



NNKcat: deep neural network to predict catalytic constants (K_{cat}) by integrating protein sequence and substrate structure with enhanced data imbalance handling

Jingchen Zhai ¹, Xiguang Qi¹, Lianjin Cai¹, Yue Liu¹, Haocheng Tang¹, Lei Xie^{2,3,*}, Junmei Wang ^{1,*}

¹Department of Pharmaceutical Sciences and Computational Chemical Genomics Screening Center, School of Pharmacy, University of Pittsburgh, 3501 Terrace St, Pittsburgh, PA 15261, United States

²Department of Computer Science, Hunter College, The City University of New York, 695 Park Ave, New York, NY 10065, United States

³Helen & Robert Appel Alzheimer's Disease Research Institute, Feil Family Brain & Mind Research Institute, Weill Cornell Medicine, Cornell University, 413 E 69th St, New York, NY 10021, United States

*Corresponding authors. Junmei Wang, Department of Pharmaceutical Sciences and Computational Chemical Genomics Screening Center, School of Pharmacy, University of Pittsburgh. E-mail: juw79@pitt.edu; Lei Xie, Department of Computer Science, Hunter College, The City University of New York and Helen & Robert Appel Alzheimer's Disease Research Institute, Feil Family Brain & Mind Research Institute, Weill Cornell Medicine, Cornell University. E-mail: lxie@iscb.org

Abstract

Catalytic constant (K_{cat}) is to describe the efficiency of catalyzing reactions. The K_{cat} value of an enzyme-substrate pair indicates the rate an enzyme converts saturated substrates into product during the catalytic process. However, it is challenging to construct robust prediction models for this important property. Most of the existing models, including the one recently published by Nature Catalysis (Li *et al.*), are suffering from the overfitting issue. In this study, we proposed a novel protocol to construct K_{cat} prediction models, introducing an intermedia step to separately develop substrate and protein processors. The substrate processor leverages analyzing Simplified Molecular Input Line Entry System (SMILES) strings using a graph neural network model, attentive FP, while the protein processor abstracts protein sequence information utilizing long short-term memory architecture. This protocol not only mitigates the impact of data imbalance in the original dataset but also provides greater flexibility in customizing the general-purpose K_{cat} prediction model to enhance the prediction accuracy for specific enzyme classes. Our general-purpose K_{cat} prediction model demonstrates significantly enhanced stability and slightly better accuracy (R^2 value of 0.54 versus 0.50) in comparison with Li *et al.*'s model using the same dataset. Additionally, our modeling protocol enables personalization of fine-tuning the general-purpose K_{cat} model for specific enzyme categories through focused learning. Using Cytochrome P450 (CYP450) enzymes as a case study, we achieved the best R^2 value of 0.64 for the focused model. The high-quality performance and expandability of the model guarantee its broad applications in enzyme engineering and drug research & development.

Keywords: K_{cat} ; enzyme turnover number; data imbalance; deep neural network; machine learning; focused learning

Introduction

Catalytic constant (K_{cat}) is a crucial parameter in enzymatic reactions [1]. It represents the maximum number of substrate molecules that a single enzyme molecule can convert per unit time under saturating substrate conditions, reflecting the enzyme's catalytic efficiency for a given substrate [2, 3]. In biological and pharmaceutical fields, the knowledge of K_{cat} value allows researchers to compare enzyme catalytic activity and the potency of small molecules in different application scenarios [4, 5]. However, obtaining K_{cat} value through various methods can be challenging. Experimentally determining K_{cat} values requires significant time and financial investment [6–9], also the measured K_{cat} values are scattered in a variety of literature sources [10–20]. Moreover, computational tools for predicting K_{cat} values are limited [5, 21], primarily due to the complexity involved in K_{cat} prediction, which depends on numerous intricate factors.

Researchers have explored various approaches for predicting K_{cat} values and related enzymatic properties [22–29]. In a recent study by Li *et al.* [30], a K_{cat} prediction model was constructed that employs a convolutional neural network (CNN) to process protein sequences and a graph neural network (GNN) to extract substrate features. Following this publication, several researchers raised concerns regarding the robustness of this model [31, 32]. We thus conducted an in-depth analysis of their model and found that the model performance is highly sensitive to the choice of random seed in splitting the training and test set. In their reported results, the final model used the random seed of “1234,” which produced satisfactory outcomes with their protocol. To replicate their study, we precisely followed their experimental steps and successfully reproduced their results. However, when we changed the random seed to “1357” and kept all other conditions unchanged, the model's performance deteriorated significantly. In their experiment, the random seed was used to split the primary dataset

Received: February 11, 2025. Revised: April 14, 2025. Accepted: April 21, 2025

© The Author(s) 2025. Published by Oxford University Press.

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<https://creativecommons.org/licenses/by-nc/4.0/>), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact journals.permissions@oup.com

(referred to as Dataset #1 hereafter) into training, validation, and test sets, leading us to conclude that the quality of their model heavily depends on how the dataset is partitioned. Furthermore, they did not thoroughly examine or report the composition of the training data.

A closer analysis of Dataset #1 revealed that, while it contains 16,618 records, sequences and substrates appear with varying degrees of repetition when forming different pairs. That said, this issue is understandable, as K_{cat} measurements are typically obtained for reactions involving specific proteins or classes of proteins with a series of substrates. As such, K_{cat} datasets often contain substantial repetition of various substrates and proteins, leading to an imbalanced dataset.

In recent years, many other studies have also focused on developing more effective K_{cat} prediction tools using various modeling technologies. A K_{cat} prediction model recently developed by Heckmann et al. [33] greatly enhanced our understanding of *Escherichia coli*'s metabolism and proteome at the genome scale, albeit the model is difficult to be applied to other species due to its species-focused enzyme types and complex input features. Later, UniPK developed by Yu et al. [5] utilized protein language models (PLMs) and SMILES transformers to analyze protein and substrate information to train models via various machine learning models including CNN and recurrent neural network. Ultimately, the authors identified the trees ensemble model, which achieved an R^2 of 0.65, as the best performer as it could distinguish mutation effects on K_{cat} prediction. Moreover, the model was extended to EF-UniPK to further incorporate pH and temperature effects. TurNup was another K_{cat} prediction model developed by Kroll et al. [34] which applied differential reaction fingerprints and a re-trained transformer network model to process chemical reactions and protein sequences, respectively. TurNup was robust as it achieved an R^2 of 0.33 for enzymes which lacked homologs, as measured by sequence identity being equal to or larger than 40%, in the training set. Recently, MPEK was developed as a pre-trained multi-task deep learning model to predict both K_{cat} and K_m simultaneously [35]. By applying protein language models and substrate language models to extract protein and substrate information, the constructed multi-task model achieved a Partial Correlation Coefficients of 0.808 for K_{cat} . However, this model required information on pH, temperature, and organism as additional input. Similarly, DLTKcat, a deep learning model developed by Qiu et al. [36] achieved an R^2 of 0.66 in temperature-dependent K_{cat} prediction. Their approach leveraged a graph attention network to extract features from substrates which were represented by graphs, and CNN to process enzyme protein sequences. In the same year, Wang et al. developed DeepEnzyme [37], which utilized a combination of transformers and graph convolutional networks (GCNs) to extract features from the sequences and 3D-structures of enzymes, and GCN to extract to process substrate graphs. DeepEnzyme achieved an R^2 of ~ 0.6 for K_{cat} prediction, outperforming the aforementioned TurNup and DLTKcat models. CatPred is one of the most recently developed prediction models by Boorla et al. for K_{cat} prediction [38]. The model achieved optimal performance of an R^2 of 0.608 and demonstrated strong predictive performance on out-of-distribution samples. Their method utilized a neural network to process substrate SMILES strings and a sequence attention algorithm along with a pre-trained protein language model to process protein sequences.

In this study, we propose a novel K_{cat} prediction protocol that utilizes substrate structures and protein amino acid sequences as input for the model. A key feature of our approach is the development of separate processors for substrates and enzyme

sequences, utilizing attentive FP and long short-term memory algorithm respectively. This design not only mitigates the impact of data imbalance but also offers greater flexibility for partial model adjustments and later customization. Our basic prediction model demonstrates significantly improved stability and accuracy when compared to the performance metrics from publications using the same dataset. We also used Cytochrome P450 (CYP450) enzymes as a case study to demonstrate how to fine-tune the general-purpose K_{cat} model into a more targeted prediction model for a specific enzyme category through focused learning. This case has highlighted the enhanced performance of a focused learning model compared to a general-purpose model, while still possessing stable performance reflected by parallel experiment results. The high-quality performance and expandability of the model guarantee its broad applications in enzyme engineering and drug research & development, while minimizing the need for extensive in vivo and in vitro experiments.

Result

Dataset distribution

The overall flowchart of this study is shown in Fig. 1. The distribution of substrates and proteins is summarized in Table 1 and shown in Fig. 2. Dataset #1 for model construction contains 16,618 different substrate-enzyme pairs with corresponding K_{cat} values (Supplemental Table S1) and Dataset #2A (Supplemental Table S2) consists of 161 substrate-enzyme pairs for external model validation. Table 1 summarized the number of records (substrate-enzyme pairs) contained within each enzyme commission (EC) class [39], as well as the number of proteins and substrates involved throughout the study.

