impact. Put simply, as of 2023 there is just not enough publicly available functional data for *supervised learning*.

The computational biology community has recently started leveraging self-supervised learning followed by transfer learning techniques in order to narrow the data gap. This is highlighted in recent structure-based frameworks, such as GearBind [13], GeoPPI [12] and Deep Local Analysis (DLA)-mutation [14]. Both GearBind and GeoPPI use self-supervision to pre-train graph neural networks on perturbed protein structures using a contrastive and reconstruction loss, respectively, and then fine-tune an MLP and GBT, respectively, on $\Delta\Delta G_{\text{bind}}$ data. However, these methods evaluate their model using complex-level cross validation, which is known to result in data leakage [16, 17], making it hard to gauge model generalization. Similarly, DLA-mutation first pre-trains a self-supervised 3D convolutional neural network (3DCNN) on masked local chemical environments (masked microenvironments) of residues and then fine-tunes an MLP on $\Delta\Delta G_{\text{bind}}$ data. Like DLA-mutation, our framework also makes use of masked microenvironments for pre-training and leverages the same test set allowing direct comparison. However, DLA-mutation requires both the wildtype and mutant structure, uses Rosetta's backrub protocol to sample the conformational landscape of the local environment, and relies on evolutionary auxiliary features — which are often not available for antibodies — to achieve state-of-the-art (SOTA) results. Furthermore, DLA-mutation fails to incorporate several other machine learning advancements that have been shown to improve model performance and speed up training and inference for supervised models.

Here, we present a sparse fine-tuning approach that can be applied to pre-trained models when the downstream functional data is limited. Our sparse fine-tuning approach, dubbed selective LoRA, synergizes layer selection via gradient norms with LoRA to optimize fine-tuning while regularizing overfitting. Additionally, we present PPI datasets with train-test splits that use a strict 30% sequence similarity threshold to avoid data leakage during model evaluation. We demonstrate that applying selective LoRA to Stability Oracle [18] — a structure-based graph transformer framework for thermodynamic $\Delta\Delta G$ prediction — enables us to quickly fine-tune on protein-protein and protein-nucleotide interface datasets. Train on one A100, it takes us $\sim$2 minutes to finetune. Our fine-tuned model, Binding Oracle, when trained with Thermodynamic Permutation data augmentation [18], achieves SOTA on the S487 test set. These results demonstrate how sparse fine-tuning techniques can enable rapid domain adaptation in protein machine learning frameworks.

2 Methods

In this section, we delve into our methodology. We begin by discussing our approach to layer fine-tuning and the intricacies of LoRA fine-tuning, as shown in Figure 1. We refer the readers to Appendix A for the introduction about our pre-trained $\Delta\Delta G$ prediction model, an exploration of our data augmentation strategies and how we make new datasets and avoid data leakage.

LoRA Although numerous methods for efficiently fine-tuning large models exist, their performance often slightly trails the traditional approaches. An exception to this trend is the Low Rank Adaptation (LoRA) [19] method, which, in certain cases, even outperforms full fine-tuning in terms of accuracy and other metrics. Instead of modifying every weight in a pre-trained model, LoRA optimizes two compact matrices that approximate the primary weight matrix. Once fine-tuned, these matrices, forming the LoRA adapter, seamlessly integrate with the pre-trained model, enhancing its predictive capabilities. In this study, we incorporate the LoRA technique for select MLP layers within the model's backbone. By integrating LoRA in this manner, we aim to harness its capabilities for efficient fine-tuning, better generalization, and optimizing the performance of these specific layers while maintaining the integrity of the pre-trained knowledge.

Finetuning Layer Selection Previous research has highlighted that fine-tuning certain layers in a neural network might not significantly impact performance [20]. Consequently, various layer selection techniques have been proposed for refining models for transfer learning [21, 22, 23, 24]. In this study, we utilized the gradient norms on our training data to select the layers for fine-tuning. We compute each criterion for each tensor based on its flattened gradient g and parameters θ. We

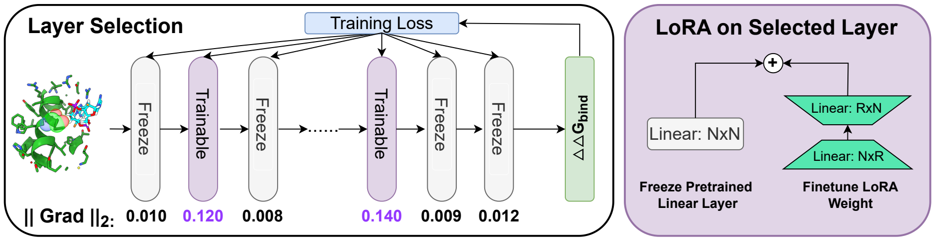

Figure 1: **Left.** The layer selection illustration: We only finetune selected layers and the final regression head. **Right.** The LoRA visualization: While we retain the pre-trained MLP weights, we fine-tune the low-rank matrices, as indicated by the purple elements in the figure. N and R denotes the feature dimension and the low-rank value, repsectively.

