# HERMES: Holographic Equivariant neuRal network model for Mutational Effect and Stability prediction

**Gian Marco Visani**
Paul G. Allen School of Computer Science and Engineering
University of Washington
gvisan01@cs.washington.edu

**Michael N. Pun**
Department of Physics
University of Washington

**William Galvin**
Paul G. Allen School of Computer Science and Engineering
University of Washington

**Eric Daniel**
Paul G. Allen School of Computer Science and Engineering
University of Washington

**Kevin Borisiak**
Department of Physics
University of Washington

**Utheri Wagura**
Department of Physics
MIT

**Armita Nourmohammad**
Department of Physics, Applied Math, and CSE
University of Washington
Fred Hutch Cancer Research Center, Seattle, WA
armita@uw.edu

## Abstract

Predicting the stability and fitness effects of amino-acid mutations in proteins is a cornerstone of biological discovery and engineering. Various experimental techniques have been developed to measure mutational effects, providing us with extensive datasets across a diverse range of proteins. By training on these data, machine learning approaches have advanced significantly in predicting mutational effects. Here, we introduce HERMES, a 3D rotationally equivariant structure-based neural network model for mutation effect prediction. Pre-trained to predict amino-acid propensities from their surrounding 3D structure atomic environments, HERMES can be efficiently fine-tuned to predict mutational effects, thanks to its symmetry-aware parameterization of the output space. Benchmarking against other models demonstrates that HERMES often outperforms or matches their performance in predicting mutation effects on stability, binding, and fitness, using either computationally or experimentally resolved protein structures. HERMES offers a versatile suit of tools for evaluating mutation effects and can be easily fine-tuned for specific predictive objectives using our open-source code.

## 1 Introduction and Related Work

Understanding the effects of amino acid mutations on a protein's function is a hallmark of biological discovery and engineering. Identifying disease-causing mutations [1, 2], enhancing enzymes' catalytic activity [3, 4], forecasting viral escape [5, 6, 7], and engineering high-affinity antibodies [8], are just some of the areas of study that rely on accurate modeling of mutational effects. Effects on protein stability are likely the most studied, as sufficient stability is usually a prerequisite of the protein's successful carrying of its function [9]. Understanding the impact of mutations on the

protein's *binding affinity* to its partner is also crucial, as most functions are mediated by binding events. These effects can be accurately measured experimentally, for example via thermal or chemical denaturation assays [10], by surface plasmon resonance [11], and, more recently, by Deep Mutational Scanning (DMS) [12, 13, 14]. These experiments are laborious and with limited throughput.

Computational modeling of mutational effects remains an attractive alternative to costly experiments. Methods based on molecular dynamics simulations are accurate for short-time (nano seconds) protein responses but are limited in predicting substantial changes in protein often inflicted by amino acid mutations [15]. Models using physical energy functions such as FoldX [16] and Rosetta [17] are well-established and remain widely used for predicting the stability effect of mutations, though they often lack accuracy and are slow [2]. Recently, machine learning models have shown substantial progress in this domain. Sequence-based [18, 19] or structure-based [20, 21, 22, 2, 23, 24], approaches are used to predict the propensity of amino acids, and by extension, the effect of mutations in a protein [18, 19, 20]. These pre-trained models serve as robust baselines, upon which additional fine-tuning on smaller protein stability datasets can significantly enhance the accuracy of predictions for mutational effects [2, 22].

Here, we introduce HERMES, which is built upon a self-supervised structure-based model H-CNN [20], and fine-tuned to predict mutational effects in proteins. Similar to H-CNN, HERMES has a 3D rotationally equivariant architecture, but with an improved performance. During pre-training, HERMES is trained to predict a residue's amino acid identity from its surrounding atomic neighborhood within a 10 Å radius in the 3D structure. To fine-tune HERMES for mutational effects, we take the pre-trained model's logits corresponding to the amino acid pair of interest, and train a model to match the experimental data for the functional difference between them. With our parametrization, HERMES automatically respects the permutational anti-symmetry in the mutational effects, which other models achieve through data augmentation [22]. We extensively benchmarked HERMES across various datasets, demonstrating state-of-the-art performance in predicting stability and highly competitive results in predicting binding effect of mutations, even when using computationally resolved protein structures. Our code is open source at `https://github.com/StatPhysBio/hermes/tree/main`, and allows users to both run the models presented in this paper, and easily fine-tune HERMES on their data.

## 2 Methods

HERMES is trained in two steps (Figure 1). First, following [20], we train an improved version of the model Holographic Convolutional Neural Network (H-CNN) to predict the identity of an amino acid from its surrounding structural neighborhood. Specifically, we remove (mask) all atoms associated with the focal residue and predict its identity using all atoms within 10 Åof the the focal residue's C-$\alpha$ (Figure 1B). Second, we develop a procedure to fine-tune HERMES on mutation effects $\Delta F$ in general, with a specific focus on predicting the stability effect of mutations $\Delta\Delta G$ (Figure 1).

**Preprocessing of protein structures.** To pre-process the protein structure data, we devise two distinct pipelines, relying either on (i) Pyrosetta [27], or (ii) Biopython [28] and other open source tools with the code adapted from [2]; see Section A.1.1 for details. The Pyrosetta pipeline is considerably faster, but requires a license, whereas the Biopython pipeline is open-source. We train models using both pipelines. Pipelines used at inference and training must match. Differences in results between the two pipelines are minimal (see Figs. S2, S5 and Table S1); for our main analyses, we report only the results using Pyrosetta.

**HERMES architecture and pre-training.** Similar to H-CNN, HERMES has a 3D rotationally equivariant architecture, with comparable number of parameters ($\sim$3.5M), but with a $\sim$2.75$\times$ improved speed in its forward pass, and a higher accuracy (Figure 1A). In short, atomic neighborhoods - i.e., featurized point clouds - are first projected onto the orthonormal Zernike Fourier basis, centered at the (masked) central residue's C-$\alpha$. We term the resulting Fourier encoding of the data an *holographic encoding*, as it presents a superposition of 3D spherical holograms [20]. Then, the resulting *holograms* are fed to a stack of SO(3)-Equivariant layers, which convert the holograms to an SO(3)-invariant embedding - i.e. a representation that is invariant to 3D rotations about the center of the initial holographic projection. These embeddings are then passed through an MLP to generate the

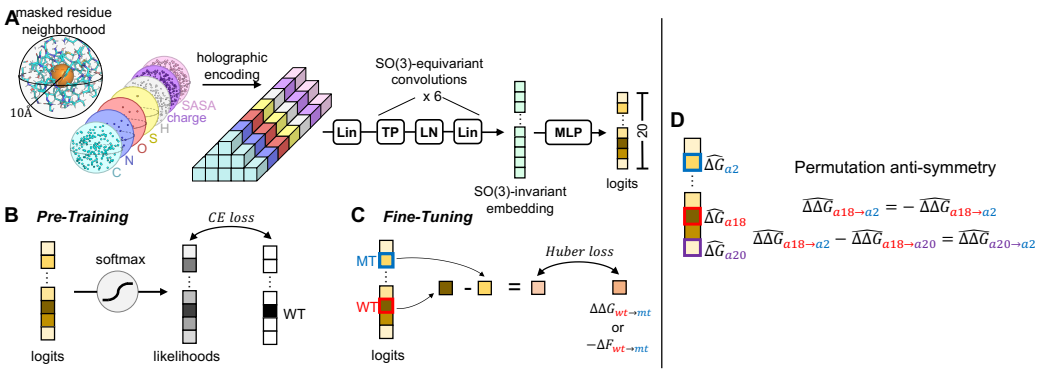


Figure 1: **Schematic of HERMES. (A)** Model architecture. We refer the reader to [25, 26] for details. **(B)** Pre-training procedure. We train HERMES to predict the identity of the central neighborhood's amino-acid, whose atoms have been masked. **(C)** Fine-tuning procedure over mutation effects. We simply fine-tune HERMES to make the difference of logits for two amino-acids regress over the corresponding mutation's score. **(D)** Our fine-tuning procedure makes the 20 logits values effectively converge to predicted $\Delta G$ (or, more broadly, $F$) values, up to a site-specific constant. This ensures that permutation anti-symmetry is respected without the need for data augmentation. This symmetry is however only approximate, as the output is conditioned on a neighborhood bearing the signature of the wildtype amino-acid.

