Generalization of Protein Foundation Models for Engineered Fluorescent Biosensors

Anonymous Authors[1]

Abstract

Protein foundation models (PFMs) are increasingly used in fitness prediction tasks in which engineers seek to identify sequences with improved function. However, these models are often evaluated under random splits, which may not reflect the generalization required for engineering superior variants. Further, results from existing classical protein fitness benchmarks may not generalize to fluorescent protein biosensors which require designing finely tuned protein dynamics. We benchmark embeddings from seven PFMs across four regression heads on a dataset of 1,314 mutated GCaMP variants, evaluating each model–head pair under a random baseline and two extrapolation regimes: novel-region splits that hold out protein sequence regions, and low-to-high fitness splits that hold out the highest-fitness variants. Extrapolative splits result in substantial performance drops across all models, and structure- and MSA-based conditioning partially mitigate this drop on novel-region splits but not on low-to-high splits. The two extrapolation regimes also differ in ranking transfer: novel-region rankings closely track random-split rankings while low-to-high rankings diverge. These findings indicate that model and architecture choices should be driven by which form of generalization the application requires, with no single configuration optimal across regimes.

1. Introduction

Genetically-encoded fluorescent biosensors are engineered proteins that enable real-time monitoring of physiological signals in live cells. A notable example is the GCaMP family of indicators, which are widely used in neuroscience to monitor the intracellular calcium transients evoked by spiking neuronal activity (Nakai et al., 2001). Biosensors are typically generated by combining existing ligand-binding and fluorescent protein domains such that binding allosterically regulates fluorescence emission (Fig. 1A). Optimization of fluorescent biosensors is an especially challenging task because function depends on the dynamic interaction of multiple domains. Successful efforts to generate GCaMP variants with improved sensitivity and faster kinetics required years of time-consuming and technically difficult measurements in live neurons (Tian et al., 2009; Chen et al., 2013; Zhang et al., 2023). Machine learning models that predict biosensor function directly from sequence offer a path to accelerating this engineering loop (Wait et al., 2024; Xiong et al., 2025).

Protein foundation models (PFMs) have emerged as strong general-purpose feature extractors for downstream biological tasks, with their representations enabling effective downstream prediction of binding affinity, stability, and toxicity (Notin et al., 2024; Lin et al., 2023; Jiang et al., 2025; Yang et al., 2023a). Prior benchmarks evaluated embedding choice on these classical engineering tasks under random or position-based splits, and showed that the choice of model affects performance though the advantage over simple baselines varied (Dallago et al., 2021; Notin et al., 2023a; Groth et al., 2023; Didi et al., 2026). Whether such benchmarks transfer to biosensor function prediction, where the relevant property emerges from dynamic interactions between protein domains rather than static structure, is an open empirical question. Engineering improved biosensors also requires generalization to variants outside the training distribution such as variants with mutations in novel regions and variants in unobserved regions of the fitness landscape (Yang et al., 2019), which are two extrapolation regimes that have typically been studied in isolation (Notin et al., 2023a) or on different datasets (Didi et al., 2026).

In this work, we benchmark embeddings from seven protein foundation models on a dataset of mutated GCaMP variants with fluorescence response measurements in a semi-supervised manner with frozen model weights. We evaluate each model–head combination under three regimes: random splits as an in-distribution baseline, novel region splits that hold out novel sequence regions, and low-to-high fitness splits that hold out the highest-fitness variants. By varying

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Submitted to the 2026 Workshop on Generative and Agentic AI for Biology (ICML 2026). Do not distribute.

all three axes simultaneously, we characterize how representation choice, architecture choice, and extrapolation regime interact with each other.

We find that PFM embeddings outperform a one-hot baseline on random splits, with structure-aware and multimodal models broadly outperforming sequence-only models and more expressive head architectures substantially outperforming simpler pooling heads. Under both novel-region and low-to-high fitness extrapolation, performance drops substantially across all model and architecture combinations, though PFM embeddings retain a clear advantage over one-hot encoding under most architectures. However, for novel-region extrapolation, the best model and architecture choices generally match those for random splits, while for fitness-space extrapolation, they do not. Our findings suggest that sequence-space and fitness-space extrapolation are qualitatively different generalization problems for protein function prediction, and that model selection should take into account the form of generalization a given engineering application requires.

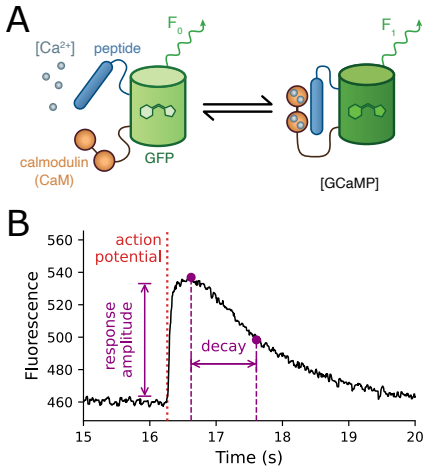

Figure 1. **(A)** *Schematic of GCaMP*: Calcium binding pulls calmodulin and the peptide together, switching GFP from dim to bright. **(B)** Labels are computed from fluorescence traces stimulated by action potentials: d' measures amplitude of the response; $decay$ measures the time from peak response to half of its value

2. Methods

2.1. Dataset

The dataset consists of 1,314 unique sequences from random mutagenesis using the jGCaMP8s sensor as the scaffold (Zhang et al., 2023). Each variant is screened in a neuronal culture-based assay (Wardill et al., 2013) for their fluorescence response to varying number of action potentials (AP) trains (Fig. 1B). The metrics – d' measuring the amplitude of the fluorescence response normalized by the baseline noise level and $decay$ measuring the half-decay time for the response to drop to half of its peak value – are extracted for each of the four AP trains and form the labels in the dataset (Appendix A).

2.2. Protein Foundation Models

We benchmark seven protein foundation models, grouped coarsely by training objective and input modality. **Protein language models** are trained on sequences alone via masked or autoregressive language modeling. We select ESM-2 (Lin et al., 2023), a transformer with masked language modeling; CARP (Yang et al., 2024), a CNN with masked language modeling; and ProGen3 (Bhatnagar et al., 2025), a sparse mixture-of-experts autoregressive model. **Multimodal models** extend language modeling with additional modalities at training or inference. We benchmark ESM-3, (Hayes et al., 2025) which jointly models sequence, structure, and functional annotations, and PoET-2 (Truong & Bepler, 2025), which is retrieval-augmented, conditioning on homologous sequences and structures. **Structure-prediction models** are trained for 3D structure prediction, with hidden representations useful for downstream tasks (Gazizov et al., 2026). We choose AlphaFold-3 (Abramson et al., 2024) and Chai-1 (Chai Discovery et al., 2024) as the candidates for this class.

We also include a one-hot baseline that has previously been found to be competitive with pre-trained embeddings (Hsu et al., 2022). For all models, weights are frozen and representations are extracted from the penultimate layer (Appendix C.1). For structural or MSA conditioning where available, we provide the jGCaMP8.410.80 crystal structure (Zhang et al., 2023) with masking at mutated positions, and a fixed MSA with the query row swapped per variant.

2.3. Head Architectures

We evaluate four head architectures commonly used in fitness prediction tasks. All heads take per-residue embeddings as input and jointly predict the eight label values (d' and $decay$ across four nAP conditions). MeanPoolMLP applies mean-pooling over residues followed by a 2-layer MLP. AttnPoolMLP replaces mean-pooling with a learned weighted mean over residues. MutationDeltaMLP uses position-wise differences between variant and wild-type embeddings as input features, thus using only the local perturbation signal. ConvMaxPool uses a 1-layer 1D CNN followed by max-pooling and an MLP (Dallago et al., 2021) (more details in Appendix D).

