# Deep learning prediction of enzyme optimum pH

Japheth E. Gado,[1,2,3,4] Matthew Knotts,[4] Ada Y. Shaw,[4] Debora Marks,[4,5] Nicholas P. Gauthier,[4,6] Chris Sander,[4,5] Gregg T. Beckham[1,2,3,*]

[1]Renewable Resources and Enabling Sciences Center, National Renewable Energy Laboratory, Golden, CO, USA
[2]BOTTLE Consortium, Golden, CO, USA
[3]Agile BioFoundry, Emeryville, CA, USA
[4]Department of Systems Biology, Harvard Medical School, Boston, MA, USA.
[5]Broad Institute of Harvard and MIT, Cambridge, MA, USA
[6]Department of Data Sciences, Dana-Farber Cancer Institute, Boston, MA, USA

* Correspondence: gregg.beckham@nrel.gov

## Abstract

The relationship between pH and enzyme catalytic activity, as well as the optimal pH ($pH_{opt}$) at which enzymes function, is crucial for biotechnological applications. Consequently, computational methods that predict $pH_{opt}$ would significantly benefit enzyme discovery and design by facilitating accurate identification of enzymes that function optimally at a specific pH, and by promoting a better understanding of how sequence affects enzyme function in relation to pH. In this study, we present EpHod (Enzyme pH optimum prediction with deep learning), which is a deep semi-supervised language model for predicting enzyme $pH_{opt}$ directly from the protein sequence. By evaluating various machine learning methods with extensive hyperparameter optimization (training over 4,000 models in total), we find that semi-supervised methods that utilize language model embeddings, including EpHod, achieve the lowest error in predicting $pH_{opt}$. From sequence data alone, EpHod learns structural and biophysical features that relate to $pH_{opt}$, including proximity of residues to the catalytic center and the accessibility of solvent molecules. Overall, EpHod presents a promising advancement in $pH_{opt}$ prediction and could potentially speed up the development of improved enzyme technologies.

## Introduction

Enzyme activity is significantly influenced by the pH of the reaction environment, often with a marked decline in activity beyond a specific pH range due to catalytic deactivation and structural instability.[1,2] While most characterized enzymes have an optimal pH for activity ($pH_{opt}$) close to the neutral value of 7.0, some enzymes function optimally at extremely acidic or alkaline conditions, with $pH_{opt}$ values as low as 1.0 or as high as 12.5, respectively.[3–5] In industrial biochemical processes, enzymes are frequently utilized, or desired to be used, far from their $pH_{opt}$, leading to a substantial reduction in activity. As a result, there is a growing interest in discovering and engineering enzymes with enhanced pH tolerance to overcome these limitations.

Exploring extreme pH metagenomes based on sequence homology with characterized enzymes provides a promising avenue to identify natural enzymes with improved pH resistance.[5–7] However, this approach is limited by the possibility of sparse or absent representation of sequences with the desired enzymatic activity in available metagenomes, and instances where homologous hits may exhibit substantially lower activity compared to known enzymes.[8] Alternatively, the pH-activity profile of a selected enzyme can be altered to suit a desired pH range through immobilization,[9,10] chemical modification,[11–13] or protein engineering.[14,15] Protein engineering entails modifying the amino acid sequence of an enzyme to derive a mutant with improved properties, and it has been the most frequently utilized approach for adjusting pH profiles of enzymes to date. However, the prohibitively large size and the extremely nonlinear nature of the combinatorial space of possible mutations, coupled with the preponderance of mutations resulting in loss of function or significantly reduced activity, make protein engineering challenging.[16] To overcome these challenges in engineering enzymes with improved pH profiles, researchers have focused on testing a small set of mutations selected to have increased likelihood of success using rational design,[9,17–22] directed evolution,[23,24] and machine learning.[7,25]

In recent years, advances in deep learning have led to remarkable improvement in predicting protein properties from amino acid sequence.[26–28] By leveraging large datasets, deep neural networks are capable of effectively learning complex, non-linear mappings between protein sequences, enabling accurate predictions of structure and function. As a result, deep learning models hold great promise for high-throughput screening of vast metagenomic datasets, such as MGnify,[29] which currently comprises nearly 3 billion non-redundant sequences, as well as large mutant protein libraries to identify proteins with desirable properties. Researchers have employed computational approaches to investigate protein pH relationships with biophysical methods,[30–32] and to predict enzyme $pH_{opt}$ using traditional machine learning models with limited datasets (fewer than 500 proteins).[7,33–39] Despite these efforts, the adoption of

large-scale deep learning to capitalize on recent breakthroughs in the field, remains limited. This is partly due to the paucity of sufficiently large enzyme $pH_{opt}$ datasets for training purposes.

Deep transfer learning offers a valuable approach for overcoming data scarcity. Deep transfer learning involves training a deep neural network on a related task with a larger labeled dataset and transferring the learned model weights to the task of interest, often followed by fine-tuning on the smaller dataset specific to the target task.[40] However, when large datasets similar to the target task are unavailable for transfer learning, semi-supervised learning can circumvent data limitations and enable improved predictive performance by utilizing unlabeled data. Semi-supervised learning typically entails training a large unsupervised model on unlabeled protein sequence data without supervision, utilizing either the masked or autoregressive objective.[41] Subsequently, the unsupervised model's hidden states offer a rich, semantic representation of unlabeled sequences (sequence embeddings), potentially leading to enhanced performance, even with limited label data, when supervised models are trained on top of these unsupervised embeddings.[42–47]

In this study, we present a deep learning model, EpHod (enzyme pH optimum prediction with deep learning), designed to predict enzyme $pH_{opt}$ directly from protein sequences. Utilizing embeddings from a protein language model (ESM-1v),[48] and pre-training on a separate dataset containing 1.9 million proteins with environmental pH values ($pH_{env}$), our method combines semi-supervised and transfer learning approaches to achieve a root mean squared error (RMSE) of 1.27 pH units on a uniformly distributed held-out testing set. We trained and evaluated multiple models based on various machine learning paradigms, demonstrating the notable efficacy of EpHod's semi-supervised transfer-learning approach in comparison to other machine learning paradigms. Additionally, we examined what EpHod learned in the training process by assessing the per-residue attention weights of the semi-supervised top model. Despite being trained solely on sequence data and lacking access to structural information during training, EpHod effectively learns to focus on residues near the catalytic center, which likely affect binding and catalytic activity, as well as residues near the protein surface, which interact with solvent molecules and ions and are likely necessary for maintaining structural stability in specific pH environments. We further demonstrate that compared to biophysical methods, including an approach that computes the optimal pH for structural stability, EpHod achieves more accurate prediction of catalytic $pH_{opt}$, and maintains robust performance on enzyme sequences that differ significantly from those encountered during training.

## Results

### Optimal pH datasets for model training

We compiled a dataset of 9,855 enzymes from BRENDA,[49] that are annotated with the pH values at which optimal enzyme activity was measured experimentally. This dataset, which we refer to as the $pH_{opt}$ dataset, comprises enzymes from organisms across the tree of life (Figure 1C, Supplementary Table 1) and includes the seven main enzyme activity classes described by enzyme commission (EC) numbers, with hydrolases (EC 3.x.x.x) being the most prevalent class (Figure 1D). We divided the $pH_{opt}$ dataset into training, validation, and testing sets, containing 7,124, 760, and 1,971 sequences respectively. Sequences in the validation set had less than 20% identity with training sequences, while the testing set had a similar distribution as the training set to enable further comparative analyses over the entire sequence distribution. Given the limited size of the $pH_{opt}$ dataset, we hypothesized that pre-training models on a larger dataset of environment pH values would improve $pH_{opt}$ prediction by exploiting the observed association between environment pH and exoenzyme optimum pH.[50] Consequently, we prepared an additional dataset of 1.9 million secreted bacterial proteins from BacDive and NCBI databases,[51,52] that are mapped to the environmental pH values for optimal growth of the source organisms. We selected only secreted proteins by predicting the presence of a signal peptide with SignalP 6.0,[44] since microbes maintain intracellular pH close to neutrality through homeostatic mechanisms.[2,53–55] We used this dataset, which we refer to as the $pH_{env}$ dataset, to examine the potential advantages of transfer learning,[26,40] by comparing models pre-trained on $pH_{env}$ values and fine-tuned on $pH_{opt}$ values with models trained only on $pH_{opt}$ values.

Previous studies have demonstrated that training with non-uniform datasets can result in overfitting of the models towards abundant regions leading to large errors on sparse regions of the label values.[56,57] Our analysis revealed that both $pH_{opt}$ and $pH_{env}$ datasets exhibit highly non-uniform label distributions (Figures 1B and 1E). Nearly 75% of the $pH_{opt}$ values lie between pH 6 to 8, and 94% are between pH 5 and 9. Considering that accurate prediction of extreme acidic and alkaline enzymes could be especially desirable for biotechnological applications, we addressed the label distribution bias by employing several techniques during model training to reweight the loss function with values inversely proportional to the label density near each sample, thus directing the models to focus on sparse extreme regions. During hyperparameter optimization for each machine learning method, we selected the optimal reweighting technique from five approaches that derive sample weights by computing the inverse of the sizes of binned target values (bin techniques) or the inverse of a kernel-smoothed distribution (label distribution smoothing, or LDS, techniques) (Figure 1B, Supplementary Table 3).[58]

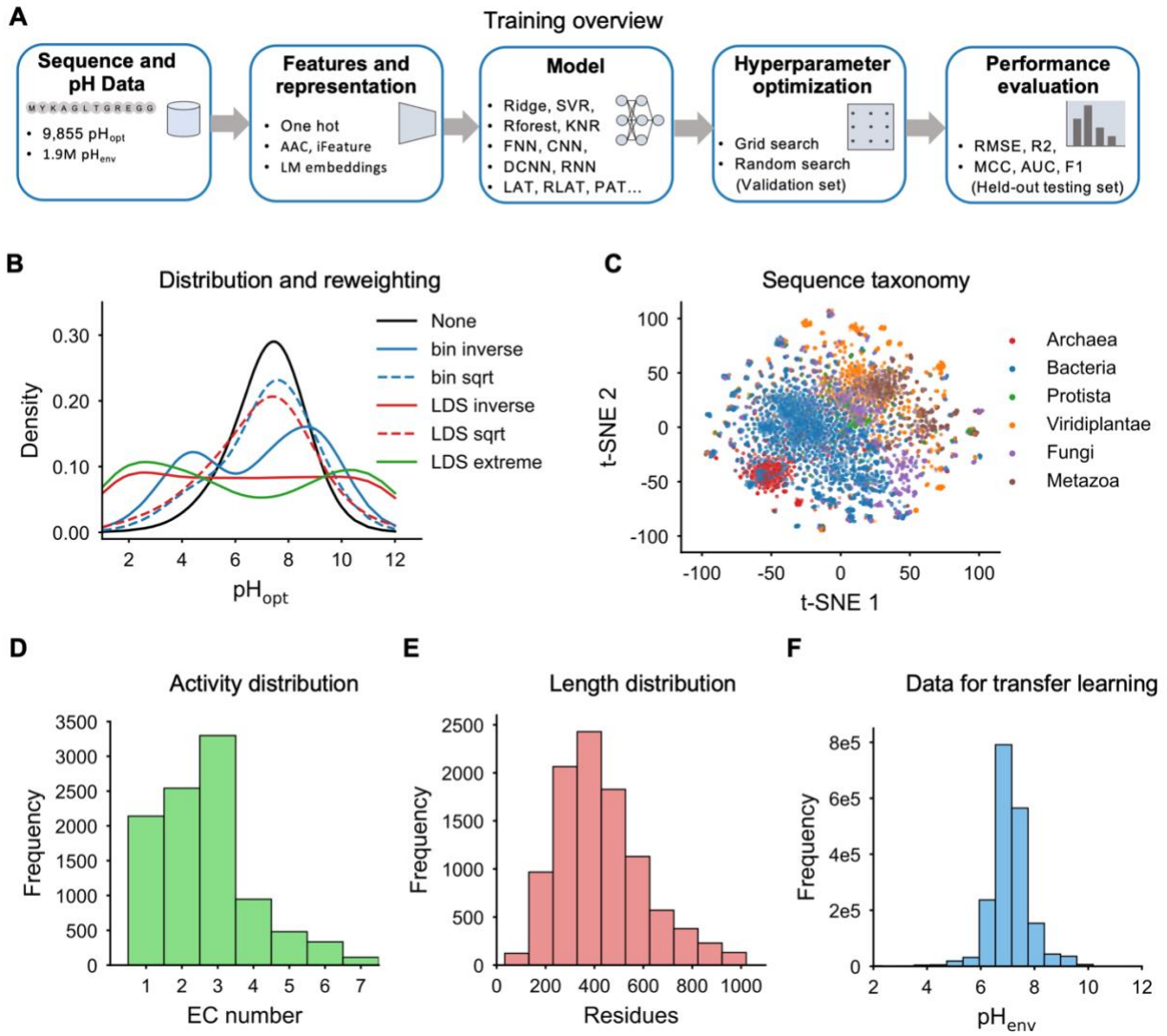


**Figure 1**. Overview of model training and the distribution of the training data. **(A)** Enzyme optimum pH ($pH_{opt}$) and organism environment pH ($pH_{env}$) are used to train various machine learning methods with different sequence representations. For each method, multiple models are trained with varying hyperparameters, and the optimal model, selected by performance on the validation set, is assessed on a held-out testing set. **(B)** Kernel density estimate (KDE) of $pH_{opt}$ for 9,855 enzymes before (None) and after applying reweighting strategies to address the non-uniformity of the label distribution (colored and dashed lines). The reweighting methods are described in the Methods. **(C)** Taxonomical diversity (kingdom) of 9,855 $pH_{opt}$ enzymes visualized by a two-dimensional t-SNE reduction of averaged ESM-1v embeddings.[48] **(D)** Distribution of catalytic activities of 9,855 $pH_{opt}$ enzymes classified by enzyme commission (EC) numbers (1: oxidoreductases, 2: transferases, 3: hydrolases, 4: lyases, 5: isomerases, 6: ligases, 7: translocases). **(E)** Distribution of sequence length of 9,855 $pH_{opt}$ enzymes. **(F)** Distribution of $pH_{env}$ of 1.9 million secreted bacterial proteins used in pre-training for transfer learning.

