LEVERAGING PROTEIN LANGUAGE MODEL EMBEDDINGS FOR CATALYTIC TURNOVER PREDICTION OF ADENYLATE KINASE ORTHOLOGS IN A LOW-DATA REGIME

Anonymous authors
Paper under double-blind review

ABSTRACT

Accurate prediction of enzymatic activity from amino acid sequences could drastically accelerate enzyme engineering for applications such as bioremediation and therapeutics development. In recent years, Protein Language Model (PLM) embeddings have been increasingly leveraged as the input into sequence-to-function models. Here, we use consistently collected catalytic turnover observations for 175 orthologs of the enzyme Adenylate Kinase (ADK) as a test case to assess the use of PLMs and their embeddings in enzyme kinetic prediction tasks. In this study, we show that nonlinear probing of PLM embeddings outperforms baseline embeddings (one-hot-encoding) and the specialized k_{cat} (catalytic turnover number) prediction models DLKcat and CatPred. We also compared fixed and learnable aggregation of PLM embeddings for k_{cat} prediction and found that transformer-based learnable aggregation of amino-acid PLM embeddings is generally the most performant. Additionally, we found that ESMC 600M embeddings marginally outperform other PLM embeddings for k_{cat} prediction. We explored Low-Rank Adaptation (LoRA) masked language model fine-tuning and direct fine-tuning for sequence-to-k_{cat} mapping, where we found no difference or a drop in performance compared to zero-shot embeddings, respectively. And we investigated the distinct hidden representations in PLM encoders and found that earlier layer embeddings perform comparable to or worse than the final layer. Overall, this study assesses the state of the field for leveraging PLMs for sequence-to-k_{cat} prediction on a set of diverse ADK orthologs.

1 INTRODUCTION

Quantitative prediction of enzymatic activity from amino acid sequence alone holds great promise to advance the field of enzyme design and engineering. Enzymes with improved activity have numerous downstream applications, ranging from bioremediation (i.e., degradation of post-consumer plastic) to therapeutics (i.e., enzyme replacement therapies). Historically, optimization of enzyme activity has been performed through directed evolution Arnold (1996). In recent years this process has been augmented via machine learning – learning a sequence-to-function model – to direct the pseudo-evolutionary processes Yang et al. (2019) Yang et al. (2025) Li et al. (2024) Jiang et al. (2024). Recent prediction methods have used embeddings generated from Protein Language Models (PLMs) as model inputs Jiang et al. (2024). PLMs are thought to reflect the 'fitness landscape' of protein sequences by learning the evolutionary conservation of amino acids contextualized by the rest of the protein sequence. The field has largely turned to PLMs under the hypothesis that these embeddings are meaningful representations for learning sequence-to-function mappings Li et al. (2022a) Boorla & Maranas (2024).

However, to our knowledge, there has been no systematic benchmarking of PLM methods for sequence-to-function mapping of enzymatic activity prediction. To address this gap, we focused on sequence-to-k_{cat} prediction with a unique, self-consistent dataset of 175 k_{cat} values for orthologs of a single enzyme, ADK Muir et al. (2024). While traditional machine-learning-guided directed evolution relies on predicting the effect of low edit distance variants from a starting wild-type se-

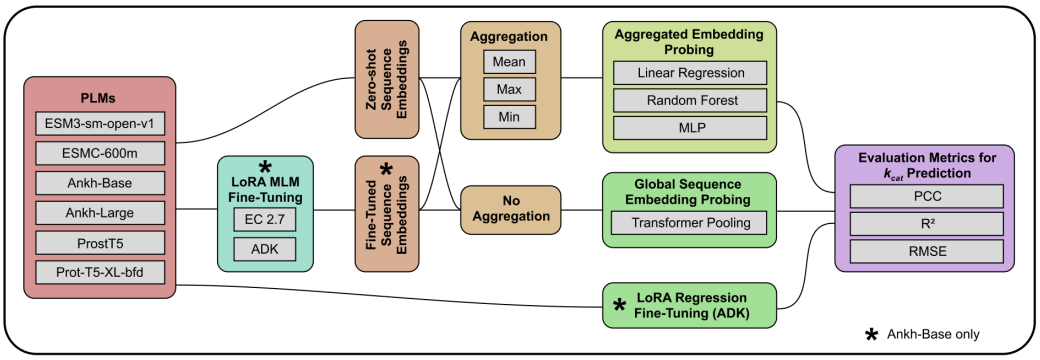

Figure 1: Overview of the experiments and benchmarking performed in this study

quence, this dataset encompasses a broader sequence space of diverse ADK orthologs (42% average pairwise sequence identity) Muir et al. (2024). This enables a new benchmarking task for testing sequence-to-function modeling in a larger, more diverse sequence space. In this study, we leveraged six state-of-the-art pre-trained PLMs and investigated fixed and learnable aggregation methods, LoRA fine-tuning to augment PLM embeddings, and direct LoRA fine-tuning for sequence-to-k_{cat} prediction. In total, we assessed the state of the field for sequence-to-function prediction leveraging PLMs in a low-data regime.