2.4. Data Splits

Fig. 2 shows the train-test splits used to evaluate the different generalization regimes. To probe *generalization to unseen regions in the sequence*, we create **novel-region** splits as defined by the "contiguous" scheme in Notin et al.

(2023b). The test set is defined by a contiguous window W on the scaffold: test variants have mutations only inside W, and training variants have mutations only outside W. Variants with mutations both inside and outside W are excluded, guaranteeing that test positions never appear in training. We select 5 windows of size 5% of the total variants that are maximally separated along the scaffold (Appendix B.1).

In addition, a **low-to-high** split is designed to evaluate *extrapolation in the fitness space*. Variants are ranked by mean rank across all eight label values. The high-fitness variants in 95$^{\text{th}}$ to 100$^{\text{th}}$ percentile are allocated to the test set, the low-fitness variants from 0$^{\text{th}}$ to 80$^{\text{th}}$ percentile are allocated to the train set, and the remaining variants are excluded from the analysis to reduce distribution overlap between the two.

As a baseline, we create 5 **random** splits by randomly holding out 5% for testing. 15% of variants are excluded from both train/test sets to match the sizes of the other test sets).

2.5. Training and Evaluation

All embedding model and architecture pairs were tuned for hyperparameters on validation correlations from a random train-test split using a grid-search (Appendix E). All training runs spanned 2000 epochs with early stopping (patience 250 epochs) and a learning rate of 10^{-4}. We use 4-fold cross-validation, ensembling the models for prediction (Lakshminarayanan et al., 2017), and report Spearman rank correlation as the primary evaluation metric (Notin et al., 2023a; Dallago et al., 2021). Statistical methodology is detailed in Appendix F.

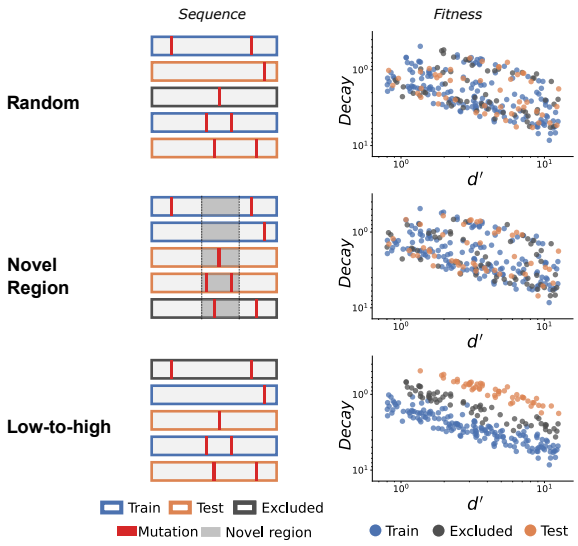

Figure 2. Schematic illustrations of sequence space (left) and fitness space (right), for the three train-test split conditions. Novel region splits are generated by holding out variants with mutations in a contiguous window. A low-to-high fitness split is generated by holding out a top fraction of variants and excluding variants in the middle-fitness range to prevent overlap.

3. Results

Fig. 3 shows Spearman rank correlations (ρ) for all models, heads, and train-test splits, values averaged across all the d' and $decay$ labels across the four nAP values.

Baseline interpolation performance For our baseline random split, all PFM embeddings outperform a one-hot encoding baseline across all head architectures (mean ρ difference between embedding and one-hot encoding across splits $\Delta\rho = +0.44, +0.43, +0.30$, and $+0.15$ for MeanPoolMLP, AttnPoolMLP, MutationDeltaMLP, and ConvMaxPool respectively). The more expressive heads (MutationDeltaMLP and ConvMaxPool) generally outperform the simpler ones, but result in a smaller advantage when used with protein models compared to one-hot encoding. This is consistent with prior observations that complex heads extract competitive signal from raw sequence input (Hsu et al., 2022). The mean Spearman correlations (across all architectures) of structure-aware and multimodal models broadly cluster above sequence-only models, with ρ ranging from 0.57 (ESM-3) to 0.65 (Chai-1) for the former and 0.42 (CARP) to 0.53 (ESM-2, ProGen-3) for the latter, indicating that structure and MSA-conditioning broadly improve performance on biosensor fitness prediction.

Extrapolation to novel regions On novel region evaluations, we find a substantial drop in performance compared to random splits across every model and head architecture with a mean fractional drop of 49.6% in ρ across protein models. The general trends from the random split results are upheld: PFM embeddings retain an advantage over the one-hot baseline under most architectures, with the largest effect under simpler pooling heads (difference between random and novel-region correlation $\Delta\rho = +0.27$ under MeanPoolMLP; $\Delta\rho = +0.19$ under AttnPoolMLP; ConvMaxPool $\Delta\rho = +0.16$) and a smaller but still significant effect under MutationDeltaMLP ($\Delta\rho = +0.08$). The model-family ordering on novel region splits closely tracks the ordering on random splits (Spearman $\rho = 0.93$ between mean model correlations across the two regimes), indicating that the PFMs that perform best in-distribution also extrapolate best to novel sequence regions. The MSA- and structure-conditioned models (AlphaFold3, Chai-1, ESM-3, PoET-2) retain strong performance in extrapolation with an average mean $\rho = 0.37$ on novel-region splits, roughly twice the $\rho = 0.18$ average of the sequence-only models. Critically, this advantage persists within model families: the structure- and MSA-conditioned configurations of AlphaFold3, Chai-1, and PoET-2 outperform the sequence-only configurations of the same models (mean within-family $\Delta\rho = +0.27, +0.11, +0.14$, all significant; ESM-3 effect is not significant), indicating that conditioning input plays an important role in this generalization advantage.

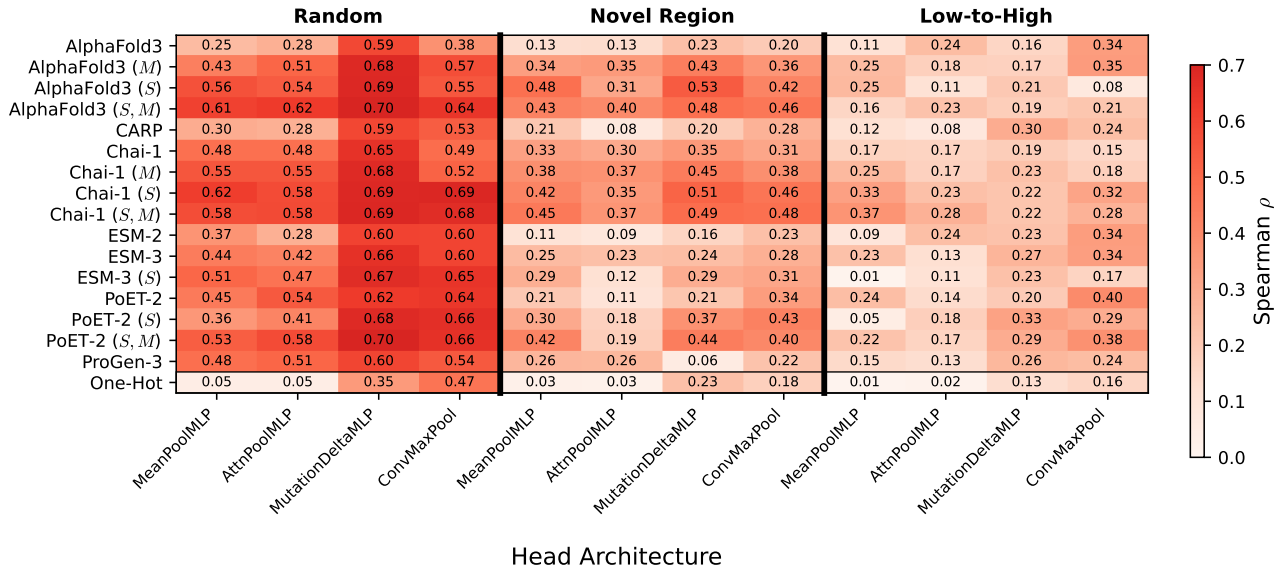

Figure 3. Mean Spearman ρ across splits and labels is plotted for all embeddings and architectures for the 3 evaluation strategies. We note a significant drop in performance in novel region and low-to-high splits. Models marked with S take wild-type structure as conditioning and M take or use MSA internally as conditioning.