**Semi-supervised language models substantially outperform alternative machine learning methods**

To predict enzyme $pH_{opt}$ values directly from the protein sequence, we trained multiple models with supervised, semi-supervised, and transfer learning paradigms. Optimal hyperparameters for each method were chosen by evaluating multiple models on the validation set, and the performance of the optimal models was assessed on the testing set (Figure 1A). The testing set was strictly withheld until after completing hyperparameter optimization to avoid overestimating the performance.[59] To ensure clarity, we note that throughout this paper, 'method' refers to a specific algorithm, while 'model' denotes a unique instance of an algorithm characterized by distinct hyperparameters, and 'paradigm' refers to a category of algorithms that share similar characteristics. We extensively evaluated the performance of various machine learning methods for predicting enzyme $pH_{opt}$ across the different paradigms by

training a total of over 4,000 models with varied hyperparameters (Supplementary Tables 3-6). Using one-hot representations, amino acid composition (AAC), and hand-engineered protein features derived with the iFeature package,[60] we trained traditional machine learning regression models including ridge regression, support vector regression (SVR), and random forest regression (RForest). We also trained supervised deep learning models with neural network architectures, including feed-forward neural networks (FNN), convolution neural networks (CNN), dilated convolution neural networks (DCNN), and recurrent neural networks (RNN), using one-hot representations of the proteins. Furthermore, we trained semi-supervised models by fitting traditional regression models (top ML) and deep neural networks (top DL) on top of embeddings derived from unsupervised large language models trained on the protein universe, namely ESM-1b,[61] ESM-1v,[48] ProtT5,[62] Tranception,[63] ProGen2,[64] and CARP (Supplementary Table 2).[41] To mitigate potential biases due to arbitrary hyperparameter selection, we trained up to 200 models for each method across the different machine learning paradigms with hyperparameters sampled using grid or random search,[65] and we selected optimal models based on performance on the validation set. The performance of the optimal models on the held-out testing set is shown in Figure 2. Details of the machine learning approaches are provided in the Methods.

The results showed that the semi-supervised methods utilizing language model embeddings markedly outperformed other paradigms, yielding consistently lower error and higher $R^2$ values on the held-out testing set (Figure 2A, Supplementary Figure 1). Surprisingly, traditional machine learning regression with expert feature engineering and selection, using the iFeature package,[60] outperformed all supervised deep learning methods.[66] This finding is consistent with a previous study that reported no performance advantage of supervised deep learning over expert feature engineering in the prediction of enzyme optimal temperatures.

**A deep semi-supervised language model to predict enzyme optimum pH directly from sequence (EpHod)**

Due to considerable computational requirements, for semi-supervised methods employing language model embeddings, we trained deep neural networks exclusively with ESM-1v embeddings. However, we trained k-neighbor (KNR) and ridge regression models with all language model embeddings (Figure 2A). We selected ESM-1v for training with deep neural networks because it yielded the lowest error and highest correlation on the $pH_{opt}$ validation set when ridge and KNR models with optimal hyperparameters were trained on averaged embeddings (i.e., averaged over residue positions) (Figure 3B-C). We evaluated various deep neural networks to predict $pH_{opt}$ from ESM-1v embeddings (See Methods). The model that achieved the lowest validation error utilized a residual light attention architecture on top of full per-residue ESM-1v embeddings.[67–69] We implemented transfer learning with this model architecture to investigate potential performance enhancements by pre-training the model on the $pH_{env}$ dataset for 533,760 steps and fine-tuning on the $pH_{opt}$ dataset for 1,776 steps (Figures 3D-E). We elected to keep the ESM-1v model frozen in both pre-training and fine-tuning stages to avoid overriding global protein properties learned during the unsupervised training.[70]

We selected this model, trained with semi-supervised and transfer learning approaches, as the final model, since it yielded the lowest error (RMSE) of 1.27 pH units and highest $R^2$ value of 0.68 on the $pH_{opt}$ held-out testing set among all models we trained (Figure 2A). We named the model, EpHod (Enzyme pH optimum prediction with deep learning). Hence, EpHod is a 699-million parameter semi-supervised language model consisting of two main components: the ESM-1v transformer model with 650 million parameters, which was pre-trained with the unsupervised masked language model objective on 98 million Uniref90 proteins,[48,61] and the supervised residual light attention top model with 49 million parameters, trained on top of frozen ESM-1v embeddings (Figure 3A).

**Distribution and size of the data notably affect predictive performance**

Having selected EpHod as the final optimal model amongst all models we trained, we investigated the effects of the distribution of both training and testing datasets on the predictive performance. To demonstrate the impact of pre-training with the larger dataset of environment pH values ($pH_{env}$), we compared EpHod with a model of identical architecture that was not pre-trained on the $pH_{env}$ dataset (EpHod-np). Our results showed that pre-training improved the performance consistently across all evaluation metrics (Figure 3E, Supplementary Table 7, Supplementary Figure 8). As it has been reported that deep learning models may perform worse than average or random predictions on proteins with less than 50% identity to training sequences,[69,71] we evaluated the performance of both EpHod and Ephod-np models in relation to the similarity between testing and training sequences, to assess whether the predictions were merely a consequence of high sequence similarity. As expected, both models showed a decrease in performance as the sequence identity of testing sequences decreased in relation to training sequences. However, even when all sequences remotely similar to any training sequence were removed from the testing set (Blast E-value $<10^4$, denoted as 0% identity), both models maintain robust performance, with RMSE less than 1.5 pH units. Furthermore, EpHod

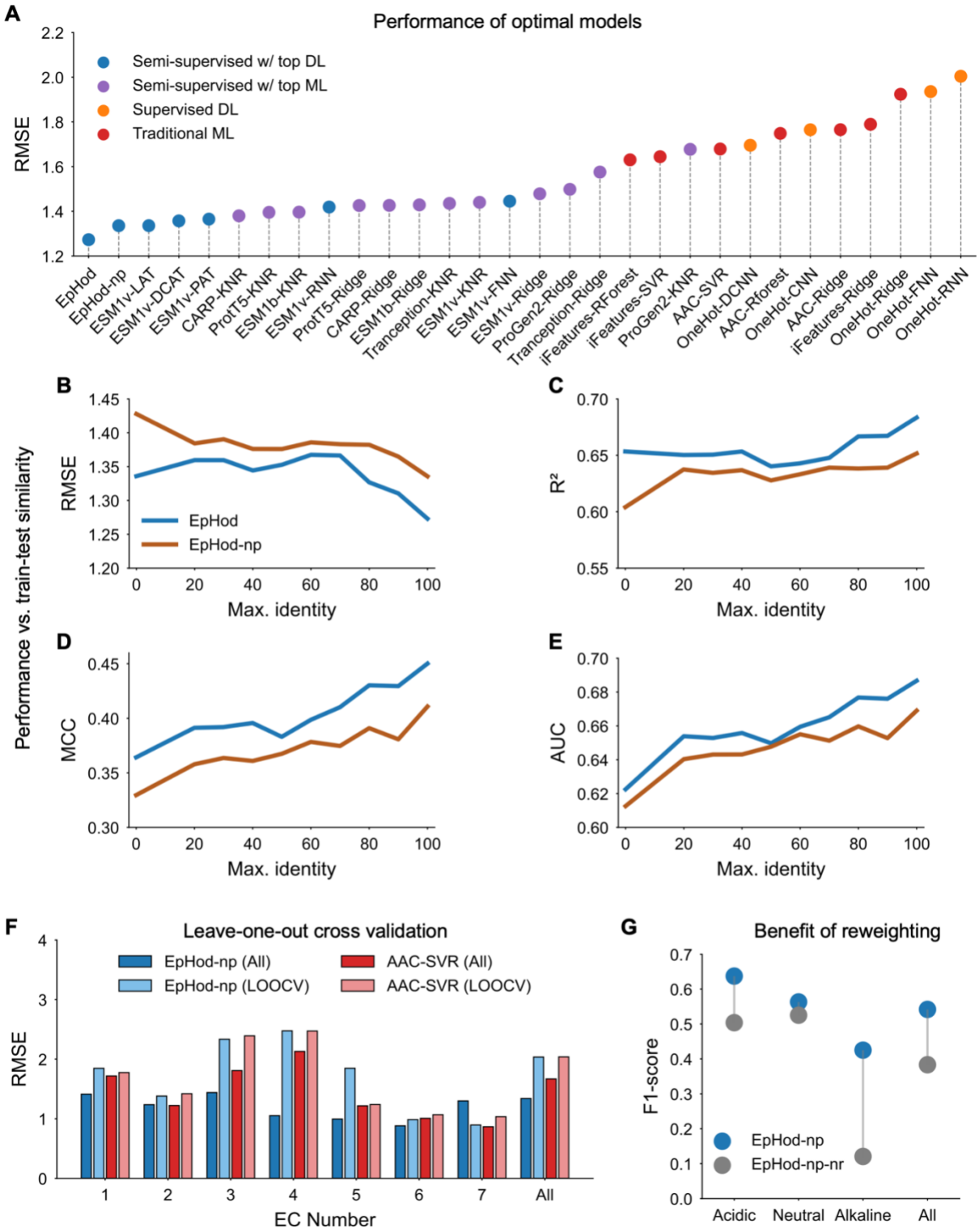


**Figure 2.** Performance of optimal models on the held-out $pH_{opt}$ testing set (n=1,971). **(A)** Root mean squared error (RMSE) of the optimal models for the various machine learning methods. Optimal models were selected from up to 200 models trained for each method with different hyperparameters based on performance on a separate validation set. Semi-supervised models utilizing language model embeddings yield lower RMSE than other methods. $R^2$ values are provided in Supplementary Figure 1. (**B-E)** Performance of EpHod compared to a model of similar architecture but without pre-training on the $pH_{env}$ dataset (EpHod-np). Pre-training leads to substantial improvement on RMSE, $R^2$, area under the receiver operating characteristic curve (AUC), and

Matthew's correlation coefficient (MCC). AUC and MCC were computed to evaluate the classification performance with classes defined as acidic (pH<5), neutral (5<pH<9) and alkaline (pH>9). The *x*-axes represent thresholds of maximum sequence identity between testing and training sequences. **(F)** RMSE evaluated with leave-one-out cross validation (LOOCV) with respect to enzyme activity classes for EpHod-np and the support vector regression model trained on amino acid composition (AAC-SVR). Both models generalize to enzyme activities not included in training, with RMSE below 2.5 pH units. However, EpHod-np performs substantially better when trained on data that includes all activity classes. **(G)** Effect of loss reweighting on performance evaluated by the F1 score of models trained with and without loss reweighting (Ephod-np and Ephod-np-nr, respectively). Loss reweighting leads to a substantial improvement in F1 score, particularly on extreme acidic and alkaline enzymes.

consistently outperformed EpHod-np at all sequence identity thresholds, demonstrating the benefit of pre-training on the larger $pH_{env}$ dataset (Figures 3B-E, Supplementary Figure 7).