measure the ratio of gradient norm to parameter norm,

$$\text{Score} = \frac{\|\theta' - \theta\|_2}{\|\theta\|_2}, \quad \text{where } \theta' = \theta_T, \theta_0 = \theta, \theta_{t+1} \leftarrow \theta_t - \alpha g_t, \tag{1}$$

where T is the time step, α is step size and g_t denotes $t-$step gradient. Here, we prioritize layers exhibiting larger gradients for selection. This strategy stems from our underlying hypothesis: layers with pronounced gradient magnitudes are likely to carry more pertinent information for the task. In our problem set, we have few number of data, and therefore it's easy to calculate the exact gradient on all the training data and accumulate it over several iterations.

Block Index		1	2	3	4
Attention	KQV	0.003	0.004	0.002	0.005
	Projection	0.004	0.003	0.004	0.003
MLP	Linear1	0.004	**0.012**	0.004	**0.013**
	Linear2	0.005	**0.011**	0.002	**0.014**

Table 1: We present the scores as defined by (1), computed for all dense layers in the four transformer blocks of our pre-trained model (SO). 'KQV' denotes the projection layer for Key, Query, Value embeddings; 'Projection' denotes the projection layer that follows the attention layer. 'Linear1' and 'Linear2' denotes the two linear layers. We define the time step as $T = 5$ and $\alpha = 10^{-3}$.

In practice, our pretrained model has a backbone architecture that consist of 4 graph-transformer blocks and we calculate the (1) for every dense layer in the backbone. As observed in Table 1, four linear layers (two in the MLP of block 2 and two in block 4) exhibit higher gradient norms than the rest. Therefore, we focus on fine-tuning only these layers with LoRA. In the future, we will extend this finetuning framework and verify whether it works for LLMs and large diffusion models.

3 Results

In order to evaluate the performance of our proposed model, Binding Oracle (BO), we carry out a series of experiments aimed at answering several important questions. 1) Can we match or outperform existing methods which make use evolutionary features? 2) What is the proper approach to fine-tune our pre-trained model on binding $\Delta\Delta G$? We refer the readers to the Appendix B for results about complex/mutation-level cross validation, data augmentation, different training and test sets, and detailed discussions about generalization to other interfaces like protein-nucleotide and protein-ligand

Comparing with Baselines on Current Benchmarks S487 [25] is a commonly-used test dataset with 487 mutations from 56 complexes. As demonstrated in Table 2, we compare the Pearson correlation coefficient and root-mean-squared-error (RMSE) for all methods on this dataset. We observe that our newly proposed Binding Oracle (BO) surpasses all existing methods, encompassing both traditional computational tools and recently introduced deep learning techniques. When compared with the pretrained model, Stability Oracle (SO), BO enhances the Pearson correlation coefficient by 22% and reduces the RMSE by 6%. More importantly, BO consistently delivers superior results when comparing with the recent SOTA method, Deep Local Analysis (DLA-mutation) [14], while having a much improved runtime. Beyond correlation metrics, it's noteworthy that DLA-mutation exhibits two distinct disadvantages relative to our approach: Firstly, DLA-mutation necessitates 3D structures for both the wild-type and mutation, adding computational overhead during data preparation. Secondly, the reliance on supplementary evolutionary features by DLA-mutation further increases the time costs, while BO does not make use of any evolutionary input features and relies solely on the structure.

Method	iSee	FoldX	mCSM	SAAMBE	DLA-mutation	Stability Oracle	Binding Oracle
Pearson ↑	0.25	0.34	0.25	0.40	0.42	0.38	**0.45**
RMSE ↓	1.32	1.53	1.35	-	1.31	1.34	**1.27**

Table 2: Pearson correlation coefficient and RMSE for several literature methods, Stability Oracle, and Binding Oracle on S487. Metrics for iSee, FoldX, mCSM literature methods are from [26, 25].

Performance of Different Finetuning Methods We showcase the impact various strategies for fine-tuning across regression and classification metrics in Table 3. The results demonstrate: ❶ End-to-end fine-tuning results in diminished performance. We suspect this is due to the small size of the training set and domain transfer from $\Delta\Delta G$ to $\Delta\Delta G_{\mathrm{bind}}$. ❷ Both layer selection and LoRA fine-tuning contribute positively. ❸ Synergizing layer selection with LoRA fine-tuning (BO) results in improved performance for every metric.