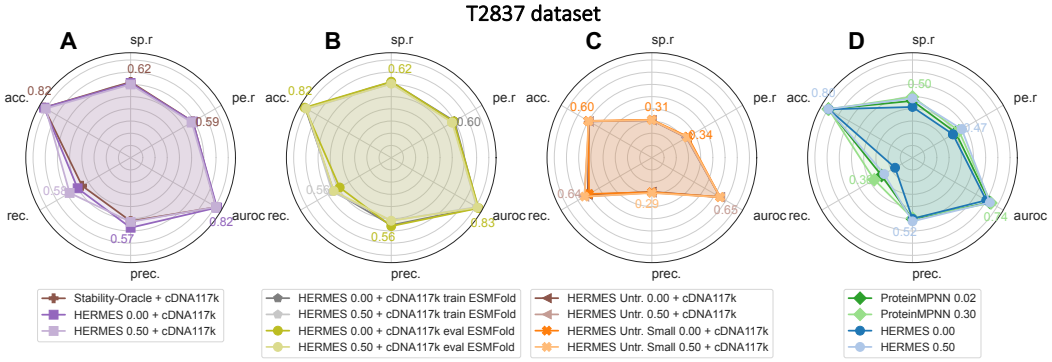


Figure 2: **Predicting stability effect of mutations in T2837 dataset.** The Pearson correlation (pe.r), Spearmann correlation (sp.r), accuracy (acc.), recall (rec.), precision (prec.), and AUROC are shown for different models. **(A)** When fine-tuned on the same dataset, HERMES models perform equivalently to the Stability-Oracle [22]. **(B)** HERMES performance does not change even when trained on ESMFold-predicted structures and evaluated on crystal structures, and vice-versa. **(C)** Non-pre-trained HERMES models perform the worst, and reducing their size from 3.5M to 50k parameters does not improve performance. **(D)** Models without fine-tuning show decent performance. Adding noise to structures during training consistently enhances their performance, though this effect is not observed in fine-tuned models.

desired predictions. Each HERMES model is an ensemble of 10 individually-trained architectures. We trained versions of HERMES after adding Gaussian noise to the 3D coordinates, with standard deviation 0.5 Å, and different random seeds for each of the 10 models. We refer the reader to [26, 25] for further details on the architecture, and the mathematical introduction to SO(3)-equivariant models in Fourier space. We implement HERMES using `e3nn` [29].

We pre-train HERMES on neighborhoods from protein chains in ProteinNet's CASP12 set with 30% similarity cutoff [30], and featurize atomic neighborhoods using atom type - including computationally-added hydrogens - partial charge, and Solvent Accessible Surface Area.

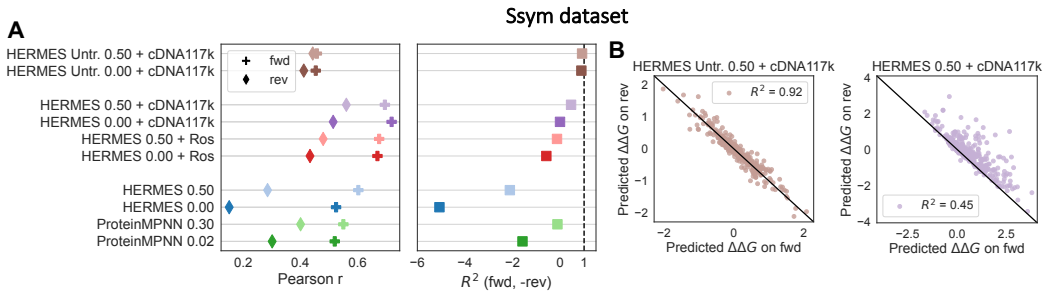


Figure 3: **Permutational anti-symmetry of stability effect of mutations. (A)** Pearson correlation between the measured stability effects of mutations from the Ssym dataset and the predictions on the forward and reverse mutations are shown (left). The effects of reverse mutations are computed using the mutant structures. The $R^2$ between the forward and (negative) reverse predictions is shown, with higher values indicating more respect for Permutational anti-symmetry (right). **(B)** Models with pre-training (right) tend to predict a larger magnitude of stability effect for forward mutations compared to reverse mutations, compared to non-pre-trained models (left).

**Predicting fitness effect of mutations with HERMES.** HERMES can be seen as a generative model for amino-acid labels for a residue, conditioned on the atomic environment surrounding the residue. Conditional generative models of amino-acid labels are shown to successfully make zero-shot predictions for mutational effects [18, 19, 20]. The log-likelihood difference between the original amino acid (often wildtype) $aa_0$ and the mutant $aa_1$ at a given residue $i$, conditioned on the surrounding neighborhood $X_i$, can well approximate mutational effects

$$F_1 - F_0 \propto \log P(aa_1|X_i^{(1)}) - \log P(aa_0|X_i^{(0)}) \tag{1}$$

The superscripts on $X_i$ indicate the structure from which the atomic neighborhood is extracted, highlighting that mutations at a residue can reorganize the surrounding structural neighborhood. Computational tools like Rosetta [27] can be used to relax the structural neighborhoods subject to mutations, when the mutant structure is not available [20]. However, this procedure can be inaccurate and computationally expensive. For practical use of HERMES, we only consider the use of a single (often the original wildtype) structure to predict all possible variant effects, approximating eq. 1 by

$$\hat{F}_1 - \hat{F}_0 \propto \log P(aa_1|X_i^{(0)}) - \log P(aa_0|X_i^{(0)}) = L_{\text{mt}|X_i^{(0)}} - L_{\text{wt}|X_i^{(0)}} \tag{2}$$

where $L_{aa|X_i^{(0)}}$ is the logit associated with the indicated amino acid, conditioned on the surrounding neighborhood for the initial amino acid $aa_0$ (e.g. wildtype). A similar approach was taken by [22].

**Fine-tuning on mutation effects.** We fine-tune pre-trained HERMES models on mutation effects, similar to prior work [2, 22]. However, unlike those works which train a separate regression head using as input embeddings from the pre-trained model, we simply fine-tune the model itself to make the predicted logit differences in eq. 2 regress over mutation effects (Figure 1C); see Section A.1.2 for details. We fine-tune HERMES on several datasets, as reported in the Results section.

**Permutational anti-symmetry for mutation effects.** The thermodynamic changes in the stability of a protein $\Delta\Delta G_{aa_0 \to aa_1}$ by a mutation $aa_0 \to aa_1$ is simply equal to the difference between the free energy of the mutant structure $\Delta G_{aa_1}$ and that of the original structure $\Delta G_{aa_0}$. Thus, the back mutation should have the opposite effect on the stability $\Delta\Delta G_{aa_1 \to aa_0} = -\Delta\Delta G_{aa_0 \to aa_1}$; a similar property occurs when considering triplets of amino acids $\Delta\Delta G_{aa_1 \to aa_0} = \Delta\Delta G_{aa_2 \to aa_0} - \Delta\Delta G_{aa_2 \to aa_1}$. The same anti-symmetric effects are present for the effect of mutations on protein fitness. HERMES is parametrized to automatically account for the anti-symmetric nature of mutations by learning a score for each of the 20 amino acids at a given site, which is associated to their fitness or thermodynamic free energy contribution, up to a site specific constant (Figure 1D). This is in contrast to other popular methods for mutation effect predictions [22, 2]. For example, Stability-Oracle only achieves this anti-symmetric property through data augmentation, by training on all the 380 possible amino acid pairs at each site, resulting in dataset augmented from 117k to 2.2M examples [22].