The distribution of substrate molecular weight (MW) and amino acid sequence length are shown in Fig. 2. In Dataset #1, the maximum MW is 6176.02 g/mol, while most of the substrates have their MW <2000. In Dataset #2A, the MW of all substrates are within 2000, with the maximum value of 1578.66 g/mol. Enzyme sequences in Dataset #1 are mostly within 1500 amino acids, with the longest one having 3712 amino acids. Sequences in Dataset #2A are all within 1200 amino acids. Even with the above differences, the distribution patterns of the two datasets are essentially similar for both MW and sequence length, suggesting that Dataset #2A is suitable to validate the performance of the models.

Substrate processor

In this study, we first constructed substrate and protein processors separately to solve the problem of data imbalance, because proteins and substrates can have different degrees of repetition in Dataset #1 during forming pairs through different combinations. We calculated the characteristic value for each substrate in Dataset #1 to serve as the response of each substrate during the construction of substrate processor model. The characteristic value of a substrate, the mean K_{cat} value for all the records of the given substrate, can indicate its overall activity when served as a substrate to different enzymes. In total, 2647 substrates with their characteristic values were obtained from Dataset #1, and were divided into two groups in a 9:1 ratio for model training and test purposes. As depicted in Fig. 3A, the best model was achieved at the 10th epoch, which achieved the lowest root-mean-square error (RMSE) on the test set, with a value of 4.297. The corresponding RMSE on the training set for this model was 3.895. The detailed performance of this model on the training and test sets are shown in Fig. 3B and C, respectively. This model

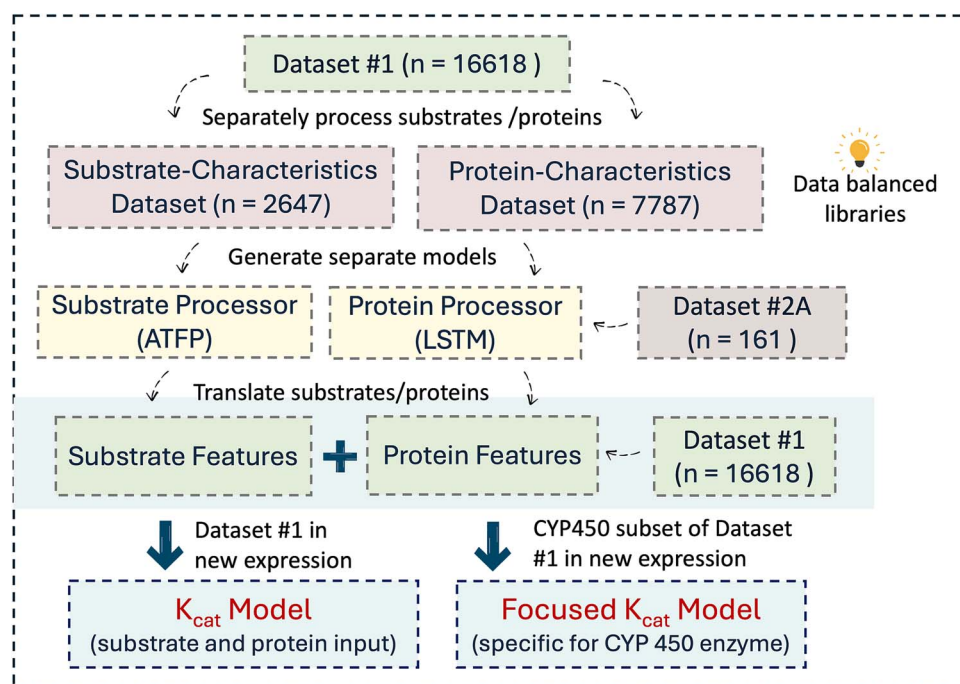


Figure 1. A flowchart highlights the key components of Kcat model development in this work. We first constructed the substrate processor and protein processor separately utilizing Dataset #1 and then conducted feature augmentation with both processors. Next, all the generated feature embeddings are combined to train the general-purpose Kcat prediction models. Three parallel experiments are conducted during the training and testing the general-purpose Kcat models. Last, Dataset #2A was further applied to objectively evaluate the general-purpose Kcat models.

Table 1. The distribution of substrate-enzyme pairs in dataset #1 and dataset #2A.

	Dataset #1 (Model training and validation)	Dataset #2A (External validation)
Oxidoreductases (EC = 1)	6528	90
Transferases (EC = 2)	4027	37
Hydrolases (EC = 3)	3147	29
Lyases (EC = 4)	1577	1
Isomerases (EC = 5)	790	2
Ligases (EC = 6)	542	0
Translocases (EC = 7)	7	2
Record total number	16 618	161
Unique Sequence	7787	144
Unique Substrate	2647	32

EC: The first part of the enzyme commission (EC) number of a sequence. E.g., EC number 1.14.14.2, is categorized to EC = 1 in this table.

was further developed in the following step to generate substrate embeddings for general-purpose K_{cat} model construction.

Protein processor

Similarly, we obtained 7787 different protein sequences with their calculated characteristic values from Dataset #1 for constructing the protein processor using the randomly generated training and test sets split at a 9:1 ratio. The model optimization process is exhibited in Fig. 4A. The best-performing model on the test set achieved an RMSE value of 4.895 at epoch 27, while the corresponding RMSE on the training set was 3.699. The prediction results of characteristic values for each protein sequence by this model versus the calculated characteristic values (mean K_{cat} value for each protein sequence) are displayed in Fig. 4B and C for the training set and test set, respectively. Apparently, the protein processor has better performance than the substrate processor, indicating that the sequence information should influence the K_{cat} values more significantly than the substrate structure

information. We also obtained the R^2 value between the characteristic values and the predicted values for sequences in both training set and test set, which are 0.462 and 0.264, respectively. This model was further developed in the following step to generate protein embeddings for the general-purpose K_{cat} model construction.

K_{cat} prediction model: data augmentation and model combination

After we have constructed and selected the top models as the substrate and protein processors, we focused on exploring their joint contribution to the general-purpose K_{cat} prediction model. Specifically, we modified the model output from a specific value to a multi-dimensional vector to convey more detailed information on the characteristics of the substrate structures or enzyme sequences [40]. We generated two 30-dimensional feature vectors for all the entries in Dataset #1 (Supplemental Table S3), one for substrate structures and the other for enzyme sequences. The two

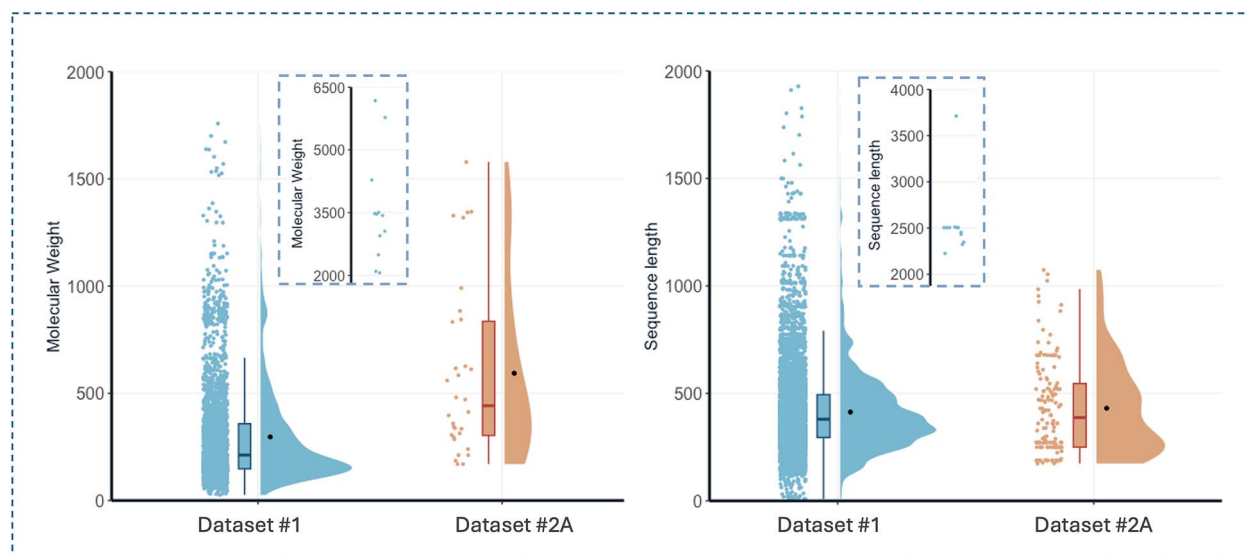


Figure 2. The distributions of Dataset #1 and Dataset #2A. Left: Distributions of substrate molecular weights in two datasets. Right: Distributions of amino acid sequence lengths of the proteins. The scatter and half-violin plots display the frequency distribution for each data group. Black dots in the half-violin plots represent the mean values for these groups. The box plots illustrate the central tendency of the data, highlighting the medians and quartiles for both groups.