Method	Pearson ↑	Spearman ↑	RMSE ↓	Accuracy ↑	Precision ↑	Recall ↑	AUC ↑
Stability Oracle (Zero-Shot)	0.38	0.38	1.34	0.76	0.49	0.27	0.70
Freeze backbone	0.37	0.38	1.27	0.75	0.47	0.26	0.70
End-to-end fine-tuning	0.34	0.34	1.42	0.70	0.42	0.25	0.68
End-to-end LoRA fine-tuning	0.42	0.42	1.30	0.77	0.48	0.28	0.71
Selected layer fine-tuning	0.39	0.39	1.32	0.76	0.47	0.28	0.70
Binding Oracle (Selective LoRA)	**0.45**	**0.45**	**1.27**	**0.78**	**0.55**	**0.29**	**0.72**

Table 3: Regression and classification metrics on the S487 test set for the fine-tuned Stability Oracle models. Freeze backbone: fine-tune only the regression head; End-to-end fine-tuning: fine-tune every layer in the backbone and regression head; End-to-end LoRA fine-tuning: fine-tune every layer in the backbone and regression head with LoRA; Selected layer fine-tuning: fine-tune the selected backbone layers; Binding Oracle: fine-tune the selected backbone layers with LoRA.

4 Conclusion

In this research, we explored the intricacies of fine-tuning Stability Oracle, a $\Delta\Delta G$ of unfolding predictor trained on a subset of the cDNA megascale dataset [27], to instead predict the impact on $\Delta\Delta G_{\mathrm{bind}}$ for point mutations at protein-protein interfaces. In order to fine-tune on a domain with limited publicly available data, we proposed a sparse fine-tuning approach that regularizes against overfitting and enhances generalization. Our empirical results underscore the superiority of our method over existing alternatives and enabled us to achieve SOTA results on the S487 test set (ρ=0.45). Compared to DLA-mutation, our method has several advantages: 1) it does not require a mutant structure; 2) it does not require expensive data preprocessing (i.e., Rosetta's backrub protocol); 3) it does not use any evolutionary auxiliary features and solely relies on structural features; 4) we observe consistent performance between mAb-AG and TCR-pMHC protein-protein interfaces and modest generalization to protein-nucleotide interfaces; 5) the fine-tuning takes $\sim$2 minutes enabling rapid experimentation of sparse-fine-tuning techniques and hyperparameter search.

Stability Oracle zero-shot capability (ρ=0.38 on S487) was somewhat surprising since the cDNA dataset consist of single domain monomeric structures. Thus, demonstrating how fine-tuning from a pre-trained model on $\Delta\Delta G$ of unfolding provides improved representations for $\Delta\Delta G_{\mathrm{bind}}$. However, in the grand scheme of things selective LoRA only enabled a marginal improvement and much more progress is still needed. This highlights how the lack of experimental PPI data, and its associated biases, is the primary hurdle for achieving significant improvements in PPI predictions. Similar to the stability community with cDNA-display proteolysis, a high throughput screening breakthrough, is needed to generate large and diverse datasets for deep learning to have its AlphaFold2 moment.

Looking forward, we intend to further refine our model by exploring proper evaluation of the protein-nucleotide interface, incorporation of $\Delta\Delta G_{\mathrm{bind}}$ data for the protein-ligand interface, and collaborations with experimentalist. Specifically, we aim to adapt and validate Binding Oracle ability to stabilize antibody-antigen interactions for B and T cells. We hope to further bridge the gap for ML-guided protein engineering, fostering an era of rapid and informed breakthrough discoveries.

References

[1] Alessia David and Michael JE Sternberg. The contribution of missense mutations in core and rim residues of protein–protein interfaces to human disease. *Journal of molecular biology*, 427(17):2886–2898, 2015.

[2] Dàmaris Navío, Mireia Rosell, Josu Aguirre, Xavier de la Cruz, and Juan Fernández-Recio. Structural and computational characterization of disease-related mutations involved in protein-protein interfaces. *International journal of molecular sciences*, 20(7):1583, 2019.

[3] Nidhi Sahni, Song Yi, Mikko Taipale, Juan I Fuxman Bass, Jasmin Coulombe-Huntington, Fan Yang, Jian Peng, Jochen Weile, Georgios I Karras, Yang Wang, et al. Widespread macromolecular interaction perturbations in human genetic disorders. *Cell*, 161(3):647–660, 2015.

[4] Christopher M Yates and Michael JE Sternberg. The effects of non-synonymous single nucleotide polymorphisms (nssnps) on protein–protein interactions. *Journal of molecular biology*, 425(21):3949–3963, 2013.