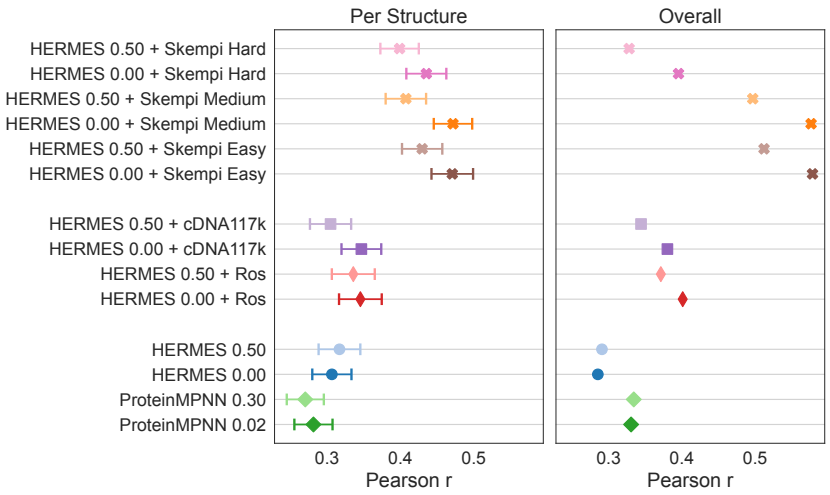


Figure 4: **Single-point mutational effects on binding affinity from SKEMPI.** Averaged "per-structure" and "overall" Pearson correlations between the predicted binding effect of mutations and the measurements from the SKEMPI data are shown, for non-fine-tuned models (bottom), models fine-tuned for stability prediction (middle), and models fine-tuned on the SKEMPI data (top).

## 3 Predicting stability effect of mutations

We evaluate the performance of HERMES on datasets used by RaSP [2] and Stability-Oracle [22]. RaSP was fine-tuned on stability effects computed with Rosetta [27] for 35 protein structures, then tested on Rosetta-computed stability effects for 10 other proteins, as well as on experimentally determined stability effects; we indicate models fine-tuned on this data by "+ Ros". Stability-Oracle was trained on a curated dataset of experimentally measured stability effects termed "cDNA117K", and tested on a dataset termed "T2837"; we indicate models fine-tuned on this data by "+ cDNA117K".

HERMES achieves state-of-the-art performance compared to RaSP (Figs. S1,S2) and Stability-Oracle (Figs. 2A,S4A,S3), using the same fine-tuning data, and without any data augmentation. Moreover, HERMES' predictions are robust to the use of structures computationally resolved by ESMFold [31] at either training or testing time (Figure 2B). Our results also indicate that pre-training on the wildtype amino-acid classification task provides significant help for downstream stability predictions. Notably, models that are pre-trained only, without any fine-tuning, perform significantly better than models trained solely on mutation effects without pre-training (Figs. 2C, 2D). In fact, we could not prevent overfitting in the non-pre-trained HERMES model, even after significantly reducing the model size from 3.5M to 50k parameters (Figure 2C).

Note that HERMES uses only the starting structure to predict mutational effects (eq. 2), and thus, its prediction are only approximately permutation anti-symmetric with respect to mutations. Specifically, the predicted effect of a forward mutation, using the initial structure, is only approximately negative of the effect of the reverse mutation, using the final structure. To assess the extent of deviation from anti-symmetry resulting from our approximation, we use the Ssym dataset, which includes measurements for the stability effect of 352 mutations across 19 different proteins structures, together with the experimentally-determined structures of all of the 352 mutants[32].

We find that HERMES models, as well as ProteinMPNN, consistently predict the stability effect of mutations in the "forward" direction (from wildtype) more accurately than in the reverse direction (Figure 3A). Although none of the Ssym structures were included in our training data, we hypothesize that this effect arises from our models being pre-trained to classify amino acids in wild-type structures, some of which may be homologues to the Ssym structures. Indeed, we observe that removing the pre-training step lessens the discrepancy between forward and reverse predictions, though this comes at the cost of reduced accuracies for both cases (Figure 3A; brown points). Moreover, HERMES models with pre-training tend to predict a larger magnitude of stability effect for forward mutations compared to reverse mutations, further underscoring the bias of these

models toward wildtype structures (Figure 3B). Adding noise during training partly mitigates the bias, as it can reduce the model's wildtype preferences (Table S1).

## 4 Predicting binding effect of mutations

We tested the accuracy of HERMES on predicting the binding effect of mutations on the SKEMPI v2.0 dataset [33], which, to our knowledge, is the most comprehensive dataset comprising mutationl effects on protein-protein binding interactions, with the associated crystal structures of the wildtype's bound complex. We evaluate pre-trained-only ProteinMPNN and HERMES models, as well as HERMES models fine-tuned on stability changes, on predicting binding affinity changes on the wild-type bound structures. Furthermore, for single-point-mutations only, we fine-tune HERMES models on SKEMPI itself using a 3-fold cross-validation scheme, thus ensuring that every point of SKEMPI is evaluated upon. Using structural homology, we provide three splitting strategies with increasing levels of difficulty; see Section A.1.5 for details.
Following [21], we report the accuracy of our predictions both across mutations within each structure individually ("per-structure" correlations), and across mutations pooled from all structure ("overall" correlations). Note that "per-structure" accuracy is particularly relevant when optimizing the binding of a specific protein to its target. As shown in Fig. 4 and Table S2, pre-trained-only models demonstrate some predictive power, and fine-tuning on stability effects enhances the accuracy of binding effect predictions, confirming that transfer learning can be leveraged between the two tasks. Fine-tuning directly on the SKEMPI dataset offers even greater improvements, achieving state-of-the-art performance for "Per-Structure" analysis and competitive results for "Overall" analysis (Table S2).

## 5 Discussion

Here, we presented HERMES, an efficient deep learning method for inferring the effects of mutations on protein function, conditioned on the local atomic environment surrounding the mutated residue. HERMES is pre-trained to model amino-acid preferences in protein structures, and can be optionally fine-tuned on arbitrary mutation effects datasets. We provide HERMES models pre-trained on a large non-reduntant chunk of the protein structure universe, as well as the same models fine-tuned on stability and binding effects of mutations. We thoroughly benchmark HERMES against other state-of-the-art models, showing robust performance on a wide variety of proteins and functions: stability effects, binding affinity, and several deep mutational scanning assays. We open-source our code and data used for experiments, where we provide easy-to-use scripts to run HERMES models on desired protein structures and mutation effects, as well as code to fine-tune our pre-trained HERMES models on the user's own mutation effect data.

## 6 Contributions

GMV led the project in both implementations and experiments. MNP effectively started the project by developing HCNN, and provided code snippets and guidance. WG assisted the implementation of preprocessing pipelines. ED collected and curated the structures from the DMS experiments. KB and UW assisted in running experiments with ESM-1v. AN supervised and designed the project, and acquired funding. GMV and AN wrote the paper.

## 7 Acknowledgement

This work has been supported by the National Institutes of Health MIRA award (R35 GM142795), the CAREER award from the National Science Foundation (grant No: 2045054), the Royalty Research Fund from the University of Washington (no. A153352), and the Allen School Computer Science & Engineering Research Fellowship from the Paul G. Allen School of Computer Science & Engineering at the University of Washington. This work is also supported, in part, through the Departments of Physics and Computer Science and Engineering, and the College of Arts and Sciences at the University of Washington.

## References


[1] Lukas Gerasimavicius, Xin Liu, and Joseph A. Marsh. Identification of pathogenic missense mutations using protein stability predictors. *Scientific Reports*, 10(1):15387, September 2020. Publisher: Nature Publishing Group.

[2] Lasse M Blaabjerg, Maher M Kassem, Lydia L Good, Nicolas Jonsson, Matteo Cagiada, Kristoffer E Johansson, Wouter Boomsma, Amelie Stein, and Kresten Lindorff-Larsen. Rapid protein stability prediction using deep learning representations. *eLife*, 12:e82593, May 2023. Publisher: eLife Sciences Publications, Ltd.

[3] Toyokazu Ishida. Effects of Point Mutation on Enzymatic Activity: Correlation between Protein Electronic Structure and Motion in Chorismate Mutase Reaction. *Journal of the American Chemical Society*, 132(20):7104–7118, May 2010. Publisher: American Chemical Society.

[4] Xiaoyu Wang, Xinben Zhang, Cheng Peng, Yulong Shi, Huiyu Li, Zhijian Xu, and Weiliang Zhu. D3DistalMutation: a Database to Explore the Effect of Distal Mutations on Enzyme Activity. *Journal of Chemical Information and Modeling*, 61(5):2499–2508, May 2021. Publisher: American Chemical Society.

[5] Nicole N. Thadani, Sarah Gurev, Pascal Notin, Noor Youssef, Nathan J. Rollins, Daniel Ritter, Chris Sander, Yarin Gal, and Debora S. Marks. Learning from prepandemic data to forecast viral escape. *Nature*, 622(7984):818–825, October 2023. Publisher: Nature Publishing Group.

[6] Marta Łuksza and Michael Lässig. A predictive fitness model for influenza. *Nature*, 507(7490):57–61, February 2014.