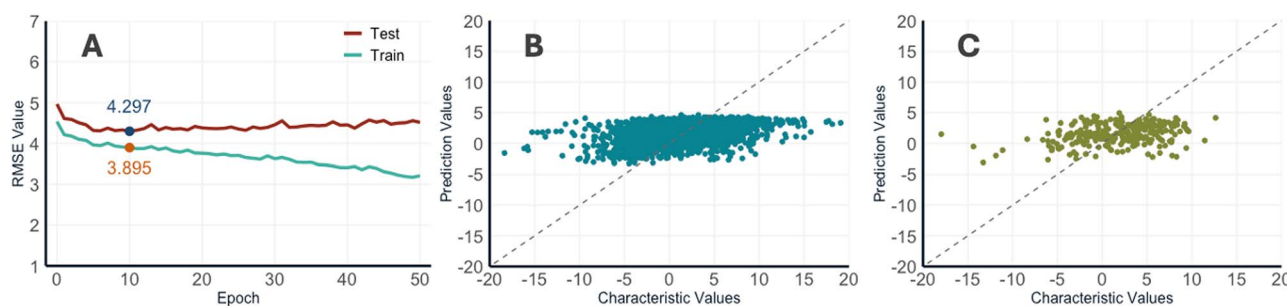


Figure 3. The performance of the substrate processor model. (A) The changing RMSE for the training and test sets during the model training process. The marked numbers are RMSE values for the best model (epoch 10). (B) The performance on the training set of the best model (epoch 10). (C) The model performance on the test set of the best model (epoch 10).

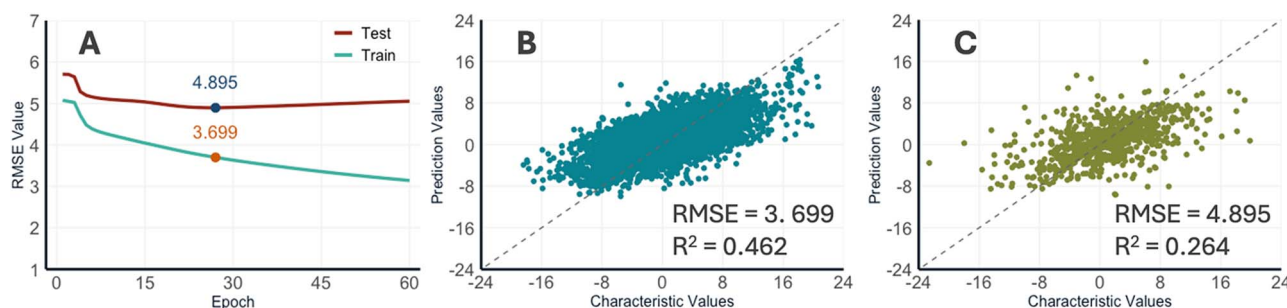


Figure 4. The performance of the protein processor. (A) The RMSE changing for the training and test sets during the model training process. The marked numbers are RMSE values for the best model (epoch 27). (B) The performance on the training set of the best model (epoch 27). (C) The performance on the test set of the best model (epoch 27).

feature vectors were then applied as descriptors to construct the general-purpose K_{cat} model. In total, we generated three random numbers to split Dataset #1 differently for model construction, and the splitting details are recorded in [Supplemental Table S4](#). The selection of random numbers is explained in the Method part.

A close examination on the performance of the prediction models using a variety of machine learning algorithms, the bagged decision tree ensemble model consistently performed the best, followed by the exponential Gaussian process regression (GPR),

and then the support vector machines (SVM). As shown in [Fig. 5](#), for each machine learning algorithm, the models of three random splits exhibit similar performance in terms of RMSE and R^2 . The mean RMSE values are 3.369, 3.521, and 3.658 for the top models trained by ensemble, GPR, and SVM, respectively; while the mean R^2 values are 0.538, 0.495 and 0.455 for the three corresponding algorithms. Apparently, the ensemble algorithm outperformed the others, thus, their models were selected for K_{cat} prediction and further expanded to predict specific enzyme classes.

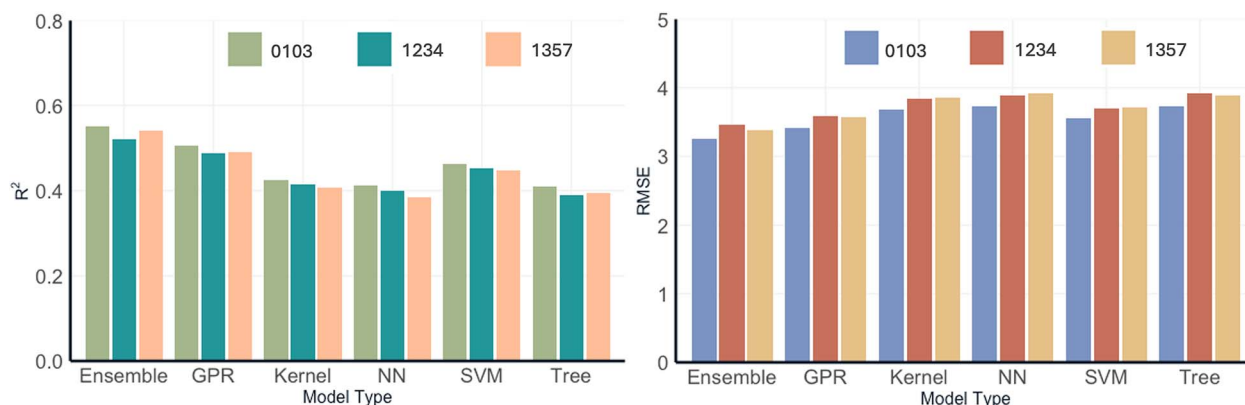


Figure 5. The model performance, measured by correlation coefficient square – R^2 (left panel) and rootmean-square errors-RMSE (right panel) of top machine learning models. Different random numbers were applied to divide dataset #1 into training and test sets. Higher R^2 indicates better correlation between predicted $\log_2 [K_{cat}]$ values and the experimental ones in dataset #1. Lower RMSE value indicates lower prediction error of $\log_2 [K_{cat}]$. GPR: Gaussian process regression; NN: Neural network; SVM: Support vector machine; tree: Decision tree.

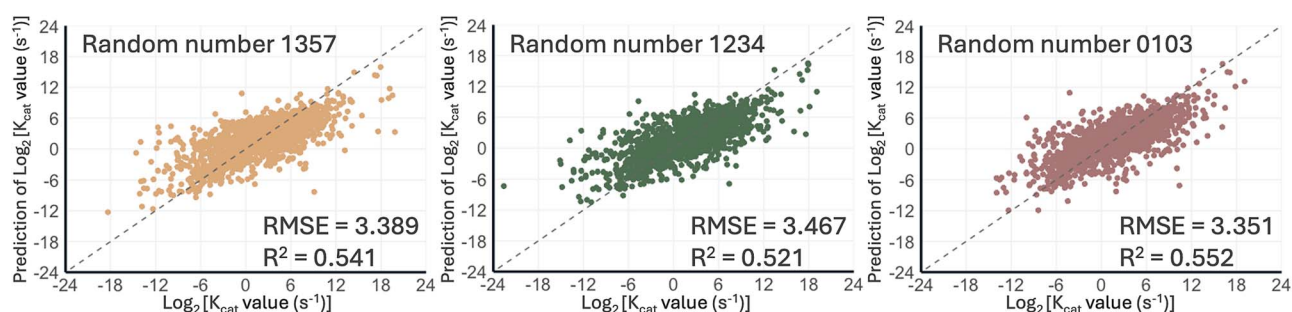


Figure 6. The test set performance of the models generated under three random splits on the dataset #1. For a comparison purpose, the performance of the Li *et al.*'s model constructed using the same dataset is listed as follows: Random number 1357: $R^2 = 0.203$; random number 1234: $R^2 = 0.516$; random number 0103: $R^2 = 0.543$. Note that we reproduced Li *et al.*'s model using the code they provided in GitHub.

We further investigated the model performance of the three ensemble models. As illustrated in Fig. 6, the scatter plots demonstrate similar patterns and exhibit similar performance, with the very small standard deviations of RMSE and R^2 , which are 0.109 and 0.016, respectively. More importantly, our models perform slightly better than the best model by Li *et al.* R^2 values of our models are slightly better than that predicted by Li *et al.*'s [30] model. As expected, the general-purpose K_{cat} prediction models show clear improvement when combining the features from both the substrate and enzyme processors (Figs. 4 and 6). Specifically, the R^2 increased from 0.264 of the protein processor to 0.538 (average) and RMSE decreased from 4.895 of the protein processor to 3.402 (average). This suggests that the feature vector of the substrates carries much valuable information for K_{cat} prediction, albeit the model performance of the substrate processor is unsatisfactory.

In summary, the results from parallel experiments indicate that models generated with our modeling protocol possess excellent prediction accuracy and high stability, basically unaffected by random data splitting. The three bagged decision tree ensemble models are exported for further external validation using Dataset #2A.

External application of K_{cat} prediction model

Dataset #2A includes 161 new yeast protein-substrate pairs, containing 144 unique yeast proteins and 32 unique small molecule substrates, as shown in Fig. 7. Considering that the protein plays the main role in the prediction of K_{cat} value for a catalytic reaction, we calculated the sequence similarity between each sequence from Dataset #2A and its most similar sequence in Dataset #1 if it

doesn't have an identical match. According to the sequence and substrate similarity between the two datasets, we divided Dataset #2A into three groups.