Extrapolation from low-to-high fitness space Performance under low-to-high fitness splits is further decreased compared to novel-region splits, with a mean fractional drop of 62% relative to random-split performance. The model-family ordering on low-to-high splits diverges from the random-split ordering (model ranking $\rho = 0.46$), with sequence-only models such as CARP and ESM-2 showing comparable performance to the best-performing models under certain heads. The conditioning advantage from the novel-region splits is absent: structure-conditioned configurations of AlphaFold-3, Chai-1, PoET-2, and ESM-3 perform comparably to or worse than their sequence-only counterparts on the highest-fitness variants, suggesting that fitness-space extrapolation tests a different generalization capability than sequence-space extrapolation.

4. Discussion

In this work, we evaluate extrapolation capabilities of embeddings from widely-used protein foundation models in sequence space and fitness space for fluorescent biosensor function. We note a substantial performance drop across all models and architectures in both extrapolation regimes , reinforcing the findings of prominent benchmarks (Dallago et al., 2021; Didi et al., 2026). However, the two regimes behave differently: while novel-region rankings closely track in-distribution rankings, fitness-space rankings transfer less consistently from random-split orderings. Whether alternative training objectives, such as different losses or head architectures, would close these extrapolation gaps is a natural direction for further investigation.

Structure and MSA conditioning are effective for novel-region extrapolation because these inputs provide the three-dimensional structural context for the held-out region of the sequence. They are less helpful, however, for fitness extrapolation. Structural templates capture only one conformational state and MSA hits tend to inform about each component domain (CaM, GFP) in isolation rather than about their non-native coupled function. This suggests that the conditioning failure on fitness extrapolation may be specific to proteins where function emerges from non-native domain interactions rather than properties of any single domain. Whether this pattern extends to other engineered proteins with non-native coupled functions, such as allosterically regulated transcription factors or engineered receptors, remains an open question for future work.

ML-assisted directed evolution crucially depends on models generalizing reliably in fitness space to predict variants with improved function, which is the regime where our results show the largest performance gaps. When structural templates or MSAs are available, structure-conditioned models provide substantial improvements under in-distribution and novel-region extrapolation. When such inputs are unavailable, sequence-only configurations of any of the benchmarked models perform comparably and outperform a one-hot baseline; the choice in this setting is more flexible. For fitness-space extrapolation, however, the conditioning advantage disappears entirely, and sequence-only models can match structure-conditioned ones. The choice of model and head architectures should therefore be driven by which form of generalization the engineering application requires.

Data & Code Availability

The data has not been released publicly at the time of writing. The code will be made availablle on GitHub.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning and its application to protein engineering. Our contributions are evaluative. We benchmark existing protein foundation models on a biosensor function prediction task to inform model selection. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., Bodenstein, S. W., Evans, D. A., Hung, C.-C., O'Neill, M., Reiman, D., Tunyasuvunakool, K., Wu, Z., Žemgulytė, A., Arvaniti, E., Beattie, C., Bertolli, O., Bridgland, A., Cherepanov, A., Congreve, M., Cowen-Rivers, A. I., Cowie, A., Figurnov, M., Fuchs, F. B., Gladman, H., Jain, R., Khan, Y. A., Low, C. M. R., Perlin, K., Potapenko, A., Savy, P., Singh, S., Stecula, A., Thillaisundaram, A., Tong, C., Yakneen, S., Zhong, E. D., Zielinski, M., Žídek, A., Bapst, V., Kohli, P., Jaderberg, M., Hassabis, D., and Jumper, J. M. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016):493–500, June 2024. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-024-07487-w. URL https://www.nature.com/articles/s41586-024-07487-w.

Bhatnagar, A., Jain, S., Beazer, J., Curran, S. C., Hoffnagle, A., Ching, K., Martyn, M., Nayfach, S., Ruffolo, J. A., and Madani, A. Scaling unlocks broader generation and deeper functional understanding of proteins. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, December 2025. URL https://openreview.net/forum?id=yvGL2HP7pU. Spotlight poster.

Chai Discovery, Boitreaud, J., Dent, J., McPartlon, M., Meier, J., Reis, V., Rogozhnikov, A., and Wu, K. Chai-1: Decoding the molecular interactions of life, October 2024. URL http://biorxiv.org/lookup/doi/10.1101/2024.10.10.615955. Preprint; no peer-reviewed journal version as of May 2026.

Chen, T.-W., Wardill, T. J., Sun, Y., Pulver, S. R., Renninger, S. L., Baohan, A., Schreiter, E. R., Kerr, R. A., Orger, M. B., Jayaraman, V., Looger, L. L., Svoboda, K., and Kim, D. S. Ultrasensitive fluorescent proteins for imaging neuronal activity. *Nature*, 499(7458):295–300, July 2013. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature12354. URL https://www.nature.com/articles/nature12354.

Dallago, C., Mou, J., Johnston, K. E. J., Wittmann, B. L., Bhattacharya, N., Goldman, S., Madani, A., and Yang, K. K. FLIP: Benchmark tasks in fitness landscape inference for proteins. In Vanschoren, J. and Yeung, S. (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/hash/2b44928ae11fb9384c4cf38708677c48-Abstract-round2.html. Round 2.

Didi, K., Alamdari, S., Lu, A. X., Wittmann, B., Johnston, K. E., Amini, A. P., Madani, A., Czeneszew, M., Dallago, C., and Yang, K. K. FLIP2: Expanding Protein Fitness Landscape Benchmarks for Real-World Machine Learning Applications, February 2026. URL http://biorxiv.org/lookup/doi/10.64898/2026.02.23.707496. Preprint; not yet peer-reviewed.

Gazizov, A., Lian, A., Goverde, C., Mou, J., Ovchinnikov, S., and Polizzi, N. F. AF2BIND: predicting small-molecule binding sites using the pair representation of AlphaFold2. *Nature Methods*, 23(3):626–635, March 2026. ISSN 1548-7091, 1548-7105. doi: 10.1038/s41592-026-03011-2. URL https://www.nature.com/articles/s41592-026-03011-2.

Groth, P. M., Michael, R., Salomon, J., Tian, P., and Boomsma, W. FLOP: Tasks for fitness landscapes of protein wildtypes. In *Advances in Neural Information Processing Systems*, volume 36, pp. 33521–33540, 2023. URL https://openreview.net/forum?id=argZAtDMMF. Datasets and Benchmarks Track.

Hayes, T., Rao, R., Akin, H., Sofroniew, N. J., Oktay, D., Lin, Z., Verkuil, R., Tran, V. Q., Deaton, J., Wiggert, M., Badkundri, R., Shafkat, I., Gong, J., Derry, A., Molina, R. S., Thomas, N., Khan, Y. A., Mishra, C., Kim, C., Bartie, L. J., Nemeth, M., Hsu, P. D., Sercu, T., Candido, S., and Rives, A. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, February 2025. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.ads0018. URL https://www.science.org/doi/10.1126/science.ads0018.

