nature communications

Article

https://doi.org/10.1038/s41467-025-57215-9

CatPred: a comprehensive framework for deep learning in vitro enzyme kinetic parameters

Received: 25 March 2024

Accepted: 14 February 2025

Published online: 28 February 2025

Check for updates

Veda Sheersh Boorla [1,2] & Costas D. Maranas [1,2]

Estimation of enzymatic activities still heavily relies on experimental assays, which can be cost and time-intensive. We present CatPred, a deep learning framework for predicting in vitro enzyme kinetic parameters, including turnover numbers (k_{cat}), Michaelis constants (K_m), and inhibition constants (K_i). CatPred addresses key challenges such as the lack of standardized datasets, performance evaluation on enzyme sequences that are dissimilar to those used during training, and model uncertainty quantification. We explore diverse learning architectures and feature representations, including pretrained protein language models and three-dimensional structural features, to enable robust predictions. CatPred provides accurate predictions with query-specific uncertainty estimates, with lower predicted variances correlating with higher accuracy. Pretrained protein language model features particularly enhance performance on out-of-distribution samples. *CatPred* also introduces benchmark datasets with extensive coverage (~23 k, 41 k, and 12 k data points for k_{cat}, K_m, and K_i respectively). Our framework performs competitively with existing methods while offering reliable uncertainty quantification.

Recent artificial intelligence (AI) algorithms are emerging as promising tools for the automated assignment of functions to uncharacterized proteins[1]. These models offer the promise of high-quality automated functional annotation of sequenced genomes[1–3]. Recently developed methods such as CLEAN[3], DeepECtransformer[4], and ProteInfer[2] have enabled accurate Enzyme Commission (EC) number recapitulation by leveraging pretrained protein Language Models[5,6] (pLM) and deep learning algorithms. However, quantification of enzyme activity is still largely dependent on costly and time-consuming biochemical assays. Such approaches cannot keep up with the pace of sequence discovery[7], leaving most computationally identified enzymes uncharacterized in terms of their kinetics, despite some progress in high-throughput screening capacity[8,9]. Therefore, predictive models that enable quantitative annotation of enzyme kinetics could be enabling for enzyme characterization in the same manner that recent fold prediction algorithms[5,10] have become for structure prediction. Even approximate estimates of enzyme kinetics on a given substrate can be very important for a diversity of tasks ranging from starting point enzyme selection in directed evolution for protein engineering[11,12], biosynthetic or biodegradation pathway pre-screening[13,14], or initialization in the parameterization of kinetic models of metabolism[15]. Enzyme engineering efforts often rely on evolutionary methods such as directed evolution that aim to ratchet up enzyme activity and/or selectivity. The selection process of the starting enzyme that undergoes directed evolution can be informed based on computationally derived enzyme kinetic estimates. De novo enzyme kinetic parameter prediction can also inform pathway assembly algorithms[16] aimed at designing entire retro-biosynthetic routes for biochemical synthesis. Kinetic parameter predictions can be used to avoid alternatives with poor enzyme turnover or enzymes that exhibit strong product inhibition, accelerating the discovery of more catalytically efficient routes. Finally, kinetic models, by relating enzyme kinetics to the concentration of metabolites and enzyme levels within a cell, can be used to both describe and redesign metabolism[17]. Advances in automated

[1]Department of Chemical Engineering, The Pennsylvania State University, University Park, PA 16802, USA. [2]The Center for Bioenergy Innovation, Oak Ridge, TN 37830, USA. e-mail: cdm8@psu.edu

functional annotation of proteins have enabled building metabolic models with a genome-wide coverage of cellular metabolism[18,19]. However, efficient kinetic parameterization to match observed fluxomic, proteomic and/or metabolomic datasets remains a bottleneck[20]. The use of reliable estimates for in vitro enzyme kinetic properties could accelerate convergence by serving as initializations of enzyme parameters[21]. These are but a handful out of the many applications that reliable enzyme parameter prediction could impact.

The catalytic turnover number and the Michaelis constant are key parameters of the Michaelis-Menten kinetics, which is a universally accepted model for quantitative assessment of enzyme function[22]. The turnover number, k_{cat}, is the *speed* of an enzyme, the maximal number of molecules of substrates converted to products per active site per unit time. The Michaelis constant, K_m, is equivalent to the concentration of a substrate at which the enzyme operates at half of its maximum catalytic rate, qualitatively describing the binding affinity between the enzyme-substrate pair. Since enzymes have evolved to cater to a wide array of cellular functions, they catalyze diverse chemical transformations and hence operate with a broad range of k_{cat} and K_m values[23]. In the presence of competitive or non-competitive inhibitors, the equivalent value of K_m can be obtained using inhibition constants (K_i). Databases such as BRENDA[24] and SABIO-RK[25] contain hundreds of thousands of in vitro kinetic measurements manually curated from primary research literature (Supplementary Table 1).

Several previous studies have focused on developing ML models for k_{cat} and K_m prediction by using these database entries as training data[26–29]. Li et al.[27] developed DLKcat by training a deep learning model on a dataset of 16,838 k_{cat} values of both natural and engineered enzymes across various species. They used a convolutional neural network (CNN) architecture to extract features of enzyme-sequence motifs and a graph neural network (GNN) to extract substrate features using their 2-dimensional (2-D) connectivity graphs. Kroll et al. trained a gradient-boosted tree model, TurNup[26], using language model features of enzymes' amino acid sequences along with reaction fingerprints for k_{cat} prediction using a dataset of 4,271 k_{cat} measurements. Although TurNup was trained on a much smaller dataset, they achieved better generalizability compared to DLKcat on test enzyme sequences dissimilar to training sequences (out-of-distribution test examples)[26]. More recently, Yu et al. developed *UniKP*[29] for ML prediction of k_{cat}, K_m, and k_{cat}/K_m values by training on previously curated datasets[27,28]. They trained a tree-ensemble regression model by utilizing pre-trained language models[6] for extracting features of both enzymes and substrates. UniKP demonstrated improved performance for k_{cat} prediction compared to DLKcat on in-distribution tests; however, no out-of-distribution examples were tested. Currently, TurNup is the only prediction framework that is systematically evaluated on out-of-distribution tests for k_{cat} prediction and outperforms DLKcat in this aspect. Predictive models that display good performance on enzyme sequences that are under-represented in the training datasets require that the models have learnt "generalizable patterns" instead of simply "memorizing" the nuances/noise present in the training data. These learnt patterns are encoded within the mathematically transformed representations of the input features. Often, these transformed feature representations (called latent spaces) can provide visualization cues through projection in 2D or 3D graphs[30].

Unlike k_{cat} values that are not directly relatable to the physical properties of the substrate, K_m values have been shown to be correlated with their molecular mass and hydrophobicity[31]. Kroll et al.[28] developed a K_m prediction model using a gradient-boosted tree algorithm by training on 11,675 in vitro measurements of natural enzyme-metabolite pairs. They used UniRep[32] for extracting enzyme features and a task-specific graph neural network derived fingerprints combined with the molecular mass and hydrophobicity properties as features for metabolites. UniRep is a Recurrent Neural Network model which was trained in an unsupervised fashion to perform next amino-acid prediction[32], akin to the recently emerged pLMs[5,6,33]. By training on large datasets of protein sequences (10-100 million or more), pLMs can predict the patterns and relationships within amino acid sequences[6]. Thus, pretrained pLMs can be employed to convert amino acid sequences into their numerical representations (also called features) which can effectively capture the complex, contextual information embedded in protein sequences. These numerical representations have been shown to encode both local and global sequence patterns and have enabled accurate prediction of protein properties and functions[34]. Yu et al.[29] trained a K_m prediction model within the UniKP framework using the training dataset curated in Kroll et al.[28] utilizing a pLM, ProtT5[6] for extracting enzyme sequence features. They demonstrated a similar performance to Kroll et al.[29]. Notably, both these existing models for K_m prediction are only evaluated on in-distribution sequences (i.e., test enzyme sequences that are not explicitly excluded from those of training datasets). Relatively fewer ML models are available for K_i prediction of enzyme-inhibitor pairs. However, such methods have been developed with the focus of predicting IC50 (the concentration of an inhibitor required to reduce the rate of an enzymatic reaction by 50%) values of drug-target pairs[35,36].