In addition, we investigated the impact of the $pH_{opt}$ distribution on model performance and the benefit of reweighting the loss function during training. To achieve this, we compared the performance of EpHod-np, which was trained with sample weights inversely proportional to the size of binned label values (*bin-inverse*) (Figure 1B), with the performance of a similar model trained without reweighting each sequence's contribution to the loss function (Ephod-np-nr). We used EpHod-np, rather than EpHod, for this analysis, since the information EpHod acquires from pre-training with the $pH_{env}$ dataset may overshadow the influence of the $pH_{opt}$ data distribution. As anticipated, we found that training without loss reweighting led to a drastically worse performance in the sparse extreme acidic and alkaline range (pH<5, pH>9) as measured by the F1 score. However, reweighting the loss function improved the F1-score by 26% and 250% on acidic and alkaline enzymes, respectively (Figure 2G). This performance gain corresponds to the scarcity of data, with only 3.9% and 2.3% of the label values falling in the extreme acidic and alkaline regions, respectively. It is worth noting that during hyperparameter optimization, we observed that reweighting techniques based on binning the label values (*bin inverse* and *bin sqrt* methods) were generally optimal for traditional machine learning models, while kernel-based methods (*LDS*) were optimal for deep learning models (Supplementary Table 3). This observation is consistent with the findings of Yang *et al*., who suggested that sample weights derived from kernel-smoothed distributions (LDS) better reflect the effective imbalance experienced by neural networks.[58]

Furthermore, we examined the variability in predictive performance for the seven main EC numbers and the ability of the models to generalize to enzyme activities not encountered during training. To this end, we compared the errors of the EpHod-np model for enzymes from each main EC number with the errors of seven similar models that were trained on a modified training set that excluded enzymes from each selected EC number. The RMSE values ranged from 0.88 to 1.44 pH units when all EC numbers were included in the training set. However, excluding sequences from each EC number led to a moderate increase in the error, up to 2.48 pH units, which was similar to the performance of a traditional machine learning model (Figure 2F). Altogether, these results demonstrate that EpHod, as well as the SVR model, learns features that are relevant to $pH_{opt}$ across different enzyme activities, enabling it to generalize to activity classes that are under-represented or absent in the training data, as well as specific features for each activity class.

**EpHod captures relevant structural features related to enzyme pH optima**

The relationship between specific protein structural features and enzyme pH optima is well-documented in literature.[2] The local distribution of amino acids in the protein structure impacts stability and crucial catalytic interactions under various pH conditions.[22] Although a universal mechanism remains elusive, the adaptation of acidic and alkaline proteins to extreme pH environments is often attributed to an abundance of negatively charged (Asp, Glu) and positively charged amino acids (Arg, Lys) on the protein surface.[2,72,73] Multiple protein engineering studies have demonstrated that altering the composition of charged residues to increase the net negative or positive charge, particularly on the protein surface, can shift the $pH_{opt}$ towards more acidic,[19,74,75] or more alkaline values,[76–78] respectively. Moreover, residues in closer proximity to catalytic sites exert greater influence on the atomic environment and protonation states (pKa) of catalytic residues, thereby dominantly affecting the pH optima.[15,21] Consequently, mutating residues near the catalytic residues to alter the pKa of catalytic residues and maintain their catalytic protonation states at extreme pH levels is recognized as a viable method for shifting the $pH_{opt}$.[14]

Based on these relationships, we next examined whether EpHod acquired these structural features during gradient-based training on the $pH_{env}$ and $pH_{opt}$ datasets. While several interpretability methods are available,[79] in this work, we assessed the attention weights per residue in the RLAT top model (output of the softmax layer) to identify the residues EpHod emphasizes for inference (Figure 3A, details in Methods). We observed that certain residues in the proteins exhibited considerably greater attention weights, suggesting that these residues play a prominent role in $pH_{opt}$ prediction (Figure 4B).

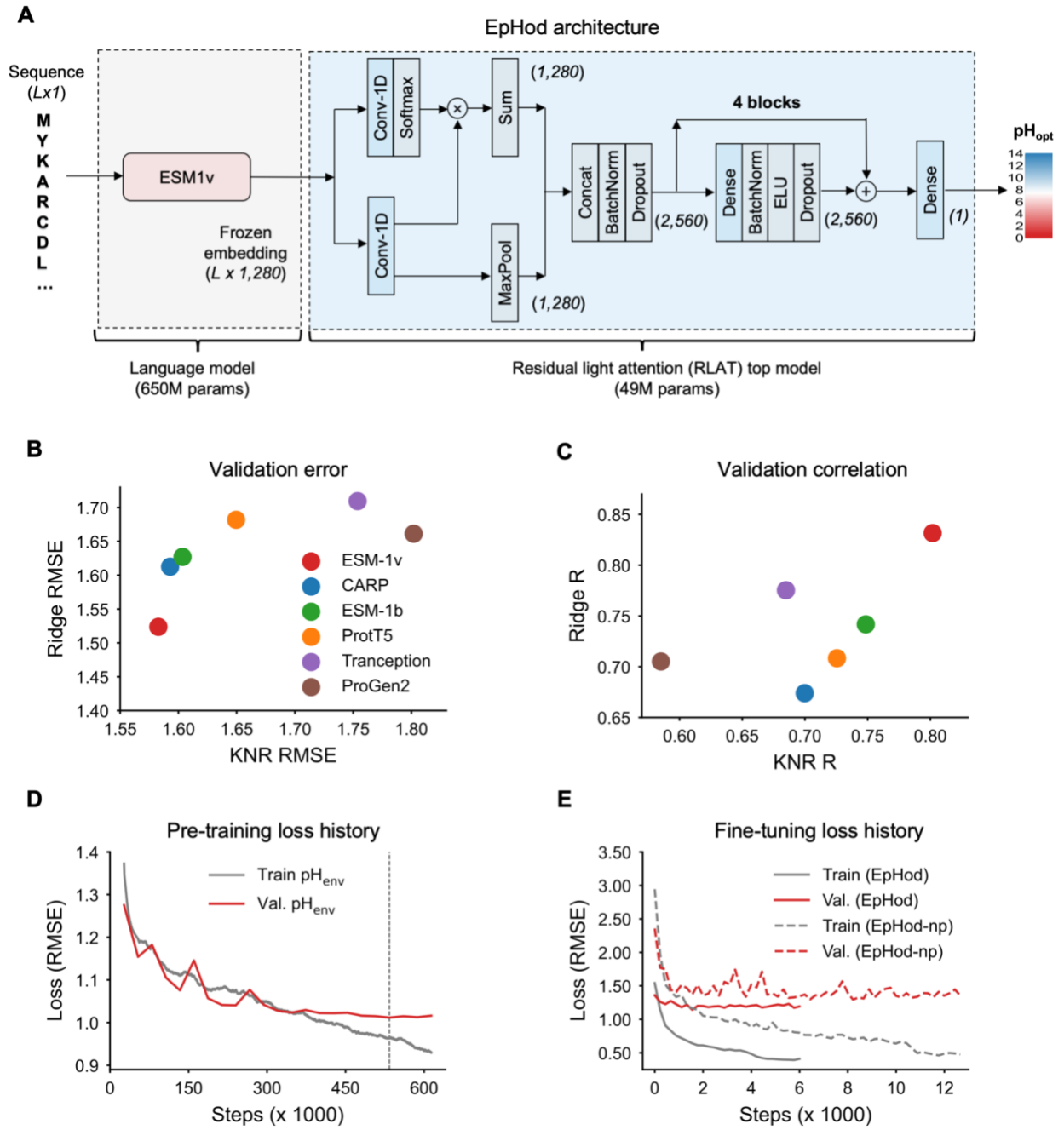


**Figure 3**. Design and training of EpHod. **(A)** Model architecture of EpHod. A residual light attention (RLAT) model with four residual blocks containing dense (fully connected) layers is trained on top of frozen ESM-1v embeddings to predict enzyme $pH_{opt}$. **(B)** Root mean squared error (RMSE) of k-neighbor regression (KNR) and ridge regression models trained on averaged embeddings from six large unsupervised models to predict the $pH_{opt}$ values of the validation set (n=760). **(C)** Pearson's correlation coefficient between predicted and experiment $pH_{opt}$ as in **B**. ESM-1v yielded the lowest error and highest correlation with both KNR and ridge models, and was subsequently selected for training with deep neural networks, including EpHod, as shown in **A**. Description of all six unsupervised models can be found in Supplementary Table 2**. (D)** RMSE loss of EpHod during pre-training on the $pH_{env}$ dataset. The training loss is shown as a moving average over one epoch or 26,688 steps, and validation loss is shown per epoch. Optimal model weights correspond to training after 533,760 steps (dashed line). **(E)** Loss of EpHod during fine-tuning on the $pH_{opt}$ dataset after pre-training on the $pH_{env}$ dataset. Dashed curves show the loss of a similar model trained only on the $pH_{opt}$ dataset without pre-training (EpHod-np).

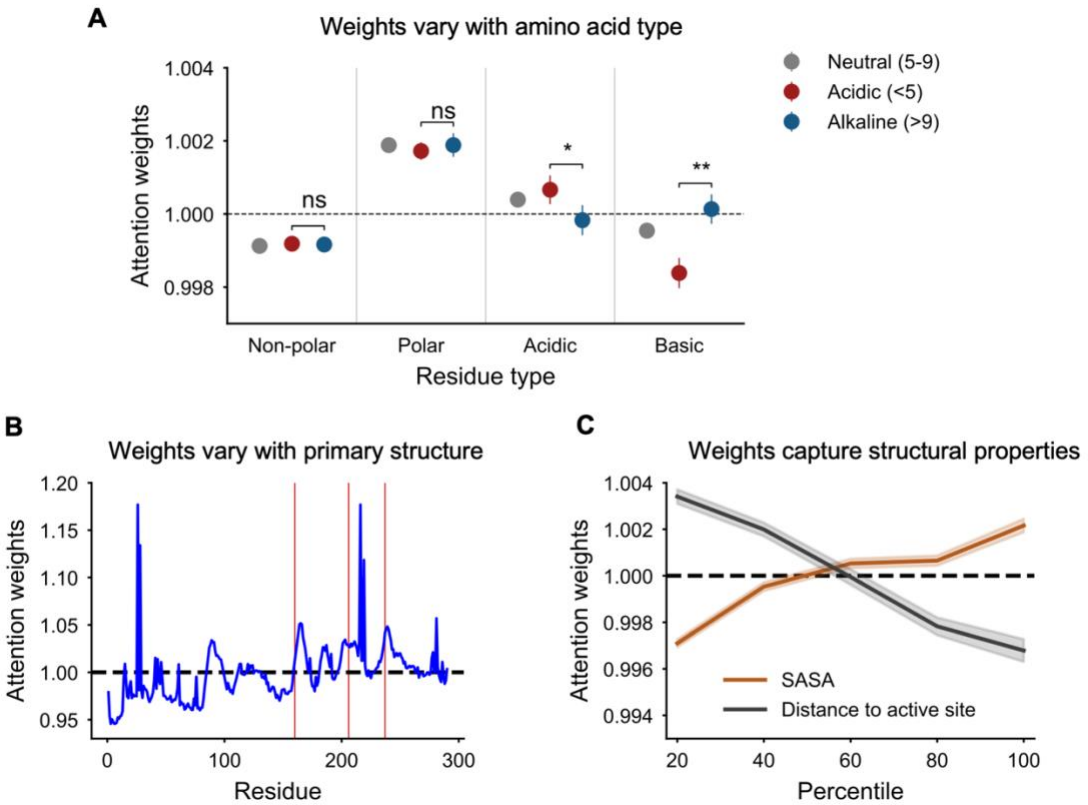


**Figure 4**. Analysis of EpHod light attention weights revealing physicochemical and structural features associated with $pH_{opt}$ using the full dataset (n=9,855). The dashed line (1.0) indicates mean weights per protein after normalization. **(A)** Average attention weights of residues classified as non-polar (Asp, Val, Leu, Ile, Pro, Phe, Trp, Met, Gly), polar (Ser, Thr, Cys, Tyr, Asn, Gln), acidic (Asp, Glu), and basic (Arg, Lys, His). Error bars indicate 95% confidence intervals. A two-sided Welch's t-test comparing average weights in acidic and alkaline enzymes reveals statistically significant difference in acidic and basic residues with *p*-values of 4.1E-3 (*) and 3.4E-9 (**), respectively, but non-significant difference in non-polar and polar residues, with p-values of 0.87 and 0.43, respectively. **(B)** Average attention weights of residues in primary structure of an example enzyme, *Ideonella sakaiensis* PETase (UniProt ID: A0A0K8P6T7).[80] Catalytic residues (Ser160, Asp206, and His237) are indicated by red lines. Certain residues exhibit weights substantially greater than average in the model, contributing majorly to the predicted $pH_{opt}$. **(C)** Averaged attention weights of residues compared with solvent accessible surface area (SASA) and distance from catalytic residues. To enable comparison across different proteins, SASA and distance values were binned into quintiles, and averaged values in each quintile are shown. The colored regions around the lines represent 95% confidence interval.