Hsu, C., Nisonoff, H., Fannjiang, C., and Listgarten, J. Learning protein fitness models from evolutionary and assay-labeled data. *Nature Biotech-*

nology, 40(7):1114–1122, July 2022. ISSN 1087-0156, 1546-1696. doi: 10.1038/s41587-021-01146-5. URL https://www.nature.com/articles/s41587-021-01146-5.

Jiang, K., Yan, Z., Di Bernardo, M., Sgrizzi, S. R., Villiger, L., Kayabolen, A., Kim, B. J., Carscadden, J. K., Hiraizumi, M., Nishimasu, H., Gootenberg, J. S., and Abudayyeh, O. O. Rapid in silico directed evolution by a protein language model with EVOLVEpro. *Science*, 387(6732):eadr6006, January 2025. doi: 10.1126/science.adr6006. URL https://www.science.org/doi/10.1126/science.adr6006.

Kim, G., Lee, S., Levy Karin, E., Kim, H., Moriwaki, Y., Ovchinnikov, S., Steinegger, M., and Mirdita, M. Easy and accurate protein structure prediction using ColabFold. *Nature Protocols*, 20(3):620–642, March 2025. ISSN 1754-2189, 1750-2799. doi: 10.1038/s41596-024-01060-5. URL https://www.nature.com/articles/s41596-024-01060-5.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In Guyon, I., Von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30, pp. 6402–6413. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/hash/9ef2ed4b7fd2c810847ffa5fa85bce38-Abstract.html.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., Dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., and Rives, A. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, March 2023. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.ade2574. URL https://www.science.org/doi/10.1126/science.ade2574.

Nakai, J., Ohkura, M., and Imoto, K. A high signal-to-noise Ca2+ probe composed of a single green fluorescent protein. *Nature Biotechnology*, 19(2):137–141, February 2001. ISSN 1087-0156, 1546-1696. doi: 10.1038/84397. URL https://www.nature.com/articles/nbt0201_137.

Notin, P., Kollasch, A. W., Ritter, D., van Niekerk, L., Paul, S., Spinner, H., Rollins, N., Shaw, A., Orenbuch, R., Weitzman, R., Frazer, J., Dias, M., Franceschi, D., Gal, Y., and Marks, D. S. ProteinGym: Large-scale benchmarks for protein fitness prediction and design. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 64331–64379. Curran Associates, Inc., 2023a. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/cac723e5ff29f65e3fcbb0739ae91bee-Abstract-Datasets_and_Benchmarks.html. Datasets and Benchmarks Track.

Notin, P., Weitzman, R., Marks, D. S., and Gal, Y. ProteinNPT: Improving protein property prediction and design with non-parametric transformers. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 33529–33563. Curran Associates, Inc., 2023b. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/6a4d5d85f7a52f062d23d98d544a5578-Abstract-Conference.html.

Notin, P., Rollins, N., Gal, Y., Sander, C., and Marks, D. Machine learning for functional protein design. *Nature Biotechnology*, 42(2):216–228, February 2024. ISSN 1087-0156, 1546-1696. doi: 10.1038/s41587-024-02127-0. URL https://www.nature.com/articles/s41587-024-02127-0.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library, 2019. URL https://arxiv.org/abs/1912.01703. Version Number: 1.

Shanehsazzadeh, A., Belanger, D., and Dohan, D. Is Transfer Learning Necessary for Protein Landscape Prediction?, 2020. URL https://arxiv.org/abs/2011.03443. Version Number: 1; also presented at the Machine Learning for Structural Biology Workshop, NeurIPS 2020 (non-archival).

Suzek, B. E., Wang, Y., Huang, H., McGarvey, P. B., Wu, C. H., and the UniProt Consortium. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, March 2015. ISSN 1367-4811, 1367-4803. doi: 10.1093/bioinformatics/btu739. URL https://academic.oup.com/bioinformatics/article/31/6/926/214968.

Tian, L., Hires, S. A., Mao, T., Huber, D., Chiappe, M. E., Chalasani, S. H., Petreanu, L., Akerboom, J., McKinney, S. A., Schreiter, E. R., Bargmann, C. I., Jayaraman,

V., Svoboda, K., and Looger, L. L. Imaging neural activity in worms, flies and mice with improved GCaMP calcium indicators. *Nature Methods*, 6(12):875–881, December 2009. ISSN 1548-7091, 1548-7105. doi: 10.1038/nmeth.1398. URL https://www.nature.com/articles/nmeth.1398.

Truong, T. F. and Bepler, T. Understanding protein function with a multimodal retrieval-augmented foundation model, 2025. URL https://arxiv.org/abs/2508.04724. Version Number: 2; PoET-2; preprint, no peer-reviewed version found.

Wait, S. J., Expòsit, M., Lin, S., Rappleye, M., Lee, J. D., Colby, S. A., Torp, L., Asencio, A., Smith, A., Regnier, M., Moussavi-Harami, F., Baker, D., Kim, C. K., and Berndt, A. Machine learning-guided engineering of genetically encoded fluorescent calcium indicators. *Nature Computational Science*, 4(3):224–236, March 2024. ISSN 2662-8457. doi: 10.1038/s43588-024-00611-w. URL https://www.nature.com/articles/s43588-024-00611-w.

Wardill, T. J., Chen, T.-W., Schreiter, E. R., Hasseman, J. P., Tsegaye, G., Fosque, B. F., Behnam, R., Shields, B. C., Ramirez, M., Kimmel, B. E., Kerr, R. A., Jayaraman, V., Looger, L. L., Svoboda, K., and Kim, D. S. A Neuron-Based Screening Platform for Optimizing Genetically-Encoded Calcium Indicators. *PLoS ONE*, 8(10):e77728, October 2013. ISSN 1932-6203. doi: 10.1371/journal.pone.0077728. URL https://dx.plos.org/10.1371/journal.pone.0077728.

Xiong, J., Nisonoff, H., Lukarska, M., Gaur, I., Oltrogge, L. M., Savage, D. F., and Listgarten, J. Guide your favorite protein sequence generative model, 2025. URL https://arxiv.org/abs/2505.04823. Version Number: 3; accepted in press at Nature Biotechnology (no DOI/volume assigned as of May 2026); v4 retitled "ProteinGuide: On-the-fly property guidance for protein sequence generative models".

Yang, K. K., Wu, Z., and Arnold, F. H. Machine-learning-guided directed evolution for protein engineering. *Nature Methods*, 16(8):687–694, August 2019. ISSN 1548-7091, 1548-7105. doi: 10.1038/s41592-019-0496-6. URL https://www.nature.com/articles/s41592-019-0496-6.

Yang, K. K., Zanichelli, N., and Yeh, H. Masked inverse folding with sequence transfer for protein representation learning. *Protein Engineering, Design and Selection*, 36:gzad015, January 2023a. ISSN 1741-0126, 1741-0134. doi: 10.1093/protein/gzad015. URL https://academic.oup.com/peds/article/doi/10.1093/protein/gzad015/7330543.

Yang, K. K., Fusi, N., and Lu, A. X. Convolutions are competitive with transformers for protein sequence pretraining. *Cell Systems*, 15(3):286–294.e2, March 2024. ISSN 24054712. doi: 10.1016/j.cels.2024.01.008. URL https://linkinghub.elsevier.com/retrieve/pii/S2405471224000292.

Yang, Z., Zeng, X., Zhao, Y., and Chen, R. AlphaFold2 and its applications in the fields of biology and medicine. *Signal Transduction and Targeted Therapy*, 8(1):115, March 2023b. ISSN 2059-3635. doi: 10.1038/s41392-023-01381-z.