2 Background

2.1 Dataset

In this study we investigate the task of predicting the catalytic turnover number (k_{cat}) of amino-acid sequences for 175 orthologs of the enzyme Adenylate Kinase (ADK) from a recently released dataset collected under consistent experimental conditions Muir et al. (2024). A unique feature of ADK activity is that the opening of the LID domain is rate-limiting for catalysis Wolf-Watz et al. (2004), and different architectures or "lid types" of ADK have significantly different activity Muir et al. (2024). To minimize potential shortcut learning or memorization of distributional differences by lid type, we used a 5-fold lid-aware train:test cross-validation strategy when training and evaluating all regression tasks (**Appendix A**). As in other k_{cat} prediction studies, we log_{10} normalize k_{cat} values for training and evaluation.

2.2 Protein Language Models

We assessed six pre-trained protein language models (PLMs): Ankh-base Elnaggar et al. (2023), Ankh-large Elnaggar et al. (2023), Prot-T5-XL-BFD Elnaggar et al. (2022), ProstT5 Heinzinger et al. (2024), ESMC-600M Team (2024), and ESM3-open Hayes et al. (2025). These protein language models all leverage masked language modeling objectives for pre-training. Some of the models, including ESM3 and ProstT5, explicitly train on protein structure data to relate sequence with structure. We focused on embeddings extracted from the encoders of each model.

2.3 Evaluation Metrics

We evaluated all models for the $log_{10}(k_{cat})$ regression task. We used the Pearson Correlation Coefficient (PCC), root mean square error (RMSE), and the Coefficient of determination (R^2) as performance metrics. We additionally used the RankMe metric Garrido et al. (2023) to quantify the effective rank of the models' embeddings to assess how embedding rank affects downstream performance. We report the metric means plus or minus the standard deviation across the five-fold cross-validation split.

3 EXPERIMENTS

We evaluated the effects of different aggregation methods, probing techniques, and fine-tuning methods on sequence to k_{cat} prediction leveraging PLMs and their embeddings. We additionally compared the optimal PLM-based sequence to k_{cat} probing method against a set of baseline models. Finally, we investigated whether intermediate layer embeddings offer better representations than the last layer embeddings for the prediction of k_{cat}.

3.1 EXPERIMENT 1: FIXED AGGREGATION PROBING OF ZERO-SHOT PLM EMBEDDINGS FOR ADK k_{cat} PREDICTION

Sequence-to-function models require an encoding method that takes in different length sequences and generates a single output. Current strategies leveraging PLM embeddings largely rely on small lightweight models built atop a protein sequence's mean aggregated amino acid embeddings. In this experiment, we extracted zero-shot embeddings from the final layer of all six pre-trained PLMs described in **Section 2.2**. We used mean, max, and min pooling to generate a single embedding per sequence. We then performed linear, random forest, and multi-layer perceptron (MLP) probing for k_{cat} evaluated by the metrics described in **Section 2.3** on the five-fold cross-validation split described in **Section 2.1**.

3.2 EXPERIMENT 2: LEARNABLE AGGREGATION PROBING OF ZERO-SHOT PLM EMBEDDINGS FOR ADK k_{cat} PREDICTION

Here, we sought to assess a learnable aggregation function for predicting k_{cat} directly form the amino acid embeddings. We tested both a lightweight single-head attention layer and a larger Transformer encoder (ViT-Tiny architecture) trained directly on amino-acid PLM embeddings padded to a sequence length of 245 (the longest sequence in the dataset). The models were trained for 1000 epochs with a linear warm-up of 50 epochs, a base learning rate of 1e-6, and a cosine decay to 1e-8. These models were evaluated following the approach outlined in **Section 3.1**.

3.3 EXPERIMENT 3: MASKED LANGUAGE MODEL LORA FINE-TUNING AND PROBING FOR ADK k_{cat} PREDICTION

We next sought to determine if parameter efficient masked language model (MLM) fine-tuning on protein subspaces closer to the ADK orthologs in the dataset is beneficial for probing on the k_{cat} regression task. Here we investigated two levels of protein subspaces.

The first dataset was generated by downloading all enzymes that transfer phosphorus-containing groups (EC 2.7) and filtering for the sequences from the organisms in the ADK ortholog dataset (n=119781). This dataset represents an enzyme sequence subspace with a broadly similar function to ADK.

The second dataset was generated by downloading all known orthologs of ADK from UniProt across all species. This dataset represents all orthologs of ADK (n=48794).

We then performed parameter-efficient MLM fine-tuning of the Ankh-base model with LoRA for 1000 epochs. Ankh-base was chosen for fine-tuning, as it has $\sim$740M parameters and is easily fine-tuned on GPUs containing as little as 24 GB of memory. We used a linear warmup of 10 epochs, a maximum learning rate of 1e-3 followed by a cosine decay to 1e-6. To determine optimal LoRA hyperparameters, we performed a sweep for the LoRA rank and the LoRA alpha parameter. We determined that a rank of 16 and an alpha of 32 was the most parameter-efficient combination as evaluated via the cross-entropy MLM loss. Amino acid embeddings were extracted from the two fine-tuned PLMs (EC 2.7, ADK) and probing for k_{cat} was performed and evaluated using the metrics defined in **Section 2.3**.