[5] Peter J Hudson and Christelle Souriau. Engineered antibodies. *Nature medicine*, 9(1):129–134, 2003.

[6] Ron Diskin, Johannes F Scheid, Paola M Marcovecchio, Anthony P West Jr, Florian Klein, Han Gao, Priyanthi NP Gnanapragasam, Alexander Abadir, Michael S Seaman, Michel C Nussenzweig, et al. Increasing the potency and breadth of an hiv antibody by using structure-based rational design. *Science*, 334(6060):1289–1293, 2011.

[7] Rebecca S Rudicell, Young Do Kwon, Sung-Youl Ko, Amarendra Pegu, Mark K Louder, Ivelin S Georgiev, Xueling Wu, Jiang Zhu, Jeffrey C Boyington, Xuejun Chen, et al. Enhanced potency of a broadly neutralizing hiv-1 antibody in vitro improves protection against lentiviral infection in vivo. *Journal of virology*, 88(21):12669–12682, 2014.

[8] Shira Warszawski, Aliza Borenstein Katz, Rosalie Lipsh, Lev Khmelnitsky, Gili Ben Nissan, Gabriel Javitt, Orly Dym, Tamar Unger, Orli Knop, Shira Albeck, et al. Optimizing antibody affinity and stability by the automated design of the variable light-heavy chain interfaces. *PLoS computational biology*, 15(8):e1007207, 2019.

[9] 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.

[10] Menglun Wang, Zixuan Cang, and Guo-Wei Wei. A topology-based network tree for the prediction of protein–protein binding affinity changes following mutation. *Nature Machine Intelligence*, 2(2):116–123, 2020.

[11] Guangyu Zhou, Muhao Chen, Chelsea JT Ju, Zheng Wang, Jyun-Yu Jiang, and Wei Wang. Mutation effect estimation on protein–protein interactions using deep contextualized representation learning. *NAR genomics and bioinformatics*, 2(2):lqaa015, 2020.

[12] Xianggen Liu, Yunan Luo, Sen Song, and Jian Peng. Pre-training of graph neural network for modeling effects of mutations on protein-protein binding affinity. *arXiv preprint arXiv:2008.12473*, 2020.

[13] Huiyu Cai, Zuobai Zhang, Mingkai Wang, Bozitao Zhong, Yanling Wu, Tianlei Ying, and Jian Tang. Pretrainable geometric graph neural network for antibody affinity maturation (preprint). 2023.

[14] Yasser Mohseni Behbahani, Elodie Laine, and Alessandra Carbone. Deep local analysis deconstructs protein–protein interfaces and accurately estimates binding affinity changes upon mutation. *Bioinformatics*, 39(Supplement_1):i544–i552, 2023.

[15] Justina Jankauskaitė, Brian Jiménez-García, Justas Dapkūnas, Juan Fernández-Recio, and Iain H Moal. Skempi 2.0: an updated benchmark of changes in protein–protein binding energy, kinetics and thermodynamics upon mutation. *Bioinformatics*, 35(3):462–469, 2019.

[16] Aron Broom, Kyle Trainor, Zachary Jacobi, and Elizabeth M. Meiering. Computational Modeling of Protein Stability: Quantitative Analysis Reveals Solutions to Pervasive Problems. *Structure*, 28(6):717–726.e3, 2020.

[17] Fabrizio Pucci, Martin Schwersensky, and Marianne Rooman. Artificial intelligence challenges for predicting the impact of mutations on protein stability. *Current opinion in structural biology*, 72:161–168, 2022.

[18] Daniel J Diaz, Chengyue Gong, Jeffrey Ouyang-Zhang, James M Loy, Jordan Wells, David Yang, Andrew D Ellington, Alex Dimakis, and Adam R Klivans. Stability oracle: A structure-based graph-transformer for identifying stabilizing mutations. *bioRxiv*, pages 2023–05, 2023.

[19] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[20] Chiyuan Zhang, Samy Bengio, and Yoram Singer. Are all layers created equal? *The Journal of Machine Learning Research*, 23(1):2930–2957, 2022.

[21] Gal Kaplun, Andrey Gurevich, Tal Swisa, Mazor David, Shai Shalev-Shwartz, and Eran Malach. Subtuning: Efficient finetuning for multi-task learning. *arXiv preprint arXiv:2302.06354*, 2023.

[22] Weijia Xu, Batool Haider, Jason Krone, and Saab Mansour. Soft layer selection with meta-learning for zero-shot cross-lingual transfer. *arXiv preprint arXiv:2107.09840*, 2021.

[23] Raphael Ngigi Wanjiku, Lawrence Nderu, and Michael Kimwele. Dynamic fine-tuning layer selection using kullback–leibler divergence. *Engineering Reports*, 5(5):e12595, 2023.