Zhang, Y., Rózsa, M., Liang, Y., Bushey, D., Wei, Z., Zheng, J., Reep, D., Broussard, G. J., Tsang, A., Tsegaye, G., Narayan, S., Obara, C. J., Lim, J.-X., Patel, R., Zhang, R., Ahrens, M. B., Turner, G. C., Wang, S. S.-H., Korff, W. L., Schreiter, E. R., Svoboda, K., Hasseman, J. P., Kolb, I., and Looger, L. L. Fast and sensitive GCaMP calcium indicators for imaging neural populations. *Nature*, 615(7954):884–891, March 2023. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-023-05828-9. URL https://www.nature.com/articles/s41586-023-05828-9.

A. Dataset

The dataset consists of a library of 1,314 unique variants generated by random and combinatorial mutagenesis around the scaffold jGCaMP8s (Zhang et al., 2023). Each sequence is 422 amino acids long. The mean Hamming distance from the scaffold is $\mu = 1.81$ with a standard deviation of $\sigma = 0.59$.

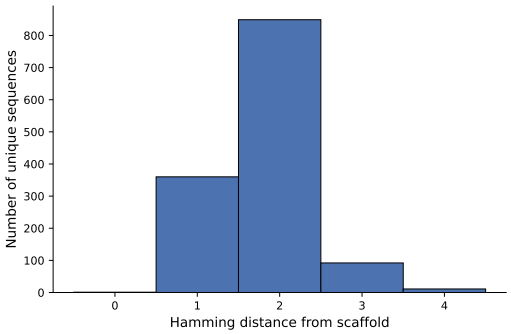

Figure 4. Hamming distance distribution within the dataset

The library is screened in a neuronal culture-based assay with electrical stimulation given in the form of trains of action potentials (APs) (Wardill et al., 2013). The number of action potentials (nAPs) given are 1, 3, 10, and 40 APs. This results in an output fluorescence trace. The metrics d' and $decay$ are then computed from this output trace individually for each nAP. Hence, for each nAP, we get 4 d' values and 4 $decay$ values that form the labels for each variant. The formula for computation of these values is shown in Fig. 5.

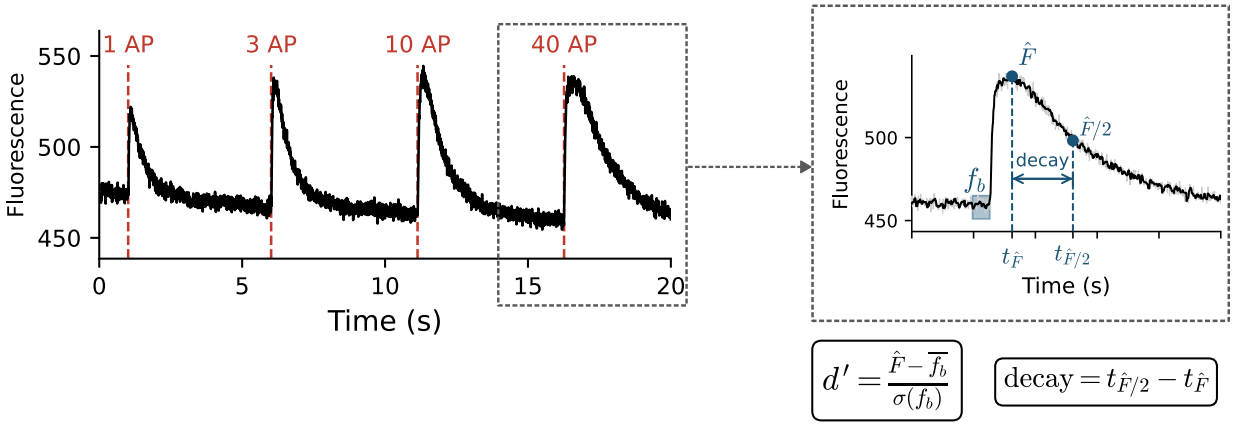

Figure 5. Fluorescence trace and label computation: The metrics are computed for each nAP input individually. $\hat{F}$ is the peak fluorescence response value. $\bar{f}_b$ is a window of 20 frames before the nAP input begins, and $\sigma(\bar{f}_b)$ is the standard deviation in this window. t_f is the time at which Fluorescence is equal to a value f. d' is the ratio of the peak response amplitude to the noise, and decay is the time taken for this response to drop to half of its maximum value.

B. Data Splits

We follow 3 split strategies: a random split for the baseline, a novel-region split to evaluate sequence-space extrapolation, and a low-to-high split to evaluate fitness-space extrapolation. The sizes and number of splits for each strategy are tabulated in Table 1.

Table 1. Dataset split statistics. Mean ± std reported across splits where applicable.

Split Strategy	Train Varinats	Test Variants	Excluded Variants	n_{splits}
Random	1051	66	197	5
Novel	963.0 ± 20.8	70.4 ± 1.2	279.6 ± 20.3	5
Low-to-high	1059	66	189	1

B.1. Novel-Region Splits

To create the novel-region splits, we follow a strategy similar to the "contiguous" split strategy outlined in Notin et al. (2023b). All sequences are of a fixed length of $L = 422$ amino acids. Each variant is represented by its set of mutated positions relative to the wild-type. We first enumerate all possible subsequence windows W across the sequence. For each (s, ℓ), we partition variants by mutation location relative to $W = [s,\ s{+}\ell{-}1]$: no mutation in W (train), mutations only in W (test), or both (excluded). Since mutation density is non-uniform, fixed-width windows yield test sets varying by $>10\times$ across the scaffold. We instead enumerate all (s, ℓ) and keep windows whose test size lies within $5\% \pm 1.25\%$ of the total number of variants. We retain only *maximal* windows (not extensible by one residue in either direction while remaining valid), then select $k{=}5$ via greedy farthest-point sampling on window midpoints to cover the scaffold with diverse unseen regions.

All pairwise differences within the test set then lie inside W, and no position mutated at test is mutated at train. We thus have 5 unique train/test splits where the variants in the test set contain mutations only within a specified subsequence (s, ℓ) and train mutations do not contain any mutations within that subsequence. The subsegment or window corresponding to each test split can be visualized as a contiguous block along the sequence length. These are visualized in Fig. 6.

B.2. Low-to-High Fitness Splits

To probe extrapolation in the fitness regime, we create splits where we train on low-fitness data and test on high-fitness data. In order to do this, we assign ranks to all variants for each individual label. Then, the ranks are averaged across the labels.

We assign the best 5% of ranks to the test set. These are the high fitness variants. Following that, we select the next 15% (in the range of 80^{th} to 95^{th} percentile) and exclude these variants completely from the analysis. This is to ensure that we are testing on strict out-of-distribution sequences and that the correlations are not bloated by the near-distribution variants. The rest of the 80% of the data is assigned to the train set.

The low-fitness, high-fitness and excluded variants can be visualized in 2D in the $decay$ vs d' space separately for each nAP. This is plotted in Fig. 7.

B.3. Random Splits

The baseline is a simple train-test split where the variants in the test set and train set are selected randomly. To keep the train and test set sizes roughly similar to the ones in the novel region and low-to-high splits, we exclude 15% of the data randomly. From the remaining 85%, 5% is chosen for the test set and the rest goes to the train set. 5 such splits are created.

C. Protein Foundation Models

We select 7 model families to use as embedding extractors. These can coarsely be grouped into 3 categories based on their training objective and input modalities. These models are tabulated in Table 2. We select models with similar parameter counts (as available) to ensure a fair comparison with the median size being 650M parameters.