Existing studies for machine learning in vitro k_{cat} and K_m values either use BRENDA[24], SABIO-RK[25], UniProt[7] or a combination of these to curate their training datasets from known measurements of kinetic parameters. However, there is a lack of complete annotations in the databases for all entries leaving significant gaps in the amount of learnable data. For example, even though there exist about 87 k, 176 k, and 46 k entries for k_{cat}, K_m, and K_i measurements, respectively, in BRENDA (Release 2022_2), many are not annotated with the corresponding enzyme sequences and/or substrate information. Owing to this, training datasets used by existing works vary significantly depending on how they handle entries with missing information. This has prompted most studies to use small, filtered subsets of the available data to mitigate this effect. For example, *TurNup* for k_{cat} prediction is trained only on 1,192 enzyme types (unique EC numbers) while the current biochemical databases contain k_{cat} values for over 3,000 enzyme types (Supplementary Table 2). Many studies have also imposed arbitrary exclusion criteria with the goal of reducing the effect of noisy measurements[26,28]. While such filtering may in part reduce the effect of noise, it could also potentially lead to information loss, biasing, and overfitting to the training datasets, especially when high-dimensional deep learning architectures are used. Filtered-out entries often correspond to infrequently occurring metabolite entries. Since they correspond to a large fraction (i.e., up to ~ 40-70%, Supplementary Table 3) of available data entries, their omission can become a missed opportunity for ML algorithms to learn on rarely seen data. Another notable source of incongruency between different datasets is the mapping process adopted of substrate names to their respective chemical connectivity information using SMILES[37] strings. Existing studies use either of, or a combination of PubChem[38], KEGG[39] or ChEBI[40] databases to map substrate names to the respective database identifiers and subsequently retrieve SMILES strings. Note that the same chemical entities can have different common names in different databases[41] and interconversion among databases can result in incorrect SMILES mapping. Such inconsistencies might lead to divergent results in some cases, thus precluding a fair comparison across studies. This motivates the need for both systematic data curation pipelines and standardized training datasets with expanded enzyme and substrate scope.

Existing ML models for k_{cat} or K_m prediction used traditional regression approaches by minimizing the mean-squared error between training data and thus output deterministic (single-valued) enzyme parameter predictions. These predictions lack any confidence metric information. In contrast, probabilistic regression approaches have the potential to offer guardrails on the reliability of predictions. Specifically, such approaches help draw inferences on prediction fidelity by estimating two classes of predictive uncertainties. One such class stems from the inherent observational noise in training data (termed aleatoric uncertainty), and the other results from the lack of training samples in each region of input space (called epistemic uncertainty). Methods that provide quantification of these uncertainties further fall into two categories Bayesian and Ensemble-based approaches. Bayesian ML algorithms such as Bayesian Neural Networks and Gaussian processes inherently output predictions as Gaussian distributions (including a mean and a variance). The mean values are treated as the model predictions, while the corresponding variances are a measure of the uncertainty (only aleatoric or both aleatoric and epistemic) in the corresponding predictions. On the other hand, ensemble-based approaches can only quantify epistemic uncertainty. This involves training identical copies of an ML model using different starting weights, and the trained models together form an ensemble model. The predictions from the ensemble members agree with each other closely or diverge depending on whether the corresponding input regions are well-represented in the training datasets or not, respectively. Uncertainty Quantification (UQ) approaches are especially useful when the experiments are time and cost prohibitive or when false-positive predictions are undesirable. For example, Hie et al.[42] used a Gaussian-process based UQ to discover several kinase inhibitors with nanomolar binding affinities. Importantly, they found that top-ranking predictions with uncertainty filters showed significantly better experimental binding affinities compared to predictions ranked without any UQ. ML models trained on noisy datasets (such as enzyme kinetic parameters) can lead to potentially unreliable predictions, especially when challenged with inputs significantly different from those that the model is trained on. This issue can partly be ameliorated by using UQ methods to offer a measure of confidence in model predictions. UQ methods have also been widely explored in the molecular property prediction domain where similar challenges with datasets exist[43].

Here we introduce a comprehensive ML framework, CatPred, for enzyme kinetic parameter prediction that addresses many of the aforementioned challenges. We first assembled an expanded set of benchmark datasets, CatPred-DB, for training and evaluating ML models using in vitro kinetic measurements of wild-type enzymes. The datasets of k_{cat}, K_m and K_i values were extracted from both BRENDA and SABIO-RK databases. Using these datasets, we train deep learning models utilizing features of different levels of complexity − enzyme sequence level (using sequence-attention and pLM features), and enzyme structure level equivariant graph neural network (E-GNN)[44] derived features. The sequence attention module allows the model to focus on specific regions within the enzyme sequence that maximally contribute towards predicting the kinetic parameters. Even though this module can learn features of individual enzyme sequences, it would not capture any enzyme relatedness connections that the pLM module can contribute. Finally, we also explored if adding 3D structure features of enzymes (using E-GNNs) can boost prediction performance. Substrate representation in CatPred relies on the directed message passing neural network (D-MPNN) approach. D-MPNNs have previously shown to be promising for a wide range of molecular property prediction tasks[45]. By leveraging an ensemble of probabilistic regression models[43] that simultaneously output means and variances of predictions, CatPred provides confidence estimates (including the contributions of both aleatoric and epistemic uncertainty) to its predictions. We systematically evaluated the predictive performances of CatPred on test datasets containing both in-distribution and out-of-distribution enzyme sequences (different from sequences encountered during training). Our results show that pLM derived features are necessary for achieving good predictive performances on out-of-distribution enzymes. CatPred performs favorably in a range of benchmarks compared to existing approaches while also offering uncertainty quantifications to its predictions.

Results

CatPred-DB: Benchmark datasets for machine learning in vitro enzyme kinetic parameters

CatPred-DB consists of a set of comprehensive benchmark datasets for training ML models, one each for k_{cat}, K_m and K_i in vitro measurements of wild-type enzymes. We curated these datasets from the BRENDA release 2022_2 and data from the SABIO-RK as of November 2023 (Fig. 1a). Initially, we parsed the databases to identify entries containing essential information, including at least one kinetic parameter value (k_{cat}, K_m, or K_i), the enzyme type (EC number), the organism of the enzyme's origin, and the names of reactants and products. To maintain the accuracy of organisms' names, we retain entries only if they are listed in the NCBI Taxonomy database[46]. We then mapped each entry to the enzyme's amino acid sequence identifier using the UniProt database (refer to "Dataset curation" section of Methods). We excluded entries that lack one or more of these annotations or if any of these annotations are incomplete. Finally, each substrate name is used to obtain a canonical SMILES string that corresponds to the 2D atom connectivity. For k_{cat} entries, all listed reactants are used to obtain a concatenated canonical SMILES string (see "Dataset curation" section in Methods).

If multiple measurements exist for any parameter belonging to an enzyme-sequence and substrate-SMILES pair, then the maximum (for k_{cat}) and the geometric mean (for K_m and K_i) value, respectively is retained. The selection of the maximum value for k_{cat} is carried out because it likely maps to the optimal growth conditions (i.e., temperature, pH, etc.). In contrast, K_m and K_i values are more directly associated with the enzyme-substrate/inhibitor affinities rather than the experimental conditions. The use of the geometric mean implies arithmetic averaging of the logarithmically transformed values used in the training process. The selection of a single value for the enzymatic parameters (for a given sequence and substrate-SMILES pair) is needed to safeguard against the ML method attempting to learn significantly different outputs for the same inputs, which can result in instabilities during training.

CatPred-DB contains 23,197 k_{cat}, 41,174 K_m and 11,929 K_i measurements spanning thousands of unique enzymes, organisms, and substrates (Table 1). Each entry in CatPred-DB is also mapped to a predicted 3D structure of the corresponding enzyme using AlphaFold-2.0 database[10]. In the absence of a 3D structure in the AlphaFold database, we used ESMFold[5] to carry out structure prediction. The coverage statistics of CatPred-DB contrasted with other efforts[27–29] are summarized in Table 1. Notably, CatPred-DB has a significantly expanded enzyme sequence space (up to 60% sequences introduced) in comparison to the existing ML datasets for k_{cat} and K_m. We find that the introduced sequences span widely across enzyme classes with no biases for specific EC classes (Fig. 1b). Moreover, k_{cat} and K_m entries in CatPred-DB have broader coverages compared to existing ML datasets across all the enzyme families as per the EC level 1 (Fig. 1c). Therefore, we envision that the enhanced sequence and EC classification coverage would make CatPred-DB a useful resource to the community for aiding systematic development and benchmarking of ML models for enzyme kinetic parameter prediction.