[24] Yoonho Lee, Annie S Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. Surgical fine-tuning improves adaptation to distribution shifts. *arXiv preprint arXiv:2210.11466*, 2022.

[25] Cunliang Geng, Anna Vangone, Gert E Folkers, Li C Xue, and Alexandre MJJ Bonvin. 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.

[26] Swagata Pahari, Gen Li, Adithya Krishna Murthy, Siqi Liang, Robert Fragoza, Haiyuan Yu, and Emil Alexov. Saambe-3d: predicting effect of mutations on protein–protein interactions. *International journal of molecular sciences*, 21(7):2563, 2020.

[27] Kotaro Tsuboyama, Justas Dauparas, Jonathan Chen, Elodie Laine, Yasser Mohseni Behbahani, Jonathan J Weinstein, Niall M Mangan, Sergey Ovchinnikov, and Gabriel J Rocklin. Mega-scale experimental analysis of protein folding stability in biology and design. *Nature*, 2023.

[28] Sarah Sirin, James R Apgar, Eric M Bennett, and Amy E Keating. Ab-bind: antibody binding mutational database for computational affinity predictions. *Protein Science*, 25(2):393–409, 2016.

[29] Junyi Liu, Siyu Liu, Chenzhe Liu, Yaping Zhang, Yuliang Pan, Zixiang Wang, Jiacheng Wang, Ting Wen, and Lei Deng. Nabe: an energetic database of amino acid mutations in protein–nucleic acid binding interfaces. *Database*, 2021:baab050, 2021.

[30] Martin Steinegger and Johannes Soding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.

[31] Grant Thiltgen and Richard A Goldstein. Assessing predictors of changes in protein stability upon mutation using self-consistency. *PloS one*, 7(10):e46084, 2012.

[32] Peter Atkins, Peter William Atkins, and Julio de Paula. *Atkins' physical chemistry*. Oxford university press, 2014.

[33] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

A Data Augmentation and Generation

Stability Oracle (SO) [18] is a structure-based deep learning framework that makes use of several innovations in data and machine learning engineering specific to stability prediction. SO uses a graph-transformer architecture that treats atoms as tokens and utilizes their pairwise distances to inject a structural inductive bias into the attention mechanism. The input to SO consists of the local chemistry surrounding a residue with the residue deleted (the masked microenvironment) and two amino acid embeddings to represent a specific point mutation. This design decision enables Stability Oracle to generate all 380 possible point mutations from a single microenvironment. In this study, we utilize SO as our pre-trained model and subsequently fine-tune it to predict K_d changes associated with single-point mutations in the binding site for a given protein.

One of the paramount advantages of the SO model deserves emphasis. Traditional structure-based methods for $\Delta\Delta G$ prediction typically necessitate both a wild-type structure and a mutated variant. Generating these structures, often done through computational tools, can be time-consuming, thereby introducing substantial delays when deploying the model. In contrast, the SO model simplifies this process by only requiring a wild-type structure. Remarkably, despite this streamlined input, SO still manages to deliver superior results compared to its counterparts.

Dataset Generation Due to known data-leakage issues plaguing most training-test splits in the literature, we curated or own training and test sets to ensure we our results are not inflated due to data leakage. First we curated all available data at the protein-protein and protein-nucleotide interfaces from the SKEMPI2.0 [15], Ab-Bind [28], and NABE databases [29]. Mutations from SKEMPI 2.0, Ab-Bind, and NABE were combined to produce a dataset with 5504 non-redundant single point mutations at protein-protein and protein-nucleotide interfaces. Before combining the datasets, Skempi had all mutations not collected by the SPR, ITC, FL, IASP, and SFFL experimental techniques, high-order mutations, and mutations from the prolactin-prolactin receptor complex removed, resulting in 3858 single mutations. From combining the three datasets, we had 5,596 mutations with 536 having duplicate entries that needed further processing. Duplicated mutations that had discrepancy between their $\Delta\Delta G$ sign were discarded. Duplicated mutations with neutral and stabilizing entries ($\Delta\Delta G < 0.5$ kcal/mol) and discrepancy larger than 0.5 were also discarded since we could not confidently discern if this mutation was neutral, destabilizing, or stabilizing. For the remaining duplicates, the $\Delta\Delta G$ with the largest magnitude was assigned as the $\Delta\Delta G$ label in order to bias predictions from being narrowly distributed around 0 kcal/mol. This deduplication procedure resulted in 92 mutations being discarded and a final dataset of 5504 mutations (B5504) from 670 PDB entries. This procedure was repeated for just the protein-protein interface databases (SKEMPI2.0 and Ab-Bind). Here, the final dataset consist of 3705 mutations (B3705) from 263 PDB entries.