Protein Language Models (PLMs) are models that are trained on amino acid sequences of proteins, typically on a masked language modeling (MLM) or autoregressive next-token prediction objectives. These are typically trained on large databases of sequences such as UniRef (Suzek et al., 2015). The learned protein representations contain information about likelihood of amino acid residues, encoding co-evolutionary signal and conservation patterns. PLMs are the most widely studied class of protein foundation models and serve as a baseline for sequence-only representation quality.

Multimodal Models extend the functionality of PLMs by incorporating additional information either in the form of structure, evolutionary information, or functional annotations. They incorporate information from these additional sources into the sequence-based model either by jointly training with these objectives or using them as conditioning during inference. This allows the model's context to go beyond what is available from sequence alone while also retaining the generality of sequence-style modeling. In realistic protein engineering scenarios when there is often information available through other modalities, multimodal models can be a useful tool to leverage all the available information regarding the protein.

Structure-Prediction Models are trained primarily to predict the 3D structure of proteins from their sequence. In the process of learning to predict structure, the representations within these models incorporate useful information for downstream tasks (Yang et al., 2023b; Gazizov et al., 2026). During inference, these models typically use information from MSAs and oftentimes take structural templates to use as a conditioning input.

We note that this classification reflects common conventions for discussing protein foundation models rather than a strict taxonomy. The categories often overlap (e.g. multimodal models are also trained on PLM objectives), and prior works have used different ways to group these models (Dallago et al., 2021; Notin et al., 2023a). We adopt this specific classification since it allows us to group these by the form of information they encode and additionally, the conditioning input they accept. This grouping is particularly useful for our analysis since we examine the effects of inference-time conditioning on extrapolation behaviour.

Table 2. Pretrained protein models evaluated. Sizes refer to the specific checkpoint used for embedding extraction; D is the per-residue embedding dimension.

Model	Size	D	Type	MSA	Structure
One-hot	—	20	—	—	—
CARP	640M	1280	Sequence-based	No	No
ESM-2	650M	1280	Sequence-based	No	No
ProGen3	762M	1024	Sequence-based	No	No
PoET-2	182M	1024	Multimodal	Yes (additional)	Inverse-folding query
ESM-3	1.4B	1536	Multimodal	No	Conditioning query
AlphaFold-3	200M	384	Structure-prediction	Yes (internal)	Template
Chai-1	1B	384	Structure-prediction	Yes (internal)	Template

C.1. Embedding Extraction

For all models, we extract per-residue hidden states from a single forward pass with frozen weights using the final layer representations. No pooling is applied at extraction time. Special tokens introduced by tokenizers and any padding are stripped so that each cached representation has shape (D, L) matching the variant sequence length $L = 422$. Representations are subsequently z-scored using residue-level statistics computed across the training partition before being passed to the heads. This is done in order to maintain the same initializations across the various PFMs. It is important to note that the z-scoring is done at the entire vector level: the same transformation is applied to every value in the embedding vector. This preserves relative structure across dimensions of the entire embedding.

Configurations that use structural or MSA conditioning share two fixed inputs across all variants: the jGCaMP8.410.80 scaffold crystal structure (PDB 7st4, chain A) for structure, and a consensus MSA for sequence homology. The MSA was computed using ColabFold (Kim et al., 2025) on the consensus sequence which is also the scaffold for the dataset - jGCaMP8s (Zhang et al., 2023). During inference, the scaffold sequence in the first row of the MSA is replaced by the variant sequence. The structural and homology context are otherwise identical across variants. For structural inputs, residues at mutated positions are masked so that the model receives the variant amino acid identity at those positions in the sequence input track but no corresponding structural input.

ESM-2. Layer 33 representations are extracted. BOS and EOS tokens are stripped after extraction.

CARP. The final convolutional block output is used. CARP introduces no special tokens.

ProGen3. Final hidden state. The tokenizer wraps each sequence as `<bos> 1 AA1 ... AAL 2 <eos>`, so per-residue activations are sliced from positions $[2,\ 2{+}L)$.

ESM-3. Post-trunk embeddings exposed via the ESM-3 API using the flag to return embeddings alongside logits. We evaluate two configurations: (i) sequence-only, with only the sequence track populated and (ii) sequence plus scaffold structure, where the scaffold's atom37 coordinates are passed alongside the sequence with NaN-masking at mutated positions and at residues unresolved in the PDB. ESM-3 internally maps NaN positions to its structure mask token.

PoET-2. The final decoder block representations are used. PoET-2's decoder is conditioned on an "encoder prompt" of homolog sequences and/or an inverse-folding query (IFQ) carrying backbone coordinates; the same model produces sequence-only outputs when the prompt is empty. We evaluate four prompt configurations: (i) empty; (ii) homologs sampled from the scaffold MSA via PoET-2's `MSASampler` with default settings (iii) IFQ built from the scaffold PDB; (iv) IFQ and the sampled homologs. For the IFQ, the scaffold's chromophore (CRO, encoded as a single fused residue in the PDB) is expanded back to three NaN-coord residues so the IFQ length matches the unfused variant length, and the IFQ sequence is fully masked so the model receives only geometry.

AlphaFold-3. The post-trunk single-track representations after 10 recycles are used. The conditioning for structure and MSA is supplied externally. AlphaFold-3 performs no MSA or template search at extraction time for compute efficiency. We evaluate four configurations: (i) no conditioning; (ii) scaffold MSA in a3m format with the query row replaced by the variant sequence during inference; (iii) scaffold mmCIF as a single template, with mutated positions removed from the template (iv) both scaffold structure and MSA.

Chai-1. The token-single trunk representations after 3 trunk recycles are used. Chai-1 uses ESM-2 input embeddings as part of its standard stack. We evaluate four configurations analogous to AlphaFold-3: (i) no conditioning, (ii) scaffold MSA with query-row swap, (iii) scaffold mmCIF as template, and (iv) both.

One-hot. This is our most simple baseline: a 20-channel binary indicator over the natural amino acids.

D. Head Architectures

We choose four head architectures for our benchmark. All heads take an input of shape (L, D) where $L = 422$ is the length of the sequence and the dimension D which is dependent on the PFM output. Each head produces an 8-dimensional vector as an output mapping to the 8 label values: $(d'_{1AP}, d'_{3AP}, ..., decay_{10AP}, decay_{40AP})$. All labels are in the natural log-space since the data distribution spans a magnitude. The MLPs in all models use ReLU activation, Xavier uniform initialization and a dropout of 0.3. No normalization is used in any head. MSE loss is used.

D.1. Model Descriptions

MeanPoolMLP. The simplest baseline. The per-residue embedding is averaged across positions into a single D-dim vector, which is passed through a 2-layer MLP with widths $[h_1, h_2]$. This head ignores positional information entirely, assuming the prediction can be made from average embedding properties alone.

AttnPoolMLP. A learned-pooling variant that lets the model decide which positions to attend to. A small position-wise scorer $\mathrm{Linear}(D, h_a) \to \tanh \to \mathrm{Linear}(h_a, 1)$ produces an attention logit per position; a softmax over the sequence axis yields weights that are used to take a weighted sum of the residue embeddings, which is then fed through a 2-layer MLP with the same structure as MeanPoolMLP. This biases the head toward a sparse subset of function-relevant residues rather than weighting all positions equally.

MutationDeltaMLP. A scaffold-relative head that operates only at mutated positions. At each mutation site p_i, the mutant and wild-type embeddings $H_{\text{mut}}[p_i], H_{\text{wt}}[p_i]$ are gathered against a fixed scaffold reference and concatenated with their difference $[H_{\text{mut}}, H_{\text{wt}}, H_{\text{mut}} - H_{\text{wt}}]$ ($3D$-dim per site). These per-site features are projected to h-dim, masked-mean-aggregated across mutation sites, and passed through a 2-layer post-aggregation MLP. This isolates the per-position perturbation introduced by mutations from the scaffold context. Site- and mutation-specific encodings of variants have been used as features in fitness regression in other works (Hsu et al., 2022). This head architecture extends this idea to learned PFM embeddings by computing position-wise differences from the wildtype representation.