[7] Richard A Neher, Colin A Russell, and Boris I Shraiman. Predicting evolution from the shape of genealogical trees. *Elife*, 3, November 2014.

[8] Brian L. Hie, Varun R. Shanker, Duo Xu, Theodora U. J. Bruun, Payton A. Weidenbacher, Shaogeng Tang, Wesley Wu, John E. Pak, and Peter S. Kim. Efficient evolution of human antibodies from general protein language models. *Nature Biotechnology*, 42(2):275–283, February 2024. Publisher: Nature Publishing Group.

[9] Gabriel J. Rocklin, Tamuka M. Chidyausiku, Inna Goreshnik, Alex Ford, Scott Houliston, Alexander Lemak, Lauren Carter, Rashmi Ravichandran, Vikram K. Mulligan, Aaron Chevalier, Cheryl H. Arrowsmith, and David Baker. Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science*, 357(6347):168–175, July 2017. Publisher: American Association for the Advancement of Science.

[10] Kresten Lindorff-Larsen and Kaare Teilum. Linking thermodynamics and measurements of protein stability. *Protein Engineering, Design and Selection*, 34:gzab002, February 2021.

[11] Robert Karlsson. SPR for molecular interaction analysis: a review of emerging application areas. *Journal of Molecular Recognition*, 17(3):151–161, 2004. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jmr.660.

[12] Douglas M. Fowler and Stanley Fields. Deep mutational scanning: a new style of protein science. *Nature Methods*, 11(8):801–807, August 2014. Publisher: Nature Publishing Group.

[13] Justin B Kinney and David M McCandlish. Massively parallel assays and quantitative Sequence–Function relationships. *Annu. Rev. Genomics Hum. Genet.*, 20(Volume 20, 2019):99–127, August 2019.

[14] Tyler N. Starr, Allison J. Greaney, Sarah K. Hilton, Daniel Ellis, Katharine H.D. Crawford, Adam S. Dingens, Mary Jane Navarro, John E. Bowen, M. Alejandra Tortorici, Alexandra C. Walls, Neil P. King, David Veesler, and Jesse D. Bloom. Deep Mutational Scanning of SARS-CoV-2 Receptor Binding Domain Reveals Constraints on Folding and ACE2 Binding. *Cell*, 182(5):1295–1310.e20, September 2020.

[15] Vytautas Gapsys, Servaas Michielssens, Daniel Seeliger, and Bert L. de Groot. Accurate and Rigorous Prediction of the Changes in Protein Free Energies in a Large-Scale Mutation Scan. *Angewandte Chemie International Edition*, 55(26):7364–7368, 2016. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/anie.201510054.

[16] Joost Schymkowitz, Jesper Borg, Francois Stricher, Robby Nys, Frederic Rousseau, and Luis Serrano. The FoldX web server: an online force field. *Nucleic Acids Research*, 33(suppl_2):W382–W388, July 2005.

[17] Elizabeth H. Kellogg, Andrew Leaver-Fay, and David Baker. Role of conformational sampling in computing mutation-induced changes in protein structure and stability. *Proteins: Structure, Function, and Bioinformatics*, 79(3):830–838, 2011. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.22921.

[18] Adam J. Riesselman, John B. Ingraham, and Debora S. Marks. Deep generative models of genetic variation capture the effects of mutations. *Nature Methods*, 15(10):816–822, October 2018. Publisher: Nature Publishing Group.

[19] Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alex Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. In *Advances in Neural Information Processing Systems*, volume 34, pages 29287–29303. Curran Associates, Inc., 2021.

[20] Michael N. Pun, Andrew Ivanov, Quinn Bellamy, Zachary Montague, Colin LaMont, Philip Bradley, Jakub Otwinowski, and Armita Nourmohammad. Learning the shape of protein microenvironments with a holographic convolutional neural network. *Proceedings of the National Academy of Sciences*, 121(6), February 2024. Publisher: Proceedings of the National Academy of Sciences.

[21] Shitong Luo, Yufeng Su, Zuofan Wu, Chenpeng Su, Jian Peng, and Jianzhu Ma. Rotamer Density Estimator is an Unsupervised Learner of the Effect of Mutations on Protein-Protein Interaction. September 2022.

[22] Daniel J. Diaz, Chengyue Gong, Jeffrey Ouyang-Zhang, James M. Loy, Jordan Wells, David Yang, Andrew D. Ellington, Alexandros G. Dimakis, and Adam R. Klivans. Stability Oracle: a structure-based graph-transformer framework for identifying stabilizing mutations. *Nature Communications*, 15(1):6170, July 2024. Publisher: Nature Publishing Group.

[23] Bian Li, Yucheng T. Yang, John A. Capra, and Mark B. Gerstein. Predicting changes in protein thermodynamic stability upon point mutation with deep 3D convolutional neural networks. *PLOS Computational Biology*, 16(11):e1008291, November 2020. Publisher: Public Library of Science.

[24] S. Benevenuta, C. Pancotti, P. Fariselli, G. Birolo, and T. Sanavia. An antisymmetric neural network to predict free energy changes in protein variants. *Journal of Physics D: Applied Physics*, 54(24):245403, March 2021. Publisher: IOP Publishing.

[25] Gian Marco Visani, William Galvin, Michael Pun, and Armita Nourmohammad. H-Packer: Holographic Rotationally Equivariant Convolutional Neural Network for Protein Side-Chain Packing. In *Proceedings of the 18th Machine Learning in Computational Biology meeting*, pages 230–249. PMLR, March 2024. ISSN: 2640-3498.

[26] Gian Marco Visani, Michael N. Pun, Arman Angaji, and Armita Nourmohammad. Holographic-(V)AE: An end-to-end SO(3)-equivariant (variational) autoencoder in Fourier space. *Physical Review Research*, 6(2):023006, April 2024. Publisher: American Physical Society.

[27] Sidhartha Chaudhury, Sergey Lyskov, and Jeffrey J. Gray. PyRosetta: a script-based interface for implementing molecular modeling algorithms using Rosetta. *Bioinformatics (Oxford, England)*, 26(5):689–691, March 2010.

[28] Peter J. A. Cock, Tiago Antao, Jeffrey T. Chang, Brad A. Chapman, Cymon J. Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, and Michiel J. L. de Hoon. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, June 2009.

[29] Mario Geiger and Tess Smidt. e3nn: Euclidean Neural Networks, July 2022. arXiv:2207.09453 [cs].

[30] Mohammed AlQuraishi. ProteinNet: a standardized data set for machine learning of protein structure. *BMC Bioinformatics*, 20(1):311, June 2019.

[31] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, March 2023. Publisher: American Association for the Advancement of Science.

[32] Corrado Pancotti, Silvia Benevenuta, Giovanni Birolo, Virginia Alberini, Valeria Repetto, Tiziana Sanavia, Emidio Capriotti, and Piero Fariselli. 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, March 2022.

[33] 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, February 2019.

[34] Peter Eastman, Mark S. Friedrichs, John D. Chodera, Randall J. Radmer, Christopher M. Bruns, Joy P. Ku, Kyle A. Beauchamp, Thomas J. Lane, Lee-Ping Wang, Diwakar Shukla, Tony Tye, Mike Houston, Timo Stich, Christoph Klein, Michael R. Shirts, and Vijay S. Pande. OpenMM 4: A Reusable, Extensible, Hardware Independent Library for High Performance Molecular Simulation. *Journal of Chemical Theory and Computation*, 9(1):461–469, January 2013. Publisher: American Chemical Society.

[35] J. Michael Word, Simon C. Lovell, Jane S. Richardson, and David C. Richardson. Asparagine and glutamine: using hydrogen atom contacts in the choice of side-chain amide orientation1. *Journal of Molecular Biology*, 285(4):1735–1747, January 1999.

[36] Jay W. Ponder and David A. Case. Force Fields for Protein Simulations. In *Advances in Protein Chemistry*, volume 66 of *Protein Simulations*, pages 27–85. Academic Press, January 2003.

[37] J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, and D. Baker. Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56, October 2022. Publisher: American Association for the Advancement of Science.