Overview of CatPred's machine learning framework

CatPred relies on the enzyme sequences/3D-structures along with the SMILES string of the corresponding substrates (reactants) as inputs and outputs machine-learned in vitro kinetic parameters. We used a concatenated SMILES string of all reactant molecules for k_{cat} prediction. For K_m or K_i prediction, the SMILES string corresponding to the relevant substrate is used. During training, the two sets of inputs are transformed into their respective feature spaces through separate feature learning modules (Fig. 2a, b). For enzyme feature learning, *CatPred* makes use of three approaches that successively add to the detail of description: (1) <u>Seq</u>uence <u>Att</u>ention (Seq-Att) (2) <u>p</u>rotein

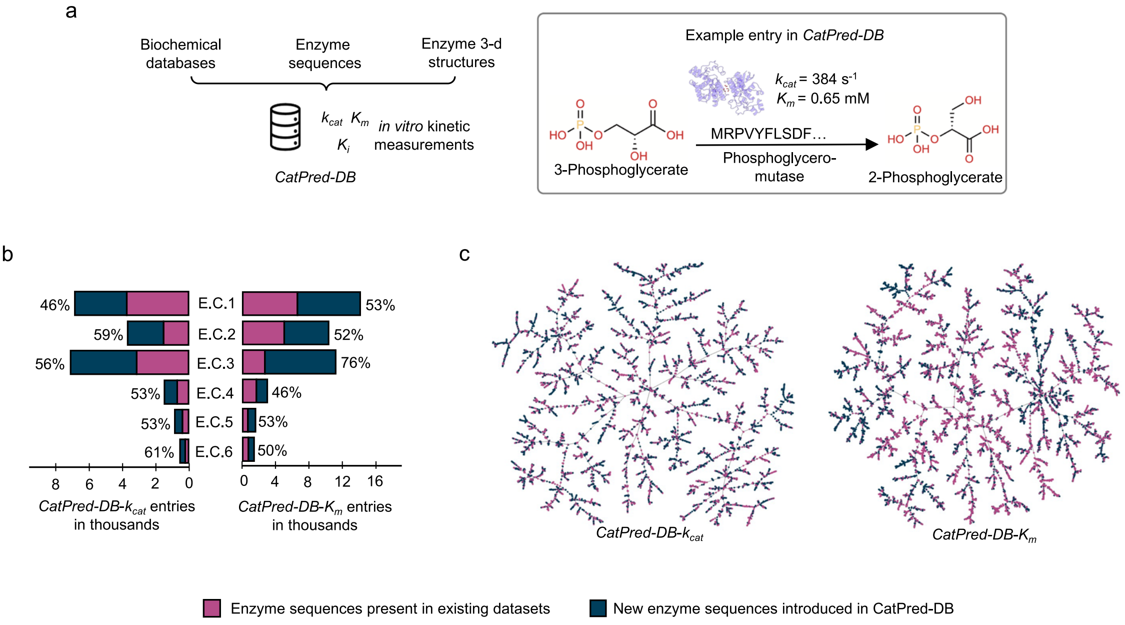


Fig. 1 | CatPred-DB - a comprehensive suite of benchmark datasets for k_{cat}, K_m and K_i in vitro measurements of enzymatic reactions curated from BRENDA and SABIO-RK databases. a For each enzymatic reaction, the datasets contain complete annotations of the molecules involved in the reaction, the enzyme sequence, the AlphaFold2.0/ESMFold predicted enzyme structure and the associated kinetic parameters. **b** Bar plot of the number of entries in the CatPred-DB - k_{cat} and - K_m datasets grouped by their Enzyme Classification (EC level 1). The labels on top of each bar show the percentages of added sequences in CatPred-DB compared to those present in existing datasets. **c** The enzyme sequence latent space plots of CatPred-DB's k_{cat} and K_m datasets visualized using the ESM-2 protein Language Model (pLM) embeddings. The sequence embeddings are converted to k-nearest neighbor graphs (k = 10) and visualized using the TMAP[63] and Faerun[64] libraries. Each point in the latent space plots corresponds to a single enzyme sequence and is colored according to whether it has been introduced in CatPred-DB or is present in existing datasets.

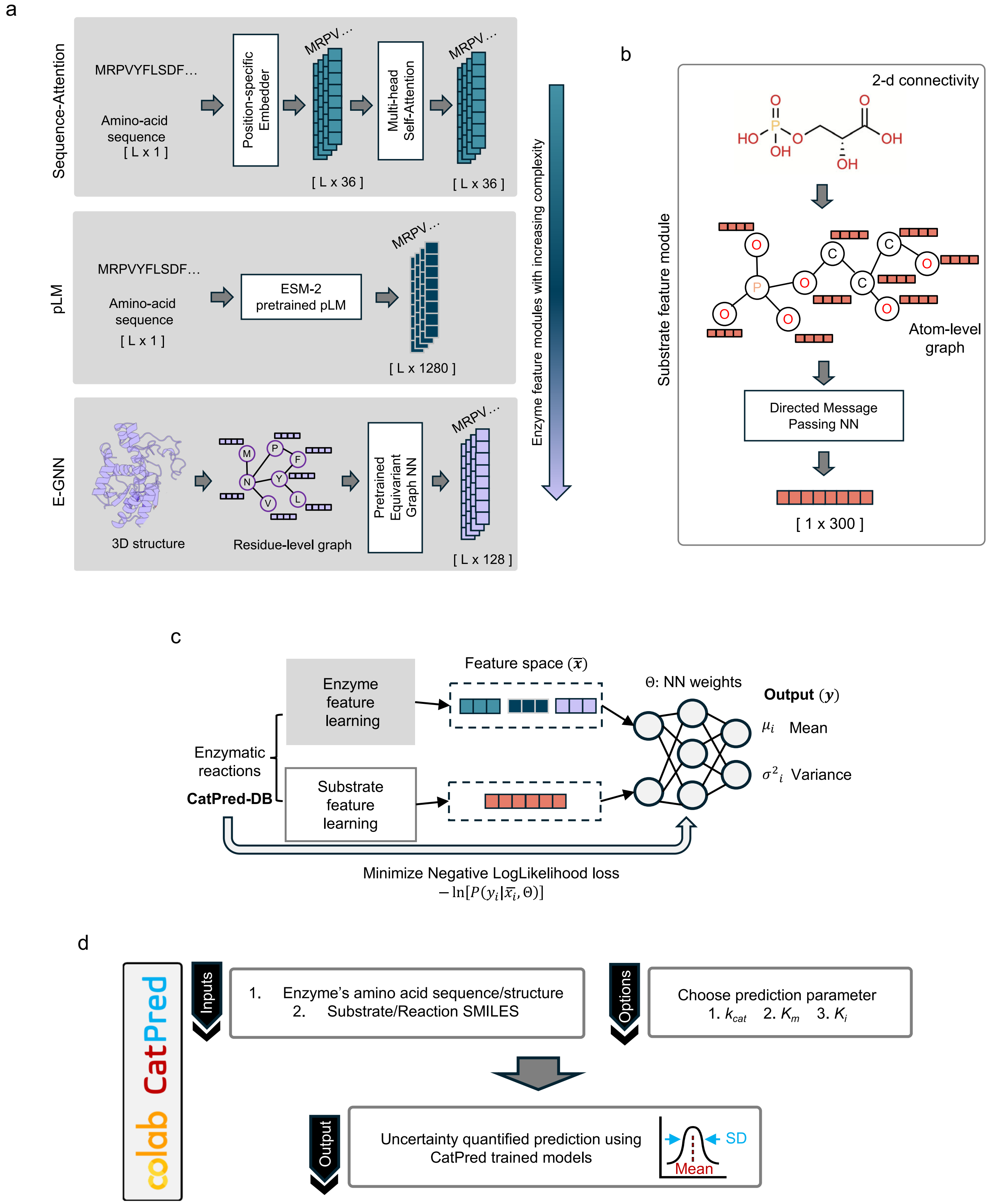
Table 1 | Coverage statistics of CatPred-DB vs. other datasets for machine learning in vitro enzyme kinetic parameter measurements

Dataset	CatPred-DB			Existing datasets	
	k_{cat}	K_m	K_i	k_{cat} (*Li.* et al.[27])	K_m (*Kroll* et al.[28])
Entries	23,197	41,174	11,929	17,010	11,722
Unique organisms	1685	2419	652	849	N/A
Unique Enzyme Classes (EC)	2657	3550	1306	1692	3690[a]
Unique enzyme sequences	7183	12,355	2829	3219	6990

[a]Predicted Enzyme Classification (EC) numbers using CLEAN.

'Unique organisms' refers to the distinct taxonomic organisms identified in the dataset. 'Unique Enzyme Classes (EC)' refers to non-redundant enzyme classification (EC) numbers based on the Enzyme Commission hierarchy. 'Unique enzyme sequences' refers to distinct amino acid sequences of enzymes in the dataset using a 100% sequence similarity threshold.