Intriguingly, we found that on average, polar residues were assigned higher weights than non-polar residues, with the weights being statistically indistinguishable for acidic and alkaline enzymes. Nonetheless, EpHod learned to selectively concentrate on negatively charged (acidic) residues in acidic enzymes and positively charged (basic) residues in alkaline enzymes, assigning significantly higher weights to acidic residues in acidic enzymes than alkaline enzymes, and higher weights to basic residues in alkaline enzymes than acidic enzymes (Figure 4A). Moreover, comparing the difference between acidic and alkaline enzymes, the rank order of individual amino acids by attention weights is considerably different from the rank order of the amino acid composition (AAC), indicating that EpHod learns the relative importance of amino acids beyond the distribution of AAC in the data (Supplementary Figure 4-5). We also evaluated the average weights in relation to solvent accessibility and proximity to catalytic residues. We found that, on average, EpHod allocated progressively higher weights to more exposed residues and to residues nearer to the catalytic center (Figure 4C), in accordance with experimental observations. This trend is discernible when the weights are projected on structures of selected enzymes (Figure 5).

Furthermore, we compared EpHod with other structural and biophysical methods, examining the ability of these methods to accurately differentiate between acidic and alkaline enzymes. We evaluated the area under the receiver operating characteristic curve (AUC) and the correlation between predicted scores and experimentally determined $pH_{opt}$ values. We considered several alternative methods, including the relative amounts of positively charged to

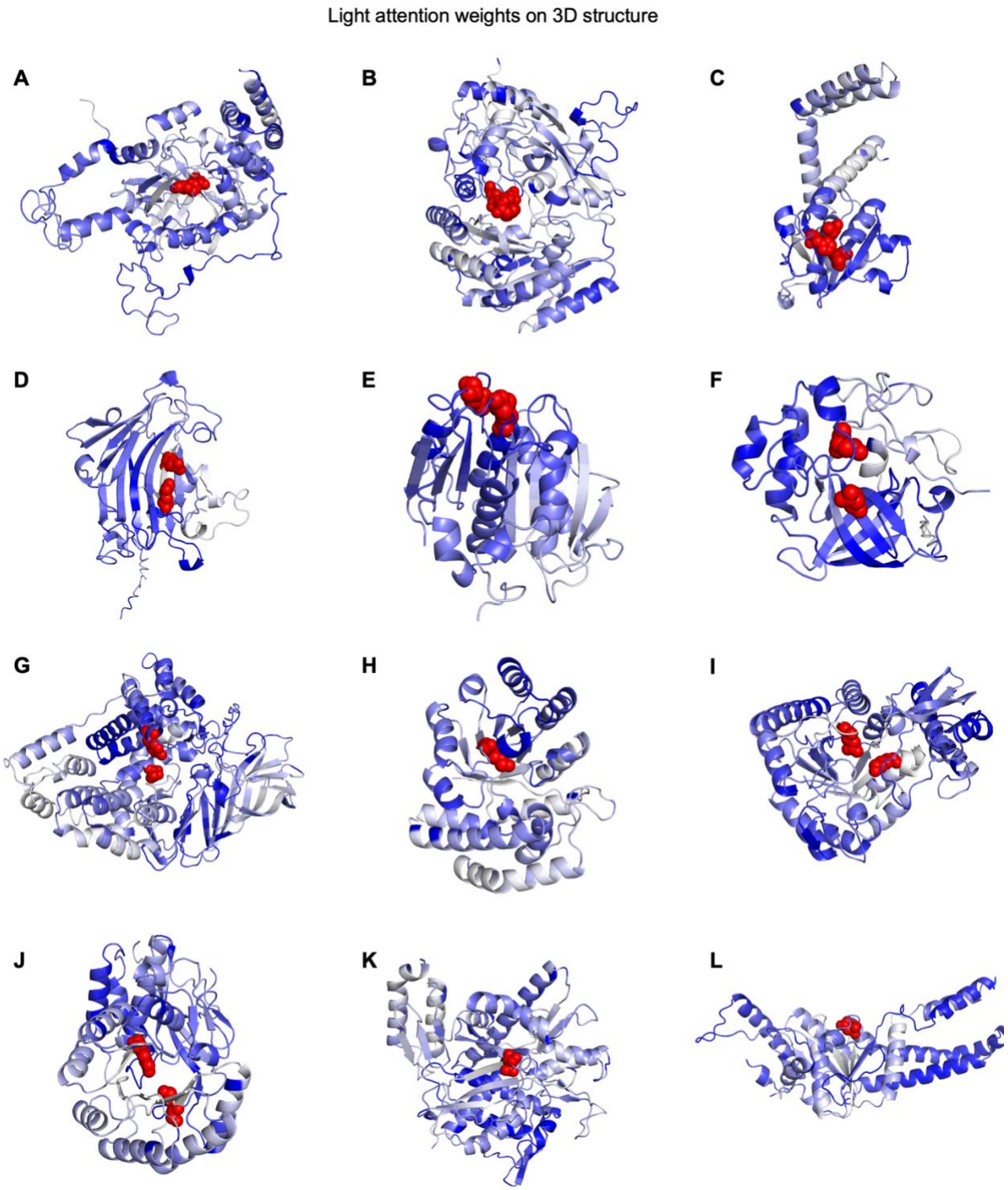


**Figure 5**. Visualization of per-residue light attention weights on selected enzyme protein structures. Darker blue regions signify residues with higher weight values, highlighting areas in the protein that EpHod emphasizes for $pH_{opt}$ predictions. Catalytic residues are shown in red. Weights were clipped to 0.5 and 99.5 percentiles to minimize the effects of outliers on the plot. Data in parenthesis indicate UniProt accession ID (UID), enzyme commission number, and the reported experiment $pH_{opt}$ in BRENDA, respectively. (**A)** *H. discus discus* catalase (UID: A5XB38, EC: 1.11.1.6, $pH_{opt}$: 10.5), **(B)** *P. simplicissimum* vanillyl-alcohol oxidase (UID: P56216, EC: 1.1.3.38, $pH_{opt}$: 7.0), **(C)** *H. sapiens* lecithin retinol acyltransferase (UID: O95237, EC: 2.3.1.135, $pH_{opt}$: 9.1), **(D)** *D. kaki* xyloglucan endotransglucosylase (UID: G5DAC7, EC: 2.4.1.207, $pH_{opt}$: 5.2), **(E)** *I. sakaiensis* poly(ethylene terephthalate) hydrolase (UID: A0A0K8P6T7, EC: 3.1.1.101, $pH_{opt}$: 9.0), **(F)** *C. antarticus* endoglucanase (UID: D3GDK4, EC: 3.2.1.4, $pH_{opt}$:

3.5), **(G)** *P. telluritireducens* aminopeptidase (UID: A0A3S6IVW5, EC 3.4.11.1, $pH_{opt}$: 7.2), (**H**) *P. mume* aldehyde lyase (UID: Q4JC35, EC: 4.1.2.55, $pH_{opt}$: 6.3), **(I)** *B. abortus* phosphopyruvate hydratase (UID: Q2YPV0, EC: 4.2.1.11, $pH_{opt}$: 8.5), **(J)** *B. subtilis* L-Ala-D/L-Glu epimerase (UID: O34508, EC: 5.1.1.20, $pH_{opt}$: 8.5), **(K)** *C. glutamicum* pup-protein ligase (UID: Q8NQE1, EC: 6.3.1.19, $pH_{opt}$: 7.5) **(L)** *C. elegans* ATPase (UID: P30632, EC:6.3.1.19, $pH_{opt}$: 7.4).

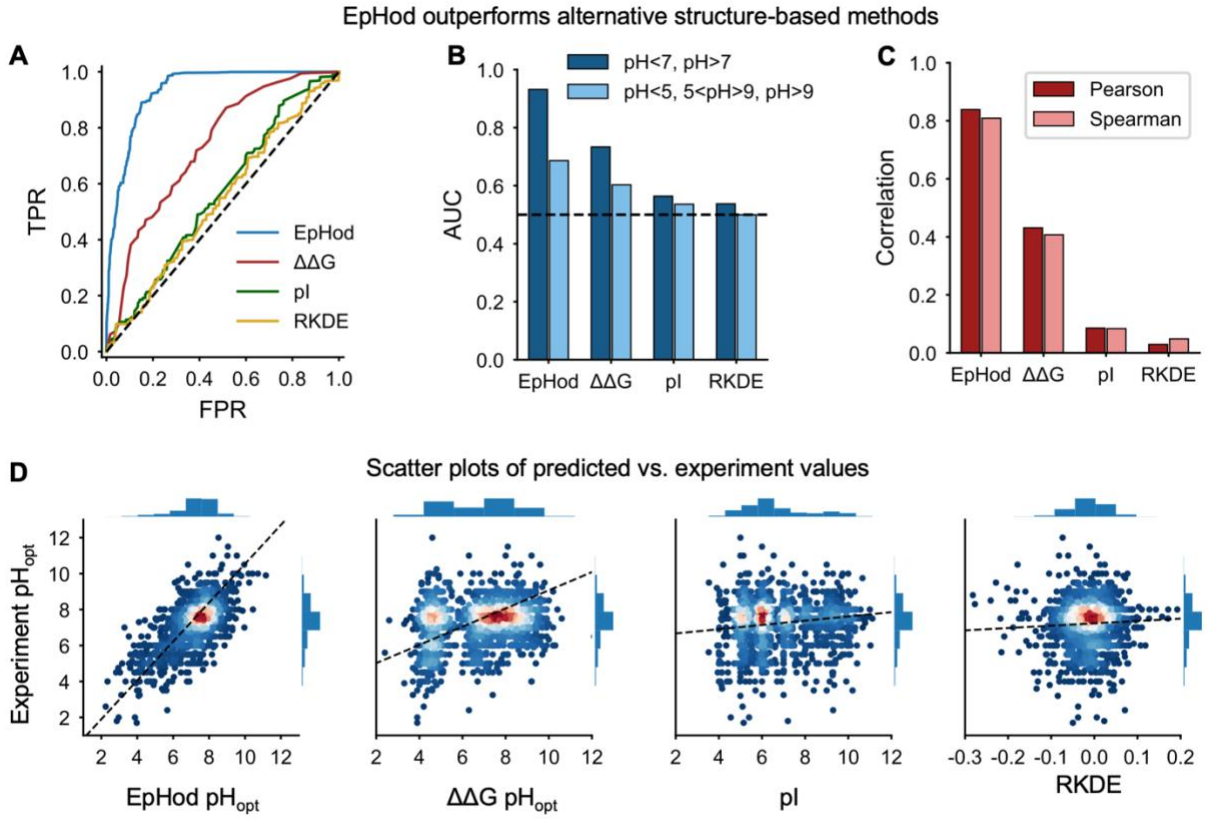


**Figure 6.** Comparison of the predictive performance of EpHod with alternative structural and biophysical methods on the held-out testing set (n=1,971). In predicting $pH_{opt}$ values of the held-out testing dataset, EpHod outperforms calculations of the optimum pH for stability with PROPKA (ΔΔG $pH_{opt}$),[81] the isoelectric point (pI), and the relative amounts of positively charged amino acids (Arg and Lys) to negatively charged amino acids (Asp and Glu) on the protein surface (RKDE). **(A)** Receiver operating characteristic (ROC) curve showing true positive rate (TPR) against false positive rate (FPR) for a binary prediction of $pH_{opt}$ (pH<7, pH>7) at varying thresholds. The dashed line corresponds to random prediction and the area under each curve (AUC) is a measure of the predictive ability. **(B)** Area under the ROC curve for binary $pH_{opt}$ prediction, shown in **A**, and for a ternary prediction computed as the average of one-vs-rest AUC values. **(C)** Spearman and Pearson correlation coefficient between predictive score and experimental $pH_{opt}$. **(D)** Scatter plots comparing predictive scores for each method and experimental $pH_{opt}$. The color gradient of the points, ranging from blue to red, signifies low-to-high density. The dashed line represents a linear regression line curve fitted to a uniformly distributed dataset with respect to experiment $pH_{opt}$.