[38] David R Armstrong, John M Berrisford, Matthew J Conroy, Aleksandras Gutmanas, Stephen Anyango, Preeti Choudhary, Alice R Clark, Jose M Dana, Mandar Deshpande, Roisin Dunlop, Paul Gane, Romana Gáborová, Deepti Gupta, Pauline Haslam, Jaroslav Koča, Lora Mak, Saqib Mir, Abhik Mukhopadhyay, Nurul Nadzirin, Sreenath Nair, Typhaine Paysan-Lafosse, Lukas Pravda, David Sehnal, Osman Salih, Oliver Smart, James Tolchard, Mihaly Varadi, Radka Svobodova-Vařeková, Hossam Zaki, Gerard J Kleywegt, and Sameer Velankar. PDBe: improved findability of macromolecular structure data in the PDB. *Nucleic acids research*, 48(D1):D335–D343, January 2020.

[39] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, August 2021. Publisher: Nature Publishing Group.

[40] Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. In *Proceedings of the 39th International Conference on Machine Learning*, pages 8946–8970. PMLR, June 2022. ISSN: 2640-3498.

[41] Sisi Shan, Shitong Luo, Ziqing Yang, Junxian Hong, Yufeng Su, Fan Ding, Lili Fu, Chenyu Li, Peng Chen, Jianzhu Ma, Xuanling Shi, Qi Zhang, Bonnie Berger, Linqi Zhang, and Jian Peng. Deep learning guided optimization of human antibody against SARS-CoV-2 variants with broad neutralization. *Proceedings of the National Academy of Sciences*, 119(11):e2122954119, March 2022. Publisher: Proceedings of the National Academy of Sciences.

[42] Pascal Notin, Mafalda Dias, Jonathan Frazer, Javier Marchena-Hurtado, Aidan N. Gomez, Debora Marks, and Yarin Gal. Tranception: Protein Fitness Prediction with Autoregressive Transformers and Inference-time Retrieval. In *Proceedings of the 39th International Conference on Machine Learning*, pages 16990–17017. PMLR, June 2022. ISSN: 2640-3498.

[43] Roshan M. Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. MSA Transformer. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8844–8856. PMLR, July 2021. ISSN: 2640-3498.

[44] Javier Delgado, Leandro G Radusky, Damiano Cianferoni, and Luis Serrano. FoldX 5.0: working with RNA, small molecules and a new graphical interface. *Bioinformatics*, 35(20):4168–4169, October 2019.

[45] Hahnbeom Park, Philip Bradley, Per Jr. Greisen, Yuan Liu, Vikram Khipple Mulligan, David E. Kim, David Baker, and Frank DiMaio. Simultaneous Optimization of Biomolecular Energy Functions on Features from Small Molecules and Macromolecules. *Journal of Chemical Theory and Computation*, 12(12):6201–6212, December 2016. Publisher: American Chemical Society.

# A Appendix

## A.1 Methods details

### A.1.1 Pre-processing details

To generate our open source pre-processed training data, we use the following procedure: we use OpenMM [34] to fix the PDB files, add missing residues and substitute non-canonical residues for their canonical counterparts; we use the reduce program [35] to add hydrogens; we take partial charges from the AMBER99sb force field [36]; we use BioPython to compute SASA [28]. Both preprocessings procedures keep atoms belonging to non-protein residues and ions, unlike RaSP [2]. Notably, our PyRosetta preprocessings does *not* replace non-canonical residues.

### A.1.2 Fine-tuning details

To greatly speed-up convergence, as a first step of fine-tuning we rescale the weight matrix and bias vector of the network's output layer so the mean and variance of the output logits become the same as that of the training scores. This step requires one initial pass through the training data to get the mean and variance, but it makes the model outputs immediately be in the same distribution as the scores, thus avoiding epochs of fine-tuning just devoted to rescaling the model outputs. We provide easy-to-use code to fine-tune our pre-trained models on arbitrary mutation effect data. Importantly, as we want to produce models using the convention that higher predicted mutation scores correspond to higher fitness, but we fine-tune on $\Delta\Delta G$ values which - since they are energy values - follow the reverse convention (lower $\Delta\Delta G$ means a more stable structure), our code fits the *negative* of Eq. 2 to the target values (wild-type score minus mutant score). In practice, to use the fine-tuning code, just make sure that lower means higher fitness, which can be done by simply flipping the sign of all the target values.

### A.1.3 Use of ESMFold

We use the ESM Metagenomic Atlas API to fold each sequence individually (`https://esmatlas.com/resources?action=fold`).

### A.1.4 Use of RaSP and Stability-Oracle datasets

**RaSP.** We use the RaSP data as provided on their github page (`https://github.com/KULL-Centre/_2022_ML-ddG-Blaabjerg`). The only difference we apply is in the Fermi transform. Since RaSP uses stability changes ($\Delta\Delta G$) computed with Rosetta, which are known to be accurate only in the [-7, 1] range, they pass them through a Fermi transform before training, which effectively "plateaus" outside the [-7, 1] range. We also use the Fermi transform, with the only difference that we *center* it so that 0 maps to zero. This is necessary since HERMES' utput space parameterization is such that the predicted stability change to the same amino-acid is zero ($\Delta\Delta G_{aa_i \to aa_i} = 0$, which is true of real $\Delta\Delta G$ also, but it is not true of the un-centered Fermi transform. Thus the equation we use is:

$$F(\Delta\Delta G) = \frac{1}{1 + e^{-\beta(\Delta\Delta G - \alpha)}} - \frac{1}{1 + e^{\beta\alpha}} \tag{S1}$$

**Stability-Oracle.** The main issue with the data provided by the authors in their github page (`https://github.com/danny305/StabilityOracle/tree/master`) is that the residue-numbers they provide do not align with the residue numbers in the original PDB files, but instead align with some post-processed representation of the structure which, at the time of writing this, is opaque and does not allow us to easily retrieve the original residue-numbers. Thus, we manually modified the datasets' csv files to have residue numbers match those found in the PDB files, and provide them in our repository.

### A.1.5 SKEMPI

After filtering duplicate experiments, the dataset includes: 5,713 $\Delta\Delta G^{\text{binding}}$ values across 331 structures, of which 4,106 are single-point mutations across 308 structures. Further filtering for mutations that belong to structures with at least 10 mutations in the dataset, 116 structures remain with 5,025 total mutations; By restricting to only single-point mutations, we arrive at 93 structures and 3,485 mutations. We consider both "Per Structure" and "Overall" correlations. For multi-point mutations, we use an additive model and neglect epistasis.

The SKEMPI dataset conveniently provides information that helps in making train-test splits without data-leakage. Specifically, each mutation is provided with two pieces of information "hold-out type" and "hold-out proteins". Verbatim from their website (`https://life.bsc.es/pid/skempi2/info/faq_and_help`):

"5) The hold-out type. Some of the complexes are classified as protease-inhibitor (Pr/PI), antibody-antigen (AB/AG) or pMHC-TCR (TCR/pMHC). This classification was introduced to aid in the cross-validation of empirical models trained using the data in the SKEMPI database, so that proteins of a similar type can be simultaneously held out during a cross-validation.

6) The hold-out proteins. This column contains the PDB identifiers (in column 1) and/or hold-out types (column 5) for all the protein complexes which may be excluded from the training when cross-validating an empirical model trained on this data, so as to avoid contaminating the training set with information pertaining to the binding site being evaluated."

For the *Easy* split, we do not consider this information at all, and just split at random. For the *Medium* split, we simply make sure that, if a mutation is in a given split, then all of its "hold-out proteins" are in the same split as well, but not necessarily all of the proteins of the same "hold-out type"; these seem to mostly include closely-related proteins, or even the same exact protein bound to a different target. For the *Hard* split instead, we make sure that, if a mutation is in a given split, then all of the proteins of the same "hold-out type" are in the same split as well. This is overkill in practice, since for instance it precludes the use of any antibody-antigen data to predict on antibody-antigen complexes; it provides, however, a great test of generalization ability. We note that sometimes there are are proteins with multiple "hold-out types"; in these cases, we randomly chose one type for the protein.

## A.2 Baselines

**H-CNN [20].** We mention H-CNN because HERMES is effectively built on top of it, with HERMES 0.00and HERMES 0.50being directly comparable to it - except for the improved speed of HERMES' forward pass, which we tested by re-implementing the H-CNN architecture in our code. H-CNN is only trained on masked amino-acid prediction - our pre-training task. Its authors showed that H-CNN learned a model akin to a physical potential, and able to predict mutation effects of stability and binding via eq. 2, albeit only on two systems.