We also calculated the RankMe scores to quantify the role of fine-tuning on amino acid embedding rank and its effect on probing performance. A RankMe score was calculated for both fine-tuned models using the amino acid embeddings from all sequences in the ADK ortholog dataset. The RankMe scores were calculated as defined in Garrido et al. (2023).

3.4 Experiment 4: Direct PLM LoRA fine-tuning for ADK k_{cat} prediction

We next evaluated direct parameter efficient fine-tuning of the Ankh-base model for k_{cat} prediction. To perform the direct LoRA fine tuning we concatenated a CLS token to all input amino acid embeddings and padded to a sequence length of 512. The CLS token was used as input into a single linear layer for the regression task with a mean squared error loss function. We use the same LoRA hyperparameters as described in **Section 3.3** with the exception of an increased LoRA dropout (0.5). This model was trained for 1000 epochs on each of the five train splits and evaluated using the metrics defined in **Section 2.3**.

3.5 Experiment 5: Comparison to general k_{cat} prediction baselines

We evaluated pre-trained models for direct k_{cat} prediction for comparison to our PLM probing methods. These models were trained on pairs of substrate SMILES strings and amino acid sequence embeddings to predict enzyme kinetic parameters. Here we evaluated DLKcat Li et al. (2022b) and CatPred Boorla & Maranas (2024) as machine learning model baselines for direct k_{cat} prediction. Both models were used in inference with ADP as the substrate input and the ADK sequences as the sequence input.

3.6 Experiment 6: Probing embeddings from all PLM hidden layers

We next evaluated whether distinct hidden layer embeddings from PLMs have differential performance as input for the k_{cat} regression task. Here we leveraged the Ankh-base PLM and extracted the hidden layer embeddings for all 49 layers. We then trained learnable aggregation models, and mean aggregation random forest models for the k_{cat} prediction.

4 Results

4.1 ESMC 600M embeddings exhibit top performance in fixed aggregation probing

The results of the 5-fold cross-validation probing with a linear regression model, a random forest model, and a 2-layer multi-layer perceptron (MLP) model are shown in **Table 1**. We omit the results of the max pooling from **Table 1** as they never performed better than mean or min pooling across all the models and probing methods. These results suggest different aggregation methods perform better for different PLM embeddings and that mean aggregation does not always generate the best representation for downstream regression tasks. In fact, the best-performing model we observed was on min pooled embeddings from the ESMC 600M model. On average, MLP probing was the most performant across the 5-fold train:test splits, outperforming linear and random forest probing. However, many of these evaluation metric scores are close and likely suggest only marginal gains in performance. In an outlier, ESM3 embeddings yielded poor performance, which we believe arises from overfitting to the train sets due to large values in the embedding vectors. We believe this is the case because we observe larger L2 norms for the ESM3 embeddings compared to other PLMs (**Appendix B**). We additionally performed a replicate training exercise with MLP probing to confirm consistent performance across multiple random seeds (**Appendix C**).

4.2 Learnable aggregation provides marginal improvement over fixed aggregation for k_{cat} prediction

The results of the learnable aggregation probing are shown in **Table 2**. For all models except ESMC 600M, we observed a marginal improvement in the RMSE compared to the fixed aggregation probing.

Table 1: Comparing fixed aggregation and probing methods on PLM embeddings for k_{cat} regression