negatively charged residues on the protein surface (i.e., Arg + Lys – Asp – Glu), the isoelectric point (pI), and an estimation of the pH for optimum stability (ΔΔG $pH_{opt}$) using the PROPKA3 package.[81] Our results demonstrate that EpHod achieved better performance compared with other approaches in the ability to predict catalytic $pH_{opt}$ on the held-out testing dataset (Figure 6). We note that although PROPKA3 computes the pH at which the protein folding free energy is minimum, the optimum pH for activity and stability are generally correlated,[82] and the optimum pH for catalytic activity has greater biochemical significance, since an enzyme needs to be sufficiently stable to carry out its function, but catalysis may be abolished regardless of stability. Our results are also in agreement with previous studies that found little or no correlation between the isoelectric point and optimum pH.[8,82,83]

## Discussion

In this study, we present a deep semi-supervised model, EpHod, for predicting enzyme $pH_{opt}$ values. Our comprehensive assessments demonstrate that EpHod outperforms numerous computational methods in $pH_{opt}$ prediction, further reinforcing the growing consensus that semi-supervised strategies employing protein language model embeddings facilitate state-of-the-art performance across various tasks.[44,67,84,85] Nevertheless, our analyses reveal that the performance of language models for $pH_{opt}$ prediction varies considerably despite extensive hyperparameter optimization (Figure 2A, 3B-C). Moreover, the relative performance of these language models for our $pH_{opt}$ task differs from other tasks.[67,86,87] This observation suggests that it may be beneficial to carefully select the most suitable language model embeddings for a specific task when developing semi-supervised models. Interestingly, we found that CARP,[41] a convolutional neural network (CNN) model comparable in size and training data to ESM-1b, a transformer-based language model, exhibited reasonably similar performance (RMSE) with ESM-1b in our evaluation analyses (Figure 2A and 3B). This finding supports the assertion from Yang *et al*. that, when trained on the same scale, CNNs can be competitive with transformer-based language models.[40]

In addition, our results emphasize the importance of considering the prospective applications and the ways in which trained models will be utilized in practical contexts to ensure optimal performance. Anticipating that the primary use of EpHod would be in identifying enzymes with high acid or alkaline tolerance, we adjusted the training process through a sample-weighted loss function and modified the evaluation using a uniformly distributed testing set. This approach mitigated biases toward prevalent neutral pH values and ensured the development of a model that excels in predicting extreme pH values (Figure 2G, Supplementary Figure 2C). Beyond the $pH_{opt}$ label distribution, we also demonstrate that EpHod maintains robust performance across enzymes of varying residue length, taxonomy, sequence similarity, and catalytic activity, yielding RMSE values of less than 2.5 pH units for all analyzed categories (Figure 1B-E, Figure 2B-F, Supplementary Figure 2). We note that, in contrast, several supervised models in the literature, developed for predicting protein properties, exhibit significant variation in performance over sub-classes of the dataset.[56,71,88,89] However, the findings from our study, corroborated by recent works,[28,44,84] suggest that semi-supervised approaches based on pre-trained protein language models demonstrate superior capability in addressing issues of data scarcity and imbalance. This advantage has been attributed to the utilization of information derived from unsupervised pre-training to enhance performance in data-sparse areas, leading to lower variability in performance.[90]

Furthermore, we anticipate that EpHod can offer multiple applications in enzyme discovery and design. For example, first, by employing EpHod to conduct high-throughput predictions of the $pH_{opt}$ of enzyme homologs from extensive sequence databases,[29,91] one can efficiently identify similar enzymes from natural diversity exhibiting enhanced acid or alkaline tolerance.[8] Second, integrating EpHod with existing protein design techniques would allow for large-scale screening of synthetically designed proteins, as well as iterative selection of variants with improved functionality within a specified pH range through adaptive learning or machine-learning directed evolution.[92–94] Third, EpHod can be incorporated into conditional generative models to directly produce novel functional proteins with the desired pH optima.[95,96] Fourth, concatenating EpHod predictions and other functional predictions with sequence representations can enable the development of augmented fitness models that navigate the fitness landscape at targeted pH levels.[97,98] Lastly, the output of the final hidden layer of EpHod can serve as sequence embeddings that richly capture enzyme pH optima and other protein properties, enabling subsequent semi-supervised training (Supplementary Figure 6). In support of training top models on EpHod embeddings is the observation that while EpHod generalizes to enzyme activities that were unencountered in the training set, integrating additional data for a specific enzyme activity consistently improves performance within that activity class (Figure 2F). This suggests that further semi-supervised training with EpHod's final embedding using supplementary data for a particular enzyme family or activity class will likely yield performance enhancements within the targeted enzyme space.

## Methods

### Data preparation

Enzyme optimum pH data were retrieved from BRENDA for 49,227 entries on May 25, 2022,[49] and protein sequences were downloaded from UniProt for 11,174 of these entries.[91] Multiple $pH_{opt}$ entries for the same sequence were replaced with the average value, if the range was no more than 1.0 pH unit, or else the sequence was discarded. To keep with the length limit of most protein language models, sequences with more than 1,022 residues or fewer than 32 residues were discarded, leaving 9,855 sequences that formed the $pH_{opt}$ dataset. A random selection of 1,971 sequences (20%) was set aside as the separate testing set. The remaining sequences were clustered with mmseqs2 at 20% identity and the clusters were randomly split at 90% and 10% (7,124 and 760 sequences) for training and validation, respectively.[99] To address concerns about $pH_{opt}$ values being derived from a single pH value or a relatively

narrow range of pH measurements in experiments, entries in the testing set that were annotated as "assay at" in BRENDA were removed to form a smaller "clean" testing set of 993 sequences as was done in a previous work.[69] Regardless of the BRENDA annotation, we decided to use the full data for training and validation, since the models can potentially benefit from the relationship between the enzyme sequence and the pH range for activity. In evaluation, we found that EpHod exhibits similar performance on the full dataset and the "clean" dataset, suggesting that the model training was robust to the experiment noise in the dataset due to some $pH_{opt}$ values being derived from a limited a pH range. (Supplementary Figure 3).

Optimum environment pH data for 5,621 bacteria were retrieved from BacDive on June 20, 2022 for the $pH_{env}$ dataset.[51] Multiple organismal entries were averaged if the range was less than 1.0 pH unit, or else they were discarded. Protein sequence data associated with each organism could be accessed for 3,715 organisms from NCBI RefSeq database and were downloaded using BioPython.[52,100] Very short and long sequences were also discarded (<32 or >1022), resulting in 17.0 million sequences. Using SignalP 6.0,[44] 2.2 million proteins were predicted to be secreted proteins possessing a signal peptide, and non-secreted proteins were discarded. All $pH_{opt}$ and $pH_{env}$ sequences were combined and clustered with mmseqs2 at 20% sequence identity.[99] Sequences in the $pH_{env}$ training set that were grouped in the same cluster as any sequence in the $pH_{opt}$ testing set were discarded to prevent data leakage between the two datasets, leaving 1.9 million proteins in the $pH_{env}$ dataset. Hence, we required that there were no sequences in the $pH_{env}$ dataset sharing more than 20% identity with sequences in the $pH_{opt}$ testing set. The remaining $pH_{env}$ clusters were randomly split into training and validation sets (90% and 10%) with 1.71 million and 188,525 sequences respectively. After clustering and splitting the datasets, there were 567 enzymes in the $pH_{opt}$ dataset that could be connected to 97 organisms with $pH_{env}$ values, with a Pearson's correlation coefficient of 0.17 between $pH_{opt}$ and $pH_{env}$ values (Supplementary Figure 11).

**Loss reweighting**

To mitigate the potentially adverse effects of the non-uniformity of $pH_{opt}$ label values,[56,57] the loss function of each model was reweighted with values derived via five methods: *bin inverse*, *LDS inverse*, *bin sqrt*, *LDS sqrt*, and *LDS extreme* (Figure 1B). In the *bin inverse* method, the label values were categorized into three distinct bins, each associated with acidic, neutral, and alkaline pH ranges. These pH ranges are designated as $B_{acidic} = \{y \mid y \leq 5\}$, $B_{neutral} = \{y \mid 5 < y < 9\}$, and $B_{alkaline} = \{y \mid y \geq 9\}$, respectively, where $y$ is the $pH_{opt}$ label. The reciprocal of the size of each bin was used as the weight for all samples within the respective bin.

$$s_i = \frac{1}{n_{B_i}} \qquad (1)$$

In the *LDS inverse* method, the reciprocal of label distribution smoothing (LDS) values were calculated according to Yang *et al*.[58] First, the target values were split into 100 uniformly spaced bins to derive the empirical label distribution. Subsequently, a Gaussian kernel was applied to the binned values to estimate the effective label distribution via kernel density estimation (KDE). Hence, the assigned weight for each sample, $s_i$, was computed as the reciprocal of the estimated density value, and is given by

$$s_i = \left[\frac{1}{nh}\sum_{j=1}^{n}\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(y-y_i)^2}{2\sigma}}\right]^{-1} \qquad (2)$$

where $y_i$ is the target value, $y$ is the center value of the bin, $n$ is the number of bins, and $h$ and $\sigma$ are the bandwidth and standard deviation of the Gaussian kernel that determine the smoothness and shape of the kernel, respectively. We used default values of 2 and 5 for $h$ and $\sigma$, respectively.[58] To attenuate the upweighting of rare acidic and alkaline samples, since inverse reweighting strategies may overcorrect the imbalance and lead to suboptimal performance,[101] we examined sample weights derived by computing the square root of the *bin inverse* and *LDS inverse* weights, termed *bin sqrt* and *LDS sqrt,* respectively. This process yielded reweighting values that were intermediate between the inverse reweighting techniques and no reweighting (Figure 1B). In contrast, we evaluated a reweighting strategy (*LDS extreme*) that amplified the *LDS inverse* weights of scarce acidic and alkaline samples twofold, relative to abundant neutral values. The optimal reweighting techniques for each machine learning method were identified through hyperparameter optimization, employing either random search or grid search,[65] and are shown in Supplementary Table 3.

For training neural networks, the root mean squared error (RMSE), reweighted by mean-normalized sample weights, was utilized as the loss function as follows:

$$\bar{s}_i = \frac{s_i}{\frac{1}{N}\sum_{i=1}^{N} s_i} \quad (3)$$

$$loss = \sqrt{\frac{1}{b}\sum_{i=1}^{b} \bar{s}_i \cdot (y_i - \hat{y}_i)^2} \quad (4)$$

where N is the training data size and b is the size of each batch. For a given learning rate, we found that normalizing the sample weights helped prevent instability arising from high variance in gradient estimates during mini-batch optimization with stochastic gradient descent.[102] In place of loss reweighting for k-neighbor regression (KNR) models, we resampled the training data, drawing from the probability distribution defined by the sample weights.

**Sequence representation**

Protein sequences were numerically represented using five distinct methods: amino acid composition (AAC), one-hot encoding of the raw unaligned sequence (One-Hot), a collection of informative hand-engineered features from the iFeature package (iFeatures),[60] average embeddings from six large unsupervised models (Supplementary Table 2), and per-residue embeddings from ESM-1v (Figure 3B-C). The AAC for each protein was computed by determining the relative abundance of the canonical amino acids, yielding a 20-dimensional vector. This low-dimensional and simplified representation serves as a baseline for assessing higher dimensional representations, given that proteins with different primary structures may have identical AACs. One-hot encoding yielded binary representations of the sequences, with ones indicating the amino acid present at each residue, and zeros indicating all other amino acids. To achieve a uniform length, all sequences were post-padded with zeros up to 1024 residues, generating the representation, $x_{onehot} \in \mathbb{R}^{1024\times 20}$, or the flattened vector, $x_{onehot} \in \mathbb{R}^{1\times 20,480}$. The iFeature package enabled automatic generation of 5,494 engineered features for each sequence (a detailed list of these features is available in Supplementary Table 8). For semi-supervised methods, the hidden states of the final transformer layer (or the final ByteNet convolution layer in CARP) were obtained from ESM-1v, [48] ESM-1b,[61] CARP640,[41] and ProtT5-U50-XL,[62] which are masked language models, and from Tranception-Large and ProGen2-base,[63,64] which are autoregressive models. The sequence embeddings derived from the hidden states, denoted as $x_{emb} \in \mathbb{R}^{L\times d}$, where $L$ represents the sequence length and, $d$ represents the hidden state dimension, were averaged over the sequence length to obtain averaged embeddings, $x_{avg} \in \mathbb{R}^{1\times d}$ (Supplementary Table 2). Embeddings from autoregressive models were obtained by processing the sequences left-to-right (or N-to-C terminus) only. While a comprehensive exploration of other language models for predicting $pH_{opt}$ could be beneficial, it falls outside the scope of this study. In this work, we selected these six language models to represent transformer- and convolution-based models, as well as masked and autoregressive models. Among these models, ESM-1v yielded the best averaged embedding, showing superior results on the validation set (Figure 3B-C). Consequently, sequences were represented as full per-residue ESM-1v embeddings, $x_{emb} \in \mathbb{R}^{L\times 1,280}$, for further assessment.