Training-Test Split Generation With our B5504 dataset, we used the S487 [25] dataset to seed our test set and guide or train-test split. We used MMSeqs2 [30] to remove any proteins with over a 30% sequence similarity in B5504 to S487, which were then added to S487. This procedure was repeated to iteratively prune any sequences originally in B5504 similar to the growing test set until no sequences were removed. This resulted in the binding training and test sets with 3542 (B3542) and 1962 (T1962) mutations, respectively, with a maximum overlap of 30% sequence similarity. The procedure was repeated with S487 and B3705, our dataset lacking protein-nucleotide mutations. This resulted in a training and test sets of solely protein-protein interactions with 1816 (B1816) and 1889 (T1889) mutations, respectively, with a maximum overlap of 30% sequence similarity.

Data Augmentation While fine-tuning with sparsity proves beneficial for few-shot transfer, incorporating data augmentation further enhances model generalization. Our training and test sets are made up of a heavily biased mutation type distribution and the $\Delta\Delta G$ labels are significantly imbalanced towards destabilizing mutations. In our work, we use the data augmentation technique, Thermodynamic Permutations (TP) proposed in Stability Oracle [18]. Thermodynamic Permutations, similar to Thermodynamic Reversibility (TR) [31], are based on the state function property of Gibbs free energy [32]. For a specific position in a protein, TP expands n empirical $\Delta\Delta G$ measurements into $n(n-1)$ thermodynamically valid measurements, which can increase a dataset by up to an order of magnitude depending on the number of residues that have multiple amino acids experimentally characterized. Below are the TP results for B3542, B1816, T1962, and T1889. When we used TP on the two training sets, we find that 40 residues in the dataset have been deep mutational scanned. Thus, these 40 residues generate 13680 additional $\Delta\Delta G$ data points and bias the datasets towards these 40 microenvironments. We refer the readers to Table 4 for details about our dataset.

Dataset	#PDB	#Mutations	#Residues	#Permutation Mutations	#All Mutations
B1816	147	1,816	885	14,532	16,348
B3542	540	3,542	2,457	15,128	18,670
T1889	116	1,889	1,647	748	2,637
T1962	130	1,962	1,720	748	2,710

Table 4: Description about the data augmentation details. B3542 and B1816 serve as our training datasets, whereas T1889 and T1962 function as test sets. The term '#PDB' represents the count of unique pdb IDs. '#Mutations' indicates the total number of mutations present in the original dataset, and '#Residues' specifies the total residue count. It's worth noting that certain residues might exhibit multiple mutations, allowing for permutation-based augmentation. The '#Permutation Mutations' denotes the volume of data that can be generated using this technique.

B Additional Experiments

Hyperparameter Configuration We trained our model using a batch size of 256, learning rate of 10^{-5}, weight decay of 10^{-5}, LoRA rank of 8 and 150 iterations with Adam [33] optimizer for all the experiments in this section. It takes approximately 2 minutes to train the model on an NVIDIA 80G A100 using a single GPU, and less than 1 minute to generate the graphs and run inference on T1962 test set (1,962 inputs). All the results in this section are averaged over three trials.

Model Size We finetune our BO model based on the Stability Oracle model, whose backbone contains 816K model parameters, and the regression head contains 202K model parameters. In this work, we tune the regression head and 8K parameters in the backbone (1% backbone parameters.)

Mutation-level and Complex-level Train-test Split In literature, some researchers make mutation-level or complex-level train-test split to verify their model performance. Here, mutation-level means the same residue does no appear in both the training and test set, complex-level means the same complex only appears in the training or test set. In Table 5, we demonstrate the performance of our Binding Oracle model in this setting, where we do 5-fold train-test split and report the performance. Mutation-level or complex-level train-test split makes the metrics much better than our setting (based on sequence similarity).

Data	Pearson ↑	Spearman ↑	RMSE ↓	Accuracy ↑	Precision ↑	Recall ↑	AUC ↑
Complex Level	0.65	0.73	1.48	0.73	0.88	0.32	0.81
Mutation Level	0.70	0.77	1.32	0.81	0.80	0.42	0.83

Table 5: We display BO's performance for 5-fold cross-validation (with mutation-level or complex-level split) on B1816 dataset.