**Stability-Oracle [22].** Similar to HERMES, Stability-Oracle is trained in two steps: first a graph attention model is pre-trained to predict masked amino-acids from their local atomic environment (i.e. "neighborhood"). The model regressing over mutation effects is then constructed and trained as follows. For a site on a structure, the masked neighborhood's embedding $h$ is extracted from the pre-trained graph attention model. This embedding is concatenated with embeddings of the "from" and "to" amino-acids separately, and the two inputs are individually fed to a transformer network, yielding the two amino-acid specific embeddings $e_{aa_{\text{from}}}$ and $e_{aa_{\text{to}}}$. These are then substracted, and $(e_{aa_{\text{to}}} - e_{aa_{\text{from}}})$ is fed to a final 2-layer MLP that outputs a scalar representing $\Delta\Delta G_{aa_{\text{from}}\to\text{to}}$. Interesingly, up to right before the MLP, the output symmetries are not yet broken, because each $e_{aa_i}$ is computed independently of any other amino-acid. The symmetries only get broken in the MLP: in fact, if the MLP were a linear layer with no bias, the symmetries would be respected. To make their model respect the symmetries, the authors train with data augmentation of reversibility and permutation.

**RaSP [2].** Similar to HERMES, RaSP is trained in two steps: first, a neural network -

specifically a 3DCNN - is pre-trained to predict masked amino-acids from their local atomic environment (i.e. "neighborhood"). Then, a small fully-connected neural network with a single output is trained to regress over mutation effects, using as input neighborhoods' embeddings from the 3DCNN, the one-hot encodings of wildtype and mutant amino-acids, and the wildtype and mutant amino-acids' frequencies in the pre-training data. RaSP is fine-tuned on the stability effect of mutations $\Delta\Delta G$, computationally determined with Rosetta [27], which we also use to fine-tune HERMES. We do not reproduce results of RaSP in this work, and instead show the values reported in the paper.

**ProteinMPNN [37].** ProteinMPNN is a tool for protein inverse-folding. The tool is most commonly used to sample amino-acid sequences conditioned on a protein's backbone structure, and optionally a partial sequence. As ProteinMPNN also outputs probability distributions of amino-acids for the sites that are to be designed, it can also be used to infer mutation effects by computing the log-likelihood ratio presented in eq. 1. Like for HERMES, we consider ProteinMPNN models trained with two noise levels: 0.02 Å(virtually no noise) and 0.30 Å. We provide scripts to infer mutation effects built upon a public fork of the ProteinMPNN repository.

**ESM-1v [19].** This is the Protein Language Model (PLM) of the ESM family trained specifically for improved zero-shot predictions of mutation effects. As the training objective is predicting amino-acids that have been masked from the sequence, mutation effects are also predicted using the log-likelihood ratio (eq. 1). To our knowledge, this is the strongest representative of PLMs for inferring mutation effects. We show a mix of previously-reported scores, and scores computed using their codebase. For our in-house ESM-1V predictions, wildtype sequences were obtained from the corresponding PDB file and verified against the European Bioinformatics Institute's PDBe database via their REST API [38]. Mutation effect predictions were computed with ESM's built-in *wildtype marginal* method; we attempted using the *masked marginal* method but ran into several errors, so we stuck to *wildtype marginal* as it was more reliable, and also had very similar performances in the few instances in which both methods worked.

**DeepSequence [18].** This is a state-of-the-art model for inferring mutation effects from sequence alone. It uses a variational auto-encoder of full protein sequences to and infers mutation effects via eq. 1. We only show previously-reported scores.

### A.3 Extended Results

#### A.3.1 Wildtype amino-acid classification

In Table S1 we show Classification Accuracy of HERMES models, when predicting the amino-acid identity of the masked residue at the center of a neighborhood. Adding noise during training, as well as fine-tuning over stability effects, reduces the model's predictions of the wild-type. Models that were *not* pre-trained on amino-acid classification, and only trained on stability effects, predict the wildtype only barely more than random. As seen in Figure 3, the models' bias in predicting the wildtype most commonly found in nature.

#### A.3.2 Results on predicting Deep Mutational Scanning assays

We evaluate model performance on 27 out of the 41 Deep Mutational Scanning (DMS) studies collected by [18] and considered by [19]. To simplify the analysis, we consider only the 37 studies containing single-point mutations only. For these, only the proteins' sequences were available to us a priori. Starting from the sequences, we augmented the dataset with both experimental structures that we identified in the RCSB website [1] and AlphaFold2 structures, either from the AlphaFold database [2], or folded using the AlphaFold2 [39] google colab with default parameters. Keeping only studies with at least one high-quality structure, we were left with 25 studies, many of which with only the AlphaFold-generated structure. Some proteins have multiple experimental structures, as in each structure they are bound to different a different and it was not obvious from the study of origin which ligand was more appropriate. We provide structures and detailed notes for each study on our github repository.

[1] `https://www.rcsb.org/`

[2] `https://alphafold.ebi.ac.uk/`

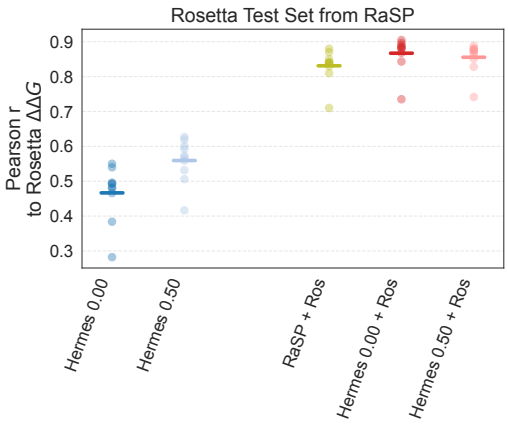


Figure S1: **Pearson correlation of predictions against RaSP's test set of Rosetta-computed stability effects for 10 proteins [2]** Each dot is a protein; the horizontal bar is the mean. HERMES models achieve better Pearson correlation using the same training data. We observe that centering the Fermi transform (Eq. S1) provided a slight boost in performance.

In Figures S7 and S8 we show absolute Pearson and Spearman correlations between model predictions and expreiments for the 27 studies, selected as described above. We use absolute values for simplicity, as assays may have either positive or negative sign associated with higher fitness. Patterns are similar to those we found for the stability effect of mutations $\Delta\Delta G$: training with noise improves pre-trained-only models, and so does pre-processing with PyRosetta. Models fine-tuned on stability effects see their performance improved. However, the best structure-based model (HERMES 0.00 + Ros 0.50 + FT with mean Pearson r of 0.40) still performs significantly worse, on average, compared to the state-of-the-art sequence-based models (DeepSequence [18] with 0.50, and ESM-1v [19] with 0.47).

Table S1: **Performance of HERMES models on wildtype amino-acid classification on 40 CASP12 test proteins.** As expected, models trained with noise have worse Accuracy. Interestingly, models fine-tuned on stability $\Delta\Delta G$ values retain part of their accuracy, whereas models that are *only trained* for stability prediction have almost no predictive power of the wildtype amino-acid. Differences between using the Pyrosetta and Biopython pipelines are negligible.

| **Model** | **Pyrosetta Accuracy** | **Biopython Accuracy** |
|---|---|---|
| HERMES 0.00 | 0.73 | 0.75 |
| HERMES 0.50 | 0.64 | 0.65 |
| HERMES 0.00 + Ros | 0.41 | 0.40 |
| HERMES 0.50 + Ros | 0.38 | 0.37 |
| HERMES 0.00 + cDNA117k | 0.47 | 0.45 |
| HERMES 0.50 + cDNA117k | 0.39 | 0.38 |
| HERMES 0.00 + cDNA117k train ESMFold | 0.46 | 0.49 |
| HERMES 0.50 + cDNA117k train ESMFold | 0.40 | 0.40 |
| HERMES Untr. 0.00 + cDNA117k | 0.09 | - |
| HERMES Untr. 0.50 + cDNA117k | 0.08 | - |

Table S2: **Results on predicting single-point mutation effects on protein-protein binding in SKEMPI.** Results above the double-line are taken from [21]; see their paper for a detail introduction of each model being compared ([40, 41, 42, 43, 44, 45]). The HERMES models most comparable - in terms of training procedure - to the models reported by [21] are the models trained on the *Easy* split: for it, we use 3-fold cross-validation on datasets split by PDB structure without further restrictions. However, we do not know which exact PDBs are in the splits for [21] and could not recover them from their codebase.