PLM Model	Aggr.	Probe	RMSE (↓)	PCC (↑)	R^2 (↑)
Ankh-base	mean	Lin Reg	0.907 ± 0.353	0.354 ± 0.269	-1.445 ± 1.924
		MLP	0.504 ± 0.051	0.570 ± 0.149	0.326 ± 0.150
		R.F.	0.517 ± 0.040	0.548 ± 0.150	0.289 ± 0.156
	min	Lin Reg	0.621 ± 0.047	0.420 ± 0.132	-0.035 ± 0.254
		MLP	0.520 ± 0.044	0.541 ± 0.138	0.284 ± 0.143
		R.F.	0.564 ± 0.077	0.443 ± 0.148	0.170 ± 0.136
Ankh-large	mean	Lin Reg	0.682 ± 0.054	0.402 ± 0.176	-0.271 ± 0.413
		MLP	0.511 ± 0.051	0.562 ± 0.106	0.314 ± 0.113
		R.F.	0.524 ± 0.054	0.551 ± 0.088	0.281 ± 0.089
	min	Lin Reg	0.578 ± 0.073	0.412 ± 0.113	0.126 ± 0.131
		MLP	0.549 ± 0.073	0.483 ± 0.054	0.220 ± 0.056
		R.F.	0.586 ± 0.083	0.390 ± 0.089	0.113 ± 0.072
ESM3 Open V1	mean	Lin Reg	0.666 ± 0.059	0.427 ± 0.184	-0.199 ± 0.392
		MLP	0.763 ± 0.105	0.434 ± 0.179	-0.622 ± 0.649
		R.F.	0.507 ± 0.054	0.583 ± 0.095	0.327 ± 0.095
	min	Lin Reg	0.569 ± 0.045	0.464 ± 0.104	0.145 ± 0.153
		MLP	3.528 ± 0.846	0.022 ± 0.137	-32.459 ± 12.702
		R.F.	0.562 ± 0.071	0.469 ± 0.149	0.176 ± 0.115
ESMC 600m	mean	Lin Reg	0.807 ± 0.068	0.334 ± 0.117	-0.740 ± 0.430
		MLP	0.501 ± 0.072	0.586 ± 0.101	0.346 ± 0.115
		R.F.	0.515 ± 0.073	0.568 ± 0.058	0.313 ± 0.071
	min	Lin Reg	0.528 ± 0.063	0.573 ± 0.050	0.271 ± 0.096
		MLP	**0.473 ± 0.069**	**0.647 ± 0.078**	**0.416 ± 0.110**
		R.F.	0.547 ± 0.057	0.505 ± 0.075	0.220 ± 0.060
ProstT5	mean	Lin Reg	0.719 ± 0.058	0.348 ± 0.114	-0.391 ± 0.365
		MLP	0.520 ± 0.046	0.529 ± 0.170	0.280 ± 0.160
		R.F.	0.546 ± 0.062	0.493 ± 0.086	0.222 ± 0.081
	min	Lin Reg	0.656 ± 0.083	0.323 ± 0.170	-0.139 ± 0.286
		MLP	0.561 ± 0.068	0.454 ± 0.076	0.183 ± 0.071
		R.F.	0.567 ± 0.075	0.441 ± 0.059	0.167 ± 0.053
ProtT5-XL-BFD	mean	Lin Reg	0.766 ± 0.113	0.388 ± 0.195	-0.557 ± 0.426
		MLP	0.502 ± 0.046	0.568 ± 0.149	0.329 ± 0.162
		R.F.	0.525 ± 0.055	0.541 ± 0.123	0.275 ± 0.121
	min	Lin Reg	0.572 ± 0.065	0.476 ± 0.188	0.100 ± 0.336
		MLP	0.496 ± 0.047	0.576 ± 0.153	0.343 ± 0.169
		R.F.	0.552 ± 0.062	0.506 ± 0.076	0.209 ± 0.047

Table 2: Comparing Learnable Pooling with Transformer vs. MLP on Aggregated Embeddings

PLM Model	Aggregation	RMSE (↓)	PCC (↑)	R^2 (↑)
Ankh-base	Agg. (mean)	0.504 ± 0.051	0.570 ± 0.149	0.326 ± 0.150
	Learned	**0.493 ± 0.045**	**0.575 ± 0.166**	**0.342 ± 0.203**
Ankh-large	Agg. (mean)	0.511 ± 0.051	0.562 ± 0.106	0.314 ± 0.113
	Learned	**0.483 ± 0.056**	**0.619 ± 0.122**	**0.380 ± 0.152**
ESM3 Open V1	Agg. (mean)	0.763 ± 0.105	0.434 ± 0.179	-0.622 ± 0.649
	Learned	**0.542 ± 0.064**	**0.499 ± 0.070**	**0.235 ± 0.086**
ESMC 600M	Agg. (min)	**0.473 ± 0.069**	**0.647 ± 0.078**	**0.416 ± 0.110**
	Learned	0.489 ± 0.066	0.608 ± 0.106	0.373 ± 0.121
ProstT5	Agg. (mean)	0.520 ± 0.046	0.529 ± 0.170	0.280 ± 0.160
	Learned	**0.497 ± 0.057**	**0.594 ± 0.128**	**0.341 ± 0.161**
ProtT5-XL-BFD	Agg. (min)	0.496 ± 0.047	0.576 ± 0.153	0.343 ± 0.169
	Learned	**0.493 ± 0.053**	**0.580 ± 0.175**	**0.349 ± 0.192**

4.3 MASKED LANGUAGE MODEL LORA FINE-TUNING DOES NOT CONSISTENTLY IMPROVE PREDICTION PERFORMANCE.

To investigate whether masked language model PLM fine-tuning could provide more performant embeddings for k_{cat} prediction, we fine-tuned on two protein subspaces (EC2.7, and ADK homologs). After fine-tuning on protein subspaces closer to the ADK orthologs, we observed a small increase in the RankMe score, particularly on the ADK dataset (**Table 3**). This suggests that MLM fine-tuning on protein subspaces does marginally increase embedding rank. However, the relative increases in the embedding rank were small and suggested that the original embedding space did not suffer from rank collapse. We compared the regression task performance using the learnable aggregation over the zero-shot embeddings vs the fine-tuned model embeddings. Although fine-tuning with an MLM objective increased the effective rank of the amino acid embeddings, it did not meaningfully change downstream k_{cat} prediction performance.

Table 3: Comparing MLM LoRA fine-tuning of Ankh-base model on protein subspaces versus zero-shot embeddings for k_{cat} prediction

Fine-tune Data	RMSE (↓)	PCC (↑)	R^2 (↑)	RankMe (↑)
ADK	0.509 ± 0.038	0.546 ± 0.146	0.308 ± 0.157	**487.09**
EC 2.7	0.500 ± 0.035	**0.582 ± 0.141**	0.332 ± 0.159	483.85
None	**0.493 ± 0.045**	0.575 ± 0.166	**0.342 ± 0.203**	475.79

4.4 DIRECT LORA FINE-TUNING DECREASES PERFORMANCE.

Direct fine-tuning of the Ankh-base led to a substantial performance decrease under five-fold cross-validation testing (**Table 4**). The performance disparity on the train versus test sets suggests the model quickly over-fits the training data. This was slightly remedied by regularization. A larger LoRA dropout increased the downstream regression performance, but it never reached the accuracy of nonlinear probing of zero-shot PLM embeddings.