After dataset clean up (details were presented in the Method section), Groups A and B have calculated protein sequence identity ranging from 34.6% to 100%. The protein sequences in Group C have identical match in Dataset #1, but they form new substrate-enzyme pairs with different new substrates in Dataset #2A. As for sequences in Groups A and B, the difference from the most similar ones in Dataset #1 fall into the following three scenarios as illustrated in Fig. 8: (1) deletion/addition at one end of the sequence; (2) single/multiple mutation of the sequence; (3) deletion/addition in the middle of the sequence. From the substrate perspective, only Group A includes existing substrates in Dataset #1, while those in Groups B and C are all new substrate entries. The inspection of Dataset #2A revealed that our external test library contains mostly new substrate entries and includes proteins with diverse sequence identity compared to those in Dataset #1.

Applying our previously constructed substrate and protein processor models, each protein-substrate pair in Dataset #2A was converted into two 30-dimensional vectors as input features. The K_{cat} value for each pair was then predicted using the three general-purpose bagged decision tree ensemble models. As shown in Fig. 9. The three models produced a similar pattern of the observed vs. predicted scatter plots and maintained very good prediction performance, with the average RMSE and R^2 values of 3.972 and 0.500, respectively. It is encouraging that the performance of external validation using Dataset #2A only slightly declined. As shown Fig. 9, scatters tend to distribute around the

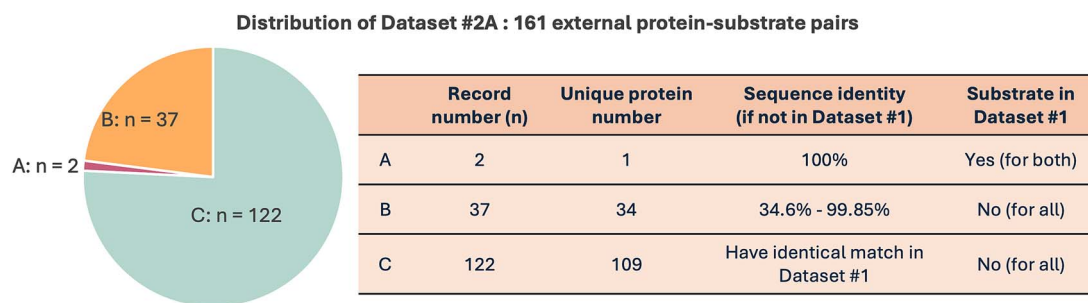


Figure 7. Similarity between substrates and protein sequences from dataset #2A and dataset #1. The sequence similarity is calculated by MUSCLE software.

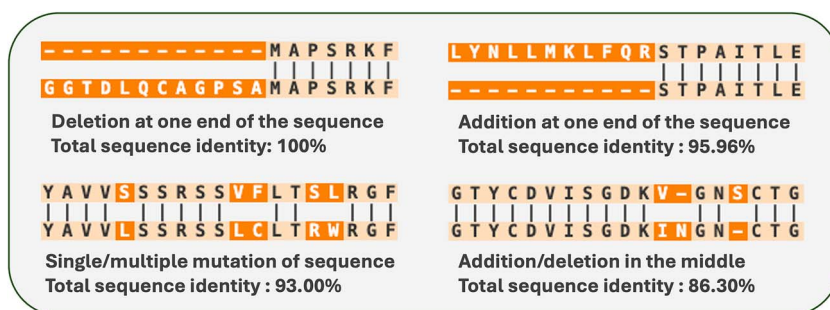


Figure 8. Illustration of sequence difference between Dataset #1 and Dataset #2A in three different scenarios. Top: Scenario 1; bottom left: Scenario 2; bottom right: Scenario 3.

Y = X trendline, and there is no obvious trend for sequence length to influence the model prediction performance. This indicates that our models are applicable to proteins with varying input sequence length. Furthermore, only two records from Dataset #2A contain substrates existing in Dataset #1, suggesting that our models can be applied to novel substrate entries.

Figure 9 shows that predictions for new K_{cat} records do not display a distinct performance between different groups, A and B versus C, as indicated by the distribution of scatter shapes (circles for Groups A and B, triangles for Group C). We further investigate the sequence identity's impact on the model performance focusing on only records in Groups A and B, allowing us to objectively evaluate our models' predictive performance when an entry containing new sequence.

For each sequence in Groups A and B, the calculated sequence identity with its most similar sequence in Dataset #1 using the Clustal Omega and MUSCLE programs are listed in Supplemental Table S5. The calculation results indicate that the MUSCLE program introduced fewer gaps during sequence alignment. Therefore, we used the identity scores generated by MUSCLE for subsequent analyses. As shown in the Fig. 10, sequences with identity >50% and those below 50% each account for roughly half of the total. Most of the scatters are still distributed around the Y = X trendline, while calculated sequence identity doesn't show significant influence on model prediction performance, as illustrated by the evenly distribution of the marker colors. This demonstrates that our model performs robustly, even when predicting proteins with low identity compared to proteins in Dataset #1.

Focused learning – CYP450 enzyme K_{cat} prediction model

Here we present a case study to illustrate how to customize our general-purpose model to improve its prediction accuracy of K_{cat} prediction for a specific enzyme category. We first filtered out all the K_{cat} records in Dataset #1 that involves CYP450 enzymes,

resulting in a sub-library containing 794 substrate-enzyme pairs. This CYP450 K_{cat} sub-library will be applied to fine-tune our general-purpose K_{cat} prediction model. The previously generated substrate and protein processors remained unchanged, as well as the feature vectors generated for training the general-purpose models. Considering the size of the sub-dataset is small for the focused learning, we split this sub-library into 8 groups and performed leave-one-group-out training to reduce random effect [41]. We also conducted three parallel experiments to randomly split this library, with the aim of investigating the robustness of our modeling protocol.

The data splitting results are exhibited through violin figures in Fig. 11A–C, with the model performance through parallel experiments shown in Fig. 11D–F. Our model maintained stable predictive performance, as evidenced by the similar R^2 and RMSE values across parallel experiments. Additionally, the model has achieved an improvement in prediction accuracy. The focused model achieved an R^2 of 0.633 on average, representing a significant improvement over the general-purpose models on both the validation set (0.538 on average) and the external test set (0.500 on average). This indicates that the substrate and protein processors can serve as pretrained models, and the general-purpose K_{cat} prediction model can be personalized for a specific enzyme category.

Discussion

This study aimed to develop robust K_{cat} prediction models. The models we developed demonstrated stable performance, unaffected by the random data splitting. We used the same dataset used by Li et al. for model construction, allowing us to directly compare the model performance of our models to theirs with the RMSE and R^2 performance metrics. As a result, our models outperformed theirs according to R^2 values from multiple parallel experiments.

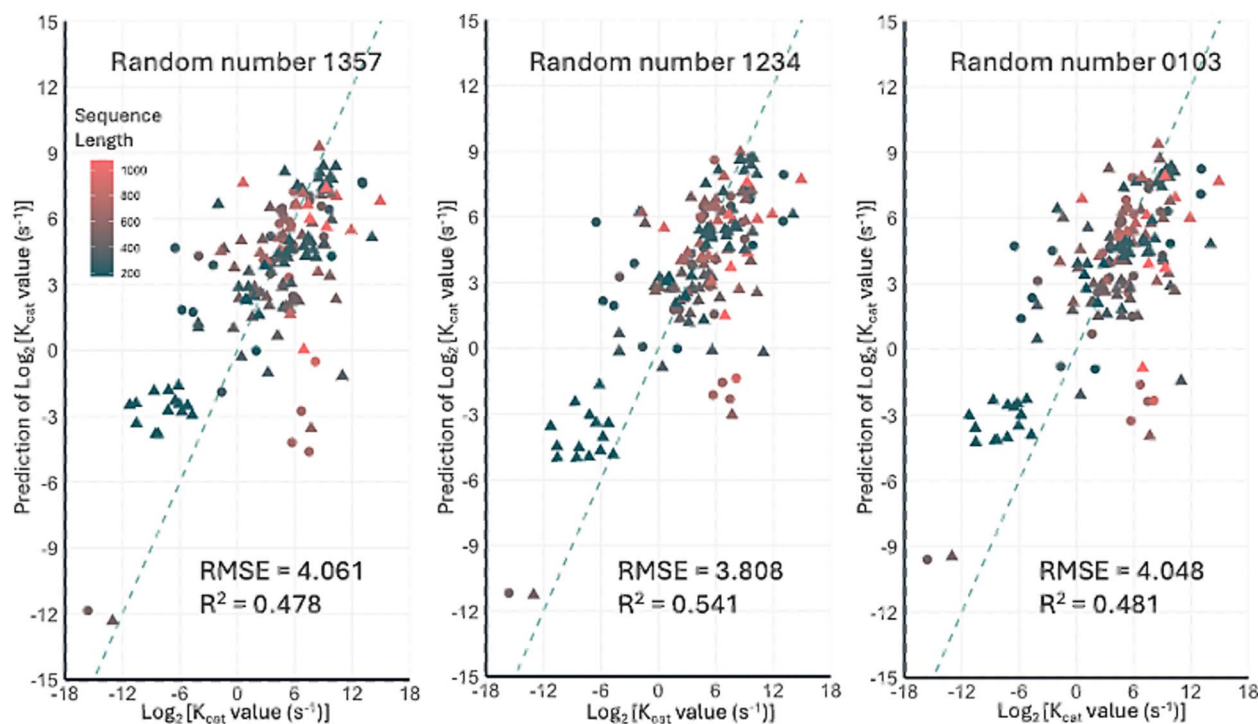


Figure 9. The performance of the three models on Dataset #2A for external validation. Sequences length ranging from 174 to 1074 amino acids are represented in different colors according to figure legend. The dash line represents the $Y=X$ trendline. A total of 122 records from groups a and B, which include new sequence inputs, are represented by circles (●). Records from group C, where all sequences have exact matches in dataset #1, are depicted as triangles (▲).