**Traditional machine learning methods**

Traditional machine learning methods including, ridge regression (Ridge), random forest regression (RForest), and support vector regression (SVR), were evaluated utilizing AAC, iFeature, or one-hot representations. As shown in Supplementary Tables 3 and 4, optimization of key hyperparameters was performed for each method, while default settings were employed for other hyperparameters. Ridge regression, a linear model, is inherently limited at capturing non-linear relationships between the features and target values. This is evident by the inferior performance of all ridge models compared with SVR, random forests, and KNR models, which are better suited for capturing non-linear associations in the data (Figure 2A, Supplementary Figure 1). Furthermore, the worse performance of the linear models demonstrates that enzyme pH optima is governed by complex, non-linear rules beyond the simple additive contributions of specific amino acids, such as charged Asp, Glu, Arg, and Lys (Figure 5, Supplementary Figure 5). Using the 5,494 hand-engineered features from iFeature, feature selection was performed to obtain an optimal subset of the features for predicting $pH_{opt}$ using recursive feature elimination (RFE).[103] In the RFE approach, a random forest with 20-trees was trained with all the features excluding AAC which yielded RMSE of 1.93 on the validation set. The features were ranked based on the random forest Gini importance, and 10% of the features with the lowest Gini scores were discarded. The training process was repeated multiple times, discarding 10% of the features in each iteration and evaluating the RMSE, until one feature was left (RMSE=2.10). The optimal feature combination corresponded to 33 features, yielding an RMSE of 1.78 (Supplementary Figure 9). The final random forest model was trained using 53

features (20 AAC and 33 RFE selection) which yielded RMSE of 1.63 on the held-out testing set, outperforming all supervised deep learning methods (Figure 2A, Supplementary Figure 10).

**Supervised deep learning methods**

Supervised deep models were constructed utilizing one-hot encoding as representations. Optimization of hyperparameters was performed for all methods as shown in Supplementary Tables 3 and 5. Feed-forward neural networks (FNN) contained up to three hidden dense layers with the same number of units in each layer, ranging from 128 to 1,024. Batch normalization was applied to the output of the dense layers, followed by a non-linear activation and dropout. An optional residual (skip) connection was employed, connecting the output of the first and last hidden layers. Convolutional neural networks (CNN) consisted of up to 8 blocks of repeating convolutional, batch normalization, non-linear activation, and pooling layers. The initial convolutional layer filter size was set at 32, and successively doubled for each subsequent layer. Pooling layers utilized a kernel size of 2 and a stride of 2. All input tensors to the convolution layers were 'same' padded. The output from the last convolution block was flattened and supplied to a feed-forward network to generate the final $pH_{opt}$ prediction. Dilated convolution neural networks (DCNN) contained of up to 6 blocks of 10 convolution layers with exponentially increasing dilation rates from 1, 2, 4, up to 512, and a kernel size of 2.[104,105] The input tensors to the convolution layers were causal padded to ensure unidirectional processing from N to C terminus. Each convolution layer was followed by batch normalization and non-linear activation layers and a dropout layer was applied after each block. The output from the last block was passed to a light attention module (*vide infra*) followed by a feed-forward neural network to predict target values. Recurrent neural networks (RNN) employed a unidirectional gated recurrent unit (GRU) processing the sequence tensors from N-to-C terminus.[106] The final hidden state of the GRU, corresponding to the C-terminal residue, was connected to a feed-forward network with one hidden layer to predict target values. It is well known that the problem of vanishing and exploding gradients pose challenges in effectively training RNNs.[107] To address these challenges, the norm of the gradients was clipped to 1.0. However, after hyperparameter optimization, the RNN consistently underperformed all other neural network methods, including FNN, CNN, and DCNN, likely due to the vanishing gradient problem, at least in part. We also trained a long short-term memory (LSTM) model with optimal hyperparameters of the GRU,[108] but this did not lead to performance gains compared with the GRU. While truncated backpropagation through time (TBPTT) may help alleviate vanishing gradients,[43] evaluation of RNNs with TBPTT was beyond the scope of this work. In any case, our results highlight the difficulty of training RNNs on biological sequences as observed previously,[104] and demonstrate that dilated convolutions may be more suitable for sequence modeling of proteins.[41]

**Semi-supervised learning methods**

The semi-supervised approaches involved training a traditional machine learning method or a neural network on averaged or full per-residue embeddings derived from protein language models (Supplementary Tables 2 and 3). Ridge regression and k-neighbor regression models were trained on averaged embeddings representing linear and non-linear methods, respectively. We used KNR in place of random forests or SVR for non-linear top models based on the observation that annotation transfer relying on Euclidean distance in embedding space has demonstrated notable performance in previous works.[67,109] A drawback with averaged embeddings (average pooling) is that valuable information regarding the relative importance of specific residues is lost, since each residue embedding contributes equally to the averaged embedding. Averaged embeddings can be expressed as a weighted sum of the per-residue embeddings with equal weights of $1/L$ as follows,

$$x_{avg} = \sum_{i=1}^{L} w_i . x_i = \sum_{i=1}^{L} \frac{1}{L} . x_i \qquad (5)$$

An alternative attention-based approach is to learn varying weights, $w_i$, that capture the relative importance of each residue, $i$. Following the light attention (LAT) module proposed by Stärk *et al.*,[67] in place of equal weights in the case of average pooling, we fit two separate convolutions with a specified filter size and 'same' padding on full per-residue embeddings, $x_{emb} \in \mathbb{R}^{L \times d}$, to derive learned attention weights and transformed values, $x_{attn} \in \mathbb{R}^{L \times d}$ and $x_{val} \in \mathbb{R}^{L \times d}$, respectively. The attention coefficients were softmax-normalized over the length dimension to obtain attention weights per residue that sum to 1.0. Consequently, the attention-weighted representation, $x_{sum} \in \mathbb{R}^{1 \times d}$, was obtained for each protein by summing the element-wise product of $x_{attn}$ and $x_{val}$ over the sequence dimension as follows,

$$x_{sum} = \sum_{i=1}^{L} x_{attn,i} . x_{val,i} \qquad (6)$$

Comparing (5) and (6), $x_{attn}$ enables the LAT model to learn varying weights for each residue, rather than $1/L$ in averaging, and potentially captures the specific contribution of each residue. Stärk *et al.* also showed that, compared to using $x_{sum}$ alone as the protein representation, concatenating $x_{sum}$ with the representation derived from max-pooling $x_{val}$ over the sequence length to yield $x_{rep} \in \mathbb{R}^{L\times 2d}$, led to performance gains in predicting protein localization.[67] Hence, we similarly concatenated attention-weighted values with max-pooled values in our implementation of the LAT architecture (Figure 3A).

Using the LAT architecture, we trained 200 models with varying hyperparameters for each model to predict $pH_{opt}$ from per-residue ESM-1v embeddings (Supplementary Table 3 and 6). We also proposed variations to the LAT architecture for possible improvements in performance. We introduced perceptive light attention (PAT) in which the average embedding, $x_{avg} \in \mathbb{R}^{1\times d}$, is repeated $L$ times and concatenated with the ESM-1v embedding tensor, $x_{emb} \in \mathbb{R}^{L\times d}$, to obtain a perceptive embedding, $x_{perc} \in \mathbb{R}^{2L\times d}$, from which we derive attention weights and values, rather than from $x_{emb}$ as in the LAT model. The motivation behind this variation was the hypothesis that the light attention's perceptive breadth is constrained by the kernel size of the convolution, which does not consider all residues in each stride. However, we hypothesized that concatenating the averaged embedding to the embedding of each residue would provide information about the full-length protein to the convolution layers, and possibly improve performance. We also proposed dilated convolution with LAT (DCAT) which incorporates a single residual block of a DCNN with 9 convolution layers that is fit on the element-wise product of the attention weights and values, $x_{attn}$ and $x_{val}$, respectively. The output of the dilated convolution block is subsequently flattened and passed to a feed-forward network to predict target values. Another variation we explored was the residual light attention (RLAT), where multiple residual dense layers replace the single hidden layer in LAT. We anticipated that this modification might provide improvements by leveraging deeper layers to better capture the nonlinear mapping between the embeddings and target values. We also examined the performance of RNNs (GRU), processing the embeddings in the N-to-C direction, and feeding the final RNN hidden states to a single hidden layer FNN. After extensive hyperparameter optimization and evaluation, the results showed that, the PAT, DCAT, and RNN methods underperformed LAT, while RLAT slightly outperformed LAT. Based on this, we further trained the RLAT model with optimal hyperparameters using a transfer learning approach, by first pre-training on 1.7 million $pH_{env}$ values and fine-tuning on 7,124 $pH_{opt}$ values, which led to substantial improvement in performance (Supplementary Figure 8).

**Training routine**

Traditional machine learning methods were implemented with the scikit-learn Python package, (version 0.23.2),[110] and neural network models were built with PyTorch.[111] Training of neural networks was carried out on the Eagle high-performance computing cluster (HPC) at the National Renewable Energy Laboratory, USA, with 16 GB NVIDIA Tesla V100 PCIe GPUs. The Adam optimization algorithm was applied with default settings, and the initial learning rate was optimized for each method (Supplementary Tables 5 and 6). To prevent overfitting and ensure efficient training, a combination of dropout, L2 regularization, batch normalization, and early stopping were implemented. A learning rate schedule was applied that lowered the learning rate by a factor of 0.5, if the validation loss did not improve after 10 epochs, and stopped the training if the validation loss did not improve after 20 epochs. A batch size of 32 was employed throughout. In the unique case of pre-training the RLAT model on the $pH_{env}$ dataset (Figure 3D), an initial learning rate of $10^{-3}$ was used, since this was the optimal learning rate in hyperparameter tuning of the RLAT model. This learning rate was reduced if the validation loss did not improve after one epoch, and the training was stopped if validation loss did not improve after five epochs.

**Performance evaluation**

Since the pH data was imbalanced, with most pH values falling within the range of 5 to 9, it was important to address this imbalance in both training and evaluation, to ensure that the trained models accurately predict the $pH_{opt}$ beyond the limited over-represented pH range, and that the estimated performance effectively represents the entire pH range. During training, a sample-weighted loss function was employed to address the imbalance as earlier discussed (Figure 1B). When evaluating the performance of the models, the testing dataset was resampled by drawing from the distribution defined by the inverse sizes of the acidic (pH<5), neutral (5<pH<9), and alkaline bins (pH>9), to achieve a more uniformly distributed testing dataset. All assessed metrics were evaluated on a resampled testing dataset, and averaged over ten resampling repetitions.[56] The regression performance was accessed using RMSE, $R^2$, and Pearson and Spearman correlation coefficients. Given the considerable variability in the scale and precision of $pH_{opt}$ measurements in experiments, the ability of the models to correctly classify the $pH_{opt}$ range (as acidic, neutral, or alkaline) was additionally evaluated by computing the percent accuracy, Matthew's correlation coefficient (MCC), F1 score, and the area under the receiver operator characteristic curve (AUC). To compute the F1 and AUC scores, which

are typically binary metrics, the performance was evaluated for each class compared with the other two classes, and the scores for all three classes were averaged. All performance metrics were computed utilizing the scikit-learn Python package.