Language Model (pLM) features, and finally (3) 3D-structure features (Fig. 2a). This is carried out to properly delineate the respective contribution to improved prediction of more sophisticated encodings. For substrate feature learning, CatPred utilizes the extensively benchmarked Directed Message Passing Neural Networks[45] (D-MPNN). D-MPNNs transform SMILES strings to 2D-graphs of atoms with bond connectivity and learn their aggregated representations using graph convolution operations[45] (Fig. 2b). For the derivation of sequence attention (Seq-Attn) features, the amino-acid sequences of enzymes are encoded into numerical representations using the rotary positional embeddings[47] akin to the encoding layer used to train the ESM-2 pLM[5]. The encoded numerical representations are then transformed using self-attention layers[48] to capture dependencies and relationships across the length of enzyme sequences (Fig. 2a). The pLM features are extracted by using the ESM-2[5] (Evolutionary Scale Modeling) model with 650 M parameters that has been pretrained on the Uniref50 dataset. The 3D structural features are extracted using the Equivariant Graph Neural Networks (E-GNN[44]) that operate on amino acid residue graphs. We integrated E-GNN from Greener et al.[49] that has been pre-trained using a supervised contrastive learning for embedding protein structures into a low-dimensional latent space (Fig. 2a). The pre-trained E-GNN's latent space clusters the embeddings of similar protein structures together whereas separating dissimilar ones away from one another[49]. We reasoned that using these E-GNN derived embeddings as features within CatPred can complement the sequence-attention and pLM features. Enzyme features learnt through these modules (Seq-Attn, pLM, E-GNN) are concatenated along with the substrate features from D-MPNNs and used to predict the respective targets (log10-transformed kinetic parameters). CatPred uses a probabilistic regression approach[50] and therefore provides kinetic parameter predictions as distributions characterized by both a mean and a variance, rather than single-value predictions. Specifically, the concatenated enzyme and substrate features are fed into a fully connected neural network that outputs a mean and variance for each input (Fig. 2c). The network is trained using a negative log likelihood (NLL) loss function on CatPred-DB's datasets (refer to "Hyperparameter tuning and training" section of Methods). For each input, the output mean value is the model prediction while the output variance


a
Sequence-Attention
MRPVYFLSDF…
Amino-acid sequence [L x 1]
Position-specific Embedder
MRPV…
[L x 36]
Multi-head Self-Attention
MRPV…
[L x 36]
pLM
MRPVYFLSDF…
Amino-acid sequence [L x 1]
ESM-2 pretrained pLM
MRPV…
[L x 1280]
E-GNN
3D structure
M
P
F
N
Y
V
L
Residue-level graph
Pretrained Equivariant Graph NN
MRPV…
[L x 128]
Enzyme feature modules with increasing complexity
b
2-d connectivity
O
P
HO
OH
O
O
OH
OH
O
O
P
O
O
C
C
C
O
O
Atom-level graph
Substrate feature module
Directed Message Passing NN
[1 x 300]
c
Enzyme feature learning
Feature space ($\bar{x}$)
Θ: NN weights
Output (y)
μ_i Mean
$\sigma^2{}_i$ Variance
Enzymatic reactions
CatPred-DB
Substrate feature learning
Minimize Negative LogLikelihood loss
$-\ln[P(y_i|\bar{x}_i, \Theta)]$
d
colab CatPred
Inputs
1. Enzyme's amino acid sequence/structure
2. Substrate/Reaction SMILES
Options
Choose prediction parameter
1. k_{cat} 2. K_m 3. K_i
Output
Uncertainty quantified prediction using CatPred trained models
SD
Mean

corresponds to the respective aleatoric uncertainty of the prediction[50]. In order to account for the epistemic uncertainty of predictions, we train ten identical copies of the models (together referred to as an ensemble model) using different random initial weights. Therefore, for a given input, there is a set of ten outputs, each constituting a mean and a variance. The average value of the ten mean predictions is taken as the final parameter prediction. And the variance of mean predictions across the ensemble corresponds to the epistemic uncertainty[43].

For each dataset in CatPred-DB, the CatPred framework is used to train ML models that minimize a negative log-likelihood loss[50] (refer to "Hyperparameter tuning and training" section of Methods) of the predicted distributions to the corresponding target values. Each CatPred-DB dataset is randomly split into 80-10-10 proportions for training-validation-testing, respectively. Because CatPred involves using both enzyme sequences/structures and substrate SMILES as inputs, the splitting is carried out so as no enzyme-substrate pair

Fig. 2 | Overview of CatPred's deep learning architecture and prediction interface. a The three different modalities with increasing level of detail explored for Enzyme feature learning. The Sequence-Attention (Seq-Att) module learns features of amino-acid embeddings using multi-head attention layers. The pLM module uses features extracted from a pre-trained protein Language Model (pLM). The Equivariant Graph Neural Network (E-GNN) module extracts features of 3 d structures of enzymes by employing equivariant graph neural networks on their amino-acid level graphs. **b** Substrate feature learning is carried out using Directed Message Passing Neural Networks (D-MPNN) that extract molecular representations by leveraging 2D atom-bond connectivity graphs. **c** CatPred models are trained on CatPred-DB datasets utilizing both substrate and enzyme feature learning modules with a probabilistic regression approach. The enzyme and substrate features are input to a fully connected neural network that predicts the kinetic parameters as outputs in the form of Gaussian distributions characterized by their respective means (μ) and variances (σ^2). **d** CatPred production models are made available through the Google-Colab interface for ease of access. The inputs are the substrate SMILES and either enzyme sequence or structure along with a choice of kinetic parameter for prediction. The interface then loads the respective trained models and outputs uncertainty quantified kinetic parameters in terms of a predicted mean and standard deviation (SD).

(enzyme and reactant-set pair in case of k_{cat}) is repeated across different partitions. Adjustable hyper-parameters in the framework are either fixed to default values or optimized by evaluating trained CatPred models on the validation sets (refer to "Hyperparameter tuning and training" section of Methods). The optimized hyperparameters are used to train the final models CatPred-k_{cat}, CatPred-K_m and CatPred-K_i using the training and validation sets and evaluated on the testing sets (see below). Production models trained on the full datasets are made available for easy access through the Google Colab interface, which can be used without requiring any local installation or specialized hardware (Fig. 2d).

Assessing CatPred's Performance Across Enzyme Feature Learning Modalities

Trained CatPred models were evaluated on two test sets – (1) "held-out" test set and (2) "out-of-distribution" test set. The evaluation criterion is based on the coefficient of determination (R^2) which quantifies the fraction of data variance in the regression target that is captured by the predicted values. For each kinetic parameter, the held-out test sets are constructed as randomly selected 10% (in size) subsets of the complete CatPred-DB dataset. As implied by their definition, the held-out test sets do not contain any enzyme-substrate pairs used for training the models. The out-of-distribution test sets are further subsets of the held-out test sets (approximately 12 to 15% thereof) with not only specific enzyme-substrate pairs but also all nearly identical enzyme sequences excluded from the training set (Fig. 3a). By construction, any enzyme sequence in the out-of-distribution set is at most 99% identical to any sequence in the training set. Therefore, prediction metrics achieved on the held-out test sets reflect the prediction fidelity for unseen enzyme-substrate pairs. Out-of-distribution test sets provide a more stringent prediction challenge by assessing prediction performance on unseen enzymes (even excluding enzymes within 99% in sequence identity).

We find that CatPred models that use substrate features along with both Seq-Attn and pLM features have the best performance across all three enzymatic parameters (Fig. 3b). Notably, using only the substrate features leads to a reasonable performance for both K_m and K_i prediction (R^2 of 0.465 and 0.525) at par with previous studies[28]. Even though the inclusion of Seq-Attn features alone slightly improves prediction performance, the combined addition of both Seq-Attn and pLM features leads to best "in-class" performance for k_{cat}, K_m and K_i prediction with R^2 values of 0.607, 0.648 and 0.637, respectively (Fig. 3b). These metrics are at least as good or better than all existing ML models for predicting k_{cat}[26,27,29] and K_m[28,29] values respectively. It is worth noting that CatPred models that use 3D-structural features extracted from the E-GNN in addition to Seq-Attn and pLM features do not improve the prediction performance compared to only using Seq-Attn and pLM. The achieved R^2 values were 0.607, 0.648 and 0.639 on the held-out test sets respectively for k_{cat}, K_m and K_i (Fig. 3b).

Importantly, CatPred models retained strong prediction performance even on "out-of-distribution" test sets for K_m ($R^2 = 0.536$), though accuracy was lower for k_{cat} and K_i ($R^2 = 0.390$ and 0.409 respectively) (Fig. 3b). It is worth noting that CatPred's R^2 value for k_{cat} prediction on out-of-distribution samples is comparable to that achieved by TurNup ($R^2 = 0.40$) in a similar evaluation setting[26]. We observe that while adding Seq-Attn features leads to improved performance for k_{cat} and K_m predictions, the improvements are not as pronounced on out-of-distribution sets. This suggests that even though the self-attention layers in Seq-Attn can successfully encode enzyme sequences by extracting local and global patterns, they cannot account for higher-order relationships across sequences that are necessary for generalization to unseen protein sequences. ESM-2 pLM can capture such features and has already been proven to be capable of encoding evolutionarily rich semantics of protein sequences[5,34] explaining their good performance on out-of-distribution samples. We further carried out an ablation study to verify if Seq-Attn features are needed despite the inclusion of pLM features. This revealed that both k_{cat} and K_m models benefit from adding Seq-Attn along with pLM features (Supplementary Table 9).