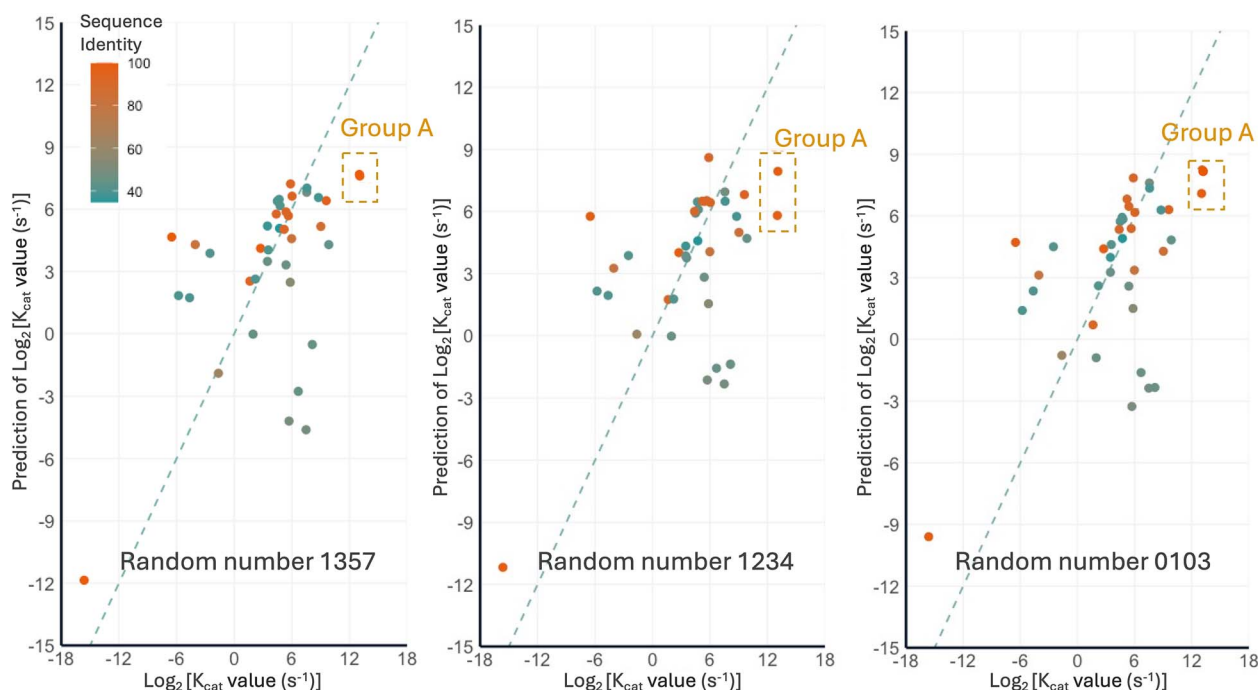


Figure 10. Application of three general-purpose models on Dataset #2A for records in group A and B. The sequence identity of a sequence in Dataset #2A and its most similar sequence in Dataset #1 was colored according to the figure legend. The sequence identity is from 34.6% to 100%.

We have successfully come up with a solution for a challenging issue in machine learning, i.e., data imbalance. Our solution is not limited to K_{cat} prediction, but also applicable in many other machine learning projects. The data imbalance issue is more critical in this project as there are many repeated protein sequences and substrates in the enzyme-substrate pairs, and the occurrence of repeated protein sequences and substrates is different from

one sequence to another or one substrate to another. Thus, it is very difficult to generate random datasets which all have similar occurrence distributions for those repeated protein sequences and substrates.

To efficiently address this issue, we preprocessed the protein sequences and substrates separately. By obtaining the mean K_{cat} value for an enzyme or a substrate as the characteristic value,

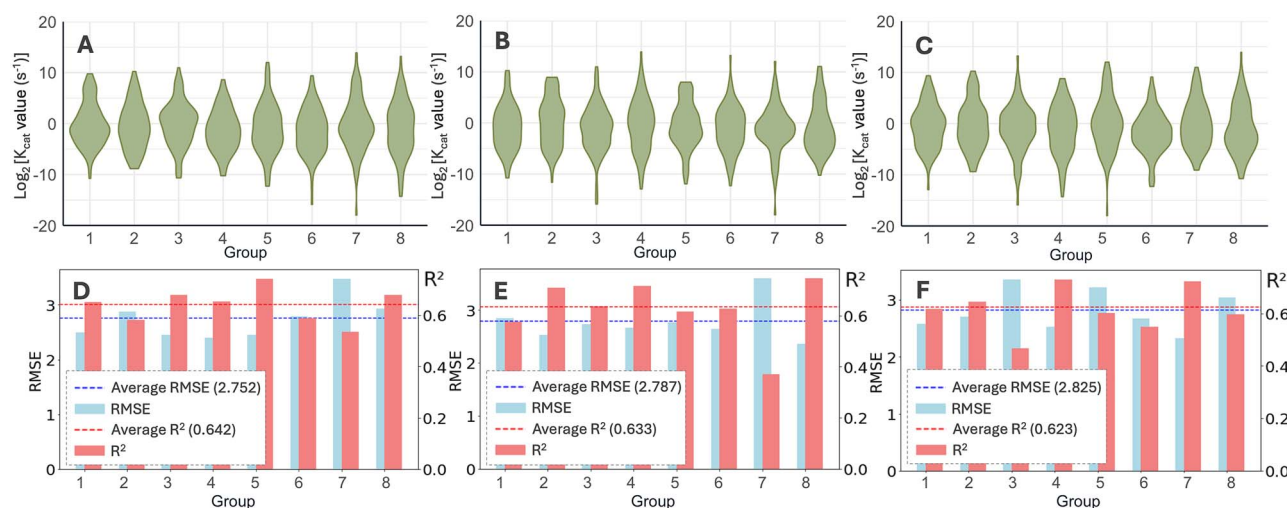


Figure 11. The performance of focused learning models from three parallel experiments. Each column represents an individual experiment. Panels A, B, and C display the distributions of K_{cat} values for the sublibrary which was randomly split into eight groups. Panels D, E, and F show the model performance under different data splitting conditions, with the axis label on the left for RMSE and the axis label on the right for R^2 . Each individual RMSE and R^2 value reflects the model performance when a specific group is used as the validation set in the leave-one-group-out approach. The average RMSE and R^2 values represent the summary statistics when each group is sequentially used as the validation set.

we were able to estimate the overall ability of each enzyme to catalyze reactions or the susceptibility of a substrate to enzyme reactions. The characteristic values served as the response of each enzyme or substrate during the construction of separate processors. This treatment effectively addressed the data imbalance issue in Dataset #1.

Building separate processors not only addressed the data imbalance issue but also increased the flexibility of our K_{cat} model. Each processor can translate an entry into a multi-dimensional vector that thoroughly characterizes it, with adjustable output dimensions to guarantee the general-purpose K_{cat} models not only accurate, but also robust. Additional descriptors can also be incorporated into current prediction model if those features are relevant to the catalytic reaction. In future work, we will develop additional descriptors to more comprehensively capture the properties of catalytic reactions, which can further enhance our model's performance.

Finally, our general-purpose K_{cat} prediction models exhibit stable performance on the validation set of Dataset #1 and external test set, Dataset #2A. Parallel experiments have demonstrated that our modeling protocol perfectly addressed a limitation of prior work that used the same dataset for model construction, and generated models whose performance was unaffected by random splits of the dataset. Additionally, using CYP450 enzymes as an example, we demonstrated that the general-purpose models can be refined for a specific enzyme category with significantly improved prediction accuracy. This focused learning strategy addressed another challenge in machine learning, data scarcity. One of the main challenges in machine learning projects is the availability of large, high-quality datasets for model training. It is known that available K_{cat} data for CYP450 enzymes are quite limited, making it difficult to develop highly reliable models purely relying on the small CYP450 dataset. This focused learning protocol can also be applied to other major enzyme categories collected by Dataset #1. By using the pretrained substrate and protein processors and fine-tuning the general-purpose models against a filtered subset of Dataset #1, we can efficiently adapt the models to specific needs. In recent years, physiologically based pharmacokinetics (PBPK) modeling plays a more and more

important role in drug development [42–46]. One needs enzymatic parameters including K_{cat} and K_m to construct a high-quality PBPK model. However, those metabolic parameters are frequently unavailable for drugs or drug candidates. Our model can assist by predicting these essential parameters, thus supporting both preclinical and clinical studies. Moreover, our models can benefit other research areas, including drug lead prioritization and drug candidate selection, and enzyme engineering.