Table 4: Comparing direct LoRA fine-tuning on the k_{cat} prediction task

Regression Method	RMSE (↓)	PCC (↑)	R^2 (↑)
Ankh-Base Direct LoRA Fine Tune	0.605 ± 0.109	0.489 ± 0.106	0.037 ± 0.243
Ankh-base Zero-Shot Learned Agg.	**0.493 ± 0.045**	**0.575 ± 0.166**	**0.342 ± 0.203**

4.5 TOP-PERFORMING PLM PROBING OUTPERFORM PRIOR SPECIALIZED k_{cat} PREDICTION MODELS

This study's top-performing models outperformed both specialized direct k_{cat} prediction models: DLKcat and CatPred. This suggests that training on large databases with less sequence coverage per protein (BRENDA contains 9 ADK sequences with k_{cat} values) does not enable meaningful prediction of k_{cat} among members of the ADK family (**Table 5**). Additionally, our PLM probing outperforms traditional encoding techniques such as BLOSUM62 and one-hot-encoding (**Appendix D**).

Table 5: Comparing against public k_{cat} prediction models

Regression Method	RMSE (↓)	PCC (↑)	R^2 (↑)
CatPred	1.443 ± 0.022	0.288 ± 0.128	-5.527 ± 3.008
DLKcat	1.873 ± 0.078	-0.093 ± 0.044	-8.408 ± 2.189
ESMC 600M (Min Agg. MLP)	**0.473 ± 0.069**	**0.647 ± 0.078**	**0.416 ± 0.110**
Ankh-large (Learnable Agg.)	0.483 ± 0.056	0.619 ± 0.122	0.380 ± 0.152

4.6 INTERMEDIATE PLM LAYER EMBEDDINGS PROVIDE NO PERFORMANCE BOOST COMPARED TO THE FINAL LAYER EMBEDDINGS FOR ANKH-BASE

We found that PLM amino acid embeddings change their magnitudes as measured by L2 norm throughout the layers, with an abrupt change at the last layer (**Figure 2A**). To assess comparative regression performance with a learnable aggregation probing method, we applied layer norm to each amino acid embedding for layers 31 to 47 to ensure numerical stability during training. Without layer norm, the embeddings are large and cause exploding gradients, which causes model training to fail. We found that probing these layers was at best comparable to probing the final layer embeddings (layer 48, **Figure 2B**).