We found that adding Seq-Attn+pLM features leads to a worse predictive performance for K_i on out-of-distribution test sets when compared to adding only Seq-Attn features (R^2 value of 0.461 vs. 0.409). This seemingly surprising finding is likely due to overfitting on the relatively small K_i dataset (approximately four-times smaller than K_m dataset, see Table 1) using high dimensional pLM features. Also, for CatPred models using E-GNN features along with Seq-Attn+pLM features, the corresponding R^2 values on the out-of-distribution test sets were 0.389, 0.538 and 0.454 for k_{cat}, K_m and K_i respectively indicating no significant improvement over using only Seq-Attn+pLM features.

Recently, Kroll et al.[26] reported that the DLKcat model for k_{cat} prediction showed a diminishing performance as a function of the similarity of test enzyme sequences to those of the training set indicating that the DLKcat model might have "memorized" the training dataset instead of "learning" meaningful patterns. They showed that the DLKcat model exhibited poor predictive performance ($R^2 = -0.61$) on sequences that are significantly dissimilar compared to those in the training set. Motivated by the need to avoid overlooking such predictive behavior, we systematically assessed the reduction in prediction performance of CatPred models as the test sets become increasingly dissimilar to the training set. For this, we evaluate model performance on subsets of out-of-distribution test points grouped according to their maximum percentage sequence identity (Max. % seq. id. cutoff) down to as low as 40%. This analysis revealed that CatPred models for K_m prediction maintain robust performance with an R^2 value of 0.48 even on out-of-distribution test sets with sequence similarities less than 40% when pLM features are enabled (Fig. 4b). Predictions by CatPred for k_{cat} values remain reasonable (i.e., $R^2 = 0.33$) even down to a seq. id. cutoff of 40% (Fig. 4a) with the contribution of pLM encodings being even more pronounced. This suggests that the CatPred models for k_{cat} and K_m (with pLM features) have learnt generalizable enzyme attributes that go beyond sequence similarities. At the same time, adding EGNN features on top of pLM features did not lead to any better performance in k_{cat} and K_m prediction (Fig. 4a, b). In contrast, for CatPred-*Ki*, the benefit of using pLM features was not observed, presumably due to overfitting caused by the relatively small training set size. However, adding EGNN features alongside pLM features appears to mitigate this overfitting effect to a certain extent. Prompted by these observations, we investigated

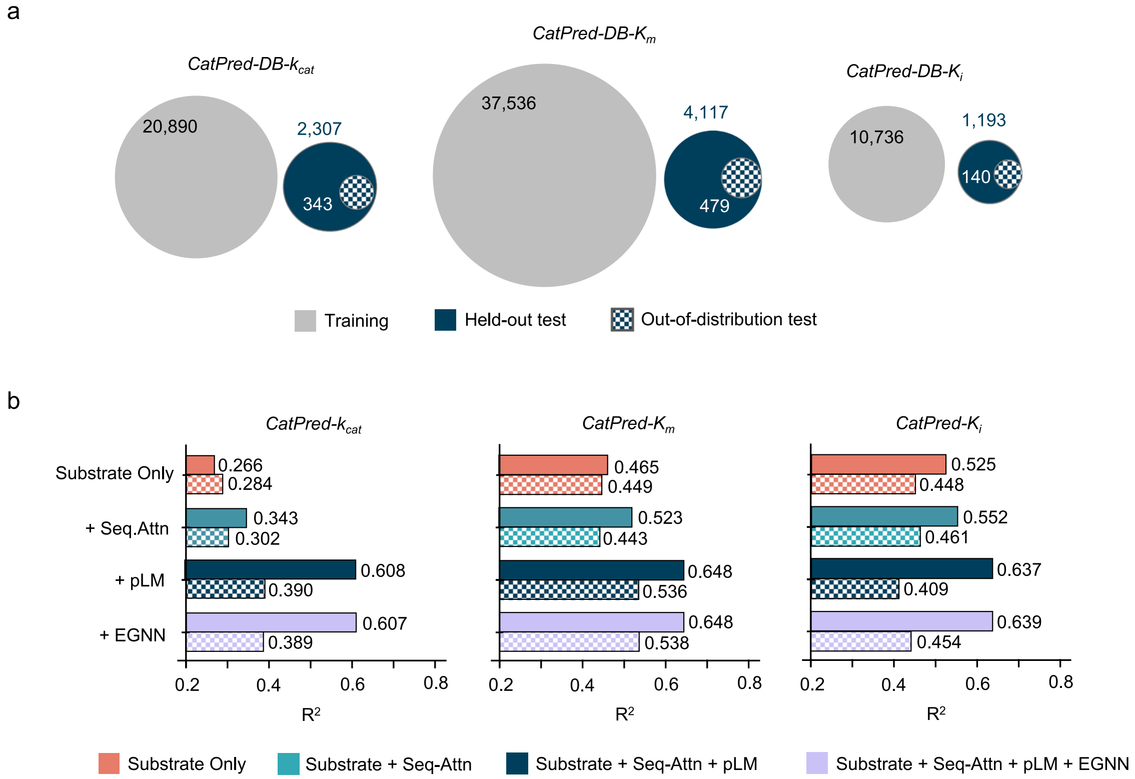


Fig. 3 | Schematic of CatPred-DB dataset splits and evaluation of CatPred models. a CatPred-DB dataset sizes used for training, held-out test and out-of-distribution test are shown as Venn diagrams. **b** Coefficient of determination (R^2) values obtained by trained CatPred models for k_{cat}, K_m and K_i prediction on held-out and out-of-distribution test sets. (**a**) by the models on (hold out) test sets (solid bars) and on (out-of-distribution) samples (patterned bars) are shown. The out-of-distribution samples are subsets of the full test-sets extracted so as no enzyme sequence in the subset is more than 99% similar to any training sequence. 'Substrate Only' refers to CatPred models trained using only the substrate features; 'Substrate+Seq-Attn' (Sequence Attention) refers to CatPred models trained using substrate features and the Seq-Attn features; 'Substrate+Seq-Attn+pLM' (protein Language Model) refers to CatPred models trained using substrate features along with both the Seq-Attn and pLM features; 'Substrate+Seq-Attn+pLM+EGNN' (Equivariant Graph Neural Networks) refers to CatPred models trained using substrate features along with Seq-Attn+pLM and EGNN features.

whether using only the EGNN structural features along with Substrate and Seq-Attn features (excluding pLM features) could benefit K_i prediction by optimally balancing model complexity and descriptive power. EGNN features(128 dimensions) are significantly lower-dimensional, whereas pLM features are 1280-dimensional (Fig. 2a). The results show that CatPred-K_i models using Substrate+Seq-Attn+EGNN achieves the best overall predictive performance metrics across held-out and out-of-distribution test sets (Supplementary Fig. 3). Therefore, the production CatPred models accessible through our Google Colab interface (Fig. 2d) are based on Substrate+Seq-Attn+pLM features for k_{cat} and K_m and Substrate+Seq-Attn+EGNN features for K_i. Also, all further mentions of CatPred-models throughout the manuscript refer to these models unless otherwise explicitly specified.

In the analyses described above we used R^2 as the sole metric of prediction quality. We have repeated almost all assessments using the mean absolute error (MAE) metric, obtaining similar trends. However, neither R^2 nor MAE provides immediate feedback to the user as to whether the predicted value for the enzyme parameter is likely to be "order of magnitude" accurate or not. Motivated by the need to provide such a metric, we introduced a metric termed p_{1mag} defined as the percent of test predictions that are within one order of magnitude error. We chose a relatively large acceptance window of one order of magnitude, as enzyme kinetic parameters span multiple orders of magnitude. To improve robustness of evaluation, we trained ten replicates of final CatPred models and Table 2 shows the corresponding performance metrics averaged over the replicates in terms of R^2 MAE and p_{1mag}. Results indicate that approximately 76%, 84% and 68% of held-out test predictions fall within an order of magnitude error (p_{1mag}) for k_{cat}, K_m and K_i predictions, respectively. They drop to 61%, 79% and 60% when evaluated on the out-of-distribution test sets. Supplementary Figs 1-3 show corresponding plots for MAE and p_{1mag} for all the analysis performed. As a further sanity check, we also evaluated the performance of CatPred models on 10 different training and evaluation sets formed by randomly splitting the full datasets such that 70–90% amounts to training while the rest is used for evaluation. This analysis revealed similar evaluation performances (see Supplementary Figs. 4–6) indicating that CatPred models are robust to evaluation on different splits of the datasets and not overfit to a single set of training and test splits.