| **Method** | **Per-Struct. Pearson** | **Per-Struct. Spearman** | **Overall Pearson** | **Overall Spearman** |
|---|---|---|---|---|
| ESM-1v | 0.0422 | 0.0273 | 0.1914 | 0.1572 |
| PSSM | 0.1215 | 0.1229 | 0.1224 | 0.0997 |
| MSA Transf. | 0.1415 | 0.1293 | 0.1755 | 0.1749 |
| Tranception | 0.1912 | 0.1816 | 0.1871 | 0.1987 |
| Rosetta | 0.3284 | 0.2988 | 0.3113 | 0.3468 |
| FoldX | 0.3908 | 0.3640 | 0.3560 | 0.3511 |
| DDGPred | 0.3711 | 0.3427 | 0.6515 | 0.4390 |
| End-to-End | 0.3818 | 0.3426 | 0.6605 | 0.4594 |
| B-factor | 0.1884 | 0.1661 | 0.1748 | 0.2054 |
| ESM-IF | 0.2308 | 0.2090 | 0.2957 | 0.2866 |
| MIF-$\Delta$logit | 0.1616 | 0.1231 | 0.2548 | 0.1927 |
| MIF-Net. | 0.3952 | 0.3479 | **0.6667** | 0.4802 |
| RDE-Linear | 0.3192 | 0.2837 | 0.3796 | 0.3394 |
| RDE-Net. | 0.4687 | **0.4333** | 0.6421 | **0.5271** |
| ProteinMPNN 0.02 | 0.2813 | 0.2824 | 0.3307 | 0.3153 |
| ProteinMPNN 0.30 | 0.2702 | 0.2549 | 0.3344 | 0.2893 |
| HERMES 0.00 | 0.3064 | 0.2866 | 0.2854 | 0.2721 |
| HERMES 0.50 | 0.3168 | 0.3075 | 0.2910 | 0.2863 |
| HERMES 0.00 + Ros | 0.3453 | 0.3072 | 0.4011 | 0.3522 |
| HERMES 0.50 + Ros | 0.3357 | 0.3069 | 0.3713 | 0.3276 |
| HERMES 0.00 + cDNA117k | 0.3467 | 0.3307 | 0.3802 | 0.3419 |
| HERMES 0.50 + cDNA117k | 0.3046 | 0.2943 | 0.3443 | 0.2881 |
| HERMES 0.00 + cDNA117k train ESMFold | 0.3405 | 0.3350 | 0.3957 | 0.3375 |
| HERMES 0.50 + cDNA117k train ESMFold | 0.3093 | 0.2939 | 0.3643 | 0.3079 |
| HERMES 0.00 + Skempi Easy | 0.4707 | 0.4331 | 0.5781 | 0.4761 |
| HERMES 0.50 + Skempi Easy | 0.4296 | 0.3892 | 0.5120 | 0.4203 |
| HERMES 0.00 + Skempi Medium | **0.4716** | 0.4302 | 0.5762 | 0.4655 |
| HERMES 0.50 + Skempi Medium | 0.4074 | 0.3676 | 0.4966 | 0.4029 |
| HERMES 0.00 + Skempi Hard | 0.4353 | 0.3979 | 0.3954 | 0.3802 |
| HERMES 0.50 + Skempi Hard | 0.3988 | 0.3592 | 0.3280 | 0.3216 |

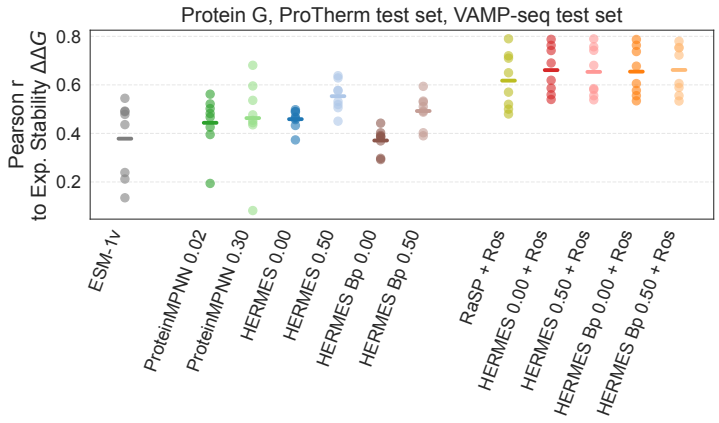


Figure S2: **Pearson correlation of predictions against RaSP's test set of experimental stability effects for 8 proteins [2].** Each dot is a protein; the horizontal bar is the mean. Zero-shot HERMES models perform similarly to ProteinMPNN models, with noise consistently improving performance. Zero-shot HERMES models using the Biopython pipeline are slightly worse. Differences between noise level and pre-processing pipeline become insignificant after fine-tuning. Notably, HERMES models achieve better Pearson correlation than RaSP using the same training data.

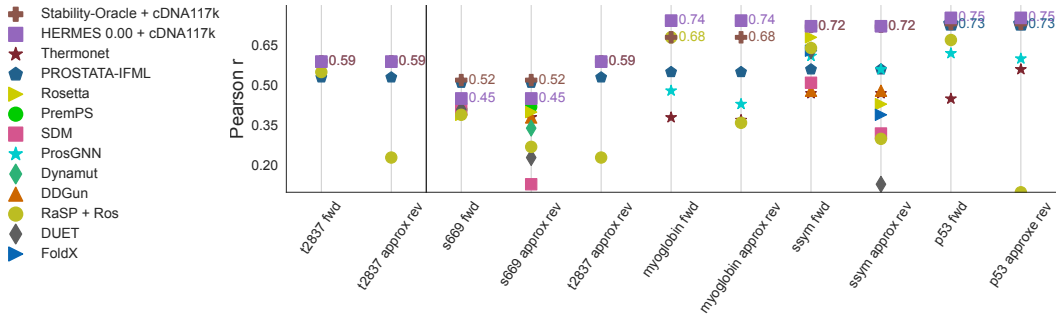


Figure S3: **Pearsonr correlation of several models' predictions and experimental stability effects, from the T2837 dataset and its subsets.** This is effectively a replica of a figure in [22]. Results of all models other than HERMES were taken from [22]. We label the correlations on "reverse" mutations as *approximate* because predictions were made with conditioning only on the wildtype structures. As discussed in the Methods section, HERMES respects approximate permutational anti-symmetry (i.e. "forward" and "reverse" mutations are anti-symmetric) by design, without the need for data augmentation.

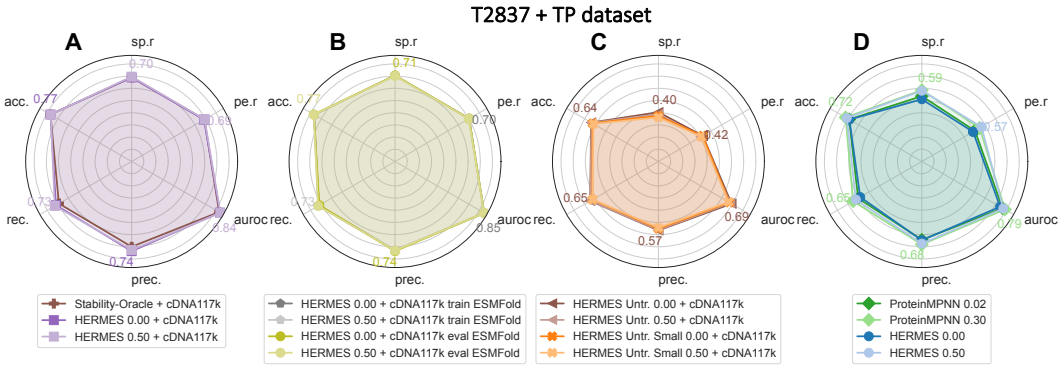


Figure S4: **Predicting stability effect of mutations in T2837 + TP dataset.** The Pearson correlation (pe.r), Spearmann correlation (sp.r), accuracy (acc.), recall (rec.), precision (prec.), and AUROC are shown for different models. "TP" is short for "Thermodynamic Permutations", i.e. the data augmentation technique of permutational anti-symmetry devised by [22]. Similar trends as in Figure 2 are observed.