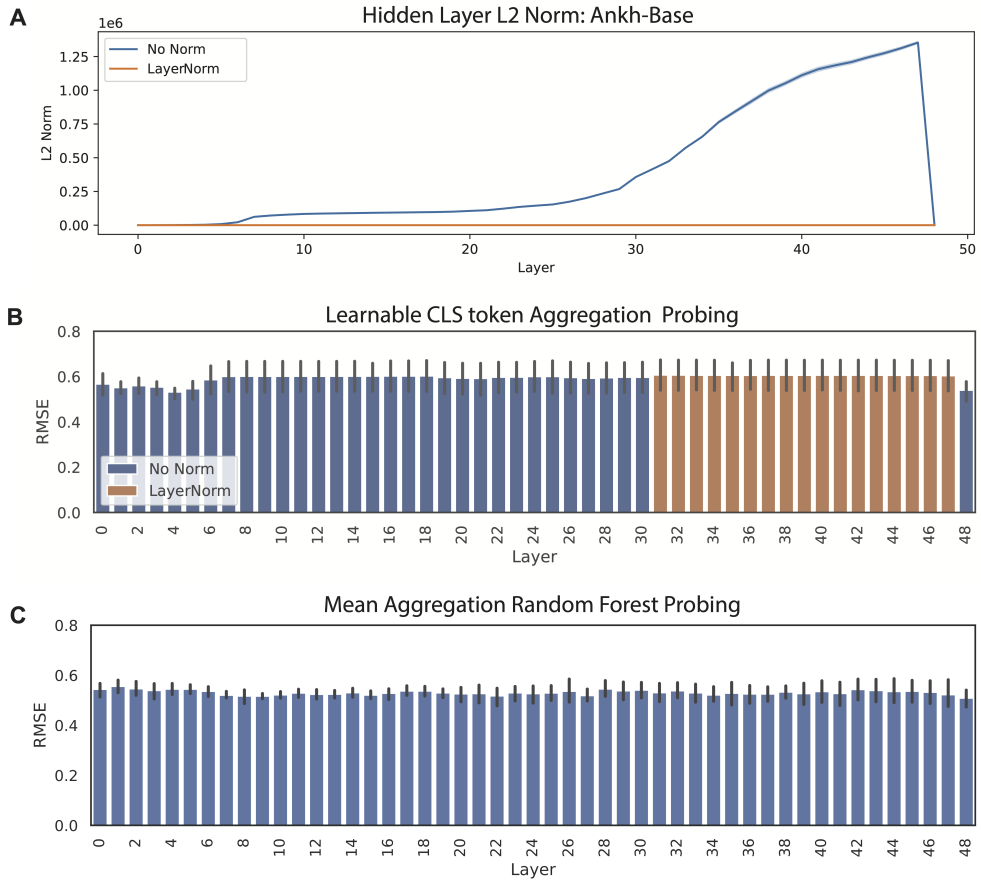

Figure 2: (A) L2 norm per layer before and after LayerNorm. (B) The mean validation RMSE for the k_{cat} regression per layer using the ViT-Tiny CLS learnable aggregation method for the Ankh-base PLM embeddings. The error bar represents the standard deviation from the 5-fold cross-validation split. Note that Layers 31 to 47 are probed after LayerNorm, as the raw embeddings are too large and induce numerical instability during model training. (C) The mean validation RMSE for the k_{cat} regression per layer using random forest probing on mean aggregated Ankh-base PLM embeddings.

Additionally, we found that random forest probing of the mean aggregated embeddings was more stable and performant across the encoder layers (**Figure 2C**). While some layers' embeddings provided equivalent performance for the k_{cat} regression task, they did not provide any benefit beyond the more widely-used last layer embeddings.

5 Discussion and Future Directions

In this study, we sought to assess the state of the field for leveraging PLMs to predict k_{cat} from enzyme sequence on a diverse set of ADK orthologs. We found that all PLM-based probing models outperformed existing larger k_{cat} prediction models trained on public databases curated across multiple studies. Differences in performance among various PLM embedding configurations were modest, but we noted several trends. First, learnable aggregation functions for full-length embeddings only marginally outperform commonly used pooling strategies for constructing fixed-length embeddings. Second, using the Ankh-base model, we showed that additional MLM fine-tuning on related protein sequences (ADKs or EC 2.7) does not improve downstream k_{cat} regression. Furthermore, directly fine-tuning using LoRA appears to be over-parametrized in the low-data regime and hinders model generalization on held-out sequences. Lastly, using embeddings from earlier layers in the Ankh-base encoder at best achieves comparable performance to using the final layer, but can introduce numerical instability.

As the field of protein engineering looks to apply advances in ML and AI to accelerate the optimization of enzymatic activity, it is imperative to understand how to predict these parameters most accurately. Our results demonstrate the utility of high-throughput enzymology datasets in training models to predict catalytic turnover. They also underscore the critical need for additional data generation to improve models and understand how to generalize across proteins. Furthermore, as new protein design methods are capable of generating highly diverse sequences, our datasets and predictive models must explore broad regions of protein sequence space beyond single variant effects. Looking forward, we envision a synergistic scale-up of high-throughput enzymology assays and advancements in model architectures to enable efficient protein function optimization.

Meaningfulness Statement

We believe that a "meaningful representation of life" is a compact encoding of biological data that reflects semantic similarity in properties such as structure, function, interactivity, and organization. These representations should enable the engineering, optimization, and fundamental understanding of biological systems. The work presented here takes a step toward this goal by generating and assessing meaningful representations of protein sequences for a sequence-to-function prediction task. This work has diverse downstream applications including but not limited to the optimization and engineering of enzymes.

References

Frances H. Arnold. Directed evolution: Creating biocatalysts for the future. *Chemical Engineering Science*, 51(23):5091–5102, 1996. ISSN 0009-2509. doi: https://doi.org/10.1016/S0009-2509(96)00288-6. URL https://www.sciencedirect.com/science/article/pii/S0009250996002886.

Veda Sheersh Boorla and Costas D. Maranas. Catpred: A comprehensive framework for deep learning in vitro enzyme kinetic parameters k cat , k m and k i, 3 2024. URL http://biorxiv.org/lookup/doi/10.1101/2024.03.10.584340.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):7112–7127, 2022. doi: 10.1109/TPAMI.2021.3095381.

Ahmed Elnaggar, Hazem Essam, Wafaa Salah-Eldin, Walid Moustafa, Mohamed Elkerdawy, Charlotte Rochereau, and Burkhard Rost. Ankh : Optimized protein language model unlocks general-purpose modelling. *arXiv [cs.LG]*, January 2023.

Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann Lecun. Rankme: Assessing the downstream performance of pretrained self-supervised representations by their rank, 2023. URL https://arxiv.org/abs/2210.02885.

Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, Rohil Badkundri, Irhum Shafkat, Jun Gong, Alexander Derry, Raul S Molina, Neil Thomas, Yousuf A Khan, Chetan Mishra, Carolyn Kim, Liam J Bartie, Matthew Nemeth, Patrick D Hsu, Tom Sercu, Salvatore Candido, and Alexander Rives. Simulating 500 million years of evolution with a language model. *Science*, 0:eads0018, 1 2025. doi: 10.1126/science.ads0018. URL https://doi.org/10.1126/science.ads0018. doi: 10.1126/science.ads0018.

Michael Heinzinger, Konstantin Weissenow, Joaquin Gomez Sanchez, Adrian Henkel, Milot Mirdita, Martin Steinegger, and Burkhard Rost. Bilingual language model for protein sequence and structure. *NAR Genomics and Bioinformatics*, 6:lqae150, 12 2024. ISSN 2631-9268. doi: 10.1093/nargab/lqae150. URL https://doi.org/10.1093/nargab/lqae150.