ConvMaxPool. A local motif detector model. A $\mathrm{Conv1d}(D{\to}f,\ k{=}5,\ \text{valid})$ with ReLU is applied along the sequence axis, followed by a position-wise $\mathrm{Linear}(f, h)$, a global max-pool over the sequence axis, and a single post-pool $\mathrm{Linear}(h, h)$ with ReLU. Max-pooling selects the strongest motif activation along the sequence in the protein, enabling the head to attend to salient local patterns rather than position-resolved aggregation. This is very similar to the convolutional model used in Shanehsazzadeh et al. (2020) and as a baseline in Dallago et al. (2021).

D.2. Hyperparameters

The hyperparameters for each head architecture and embedding pair were selected via evaluation over three discrete capacity tiers. Since the exact parameter counts vary depending on the embedding dimension D, we report the counts for the median embedding dimension $D = 1536$: low at $\approx 2M$, medium at $\approx 5M$, high at $\approx 14M$. For each head, the searched dimensions were the MLP hidden widths (`hidden_dims` for MeanPoolMLP and AttnPoolMLP, `hidden_dim` for MutationDeltaMLP, and `(num_filters, linear_dim)` for ConvMaxPool). The other architectural hyperparameters such as the attention scorer hidden dimension for AttnPoolMLP at 256, the convolution kernel size for ConvMaxPool at 5, and the post-pooling MLP depth for MutationDeltaMLP at 2 layers were fixed. Selection was performed using the mean validation Spearman correlation (across the 4 folds) from the training results on `split_0` from the random split regime. Training-specific hyperparameters were not varied. The chosen hyperparameters are tabulated in Table 3.

Table 3. Chosen parameter counts for the four head architectures (at $D = 1536$) for the most frequently selected configuration.

Head	Params
MeanPoolMLP	14.7M
AttnPoolMLP	15.1M
MutationDeltaMLP	14.7M
ConvMaxPool	2.4–14.2M

D.3. One-hot Encoding Behavior under Different Heads

The one-hot baseline interacts differently with each head architecture, and the differences are worth understanding because they affect how to interpret the PFM-vs-one-hot performance gap. Under *MeanPoolMLP*, mean-pooling a one-hot embedding produces the amino acid composition of the sequence: each entry of the resulting vector is the frequency of an amino acid in the variant. For variants of the same scaffold differing by only a few mutations, these composition vectors differ very little. A single mutation A→V shifts only the A and V entries, leaving the remaining entries unchanged. The MeanPoolMLP head therefore receives almost identical inputs across variants and cannot meaningfully discriminate between them, producing $\rho \approx 0.05$ on random splits.

Under *AttnPoolMLP*, the attention scorer takes each position's one-hot vector as input and produces a scalar attention weight. The scorer's input depends only on amino acid identity (and not on which position it occurs at). It therefore learns one effective weight per amino acid type — but cannot distinguish two positions that contain the same amino acid. This results in a weighted composition vector from the pooling, where each amino acid's contribution is scaled by a learned weight. This is more flexible than uniform composition under the MeanPoolMLP but still does not encode any positional information. Empirically, AttnPoolMLP achieves $\rho \approx 0.05$ similar to the MeanPoolMLP.

Under *MutationDeltaMLP*, position information is utilized explicitly. The head gathers embeddings at specified mutation sites p_i relative to a fixed scaffold. With one-hot inputs, the per-site features encode information on which mutation occurred at which position, which is informative even without learned representations. This produces a substantially better $\rho \approx 0.35$ on random splits.

Under *ConvMaxPool*, we have a local motif detector of length 5. Max-pooling then selects the strongest activation across the sequence. With one-hot inputs, this means that the head learns short amino acid motifs and identifies the strongest motif anywhere in the sequence. Although the absolute position of the motif is lost during pooling, local positional context within the kernel window is preserved. The model hence achieves a $\rho \approx 0.47$ on random splits with one-hot inputs.

The one-hot baseline performance gap measures different things under different heads. Under pooling heads, it tests whether PFM embeddings encode any information beyond amino acid composition. Under MutationDeltaMLP, it tests whether PFM embeddings encode information beyond raw position-and-identity at mutation sites. Under ConvMaxPool, it tests whether PFM embeddings encode information beyond local sequence motifs. These are all different questions, and the

architecture-conditioned $\Delta\rho$ values in Section 3 reflect this. We find that the gap is the largest under heads where the one-hot reduction loses the most amount of information (simple pooling heads), and it is the smallest where one-hot retains substantial information (ConvMaxPool).

E. Training

All model training was done in Pytorch (Paszke et al., 2019). The optimizer used was Adam with a learning rate of 10^{-4} and a batch size of 1024 and gradient clipping at norm 1.0. Automatic Mixed Precision (AMP) was used for efficient GPU usage. Each training run was set for 2000 epochs with early stopping on validation loss at a patience of 250 epochs and minimum $\Delta = 10^{-4}$. The best checkpoint was selected by validation loss. For each training run, we do 4-fold cross-validation on the train partition (with group K-fold to prevent any variant appearing in multiple folds). During evaluation, the predictions from these 4 fold-models are averaged and the correlations are computed between the ensemble predictions and the ground truth of the variant. Mean squared error (MSE) loss was used throughout. The loss averages over valid labels (instead of a sum of the full 8-length vector) in order to account for missing labels.

F. Evaluation

We use Spearman rank correlation as the standard metric throughout this analysis, as is common practice in protein modeling studies (Notin et al., 2023a; Dallago et al., 2021; Didi et al., 2026), since the relative ranking of variants is more relevant for engineering applications than predicting the absolute fitness scale. For each training run, we get a Spearman rank correlation over all 8 labels - 4 nAPs for d' and 4 nAPs for $decay$. We then average across the four nAP conditions to obtain $\rho_{d'}$ and a ρ_{decay} for each trained model. Each ρ value therefore corresponds to a unique combination of: (split type, head architecture, protein foundation model, split index, label type). Split type is one of random, novel-region, or low-to-high. Split index for random and novel-region $\in \{0, 1, 2, 3, 4\}$, and there is only a single split for low-to-high. Label type is either d' or $decay$.

For statistical comparisons, we use paired Wilcoxon signed-rank tests on the nAP-averaged ρ values that are defined above. Comparisons on random and novel-region splits are paired by (split, label), giving $n = 10$ paired observations per test. For low-to-high splits, which have only a single split, we instead pair by (architecture, label), giving $n = 8$ paired observations per test. When multiple tests are reported within a single comparison family, we apply Holm-Bonferroni correction within that family. Test results are summarized in Table 4.

Table 4. Statistical test results for pairwise comparisons reported in Section 3. Holm-Bonferroni correction is applied within each comparison family, separated by horizontal rules. Bold p_{Holm} indicates values of $p < 0.05$.