**Interpretability analyses and structural evaluation**

To assess what EpHod learned in training, the attention weights generated by the softmax layer, $x_{attn} \in \mathbb{R}^{L\times 1,280}$, were evaluated. Normalized weights per residue for each sequence were derived by initially averaging the weights across the feature dimension to produce $x_{attn1} \in \mathbb{R}^{L\times 1}$, and normalizing by dividing with the mean value. To understand important residues EpHod focused on for prediction, the normalized attention weights for each residue were compared with the amino acid type, the distance from the active site, and the solvent accessible surface area (SASA). Three-dimensional structures for enzymes in the $pH_{opt}$ dataset were retrieved from the AlphaFold protein structure database (AlphaFoldDB).[112] For consistency, AlphaFold structures were utilized for all structural analyses, even when a crystal structure was available. Enzymes lacking a structure in AlphaFoldDB were folded with ColabFold.[113] To compare per-residue attention weights with distance to the active site, 3,074 enzymes in the dataset which had at least one residue annotated as "Active Site" in UniProt were considered. In the AlphaFold structure, the distance to the active site for each residue was computed as the Euclidean distance between the alpha carbon of the residue and the centroid of the alpha carbons of all annotated active-site residues. SASA calculations were performed on the AlphaFold structures using the Shrake and Rupley algorithm with default parameters via the Biopython module.[100,114] To distinguish surface or exposed residues from buried residues, the relative SASA was computed by dividing the SASA by the maximum SASA from the Sander and Rost scale,[115] and a residue was classified as buried if the relative SASA was less than 20%.[116] The relative composition of positively to negatively charged amino acids on the protein surface (RKDE) was computed by evaluating the amino acid composition of residues among surface residues, and computing, $c_{Arg} + c_{Lys} - c_{Asp} - c_{Glu}$, where $c_a$ is the percent composition of amino acid, $a$, on the protein surface. The pH for minimum (optimum) free energy of folding, $\Delta\Delta G$ $pH_{opt}$, and the isoelectric point, pI, were computed using PROPKA3 with default settings on the AlphaFold structures.[81]

**Data availability**

Data for training the models, as well as the weights of the optimal EpHod model, are available at https://doi.org/10.5281/zenodo.8011248.

**Code availability**

Code for EpHod and training the other models is available at https://github.com/beckham-lab/EpHod.

**Acknowledgements**

We thank Ralph Estanboulieh and Rose Orenbuch for their support in visualizations and analyses. We thank Jeffery Law, Aaron Kollasch, Daniel Ritter, Pascal Notin, Jonathan Frazer, and Malfalda Dias for helpful discussions. This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Biological and Environmental Research, Genomic Science Program under Award Number DE-SC0022024 to NPG. This work was authored in part by Alliance for Sustainable Energy, LLC, the manager and operator of the National Renewable Energy Laboratory for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Partial funding to JEG and GTB was provided by the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, Advanced Materials and Manufacturing Technologies Office (AMMTO) and Bioenergy Technologies Office (BETO) as part of the BOTTLE™ Consortium and was supported by AMMTO and BETO under contract no. DE-AC36-08GO28308 with the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC. Partial funding to JEG and GTB was also provided by the U.S. Department of Energy Office of Energy Efficiency and Renewable Energy Bioenergy Technologies Office (BETO) for the Agile BioFoundry. We thank Gayle Bentley at DOE and members of the Agile BioFoundry for helpful discussions. Partial funding to JEG and GTB was also provided by the U.S. Department of Energy Office of Science Biological and Environmental Research via DE-SC0023278. The views expressed herein do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains, and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a non-exclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

**Author contributions**

GTB and JEG conceived of the project, and JEG, DM, NPG, CS, and GTB designed the study. JEG trained the predictive models, and AYS, MK, and JEG evaluated the models. The manuscript was written by JEG and edited and approved by all authors.

**Competing interests**

DM is an advisor for Dyno Therapeutics, Octant, Jura Bio, Tectonic Therapeutics, and Genentech, and a cofounder of Seismic Therapeutics. CS is an advisor for CytoReason Ltd. The remaining authors declare no competing interests.

## References


1. Barroca, M. *et al.* Deciphering the factors defining the pH-dependence of a commercial glycoside hydrolase family 8 enzyme. *Enzyme Microb. Technol.* **96**, 163–169 (2017).
2. Reed, C. J., Lewis, H., Trejo, E., Winston, V. & Evilia, C. Protein adaptations in archaeal extremophiles. *Archaea* **2013**, e373275 (2013).
3. Protze, J. *et al.* An extracellular tetrathionate hydrolase from the thermoacidophilic archaeon *Acidianus ambivalens* with an activity optimum at pH 1. *Front. Microbiol.* **2**, (2011).
4. Pradeep, G. C. *et al.* An extremely alkaline novel chitinase from *Streptomyces sp.* CS495. *Process Biochem.* **49**, 223–229 (2014).
5. Ferrer, M., Golyshina, O., Beloqui, A. & Golyshin, P. N. Mining enzymes from extreme environments. *Curr. Opin. Microbiol.* **10**, 207–214 (2007).
6. Verma, D. & Satyanarayana, T. Xylanolytic extremozymes retrieved from environmental metagenomes: Characteristics, genetic engineering, and applications. *Front. Microbiol.* **11**, (2020).
7. Shahraki, M. F. *et al.* A computational learning paradigm to targeted discovery of biocatalysts from metagenomic data: A case study of lipase identification. *Biotechnol. Bioeng.* **119**, 1115–1128 (2022).
8. Erickson, E. *et al.* Sourcing thermotolerant poly(ethylene terephthalate) hydrolase scaffolds from natural diversity. *Nat. Commun.* **13**, 7850 (2022).
9. Wang, C.-H., Liu, X.-L., Huang, R.-B., He, B.-F. & Zhao, M.-M. Enhanced acidic adaptation of *Bacillus subtilis* Ca-independent alpha-amylase by rational engineering of pKa values. *Biochem. Eng. J.* **139**, 146–153 (2018).
10. dos Santos, J. P., Zavareze, E. da R., Dias, A. R. G. & Vanier, N. L. Immobilization of xylanase and xylanase–β-cyclodextrin complex in polyvinyl alcohol via electrospinning improves enzyme activity at a wide pH and temperature range. *Int. J. Biol. Macromol.* **118**, 1676–1684 (2018).
11. Giri, P., Pagar, A. D., Patil, M. D. & Yun, H. Chemical modification of enzymes to improve biocatalytic performance. *Biotechnol. Adv.* **53**, 107868 (2021).
12. Xue, Y. *et al.* Chemical modification of stem bromelain with anhydride groups to enhance its stability and catalytic activity. *J. Mol. Catal. B Enzym.* **63**, 188–193 (2010).
13. Li, C. Effects of chemical modification by chitooligosaccharide on enzyme activity and stability of yeast β-D-fructofuranosidase. *Enzyme Microb. Technol.* **64–65**, 24–32 (2014).
14. Li, S.-F., Cheng, F., Wang, Y.-J. & Zheng, Y.-G. Strategies for tailoring pH performances of glycoside hydrolases. *Crit. Rev. Biotechnol.* **43**, 121–141 (2023).
15. Shi, X., Wu, D., Xu, Y. & Yu, X. Engineering the optimum pH of β-galactosidase from Aspergillus oryzae for efficient hydrolysis of lactose. *J. Dairy Sci.* **105**, 4772–4782 (2022).
16. Wittmund, M., Cadet, F. & Davari, M. D. Learning epistasis and residue coevolution patterns: Current trends and future perspectives for advancing enzyme engineering. *ACS Catal.* **12**, 14243–14263 (2022).
17. Zhou, S. *et al.* Improving catalytic efficiency and maximum activity at low pH of *Aspergillus neoniger* phytase using rational design. *Int. J. Biol. Macromol.* **131**, 1117–1124 (2019).
18. Yan, Y. *et al.* Surface charge modifications modulate glucose oxidase pH-activity profiles for efficient gluconic acid production. *J. Clean. Prod.* **372**, 133817 (2022).
19. Wang, Y., Xu, M., Yang, T., Zhang, X. & Rao, Z. Surface charge-based rational design of aspartase modifies the optimal pH for efficient β-aminobutyric acid production. *Int. J. Biol. Macromol.* **164**, 4165–4172 (2020).
20. Pokhrel, S., Joo, J. C., Kim, Y. H. & Yoo, Y. J. Rational design of a <i>Bacillus circulans<i> xylanase by introducing charged residue to shift the pH optimum. *Process Biochem.* **47**, 2487–2493 (2012).
21. Pokhrel, S., Joo, J. C. & Yoo, Y. J. Shifting the optimum pH of *Bacillus circulans* xylanase towards acidic side by introducing arginine. *Biotechnol. Bioprocess Eng.* **18**, 35–42 (2013).
22. Suplatov, D. *et al.* Computational design of a pH stable enzyme: understanding molecular mechanism of penicillin acylase's adaptation to alkaline conditions. *PLOS ONE* **9**, e100643 (2014).

23. Li, Y. *et al.* Directed evolution of a β-mannanase from Rhizomucor miehei to improve catalytic activity in acidic and thermophilic conditions. *Biotechnol. Biofuels* **10**, 143 (2017).
24. Hu, Y. *et al.* Engineering a recombination neutral protease I from *Aspergillus oryzae* to improve enzyme activity at acidic pH. *RSC Adv.* **10**, 30692–30699 (2020).
25. Ma, F. *et al.* Sequence homolog-based molecular engineering for shifting the enzymatic pH optimum. *Synth. Syst. Biotechnol.* **1**, 195–206 (2016).
26. Li, G. *et al.* Learning deep representations of enzyme thermal adaptation. *Protein Sci.* **31**, e4480 (2022).
27. Wang, R., Jin, J., Zou, Q., Nakai, K. & Wei, L. Predicting protein–peptide binding residues via interpretable deep learning. *Bioinformatics* **38**, 3351–3360 (2022).
28. Kroll, A., Hu, X.-P., Liebrand, N. A. & Lercher, M. J. Turnover number predictions for kinetically uncharacterized enzymes using machine and deep learning. 2022.11.10.516024 Preprint at https://doi.org/10.1101/2022.11.10.516024 (2022).
29. Richardson, L. *et al.* MGnify: the microbiome sequence data analysis resource in 2023. *Nucleic Acids Res.* **51**, D753–D759 (2023).
30. Hebditch, M. & Warwicker, J. Web-based display of protein surface and pH-dependent properties for assessing the developability of biotherapeutics. *Sci. Rep.* **9**, 1969 (2019).
31. Schmitz, M. *et al.* patcHwork: a user-friendly pH sensitivity analysis web server for protein sequences and structures. *Nucleic Acids Res.* **50**, W560–W567 (2022).
32. Oeller, M. *et al.* Sequence-based prediction of pH-dependent protein solubility using CamSol. *Brief. Bioinform.* **24**, bbad004 (2023).
33. Zhang, G., Li, H. & Fang, B. Discriminating acidic and alkaline enzymes using a random forest model with secondary structure amino acid composition. *Process Biochem.* **44**, 654–660 (2009).
34. Lin, H., Chen, W. & Ding, H. AcalPred: A sequence-based tool for discriminating between acidic and alkaline enzymes. *PLOS ONE* **8**, e75726 (2013).
35. Fan, G.-L., Li, Q.-Z. & Zuo, Y.-C. Predicting acidic and alkaline enzymes by incorporating the average chemical shift and gene ontology informations into the general form of Chou's PseAAC. *Process Biochem.* **48**, 1048–1053 (2013).
36. Khan, Z. U., Hayat, M. & Khan, M. A. Discrimination of acidic and alkaline enzyme using Chou's pseudo amino acid composition in conjunction with probabilistic neural network model. *J. Theor. Biol.* **365**, 197–203 (2015).
37. Yan, S. & Wu, G. Predicting pH optimum for activity of beta-glucosidases. *J. Biomed. Sci. Eng.* **12**, 354–367 (2019).
38. Wang, X., Li, H., Gao, P., Liu, Y. & Zeng, W. Combining support vector machine with dual g-gap dipeptides to discriminate between acidic and alkaline enzymes. *Lett. Org. Chem.* **16**, 325–331 (2019).
39. Li, X. *et al.* A sequence embedding method for enzyme optimal condition analysis. *BMC Bioinformatics* **21**, 512 (2020).
40. Tan, C. *et al.* A Survey on deep transfer learning. Preprint at https://doi.org/10.48550/arXiv.1808.01974 (2018).
41. Yang, K. K., Lu, A. X. & Fusi, N. Convolutions are competitive with transformers for protein sequence pretraining. 2022.05.19.492714 Preprint at https://doi.org/10.1101/2022.05.19.492714 (2022).
42. Rao, R. *et al.* Evaluating protein transfer learning with TAPE. Preprint at https://doi.org/10.48550/arXiv.1906.08230 (2019).
43. Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M. & Church, G. M. Unified rational protein engineering with sequence-based deep representation learning. *Nat. Methods* **16**, 1315–1322 (2019).
44. Teufel, F. *et al.* SignalP 6.0 predicts all five types of signal peptides using protein language models. *Nat. Biotechnol.* **40**, 1023–1025 (2022).
45. Thumuluri, V. *et al.* NetSolP: predicting protein solubility in *Escherichia coli* using language models. *Bioinforma. Oxf. Engl.* **38**, 941–946 (2022).
46. Marquet, C. *et al.* Embeddings from protein language models predict conservation and variant effects. *Hum. Genet.* **141**, 1629–1647 (2022).
47. Pokharel, S., Pratyush, P., Heinzinger, M., Newman, R. H. & Kc, D. B. Improving protein succinylation sites prediction using embeddings from protein language model. *Sci. Rep.* **12**, 16933 (2022).
48. Meier, J. *et al.* Language models enable zero-shot prediction of the effects of mutations on protein function. 2021.07.09.450648 Preprint at https://doi.org/10.1101/2021.07.09.450648 (2021).
49. Schomburg, I. *et al.* The BRENDA enzyme information system–From a database to an expert system. *J. Biotechnol.* **261**, 194–206 (2017).
50. Puissant, J. *et al.* The pH optimum of soil exoenzymes adapt to long term changes in soil pH. *Soil Biol. Biochem.* **138**, 107601 (2019).
51. Reimer, L. C. *et al.* BacDive in 2022: the knowledge base for standardized bacterial and archaeal data. *Nucleic Acids Res.* **50**, D741–D746 (2022).
52. Sayers, E. W. *et al.* Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.* **49**, D10–D17 (2021).
53. Booth, I. R. Regulation of cytoplasmic pH in bacteria. *Microbiol. Rev.* **49**, 359–378 (1985).