Kaiyi Jiang, Zhaoqing Yan, Matteo Di Bernardo, Samantha R Sgrizzi, Lukas Villiger, Alisan Kayabolen, B J Kim, Josephine K Carscadden, Masahiro Hiraizumi, Hiroshi Nishimasu, Jonathan S Gootenberg, and Omar O Abudayyeh. Rapid in silico directed evolution by a protein language model with EVOLVEpro. *Science*, pp. eadr6006, November 2024.

Feiran Li, Le Yuan, Hongzhong Lu, Gang Li, Yu Chen, Martin K M Engqvist, Eduard J Kerkhoven, and Jens Nielsen. Deep learning-based kcat prediction enables improved enzyme-constrained model reconstruction. *Nature Catalysis*, 5:662–672, 2022a. ISSN 2520-1158. doi: 10.1038/s41929-022-00798-z. URL https://doi.org/10.1038/s41929-022-00798-z.

Feiran Li, Le Yuan, Hongzhong Lu, Gang Li, Yu Chen, Martin K M Engqvist, Eduard J Kerkhoven, and Jens Nielsen. Deep learning-based kcat prediction enables improved enzyme-constrained model reconstruction. *Nature Catalysis*, pp. 1–11, June 2022b.

Francesca-Zhoufan Li, Jason Yang, Kadina E. Johnston, Emre Gürsoy, Yisong Yue, and Frances H. Arnold. Evaluation of machine learning-assisted directed evolution across diverse combinatorial landscapes, 10 2024. URL http://biorxiv.org/lookup/doi/10.1101/2024.10.24.619774.

Duncan F. Muir, Garrison P. R. Asper, Pascal Notin, Jacob A. Posner, Debora S. Marks, Michael J. Keiser, and Margaux M. Pinney. Evolutionary-scale enzymology enables biochemical constant prediction across a multi-peaked catalytic landscape, 10 2024. URL http://biorxiv.org/lookup/doi/10.1101/2024.10.23.619915.

ESM Team. Esm cambrian: Revealing the mysteries of proteins with unsupervised learning. EvolutionaryScale Website, December 2024. URL https://evolutionaryscale.ai/blog/esm-cambrian.

Magnus Wolf-Watz, Vu Thai, Katherine Henzler-Wildman, Georgia Hadjipavlou, Elan Z Eisenmesser, and Dorothee Kern. Linkage between dynamics and catalysis in a thermophilic-mesophilic enzyme pair. *Nat. Struct. Mol. Biol.*, 11(10):945–949, October 2004.

Jason Yang, Ravi G Lal, James C Bowden, Raul Astudillo, Mikhail A Hameedi, Sukhvinder Kaur, Matthew Hill, Yisong Yue, and Frances H Arnold. Active learning-assisted directed evolution. *Nature Communications*, 16:714, 2025. ISSN 2041-1723. doi: 10.1038/s41467-025-55987-8. URL https://doi.org/10.1038/s41467-025-55987-8.

Kevin K Yang, Zachary Wu, and Frances H Arnold. Machine-learning-guided directed evolution for protein engineering. *Nature Methods*, 16:687–694, 2019. ISSN 1548-7105. doi: 10.1038/s41592-019-0496-6. URL https://doi.org/10.1038/s41592-019-0496-6.

A Cross-Fold Validation: ADK Sequence Splits Stratified by Lid Type

Table 6: Organism Names for Orthologous ADK Splits (Lid type stratified)

Split 1	Split 2	Split 3	Split 4	Split 5
A. salm	A. mari	A. arab	A. sera	F. stag
A. deha	A. pleu	A. flav	A. muco	H. halo
A. caul	A. subb	A. ferr	A. guan	K. avic
B. hens	A. hain	A. bayl	A. medi	L. eryt
B. bact	B. cere	A. ehrl	B. aggr	M. hyor
C. trac	B. lich	A. orem	C. arse	M. mobi
C. urea	B. subt	A. fulg	D. alke	M. pneu
D. dese	B. quin	C. subt	D. acet	O. mess
E. minu	B. coag	C. japo	D. ther	P. bovi
E. tasm	B. petr	C. parv	E. pohl	P. umbo
F. nodo	B. pilo	C. apic	E. cate	P. medi
G. diaz	B. vulp	D. aura	E. coli	P. hydr
K. olea	C. ochr	D. prot	F. acid	P. maca
L. inte	C. pine	D. hafn	F. isla	P. mari
M. silv	C. botu	D. atla	G. aqua	P. debo
M. myco	E. faec	D. ther	G. puni	P. rhiz
P. magn	G. kaus	D. alim	H. acin	P. meth
P. mobi	G. stea	F. rode	H. rode	R. fulv
P. prof	G. ther	F. inte	H. ther	R. floc
P. chlo	G. dalt	F. hyda	L. mult	S. rube
P. atla	H. mode	H. mari	L. cris	S. paci
P. lett	H. somn	H. sali	M. naut	S. muco
P. furi	L. delb	J. mari	M. pauc	S. pleo
S. frig	L. lact	M. arvo	M. floc	S. subs
S. hali	L. pneu	M. caps	M. phyc	S. ther
S. cell	L. reut	M. ther	M. frap	S. alba
S. alas	L. spha	N. ther	M. choc	S. subr
T. sibi	M. burt	P. ingr	M. viri	S. elon
T. afri	M. acet	S. aren	M. argi	T. atla
T. mela	N. gono	S. loih	N. ulva	T. comm
T. mari	P. naph	S. pneu	O. oeni	T. rose
T. neap	P. ento	S. coel	P. hart	T. puti
V. chol	P. syri	S. ther	P. exov	V. semi
V. para	S. ther	S. wolf	R. mass	V. apor
X. camp	T. lovl	T. yell	R. radi	V. dokd