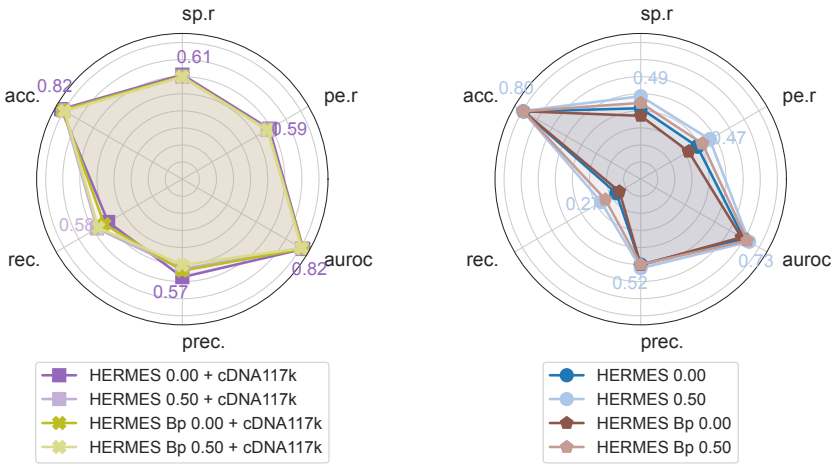


Figure S5: **Predicting stability effect of mutations in T2837: comparison between Pyrosetta and Biopython pipelines.** Similar to the results on RaSP data (Figure S2) models using Biopython preprocessing perform a bit worse than those using Pyrosetta, but the difference is rendered insignificant after fine-tuning.

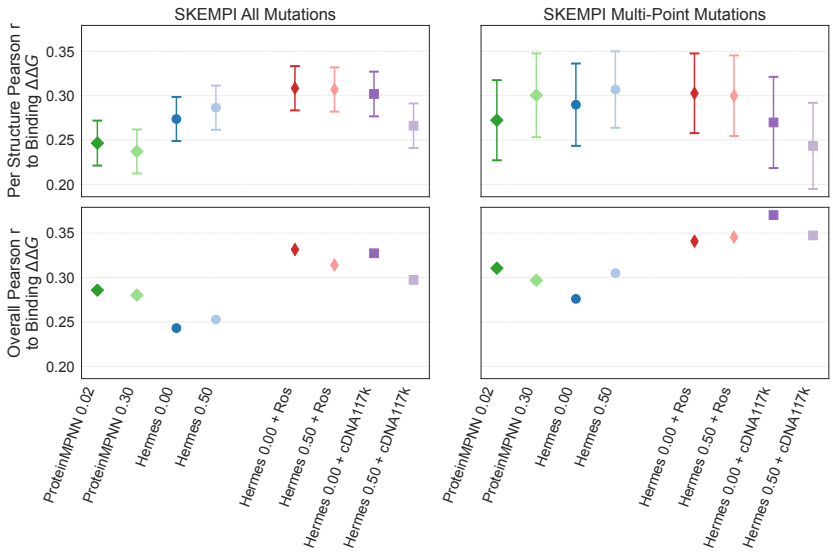


Figure S6: **Pearson correlation on SKEMPI multi-point mutations.**.

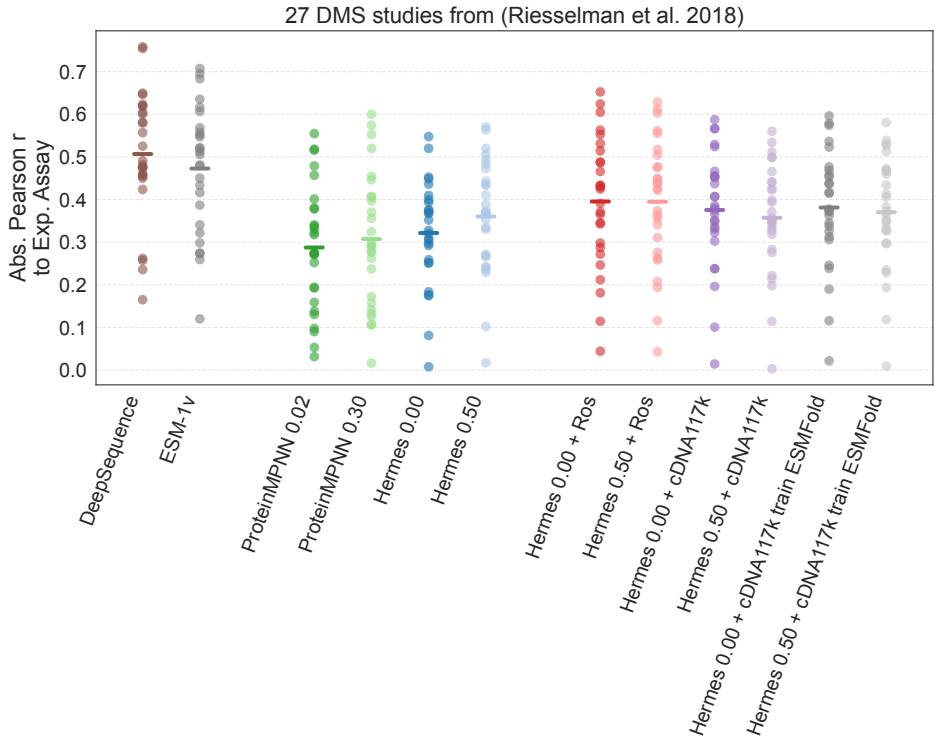


Figure S7: **Pearson correlation of models' predictions against DMS experimental assays from [18].** Each point is a study (single protein), and horizontal bars are mean values. Fine-tuning HERMES models on stability $\Delta\Delta G$ values improves performance, but it does not enable them to reach the levels of state-of-the-art sequence-based models DeepSequence and ESM-1v.

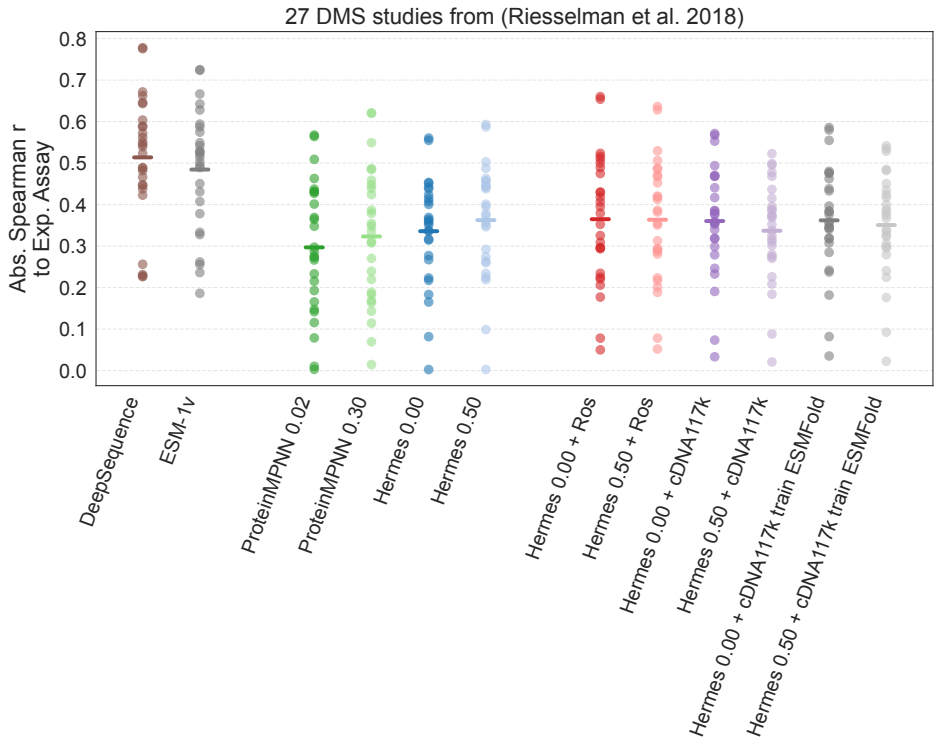


Figure S8: **Spearman correlation of models' predictions against DMS experimental assays from [18].** Each point is a study (single protein), and horizontal bars are mean values. Fine-tuning HERMES models on stability $\Delta\Delta G$ values improves performance, but it does not enable them to reach the levels of state-of-the-art sequence-based models DeepSequence and ESM-1v.