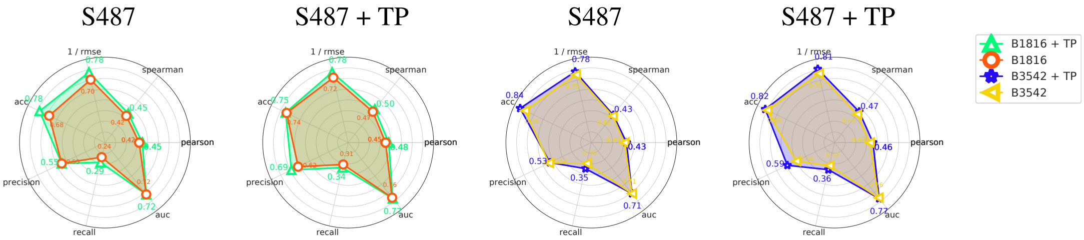

Figure 2: Evaluating the impact of Thermodynamic Permutation data augmentation of Binding Oracle, when training on B3542 and B1816 datasets.

Performance of Different Training Datasets In Table 6, we showcase the efficacy of models when trained on various datasets. The 'iSee Training' dataset, proposed by iSee [25] and used by

DLA-mutation [14], was used for training here as well in order to have direct comparison with these two ML frameworks. Additionally, we introduced two new training sets with less than 30% sequence similarity to S487: B1816 and B3542. Training on both B1816 and B3542 results in improved regression and classification and B3542 extends generalization to protein-nucleotide interactions. In the future we plan to propose a formal test set for evaluating the protein-nucleotide interactions.

Dataset	Pearson ↑	Spearman ↑	RMSE ↓	Accuracy ↑	Precision ↑	Recall ↑	AUC ↑
Stability Oracle (Zero-Shot)	0.38	0.38	1.34	0.76	0.49	0.27	0.70
Binding Oracle (iSee Training)	0.42	0.43	1.31	0.80	0.41	0.28	0.71
Binding Oracle (B3542)	0.43	0.43	**1.27**	**0.84**	0.53	**0.35**	0.71
Binding Oracle (B1816)	**0.45**	**0.45**	**1.27**	0.78	**0.55**	0.29	**0.72**

Table 6: Binding Oracle's performance on several regression and classification metrics when trained on different datasets and evaluated on S487.

Thermodynamic Permutation Data Augmentation Due to the limited amount of training data, we explored if Thermodynamic Permutations (TP) could alleviate known biases that plague experimental $\Delta\Delta G$ datasets curated from the literature, such as incomplete mutation type sampling, oversampling of mutations "to", and the strong imbalance between stabilizing and destabilizing mutations [16]. TP expands the B1816 dataset with 14532 additional mutations and results in improved performance (Figure 2). This improvement is also observed when applied to the B3542 dataset, which contains an additional 1.7k protein-nucleotide mutations (Figure 2). This results demonstrate how TP augmentation is particularly advantageous when dealing with small and bias datasets. It should be noted that due to the particular composition of SKEMPI2.0, most of the TP-based mutations (13680 of the 14532) come from 40 residues that have had all 20 amino acids characterized. Thus, in this situation, TP biases the training towards a small amount of microenvironment that have had all 20 amino acids characterized (microenvironment bias). Nonetheless, TP augmentation results in improved performance on test sets with less than 30% sequence similarity.

Performance on New Test Sets While S487 allows us to evaluate performance against the literature, the test set is small and is heavily biased with mutations "to" alanine . To attenuate these biases and better assess the ability of Binding Oracle to generalize, we extend S487 with any additional $\Delta\Delta G$ mutational data from similar proteins (>30% sequence similarity) to generate T1889 and T1962, where T1962 includes a small amount of protein-nucleotide mutations. Figure 3 depicts the performance of the pre-trained model, Stability Oracle, and Binding Oracle on a variety of training sets when evaluated on T1889, T1962, and their TP augmented versions. These results depict that sparse fine-tuning of Stability Oracle to $\Delta\Delta G_{\text{bind}}$ datasets can provide marginal improvements and highlight how data-scarcity is the primary limitation for making robust and generalizable models. Furthermore, Binding Oracle performance drops when compared to the smaller S487 test set (shown in Table 3), demonstrating the need for larger more diverse test sets to better gauge model generalization.

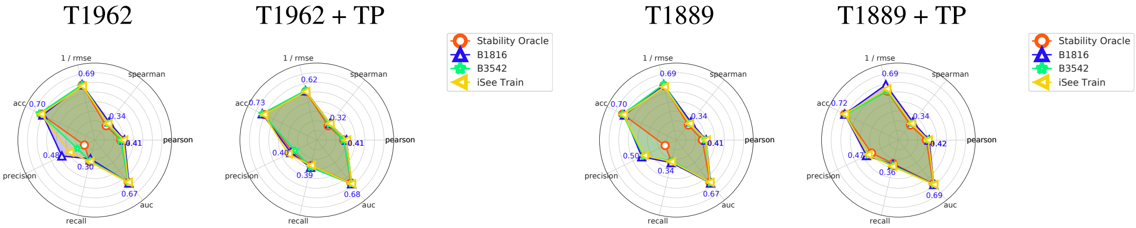

Figure 3: Comparative Analysis of the Binding Oracle Trained on Diverse Datasets, Evaluated on T1962 and T1889. Detailed numbers are shown in Table 7 in appendix.