Benchmarking CatPred against existing ML frameworks

To further contrast the predictive performance of CatPred in direct relation to existing ML frameworks, we sought to train and evaluate models using CatPred-DB datasets. We used DLKcat and UniKP for comparison. DLKcat is one of the first frameworks of its kind whereas UniKP stands as the current state-of-the-art for k_{cat} and K_m prediction. Comparing these metrics with a simple (untrained) baseline model can help shed light on the efficacy of our models. For this, we adopt an approach proposed in Kroll et al.[26]. Specifically, for each test point, we searched the training datasets for the three most similar enzyme sequences. The geometric mean of the parameter values of the found training entries is assigned as the predicted value. We trained and evaluated DLKcat and UniKP using the pipelines provided in their original works[27,29] (Refer to "Uncertainty quantification" section of Methods). For robust evaluation, we trained ten replicates of all models and compare the corresponding performance metrics

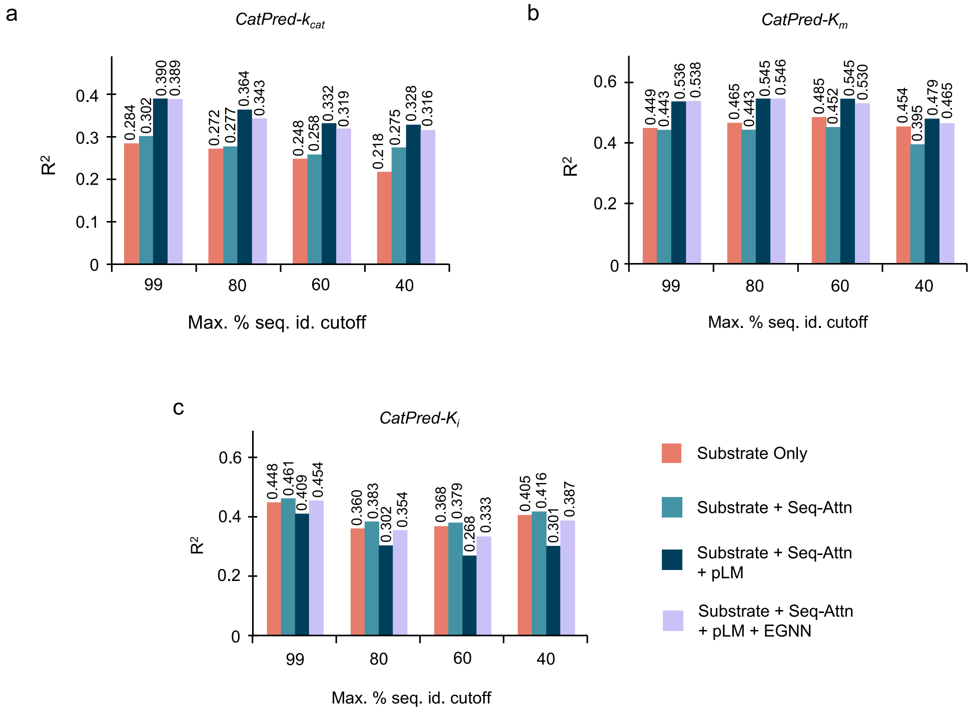


Fig. 4 | Evaluation of CatPred models on out-of-distribution sets with decreasing enzyme sequence similarities to training sequences. Evaluation metrics for (**a**) k_{cat} (**b**) K_m and (**c**) K_i are plotted. Each group on the X-axis represents the coefficient of determination (R^2) obtained on subsets of held-out tests selected using a maximum percent sequence identity cutoff (Max. % seq. id. cutoff) to training sequences. 'Substrate Only' refers to CatPred models trained using only the substrate features. 'Substrate+Seq-Attn' (Sequence Attention) refers to CatPred models trained using substrate features and the Seq-Attn features. 'Substrate+Seq-Attn+pLM' (protein Language Model) refers to CatPred models trained using substrate features along with both the Seq-Attn and pLM features. 'Substrate+Seq-Attn +pLM+EGNN' (Equivariant Graph Neural Networks) refers to CatPred models trained using substrate features along with Seq-Attn+pLM and EGNN features.

averaged over the replicates in terms of R^2, MAE and p_{1mag}. The plots for $R^2$are shown in Fig. 5 while those of MAE and p_{1mag} are provided in Supplementary Fig. 7.

We find that CatPred performs favorably against the other methods for all prediction targets both on held-out and out of distribution datasets. The improvement in CatPred's prediction performance over UniKP is particularly pronounced on out-of-distribution datasets. For example, CatPred's k_{cat} predictions explain ~40.5% more variance in the ground truth values compared to UniKP on enzyme sequences with only up to 40% similarity to training sequences ($R^2 = 0.365$ vs. $R^2 = 0.260$ in Fig. 5a). At the same time, R^2 values for K_m and K_i prediction by CatPred are greater than those of UniKP by an average of ~8% and ~5% respectively (Fig. 5b). For K_i prediction, CatPred outperforms UniKP on held-out and out-of-distribution sets up to 60% sequence similarity to training (Fig. 5c). However, for K_i prediction on the out-of-distribution set with a 40% max. seq. id. cutoff, we found no statistically significant difference between R^2 values of UniKP and CatPred (p-value = 0.526 for a Welch's two-sample t-test). In summary, CatPred models perform competitively against the compared methods in all metrics both on held-out and out-of-distribution test sets for k_{cat}, K_m and K_i prediction.

The baseline method also yields a decent predictive performances with R^2 values of 0.315, 0.380 and 0.332 respectively on held-out samples for k_{cat}, K_m and K_i prediction although the prediction performances fall inadequate for out-of-distribution samples. Therefore, the benefit of predictive ML models is maximally revealed for out-of-distribution samples. UniKP and CatPred outperform both the baseline and DLKcat in all metrics, implying that the use of pLMs is probably crucial for predictive performance. At the same time, DLKcat is outperformed by the simple baseline model for k_{cat} prediction but not for K_m and K_i prediction. This presumably indicates that more complex features such as those from pLM are essential for k_{cat} prediction. DLKcat has also been shown to overfit to nuances in the dataset by other studies[27,51]. It is worth noting that although CatPred and UniKP both use enzymatic features obtained from pLMs, CatPred often outperforms. This gain of accuracy is potentially due to the ensemble approach employed in CatPred (see Supplementary Table 10 for a comparison between single CatPred models vs. ensemble CatPred models). The ensemble-based approach with neural networks has also been previously shown to improve enzyme classification prediction[2]. At the same time, improvements with CatPred models over UniKP are often larger for out-of-distribution datasets. This could probably be attributed to the better associations between features found by the deep neural network approach employed in CatPred leading to better generalization capacity. In summary, CatPred models prove to perform competitively against the compared methods in all metrics both on held-out and out-of-distribution test sets for k_{cat}, K_m and K_i prediction. The MAE and p_{1mag} values obtained in this benchmark also reflect similar trends (The tabulated values and plots are provided in Supplementary Table 8 and Supplementary Fig. 7 respectively). However, the improvements in MAE and p_{1mag} values by CatPred models are less pronounced compared to R^2 values. This suggests that the prediction accuracy may be limited by the inherent noise in the training datasets. We next describe how leveraging CatPred's probabilistic regression framework can be used to quantify uncertainty of individual predictions, capturing the influence of noisy data on individual predictions. The uncertainty estimates reflect a measure of confidence in each prediction. Reliable confidence estimates can help segregate predictions with small errors from those with larger ones, thus alleviating the impact of uncertain predictions.

Uncertainty estimates for predictions using CatPred models

Regression models used in existing works for training ML models of k_{cat} and K_m relied on a mean-squared error loss function[26–29]. This

Table 2 | The performance metrics obtained by CatPred models as quantified using the coefficient of determination (R^2), the mean absolute error (MAE), and the percent of predictions within test sets that are within one order of magintude error (p_{1mag})

	CatPred-k_{cat}		CatPred-K_m		CatPred-K_i	
	Held-out	Out-of-distribution	Held-out	Out-of-distribution	Held-out	Out-of-distribution
R^2	0.6023 (0.0007)	0.4042 (0.0038)	0.6422 (0.0010)	0.5332 (0.0022)	0.6319 (0.0006)	0.4462 (0.0012)
MAE	0.7138 (0.0007)	0.9870 (0.0034)	0.5582 (0.0012)	0.6528 (0.0020)	0.8758 (0.0006)	0.9875 (0.0013)
p_{1mag}	75.91 (0.09)	61.43 (0.44)	83.91 (0.08)	79.04 (0.30)	67.84 (0.15)	60.50 (0.34)

Prediction metrics obtained on both held-out test sets and out-of-distribution sets are listed. Metrics listed correspond to the mean and standard error values of ten replicate models trained using different seeds.