B Average L2 Norm of Final Layer PLM Embeddings

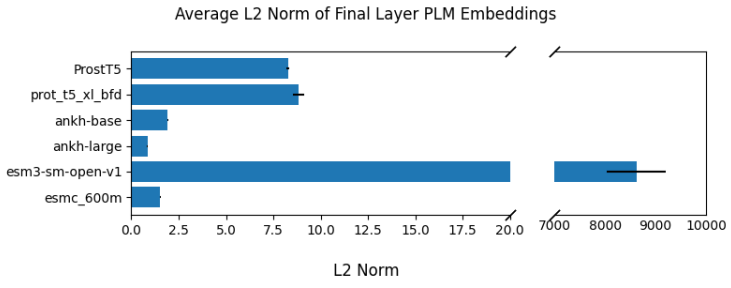

Figure 3: PLM amino acid embedding L2 norms. Error bars represent std. dev.

C Reproducibility of MLP Results Across Initialization Random Seeds

To assess the reproducibility of the MLP probing results, we used two random seeds and plotted the consistency of the RMSE on the regression task. Models trained on ESM3 are not plotted here due to their large error values, likely stemming from large-magnitude embeddings which induced numerical instability during training.

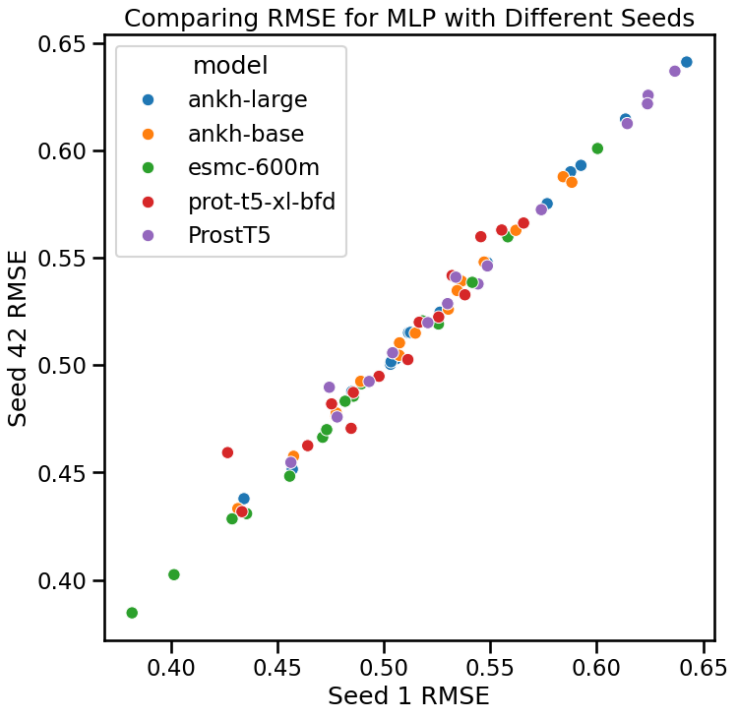

Figure 4: Comparison of random seed initialization for MLP Training with each point representing a validation fold for each of the models probed with an MLP.

D Comparing probing of PLM embeddings to baselines

Although this study focuses on experiments with PLM embeddings, we also evaluated baselines that do not rely on them. The first baseline we assessed was randomly initialized orthogonal embeddings for each amino acid matching the 768 dimensionality of the Ankh-base embeddings. These random embeddings sought to act as a pseudo-one-hot-encoding method. We also generated amino acid embeddings with cosine distances equal to the BLOSUM62 matrix via optimization with the Adam optimizer. Using random forest probing we found that the Ankh-base PLM embeddings performed superior by all metrics for the downstream tasks compared to the baseline embedding methods.

Table 7: Comparing the random forest probing performance for Ankh-base embeddings with randomly distributed embeddings of the same dimension and embeddings generated to have cosine distances equal to the BLOSUM62 matrix

Embeddings	RMSE (↓)	PCC (↑)	R^2 (↑)
Ankh-Base Zero Shot (Mean Agg.)	**0.509 ± 0.043**	**0.574 ± 0.142**	**0.312 ± 0.159**
Pseudo BLOSUM62 (Mean Agg.)	0.539 ± 0.050	0.515 ± 0.133	0.233 ± 0.129
Pseudo one-hot (Mean Agg.)	0.553 ± 0.030	0.464 ± 0.198	0.178 ± 0.205