54. Baker-Austin, C. & Dopson, M. Life in acid: pH homeostasis in acidophiles. *Trends Microbiol.* **15**, 165–171 (2007).
55. Hough, D. W. & Danson, M. J. Extremozymes. *Curr. Opin. Chem. Biol.* **3**, 39–46 (1999).
56. Gado, J. E., Beckham, G. T. & Payne, C. M. Improving enzyme optimum temperature prediction with resampling strategies and ensemble learning. *J. Chem. Inf. Model.* **60**, 4098–4107 (2020).
57. Branco, P., Torgo, L. & Ribeiro, R. P. Pre-processing approaches for imbalanced distributions in regression. *Neurocomputing* **343**, 76–99 (2019).
58. Yang, Y., Zha, K., Chen, Y.-C., Wang, H. & Katabi, D. Delving into deep imbalanced regression. Preprint at https://doi.org/10.48550/arXiv.2102.09554 (2021).
59. Chicco, D. Ten quick tips for machine learning in computational biology. *BioData Min.* **10**, 35 (2017).
60. Chen, Z. *et al.* iFeature: a Python package and web server for features extraction and selection from protein and peptide sequences. *Bioinforma. Oxf. Engl.* **34**, 2499–2502 (2018).
61. Rives, A. *et al.* Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci.* **118**, e2016239118 (2021).
62. Elnaggar, A. *et al.* ProtTrans: toward understanding the language of life through self-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 7112–7127 (2022).
63. Notin, P. *et al.* Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval. Preprint at https://doi.org/10.48550/arXiv.2205.13760 (2022).
64. Nijkamp, E., Ruffolo, J., Weinstein, E. N., Naik, N. & Madani, A. ProGen2: exploring the boundaries of protein language models. Preprint at https://doi.org/10.48550/arXiv.2206.13517 (2022).
65. Bergstra, J. & Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **13**, 281–305 (2012).
66. Jiang, Y. *et al.* Expert feature-engineering vs. deep neural networks: 19th International Conference on Artificial Intelligence in Education, AIED 2018. *Artif. Intell. Educ.* 198–211 (2018) doi:10.1007/978-3-319-93843-1_15.
67. Stärk, H., Dallago, C., Heinzinger, M. & Rost, B. Light attention predicts protein location from the language of life. *Bioinforma. Adv.* **1**, vbab035 (2021).
68. Villegas-Morcillo, A., Gomez, A. M. & Sanchez, V. An analysis of protein language model embeddings for fold prediction. *Brief. Bioinform.* **23**, bbac142 (2022).
69. Li, G. *et al.* Performance of Regression Models as a Function of Experiment Noise. *Bioinforma. Biol. Insights* **15**, 11779322211020316 (2021).
70. Detlefsen, N. S., Hauberg, S. & Boomsma, W. Learning meaningful representations of protein sequences. *Nat. Commun.* **13**, 1914 (2022).
71. Kroll, A. & Lercher, M. J. Machine learning models for the prediction of enzyme properties should be tested on proteins not used for model training. 2023.02.06.526991 Preprint at https://doi.org/10.1101/2023.02.06.526991 (2023).
72. Huang, Y., Krauss, G., Cottaz, S., Driguez, H. & Lipps, G. A highly acid-stable and thermostable endo-beta-glucanase from the thermoacidophilic archaeon *Sulfolobus solfataricus*. *Biochem. J.* **385**, 581–588 (2005).
73. Mamo, G., Thunnissen, M., Hatti-Kaul, R. & Mattiasson, B. An alkaline active xylanase: insights into mechanisms of high pH catalytic adaptation. *Biochimie* **91**, 1187–1196 (2009).
74. Jakob, F. *et al.* Surface charge engineering of a *Bacillus gibsonii* subtilisin protease. *Appl. Microbiol. Biotechnol.* **97**, 6793–6802 (2013).
75. Yang, T. *et al.* N20D/N116E combined mutant downward shifted the pH optimum of *Bacillus subtilis* NADH Oxidase. *Biology* **12**, 522 (2023).
76. Masui, A., Fujiwara, N., Yamamoto, K., Takagi, M. & Imanaka, T. Rational design for stabilization and optimum pH shift of serine protease AprN. *J. Ferment. Bioeng.* **85**, 30–36 (1998).
77. Turunen, O., Vuorio, M., Fenel, F. & Leisola, M. Engineering of multiple arginines into the Ser/Thr surface of *Trichoderma reesei* endo-1,4-β-xylanase II increases the thermotolerance and shifts the pH optimum towards alkaline pH. *Protein Eng. Des. Sel.* **15**, 141–145 (2002).
78. Li, Q., Jiang, T., Liu, R., Feng, X. & Li, C. Tuning the pH profile of β-glucuronidase by rational site-directed mutagenesis for efficient transformation of glycyrrhizin. *Appl. Microbiol. Biotechnol.* **103**, 4813–4823 (2019).
79. Carvalho, D. V., Pereira, E. M. & Cardoso, J. S. Machine Learning Interpretability: A Survey on Methods and Metrics. *Electronics* **8**, 832 (2019).
80. Austin, H. P. *et al.* Characterization and engineering of a plastic-degrading aromatic polyesterase. *Proc. Natl. Acad. Sci. U. S. A.* **115**, E4350–E4357 (2018).
81. Olsson, M. H. M., Søndergaard, C. R., Rostkowski, M. & Jensen, J. H. PROPKA3: Consistent treatment of internal and surface residues in empirical pKa predictions. *J. Chem. Theory Comput.* **7**, 525–537 (2011).
82. Talley, K. & Alexov, E. On the pH-optimum of activity and stability of proteins. *Proteins* **78**, 2699–2706 (2010).
83. Alexov, E. Numerical calculations of the pH of maximal protein stability. The effect of the sequence composition and three-dimensional structure. *Eur. J. Biochem.* **271**, 173–185 (2004).

84. Yu, T. *et al.* Enzyme function prediction using contrastive learning. *Science* **379**, 1358–1363 (2023).
85. Pak, M. A., Dovidchenko, N. V., Sharma, S. M. & Ivankov, D. N. New mega dataset combined with deep neural network makes a progress in predicting impact of mutation on protein stability. 2022.12.31.522396 Preprint at https://doi.org/10.1101/2022.12.31.522396 (2023).
86. Unsal, S. *et al.* Learning functional properties of proteins with language models. *Nat. Mach. Intell.* **4**, 227–245 (2022).
87. Yeung, W., Zhou, Z., Li, S. & Kannan, N. Alignment-free estimation of sequence conservation for identifying functional sites using protein sequence embeddings. *Brief. Bioinform.* **24**, bbac599 (2023).
88. Ferdous, S., Shihab, I. F. & Reuel, N. F. Effects of sequence features on machine-learned enzyme classification fidelity. *Biochem. Eng. J.* **187**, 108612 (2022).
89. Li, G., Rabe, K. S., Nielsen, J. & Engqvist, M. K. M. Machine Learning Applied to Predicting Microorganism Growth Temperatures and Enzyme Catalytic Optima. *ACS Synth. Biol.* **8**, 1411–1420 (2019).
90. Liu, H., HaoChen, J. Z., Gaidon, A. & Ma, T. Self-supervised learning is more robust to dataset imbalance. Preprint at https://doi.org/10.48550/arXiv.2110.05025 (2022).
91. UniProt Consortium. UniProt: the Universal Protein Knowledgebase in 2023. *Nucleic Acids Res.* **51**, D523–D531 (2023).
92. Hopf, T. A. *et al.* Mutation effects predicted from sequence co-variation. *Nat. Biotechnol.* **35**, 128–135 (2017).
93. Hie, B. L. & Yang, K. K. Adaptive machine learning for protein engineering. *Curr. Opin. Struct. Biol.* **72**, 145–152 (2022).
94. Yang, K. K., Wu, Z. & Arnold, F. H. Machine-learning-guided directed evolution for protein engineering. *Nat. Methods* **16**, 687–694 (2019).
95. Hawkins-Hooker, A. *et al.* Generating functional protein variants with variational autoencoders. *PLoS Comput. Biol.* **17**, e1008736 (2021).
96. Strokach, A. & Kim, P. M. Deep generative modeling for protein design. *Curr. Opin. Struct. Biol.* **72**, 226–236 (2022).
97. Hsu, C., Nisonoff, H., Fannjiang, C. & Listgarten, J. Learning protein fitness models from evolutionary and assay-labeled data. *Nat. Biotechnol.* **40**, 1114–1122 (2022).
98. Wittmann, B. J., Yue, Y. & Arnold, F. H. Informed training set design enables efficient machine learning-assisted directed protein evolution. *Cell Syst.* **12**, 1026-1045.e7 (2021).
99. Steinegger, M. & Söding, J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat. Biotechnol.* **35**, 1026–1028 (2017).
100. Cock, P. J. A. *et al.* Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **25**, 1422–1423 (2009).
101. Cui, Y., Jia, M., Lin, T.-Y., Song, Y. & Belongie, S. Class-balanced loss based on effective number of samples. in 9268–9277 (2019).
102. Riesselman, A. J., Ingraham, J. B. & Marks, D. S. Deep generative models of genetic variation capture the effects of mutations. *Nat. Methods* **15**, 816–822 (2018).
103. Guyon, I., Weston, J., Barnhill, S. & Vapnik, V. Gene selection for cancer classification using support vector machines. *Mach. Learn.* **46**, 389–422 (2002).
104. Shin, J.-E. *et al.* Protein design and variant prediction using autoregressive generative models. *Nat. Commun.* **12**, 2403 (2021).
105. Oord, A. van den *et al.* WaveNet: a generative model for raw audio. Preprint at https://doi.org/10.48550/arXiv.1609.03499 (2016).
106. Cho, K. *et al.* Learning phrase representations using RNN encoder–decoder for statistical machine translation. in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* 1724–1734 (Association for Computational Linguistics, 2014). doi:10.3115/v1/D14-1179.
107. Pascanu, R., Mikolov, T. & Bengio, Y. On the difficulty of training recurrent neural networks. in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28* III-1310-III–1318 (JMLR.org, 2013).
108. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997).
109. Littmann, M., Heinzinger, M., Dallago, C., Olenyi, T. & Rost, B. Embeddings from deep learning transfer GO annotations beyond homology. *Sci. Rep.* **11**, 1160 (2021).
110. Pedregosa, F. *et al.* Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
111. Paszke, A. *et al.* PyTorch: an imperative Style, high-performance deep learning library. in *Advances in Neural Information Processing Systems* vol. 32 (Curran Associates, Inc., 2019).
112. Varadi, M. *et al.* AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Res.* **50**, D439–D444 (2022).
113. Mirdita, M. *et al.* ColabFold: making protein folding accessible to all. *Nat. Methods* **19**, 679–682 (2022).
114. Shrake, A. & Rupley, J. A. Environment and exposure to solvent of protein atoms. Lysozyme and insulin. *J. Mol. Biol.* **79**, 351–371 (1973).

115. Rost, B. & Sander, C. Conservation and prediction of solvent accessibility in protein families. *Proteins* **20**, 216–226 (1994).

116. Savojardo, C., Manfredi, M., Martelli, P. L. & Casadio, R. Solvent accessibility of residues undergoing pathogenic variations in humans: from protein structures to protein sequences. *Front. Mol. Biosci.* **7**, (2021).