Section	Comparison	Split	n	$\Delta\rho$	p_{Holm}
PFM-aggregated vs. one-hot, per architecture					
3	MeanPoolMLP	Random	10	+0.44	**0.008**
§3.1	AttnPoolMLP	Random	10	+0.43	**0.008**
§3.1	MutationDeltaMLP	Random	10	+0.30	**0.008**
§3.1	ConvMaxPool	Random	10	+0.15	**0.008**
§3.2	MeanPoolMLP	Contiguous	10	+0.27	**0.008**
§3.2	AttnPoolMLP	Contiguous	10	+0.19	**0.008**
§3.2	MutationDeltaMLP	Contiguous	10	+0.08	**0.027**
§3.2	ConvMaxPool	Contiguous	10	+0.16	**0.008**
§3.3	PFM-aggregated (all heads)	Low-to-high	8	+0.13	**0.008**
Within-family conditioning (best vs. sequence-only)					
§3.2	AlphaFold3	Contiguous	10	+0.27	**0.008**
§3.2	Chai-1	Contiguous	10	+0.11	**0.039**
§3.2	ESM-3	Contiguous	10	+0.00	0.922
§3.2	PoET-2	Contiguous	10	+0.14	**0.008**
§3.3	AlphaFold3	Low-to-high	8	−0.02	1.000
§3.3	Chai-1	Low-to-high	8	+0.10	0.063
§3.3	ESM-3	Low-to-high	8	−0.11	0.234
§3.3	PoET-2	Low-to-high	8	+0.02	1.000

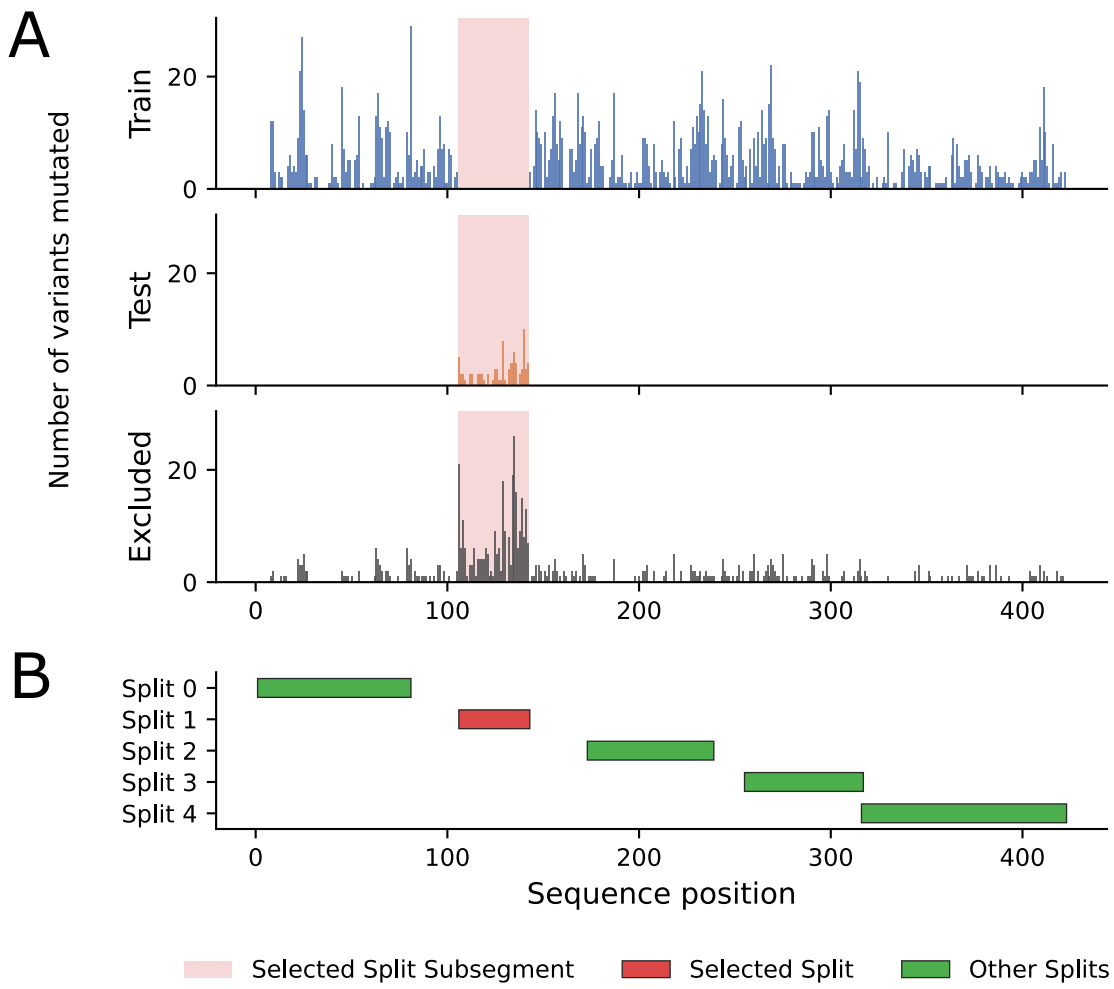

Figure 6. **(A)** Number of variants mutated at each position in the train, test, and excluded sets of a representative novel-region split. The shaded region represents the novel subsegment for the particular split. There are no mutations in the train set within the selected subsegment, and the test set variants contain mutations only within this region. The variants with mutations both inside and outside the subsegment are excluded from both train and test sets. **(B)** Subsegments for novel-region splits across the sequence. The split for the selected subsegment visualized in (A) is highlighted in red.

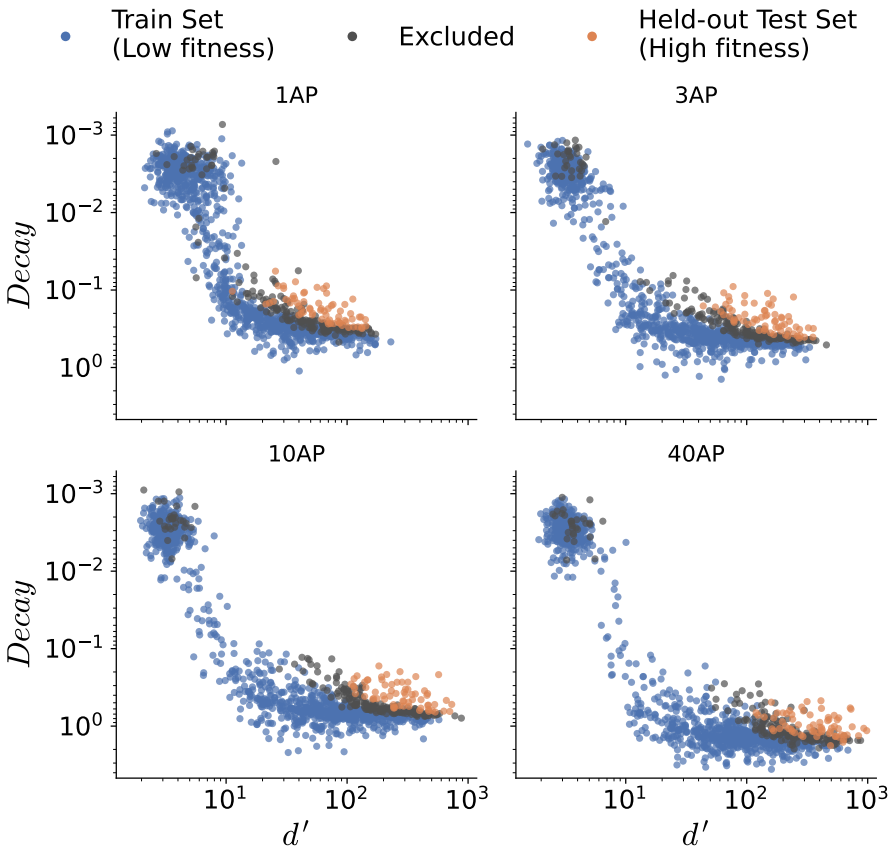

Figure 7. Low-to-high fitnesss splits can be visualized in 2D for each nAP. The excluded variants form a medium fitness buffer band between the low and high fitness variants to ensure a challenging out-of-distribution problem. The $decay$ and d' values are plotted in log-space. Towards the top and to the right is the direction of desired higher fitness. Note that a lower value is better for decay (axis is inverted).