Results on Different Biological Function Types To further assess the generalization of the Binding Oracle framework, we showcase its performance across the functionally distinct interfaces in the T1962 test set. In Table 8, we can observe that the BO model has similar performance across a spectrum of functional interfaces. Notably, the results for the protein-nucleotide interface are worse compared to the two protein-protein interfaces (mAb-Ag/TCR-pMHC), in particular the regression metrics. We hypothesize this is due the small size of the evaluation set. Thus, making it difficult to assess how well Binding Oracle generalizes to protein-nucleotide interfaces and is an avenue

Model	Pearson ↑	Spearman ↑	RMSE ↓	Accuracy ↑	Precision ↑	Recall ↑	AUC ↑
T1962							
Stability Oracle (Zero-Shot)	0.37	0.29	1.46	0.72	0.22	**0.33**	0.66
Binding Oracle (B1816)	**0.41**	**0.34**	**1.45**	0.70	**0.48**	0.30	**0.67**
Binding Oracle (B3542)	0.37	0.32	**1.45**	**0.74**	0.30	0.32	0.66
Binding Oracle (iSee Training)	0.40	0.32	1.48	0.72	0.39	**0.33**	0.66
T1962 + TP							
Stability Oracle (Zero-Shot)	0.38	0.29	1.62	0.71	0.38	0.37	0.67
Binding Oracle (B1816)	**0.41**	**0.32**	1.60	**0.73**	0.40	**0.39**	**0.68**
Binding Oracle (B3542)	0.38	0.31	**1.59**	**0.73**	0.35	**0.39**	**0.68**
Binding Oracle (iSee Training)	**0.41**	**0.32**	1.62	0.71	**0.43**	0.37	**0.68**
T1889							
Stability Oracle (Zero-Shot)	0.37	0.30	1.47	0.72	0.23	0.33	0.66
Binding Oracle (B1816)	**0.41**	**0.34**	1.45	0.70	**0.50**	**0.34**	**0.67**
Binding Oracle (B3542)	0.40	0.32	**1.44**	**0.72**	0.48	0.32	**0.67**
Binding Oracle (iSee Training)	**0.41**	0.32	1.50	0.71	0.47	0.33	0.66
T1889 + TP							
Stability Oracle (Zero-Shot)	0.39	0.30	1.61	**0.72**	0.41	0.35	0.68
Binding Oracle (B1816)	**0.42**	**0.34**	**1.45**	**0.72**	**0.47**	0.36	**0.69**
Binding Oracle (B3542)	0.41	0.32	1.60	0.71	0.45	0.37	**0.69**
Binding Oracle (iSee Training)	0.41	0.32	1.55	**0.72**	0.45	**0.38**	**0.69**

Table 7: Comparative Analysis of the Binding Oracle trained on Diverse Datasets. The values presented are averages from three trials.

we will further explore for improvements in the future. DLA-mutation[14] did a similar analysis using their complex level train-test split model where they report pearson correlations of 0.11 and 0.24 for mAb-AG and TCR-pMHC, respectively. And, although we achieve 0.31 and 0.32 person correlations for mAb-AG and TCR-pMHC, respectively, these results are not comparable to DLA-mutation's results due to their smaller test set and data leakage from their complex level splitting. However, what is comparable is the discrepancy in performance between mAb-AG and TCR-pMHC for DLA-mutation and Binding Oracle. Here, Binding Oracle seems to generalize well while DLA-mutation has significantly worse performance on mAb-AG compared to TCR-pMHC. This demonstrates how Binding Oracle is much better at generalizing across the two protein-protein interfaces compared to DLA-mutation. We were unable to obtain protease-inhibitor metadata to conduct that comparison and the remaining data in T1962 is labeled with unknown.

Data	#Mutation	Pearson ↑	Spearman ↑	Accuracy ↑	Recall ↑	Precision ↑	AUC ↑
Pr-NA	73	0.12	0.16	0.72	0.67	0.31	0.69
mAb-AG	489	0.31	0.31	0.78	0.64	0.37	0.73
TCR-pMHC	397	0.32	0.32	0.76	0.49	0.22	0.70
Unknown	1003	0.35	0.34	0.79	0.53	0.27	0.71

Table 8: We present the Binding Oracle (trained on B1816 with TP) performance on different biology function types in our test set.