Our future studies will investigate alternative approaches to construct the two types of processors. For example, chemical bidirectional transformers [47], represented by ChemBERT [48], can be applied to extract substrate information, while protein language models, represented by ProtBERT and ESM2 model [49, 50], can be applied to characterize the protein sequences. We will identify the combination which achieves the best model performance. In addition, a great effort will be put to curate K_{cat} data in the public databases, as we found that there are large discrepancies of the measured K_{cat} values of the same substrate-enzyme pairs in different databases. All the curated datasets from different sources will be merged to construct one of the largest K_{cat} datasets, to facilitate us to construct more robust models for K_{cat} prediction.

Conclusions

In this study, we successfully resolved two major challenges in enzymatic turn-over rate prediction, namely, data imbalance and data scarcity. For the first challenge, we separately constructed processors for enzyme proteins and substrates, to generate feature vectors for the construction of the general-purpose K_{cat} models. This treatment enabled us to generate highly robust prediction models for K_{cat} prediction, as the model performance is basically independent from random data splitting. Our K_{cat} models achieved the prediction R^2 of 0.538 on average, which is slightly better than the model constructed with the same database from recent publication. Moreover, our models are very robust as their performance is insensitive to the sequence length and sequence identity of the enzyme protein and whether the substrate structures exist in the training set or not. For the second

challenge, we utilized a focused learning strategy to fine tune the general-purpose models to enhance the model performance for a specific enzyme category. Indeed, our focused model for CYP450 enzymes achieved a very encouraging performance with an average R^2 of 0.633.

Methods

Dataset preparation

The overall dataset for model training, validation and external test are extracted from the publication of Li *et al.* [30]. To ensure comparability between their study and ours, records in the datasets were only reduced when necessary—primarily to remove duplicates and occasionally to exclude conflicting records within the dataset.

In this study, separate protein and substrate data libraries were obtained from Dataset #1 to construct separate processor models (Fig. 1). The separate libraries were created as follows: for each unique substrate or sequence in the original dataset, the average of all associated K_{cat} values was calculated. This averaged value served as the characteristic value for the substrate or sequence. The characteristic values were used as the response or the dependent variables in training the processor model.

Substrate processor: Attentive FP model for small molecules

The original attentive fingerprint model (AtFP) was proposed by Xiong *et al.* [51, 52], which was a state-of-art graph-based neural network modeling method aiming at the prediction of molecular characteristics [53–56]. We implemented the code from GitHub (<https://github.com/OpenDrugAI/AttentiveFP>) and constructed prediction models to process substrate input in SMILES string format. All the models were generated using the PyTorch package [57]. Model performances were evaluated by using RMSE and correlation coefficient square (R^2) between the characteristic values and model predicted values for substrates.

Protein processor: LSTM model for proteins

The protein sequence was processed using a long short-term memory (LSTM) model to generate 30 sequenced descriptive features. The LSTM model was built with a single layer and 128-dimension hidden layer by Python with package PyTorch [57]. The lengths of all input sequences were adjusted to the length of the longest sequence by padding zeros. Loss function was set to mean squared error (MSE) and stochastic gradient descent (SGD) was chosen as the optimizer. The learning rate and weight decay were set to 0.0001 and 0.011, respectively. Feature generation was constructed by applying a linear transformation to the tensor in hidden dimension, producing the specified number of descriptive features. Model performances were evaluated by using RMSE and R^2 between the characteristic values and model predicted values for protein sequences.

K_{cat} prediction model: Feature augmentation and model combination

The construction of general-purpose K_{cat} prediction models utilized the regression learner module in Matlab (version R2024a), while the top-ranking models were chosen as the general-purpose K_{cat} prediction models, whose performance was further validated by the external test set (Dataset #2A). All models were evaluated by calculating the RMSE and R^2 between the experimental K_{cat} values and the predicted ones for all substrate-enzyme pairs. To ensure the robustness of the generated models, we divided

Dataset #1 into training, validation and test sets using three different random numbers, with a splitting ratio of 8: 1: 1. The data splitting ratio is the same to the original publication by Li *et al.* [30]. To facilitate cross-study comparability, we listed the three random numbers below:

- (1) Random number 1234. This random number is derived from the publication by Li *et al.*, ensuring that the split of Dataset #1 matches exactly with the training, validation, and test splitting results reported in their study.
- (2) Random number 1357. This random number was used during our reproduction of Li *et al.*'s experiments, where we observed that the choice of random number significantly affected their model performance. When we replaced the random number with 1357 to split Dataset #1, their model's R^2 value decreased from 0.5 to 0.2.
- (3) Time-dependent random seed. This was a random order generated based on the timestamp during the experiment and was used for splitting Dataset #1. The generated order was recorded to enable reproducibility of the experimental results. In subsequent experiments, for ease of reference, this order was named "random number 0103" (R^2 for test set is 0.52 for Li's model under this order).

The evaluation of model performance

The performances of both independent substrate and protein processors, as well as the comprehensive K_{cat} prediction model, was assessed by overall RMSE and R^2 value between the characteristic/ K_{cat} values and the model predicted values as previously described, based on the equations below [58–60]:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum (y_{pred} - y_{obe})^2}$$

Where n is the number of records, y_{obe} represents the characteristic value for a single substrate, a protein entry or the K_{cat} value of a substrate-enzyme pair, and y_{pred} stands for the predicted value for each input/pair. R^2 will be calculated by the equation below:

$$R^2 = 1 - \frac{\sum (y_{obe} - y_{pred})^2}{\sum (y_{obe} - \bar{y})^2}$$

Where y_{obe} and y_{pred} represents the same with previous, \bar{y} is the mean of all y_{obe} values involved in respective calculation.

External test of K_{cat} prediction model

In Li *et al.*'s publication, they mentioned another dataset (Dataset #2) involving 343 yeast/fungi species, which they used for the external application of their K_{cat} prediction model. We also utilized this dataset as an external test set to critically validate our models. However, during our experiments, we discovered that many records in Dataset #2 were duplicated with entries in Dataset #1, with identical substrate-enzyme pairs and corresponding K_{cat} values. Therefore, we removed all duplicate records from Dataset #2, resulting in a clean external dataset (Dataset #2A), which allowed us to objectively evaluate the model performance [61].

We compared the sequences from Dataset #2A with the proteins contained in Dataset #1 to assess their identities. For those protein sequences only occurring in Dataset #2, we calculated its sequence identity to every sequence in Dataset #1. Sequence alignment and subsequent sequence identity calculations were performed using two programs: Muscle and Clustal [62–68]. Note that a short protein sequence in Dataset #1 which has fewer than 50 amino acid residues were neglected during the above sequence

comparison and sequence identity calculations. Sequence identity was calculated as the ratio of the number of identical amino acids at aligned positions to the number of all non-gap aligned positions [69–72]. The sequence identity value obtained for a pair of sequence alignment containing gaps >20% of the shorter sequence length was also discarded [73–75]. Subsequently, we collected and ranked all the sequence identity results with an aim to identify the most similar sequence in Dataset #1 for each protein sequence in Dataset #2A. To judge if two substrates are actually the same molecule, we converted all substrates into canonical SMILES strings prior to comparison [76, 77], as every substrate has a unique canonical SMILES string.

Focused learning

We applied the focused learning strategy to fine-tune the general-purpose K_{cat} prediction models obtained in the previous step to enhance the prediction accuracy for a specific enzyme category, such as Cytochromes P450. To conduct focused learning, one needed to first either extract a subset from Dataset #1 or prepare a separate dataset for the targeted enzyme category. Next, protein and substrate embeddings should be generated using the developed protein and substrate processors, respectively. Last, the new model was trained solely using the dataset of the enzyme category.

Key Points

- We proposed a novel protocol to construct K_{cat} prediction models, introducing an intermedia step to separately develop substrate and protein processors.
- The separation of the substrate processor and protein processor addressed the problem of data imbalance in the original dataset, caused by varying degrees of repetition of enzyme proteins and substrates when forming different pairs.
- The separation of the processors also provides greater flexibility in customizing the general-purpose K_{cat} prediction models to enhance the prediction accuracy for specific enzyme classes.
- The fine-tuned K_{cat} models for a specific enzyme category through focused learning address the challenge of data scarcity during the construction of prediction model for an individual enzyme category.
- The general-purposed K_{cat} prediction models, including the separate processors, are available for application and customization upon request.

Acknowledgments

The authors would like to thank the computing resources provided by the Center for Research Computing (facility RRID: SCR_022735) at the University of Pittsburgh (NSF award number OAC-2117681), and the Pittsburgh Supercomputer Center (grant number BIO210185).

Author contributions

J.W. and L.X. conceived the project and contributed to securing funding. J.Z. and X.Q. developed and trained the machine learning models and analyze the results. L.C. collected the data for model development. J.Z. and J.W. wrote the manuscript. All authors discussed the results and revised the manuscript.

Supplementary data

Supplementary data is available at Briefings in Bioinformatics online.

Conflict of interest: All authors declare no competing interests.

Funding

This work was supported by the following funds from the National Institutes of Health (NIH) and the National Science Foundation (NSF): H1H R01GM149705, NIH R01AG057555, and NSF 1955260.