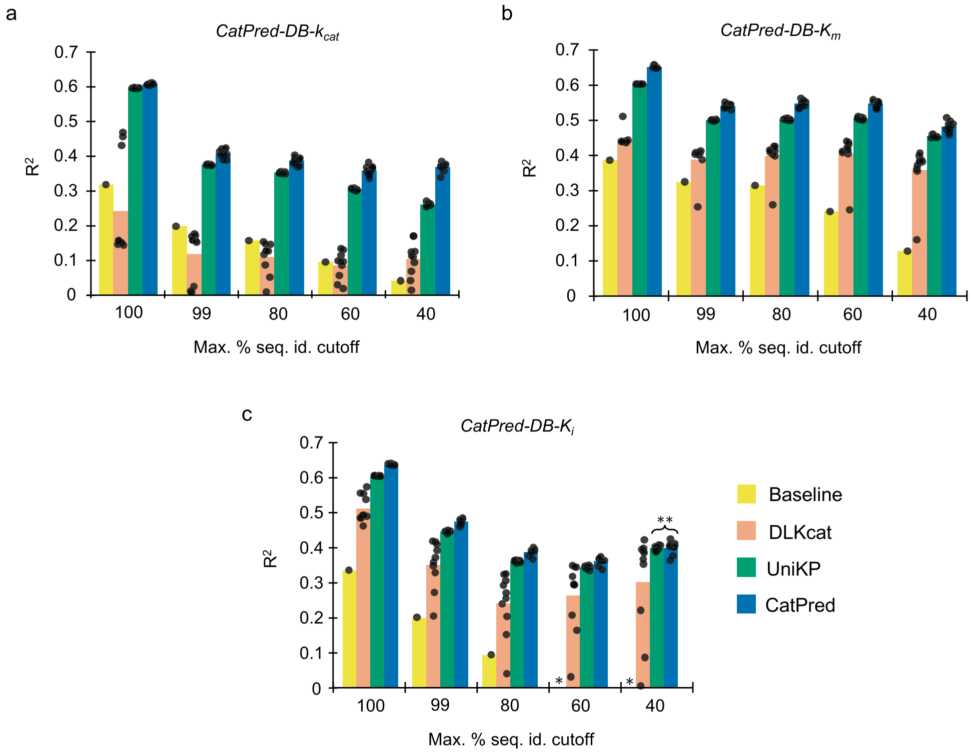


Fig. 5 | Comparative evaluation of CatPred against existing ML frameworks using CatPred-DB benchmark datasets. Baseline, DLKcat, UniKP and CatPred were evaluated. The values of coefficient of determination (R^2) obtained on held-out and out-of-distribution tests at decreasing levels of enzyme sequence similarity to training sequences are plotted. R^2 values obtained for benchmarking on (**a**) CatPred-DB-k_{cat} (**b**) CatPred-DB-K_m and (**c**) CatPred-DB-K_i are shown. Each group on X-axes indicates the test set formed by using a maximum percent sequence identity cutoff (Max. % seq. id. cutoff) to training sequences. The set with 100% Max. seq. id. cutoff refers to held-out test and the rest refer to out-of-distribution sets. The heights of each bar denote the mean metric value of ten replicate models for DLKcat, UniKP, and CatPred, respectively, while the overlaid points denote the metric values of individual replicates. There are no replicates for the Baseline. For CatPred-DB-K_i evaluation, ** indicates statistically insignificant p-value (=0.539) from a Welch's two-sample t-test (two-sided) assuming unequal variances. The results were: t(17.3) = 0.626, p = 0.539 and 95% CI for the difference in means = (−0.0061, 0.0075). No adjustments were made for multiple comparisons. * is the placeholder used for 'Baseline' bars that have a negative R^2 value (−0.047 and −0.106 for Out-of-distribution evaluation for 60% and 40% Max. % seq. id. Cutoff respectively).

approach precludes quantifying the level of uncertainty of predictions for individual enzyme-substrate pairs. The metrics such as R^2, MAE and p_{1mag} are assessed for the entire evaluation set (i.e., held-out or out-of-distribution) and not for individual predictions. Either lack of measurements or noisy data can adversely affect predictions for enzyme-substrate pairs. This implies that not all predictions would have the same fidelity. Using a probabilistic description allows CatPred to quantify the uncertainty in prediction for individual enzyme-substrate pairs. There are two sources of encountered uncertainty (i.e., aleatoric and epistemic[43]). Aleatoric uncertainty arises from noise in the training data which can be due to several factors. For example, measurements made at different assay conditions and/or using different protocols, systematic/random experimental errors and curation errors in databases, can all contribute to aleatoric uncertainty. This leads to uncharacteristic fluctuations in the value of the output even for small changes in the input (Fig. 6c). On the other hand, epistemic uncertainty arises due to the lack (or insufficiency) of training data in certain regions of the input space (Fig. 6c).

Aleatoric uncertainty is captured using the probabilistic regression approach in CatPred. By training the neural networks with a negative log likelihood (NLL) loss function, each CatPred model estimate is a Gaussian distribution characterized by a mean and a variance (Fig. 6a). This variance corresponds to the aleatoric uncertainty of each prediction (Fig. 6b). Epistemic uncertainty on the other hand, can be estimated from the variance in the set of mean predictions made from the ensemble of identical neural network models trained using different initializations (Fig. 6b). Individual models in the ensemble would provide dissonant predictions for inputs corresponding to regions with insufficient training data (Fig. 6c). The extent of the disagreement thus quantifies the associated epistemic uncertainty. For each kinetic parameter prediction made by CatPred, the combined uncertainty (the sum of aleatoric and epistemic contributions) is provided (Fig. 6b). The aleatoric uncertainty is quantified as the square root of the arithmetic mean of ensemble variances (Fig. 6b) whereas the