Data and code availability

The data and source code for model construction, validation, dataset splitting, and the generation of substrate vectors and protein vectors in this study are available at GitHub (<https://github.com/jiczhh/NNKcat/tree/main>). A Windows program (`kcat_predict.exe`) was developed to predict K_{cat} values using the trained bagged decision tree ensemble model by Matlab.

References

1. Koshland DE Jr. The application and usefulness of the ratio k_{cat}/K_M . *Bioorg Chem* 2002;**30**:211–3. <https://doi.org/10.1006/bioo.2002.1246>
2. Eisenthal R, Danson MJ, Hough DW. Catalytic efficiency and k_{cat}/K_M : a useful comparator? *Trends Biotechnol* 2007;**25**:247–9. <https://doi.org/10.1016/j.tibtech.2007.03.010>
3. Lorsch JR. Methods in Enzymology. Laboratory methods in enzymology: protein part A. Preface *Methods Enzymol* 2014;**536**:xv. <https://doi.org/10.1016/B978-0-12-420070-8.09988-8>
4. Carrillo N, Ceccarelli E, Roveri O. Usefulness of kinetic enzyme parameters in biotechnological practice. *Biotechnol Genet Eng Rev* 2010;**27**:367–82. <https://doi.org/10.1080/02648725.2010.10648157>
5. Yu H, Deng H, He J. et al. UniKP: a unified framework for the prediction of enzyme kinetic parameters. *Nat Commun* 2023;**14**:8211. <https://doi.org/10.1038/s41467-023-44113-1>
6. Renard M, Fersht AR. Anomalous pH dependence of k_{cat}/K_M in enzyme reactions. Rate constants for the association of chymotrypsin with substrates. *Biochemistry* 1973;**12**:4713–8. <https://doi.org/10.1021/bi00747a026>
7. Davidi D, Noor E, Liebermeister W. et al. Global characterization of in vivo enzyme catalytic rates and their correspondence to in vitro k_{cat} measurements. *Proc Natl Acad Sci USA* 2016;**113**:3401–6. <https://doi.org/10.1073/pnas.1514240113>
8. Ou Y, Wilson RE, Weber SG. Methods of measuring enzyme activity ex vivo and In vivo. *Annu Rev Anal Chem (Palo Alto Calif)* 2018;**11**:509–33. <https://doi.org/10.1146/annurev-anchem-061417-125619>
9. Boeckx J, Hertog M, Geeraerd A. et al. Kinetic modelling: an integrated approach to analyze enzyme activity assays. *Plant Methods* 2017;**13**:69. <https://doi.org/10.1186/s13007-017-0218-y>
10. Testore G, Colombatto S, Silvagno F. et al. Purification and kinetic characterization of γ -aminobutyraldehyde dehydrogenase from rat liver. *Int J Biochem Cell Biol* 1995;**27**:1201–10. [https://doi.org/10.1016/1357-2725\(95\)00075-Z](https://doi.org/10.1016/1357-2725(95)00075-Z)
11. Paragas EM, Choughule K, Jones JP. et al. Enzyme kinetics, pharmacokinetics, and inhibition of aldehyde oxidase. *Enzyme Kinetics Drug Metabol Fundamentals Appl* 2021;**2342**:257–84. https://doi.org/10.1007/978-1-0716-1554-6_10

12. Masimirembwa CM, Otter C, Berg M. et al. Heterologous expression and kinetic characterization of human cytochromes P-450: validation of a pharmaceutical tool for drug metabolism research. *Drug Metab Dispos* 2018;**27**:1117–22. [https://doi.org/10.1016/S0090-9556\(24\)15034-9](https://doi.org/10.1016/S0090-9556(24)15034-9)
13. Parés X, Vallee BL. New human liver alcohol dehydrogenase forms with unique kinetic characteristics. *Biochem Biophys Res Commun* 1981;**98**:122–30. [https://doi.org/10.1016/0006-291X\(81\)91878-7](https://doi.org/10.1016/0006-291X(81)91878-7)
14. Kellner DG, Hung S-C, Weiss KE. et al. Kinetic characterization of compound I formation in the thermostable cytochrome P450 CYP119. *J Biol Chem* 2002;**277**:9641–4. <https://doi.org/10.1074/jbc.C100745200>
15. Ge G-B, Ning J, Hu LH. et al. A highly selective probe for human cytochrome P450 3A4: isoform selectivity, kinetic characterization and its applications. *Chem Commun* 2013;**49**:9779–81. <https://doi.org/10.1039/c3cc45250f>
16. Kirkwood L, Nation R, Somogyi A. Characterization of the human cytochrome P450 enzymes involved in the metabolism of dihydrocodeine. *Br J Clin Pharmacol* 1997;**44**:549–55. <https://doi.org/10.1046/j.1365-2125.1997.t01-1-00626.x>
17. Denisov IG, Grinkova YV, Baas BJ. et al. The ferrous-dioxygen intermediate in human cytochrome P450 3A4: substrate dependence of formation and decay kinetics. *J Biol Chem* 2006;**281**:23313–8. <https://doi.org/10.1074/jbc.M605511200>
18. Gallagher EP, Kunze KL, Stapleton PL. et al. The kinetics of aflatoxin B1 oxidation by human cDNA-expressed and human liver microsomal cytochromes P450 1A2 and 3A4. *Toxicol Appl Pharmacol* 1996;**141**:595–606. <https://doi.org/10.1006/taap.1996.0326>
19. Shabtai Y, Jubran H, Nassar T. et al. Kinetic characterization and regulation of the human retinaldehyde dehydrogenase 2 enzyme during production of retinoic acid. *Biochem J* 2016;**473**:1423–31. <https://doi.org/10.1042/BCJ20160101>
20. Zhong Y-S, Kong QH, Wang J. et al. Discovery and enzyme kinetic characterization of novel CYP2D6 variants. *Chem Res Toxicol* 2024;**37**:1903–10. <https://doi.org/10.1021/acs.chemrestox.4c00298>
21. Boorla, V. S. & Maranas, C. D. CatPred: A comprehensive framework for deep learning in vitro enzyme kinetic parameters kcat, Km and Ki. *bioRxiv*, 2024.2003.2010.584340 (2024).
22. Borger S, Liebermeister W, Klipp E. Prediction of enzyme kinetic parameters based on statistical learning. *Genome Inform* 2006;**17**:80–7.
23. Galanakis CM, Patsioura A, Gekas V. Enzyme kinetics modeling as a tool to optimize food industry: a pragmatic approach based on amylolytic enzymes. *Crit Rev Food Sci Nutr* 2015;**55**:1758–70. <https://doi.org/10.1080/10408398.2012.725112>
24. Boyacı İH. A new approach for determination of enzyme kinetic constants using response surface methodology. *Biochem Eng J* 2005;**25**:55–62. <https://doi.org/10.1016/j.bej.2005.04.001>
25. Vasic-Racki D, Kragl U, Liese A. Benefits of enzyme kinetics modelling. *Chem Biochem Eng Q* 2003;**17**:7–18.
26. Seibert E, Tracy TS. Different enzyme kinetic models. *Enzyme Kinetics Drug Metabolism Fundamentals Appl* 2014;**1113**:23–35. https://doi.org/10.1007/978-1-62703-758-7_3
27. Gabdoulline RR, Stein M, Wade RC. qPIPSA: relating enzymatic kinetic parameters and interaction fields. *BMC Bioinform* 2007;**8**:1–16. <https://doi.org/10.1186/1471-2105-8-373>
28. Gollub MG, Backes T, Kaltenbach H-M. et al. ENKIE: a package for predicting enzyme kinetic parameter values and their uncertainties. *Bioinformatics* 2024;**40**:btae652. <https://doi.org/10.1093/bioinformatics/btae652>
29. Shen X, Cui Z, Long J. et al. EITLEM-kinetics: a deep-learning framework for kinetic parameter prediction of mutant enzymes. *Chem Catalysis* 2024;**4**:101094. <https://doi.org/10.1016/j.checat.2024.101094>
30. Li F, Yuan L, Lu H. et al. Deep learning-based kcat prediction enables improved enzyme-constrained model reconstruction. *Nature Catalysis* 2022;**5**:662–72. <https://doi.org/10.1038/s41929-022-00798-z>
31. Kroll, A. & Lercher, M. J. DLKcat cannot predict meaningful k_{cat} values for mutants and unfamiliar enzymes. *bioRxiv* 2024;2023.2002.2006.526991. <https://doi.org/10.1101/2023.02.06.526991>
32. Kroll A, Lercher MJ. Machine learning models for the prediction of enzyme properties should be tested on proteins not used for model training. *bioRxiv* 2023;2023.2002.2006.526991. <https://doi.org/10.1101/2023.02.06.526991>
33. Heckmann D, Lloyd CJ, Mih N. et al. Machine learning applied to enzyme turnover numbers reveals protein structural correlates and improves metabolic models. *Nat Commun* 2018;**9**:5252. <https://doi.org/10.1038/s41467-018-07652-6>
34. Kroll A, Rousset Y, Hu X-P. et al. Turnover number predictions for kinetically uncharacterized enzymes using machine and deep learning. *Nat Commun* 2023;**14**:4139. <https://doi.org/10.1038/s41467-023-39840-4>
35. Wang J, Yang Z, Chen C. et al. MPEK: a multitask deep learning framework based on pretrained language models for enzymatic reaction kinetic parameters prediction. *Brief Bioinform* 2024;**25**:bbad506. <https://doi.org/10.1093/bib/bbae387>
36. Qiu S, Zhao S, Yang A. DLTKcat: deep learning-based prediction of temperature-dependent enzyme turnover rates. *Brief Bioinform* 2024;**25**:bbad506. <https://doi.org/10.1093/bib/bbad506>
37. Wang T, Xiang G, He S. et al. DeepEnzyme: a robust deep learning model for improved enzyme turnover number prediction by utilizing features of protein 3D-structures. *Brief Bioinform* 2024;**25**:bbae409. <https://doi.org/10.1093/bib/bbae409>
38. Boorla VS, Maranas CD. CatPred: a comprehensive framework for deep learning in vitro enzyme kinetic parameters. *Nat Commun* 2025;**16**:2072. <https://doi.org/10.1038/s41467-025-57215-9>
39. Tian W, Skolnick J. How well is enzyme function conserved as a function of pairwise sequence identity? *J Mol Biol* 2003;**333**:863–82. <https://doi.org/10.1016/j.jmb.2003.08.057>
40. Shen, R., Bubeck, S. & Gunasekar, S. In *International conference on machine learning*. 19773–808 (PMLR).
41. Ye J, Borovykh A, Hayou S. et al. Leave-one-out distinguishability in machine learning. *arXiv preprint arXiv:2309.17310*. 2023.
42. Zhuang X, Lu C. PBPK modeling and simulation in drug research and development. *Acta Pharmaceutica Sinica B* 2016;**6**:430–40. <https://doi.org/10.1016/j.apsb.2016.04.004>
43. Zhao P, Zhang L, Grillo JA. et al. Applications of physiologically based pharmacokinetic (PBPK) modeling and simulation during regulatory review. *Clin Pharmacol Therapeut* 2011;**89**:259–67. <https://doi.org/10.1038/clpt.2010.298>
44. Kostewicz ES, Aarons L, Bergstrand M. et al. PBPK models for the prediction of in vivo performance of oral dosage forms. *Eur J Pharm Sci* 2014;**57**:300–21. <https://doi.org/10.1016/j.ejps.2013.09.008>
45. Peters, S. A. *Physiologically Based Pharmacokinetic (PBPK) Modeling and Simulations: Principles, Methods, and Applications in the Pharmaceutical Industry*. Hoboken, NJ: John Wiley & Sons, 2021. <https://doi.org/10.1002/9781119497813>
46. Jones HM, Mayawala K, Poulin P. Dose selection based on physiologically based pharmacokinetic (PBPK) approaches. *AAAPS J* 2013;**15**:377–87. <https://doi.org/10.1208/s12248-012-9446-2>
47. Hayes N, Merkurjev E, Wei G-W. Graph-based bidirectional transformer decision threshold adjustment algorithm for

- class-imbalanced molecular data. *J Comput Biophys Chem* 2024;**23**:1339–58. <https://doi.org/10.1142/s2737416524500479>
48. Wu Z, Jiang D, Wang J. et al. Knowledge-based BERT: a method to extract molecular features like computational chemists. *Brief Bioinform* 2022;**23**:bbac131. <https://doi.org/10.1093/bib/bb ac131>
 49. Lin Z, Akin H, Rao R. et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* 2023;**379**:1123–30. <https://doi.org/10.1126/science.ade 2574>
 50. Zhang Z, Wayment-Steele HK, Brixi G. et al. Protein language models learn evolutionary statistics of interacting sequence motifs. *Proc Natl Acad Sci* 2024;**121**:e2406285121. <https://doi.org/10.1073/pnas.2406285121>
 51. Xiong Z, Wang D, Liu X. et al. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *J Med Chem* 2020;**63**:8749–60. <https://doi.org/10.1021/acs.jmedchem.9b00959>
 52. Cai H, Zhang H, Zhao D. et al. FP-GNN: a versatile deep learning architecture for enhanced molecular property prediction. *Brief Bioinform* 2022;**23**:bbac408. <https://doi.org/10.1093/bib/bb ac408>
 53. Lei, Y., Hu, J., Zhao, Z. & Ye, S. In *International Conference on Intelligent Computing*. 507–16 (Springer).
 54. Jiang D, Wu Z, Hsieh CY. et al. Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models. *J Chem* 2021;**13**:1–23. <https://doi.org/10.1186/s13321-020-00479-8>
 55. Zhang Z, Guan J, Zhou S. FraGAT: a fragment-oriented multi-scale graph attention model for molecular property prediction. *Bioinformatics* 2021;**37**:2981–7. <https://doi.org/10.1093/bioinformatics/btab195>
 56. Wu J, Wang J, Wu Z. et al. ALipSol: an attention-driven mixture-of-experts model for lipophilicity and solubility prediction. *J Chem Inf Model* 2022;**62**:5975–87. <https://doi.org/10.1021/acs.jcim.2c01290>
 57. Paszke A, Gross S, Massa F. et al. Pytorch: an imperative style, high-performance deep learning library. *Adv Neural Inform Processing Syst* 2019;**721**:8026–37. <https://dl.acm.org/doi/10.5555/3454287.3455008>
 58. Chicco D, Warrens MJ, Jurman G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Comput Sci* 2021;**7**:e623. <https://doi.org/10.7717/peerj-cs.623>
 59. Chai T, Draxler RR. Root mean square error (RMSE) or mean absolute error (MAE)?—arguments against avoiding RMSE in the literature. *Geosci Model Dev* 2014;**7**:1247–50. <https://doi.org/10.5194/gmd-7-1247-2014>
 60. Hodson TO. Root mean square error (RMSE) or mean absolute error (MAE): when to use them or not. *Geosci Model Develop Discuss* 2022;**2022**:1–10.
 61. Schuurmann G, Ebert R-U, Chen J. et al. External validation and prediction employing the predictive squared correlation coefficient test set activity mean vs training set activity mean. *J Chem Inf Model* 2008;**48**:2140–5. <https://doi.org/10.1021/ci800253u>
 62. Sievers F, Higgins DG. Clustal omega, accurate alignment of very large numbers of sequences. *Multiple Seq Alignment Methods* 2014;**1079**:105–16. https://doi.org/10.1007/978-1-62703-646-7_6
 63. Sievers F, Higgins DG. The clustal omega multiple alignment package. *Multiple Seq Alignment: Methods Protocols* 2021;**2231**:3–16. https://doi.org/10.1007/978-1-0716-1036-7_1
 64. Griffin AM, Griffin HG, Higgins DG. CLUSTAL V: multiple alignment of DNA and protein sequences. *Comput Anal Seq Data: Part II* 1994;**25**:307–18. <https://doi.org/10.1385/0896032760>
 65. Edgar RC. MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinform* 2004;**5**:1–19. <https://doi.org/10.1186/1471-2105-5-113>
 66. Edgar RC. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res* 2004;**32**:1792–7. <https://doi.org/10.1093/nar/gkh340>
 67. Elzinga, C. H. In: Blanchard P, Buhlmann F, Gauthier J-A (eds.), *Advances in Sequence Analysis: Theory, Method, Applications* 51–73. Springer, 2014. https://doi.org/10.1007/978-3-319-04969-4_4
 68. Borozan I, Watt S, Ferretti V. Integrating alignment-based and alignment-free sequence similarity measures for biological sequence classification. *Bioinformatics* 2015;**31**:1396–404. <https://doi.org/10.1093/bioinformatics/btv006>
 69. Kosloff M, Kolodny R. Sequence-similar, structure-dissimilar protein pairs in the PDB. *Proteins: Struct Func Bioinform* 2008;**71**:891–902. <https://doi.org/10.1002/prot.21770>
 70. Zhang, Y. & Yu, X. in 2010 *IEEE fifth international conference on bio-inspired computing: theories and applications (BIC-TA)*. 1255–8 (IEEE).
 71. Jovanovic JT. New method for sequence similarity analysis based on the position and frequency of statistically significant repeats. *Curr Bioinform* 2021;**16**:1299–310. <https://doi.org/10.2174/1574893616999210805165628>
 72. Munjal G, Sharma P, Gaur D. Sequence similarity using composition method. *Int J Data Sci* 2018;**3**:19–28. <https://doi.org/10.1504/IJDS.2018.10011822>
 73. Jin X, Jiang Q, Chen Y. et al. Similarity/dissimilarity calculation methods of DNA sequences: a survey. *J Mol Graph Model* 2017;**76**:342–55. <https://doi.org/10.1016/j.jmgm.2017.07.019>
 74. Pearson WR. Empirical statistical estimates for sequence similarity searches. *J Mol Biol* 1998;**276**:71–84. <https://doi.org/10.1006/jmbi.1997.1525>
 75. Triandini, E., Fauzan, R., Siahaan, D. O. & Rochimah, S. in 2019 *16th International Joint Conference on Computer Science and Software Engineering (IJCSSE)*. 348–51 (IEEE).
 76. O'Boyle NM. Towards a universal SMILES representation—a standard method to generate canonical SMILES based on the InChI. *J Chem* 2012;**4**:1–14. <https://doi.org/10.1186/1758-2946-4-22>
 77. Bjerrum EJ. SMILES enumeration as data augmentation for neural network modeling of molecules. arXiv preprint arXiv:1703.07076. 2017.