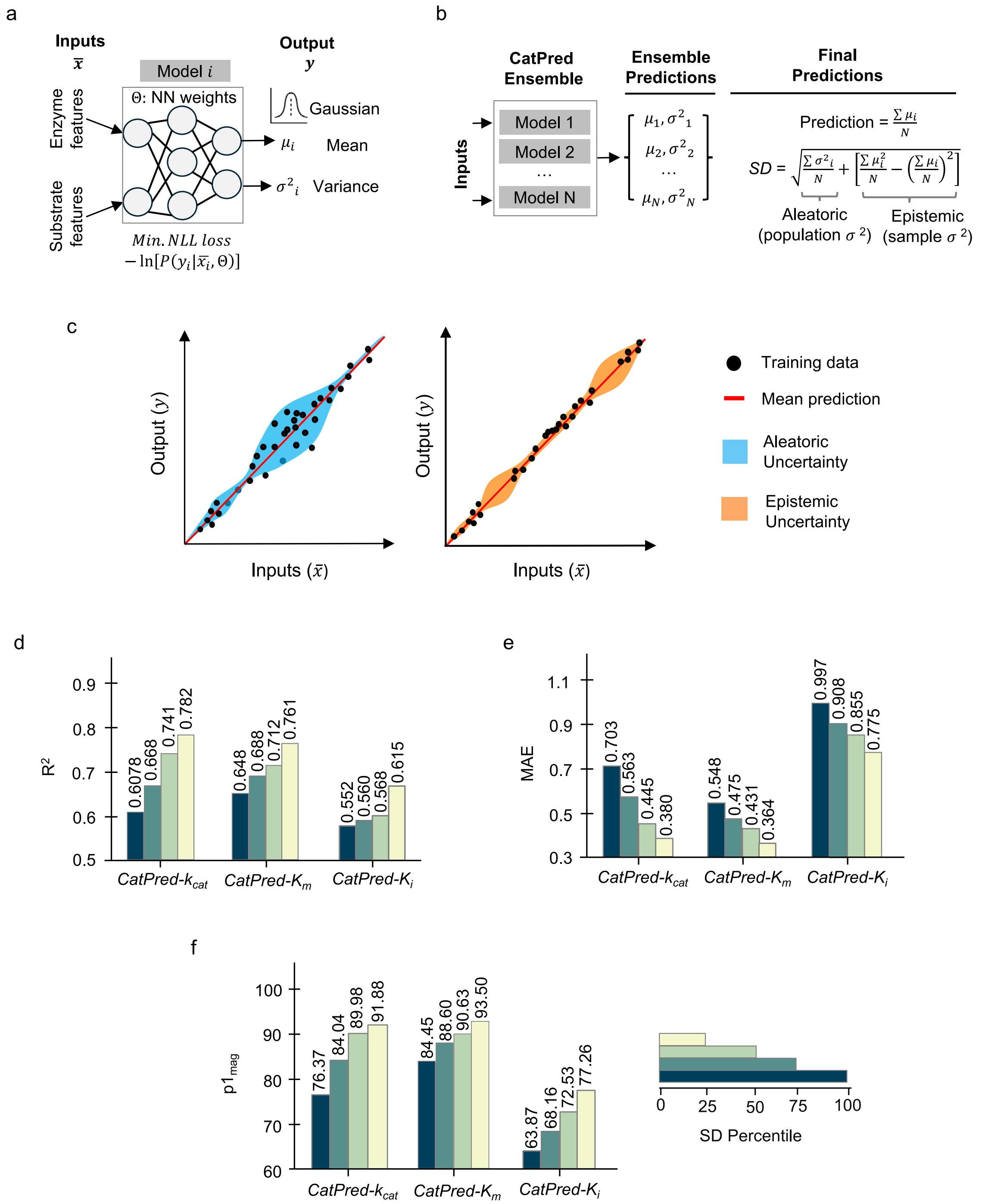


Fig. 6 | Uncertainty quantification in CatPred: Framework and Performance Analysis. a CatPred uses as inputs enzyme and substrate features and outputs kinetic parameters as Gaussians distributions characterized by a mean and a variance. When training an ensemble of models, 'Model *i*' corresponds to the *i*th set of randomly initialized weights. **b** Uncertainty prediction pipeline in CatPred. An ensemble of 'N' independent models (each with a set of randomly initialized weights) is trained for each prediction target k_{cat}, K_m and K_i. Each model outputs a mean and a variance for a given set of inputs. The final prediction is the arithmetic average of ensemble means, and the final uncertainty is the sum of aleatoric and epistemic contributions. **c** Schematic depicting the two kinds of uncertainties: aleatoric and epistemic. Aleatoric uncertainty is higher in areas with larger spread of the regression target variable, *y*, with respect to the input latent space $\bar{x}$. Epistemic uncertainty is higher in areas with absence of knowledge of *y* within the training data. The circles in plots refer to training data and the solid red line indicates the mean prediction by trained models. The performance metrics (**d**) coefficient of determination (R^2), **e** mean absolute error (MAE) and (**f**) percent of predictions within one oder of magnitude error (p_{1mag}) achieved by CatPred-k_{cat}, CatPred-K_m and CatPred-K_i models on populations of the held-out tests binned in order of their predicted uncertainty values (sum of aleatoric and epistemic uncertainty). Each colored bar denotes a population of the held-out set with standard deviation (SD) less than the 100th, 75th, 50th, and 25th percentile respectively.

epistemic uncertainty is the sample standard deviation of the ensemble means (Fig. 6b). It is important to note that because the model training is performed using log10-transformed kinetic parameter values, the corresponding standard deviations estimated are also on a log10-scale (refer to "Uncertainty quantification" section of Methods). A similar framework for uncertainty description was used in molecular property prediction[43].

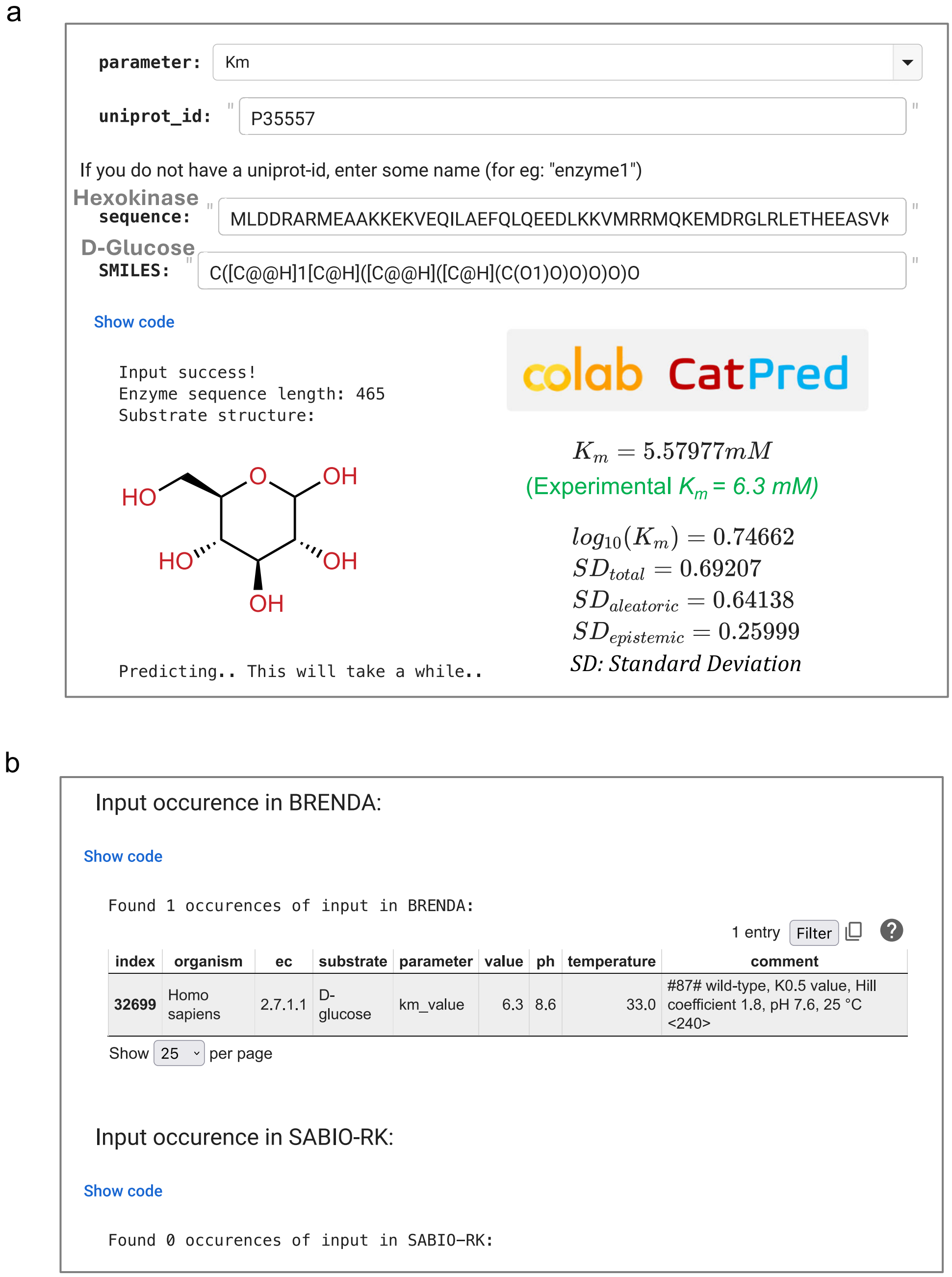
We first verified whether the predicted uncertainty values are consistent with the absolute errors for predictions made by the CatPred trained models on held-out test sets. The goal was to ensure that the predicted uncertainties can be used to discriminate between highly confident predictions from the lesser confident ones. To this end, the held-out test sets were partitioned in four subsets each consisting of predictions with uncertainty (SD) values less than the 100th, 75th, 50th and 25th percentile, respectively. This means that each subset becomes progressively enriched with predictions of higher confidence. Performance metrics R^2, MAE and p_{1mag} are calculated separately within each subset (Fig. 6d–f respectively). We perform these analyses on CatPred production models i.e., based on Substrate+Seq-Attn+pLM for k_{cat} and K_m and Substrate+Seq-Attn+EGNN for K_i. We observe that the prediction metrics monotonically improved when held-out subsets with smaller predicted uncertainties are assessed. We note that R^2 values for the (25th percentile) set are improved to 0.78, 0.76 and 0.61 for CatPred-k_{cat}, CatPred-K_m and CatPred-K_i models, respectively. Similarly, the MAE drops by approximately 46%, 37% and 29% (respectively for k_{cat}, K_m and K_i prediction) for the 25th percentile set compared to the 100th percentile set. This trend is also reflected by the increase in p_{1mag} values (Fig. 6f) showing that more than 90% of predictions in the highest confident subset (i.e., 25th percentile subset) are within an order of magnitude error for k_{cat} and K_m prediction. We also carried out this analysis for the out-of-distribution tests and we observed similar trends for MAE and p_{1mag} (Supplementary Fig. 9). These results imply that the probabilistic description of CatPred correctly assigns lower standard deviations for predictions associated with higher confidence evaluation sets. For reference, we tabulate the uncertainty values (SD) corresponding to the percentiles (used in Fig. 6) and respective MAE values obtained separately for held-out and out-of-distribution sets in Supplementary Tables 5–7. These tables can be used to lookup an estimate of the expected error associated with the predicted SD.

To facilitate the usage of CatPred, we developed an easy-to-use interface on Google Colab (Fig. 7). This interface allows for remote computations in a web browser and does not require any specialized hardware or local installation, facilitating use by a wider scientific community. The CatPred interface input area has input fields for the amino-acid sequence of the enzyme and the substrate SMILES string (Fig. 7a). In the case of k_{cat} prediction, the substrate SMILES string must contain the concatenation of the SMILES strings associated with all reactants. As discussed previously, this is needed as we discovered that not only the primary substrate but also the co-substrates (such as secondary substrates, cofactors etc.) contain information relevant to k_{cat} prediction. Unsurprisingly, this is not the case for K_m and K_i where only substrate connectivity information is a sufficient predictor. Once the enzyme parameter of interest is chosen and the inputs are entered, they are validated for correct formatting. If the enzyme sequence contains characters other than the natural amino-acid alphabet or if the SMILES string is invalid, then an error prompt is displayed asking for re-entry of inputs. After input validation, the relevant enzyme parameter prediction value along with the estimated uncertainty (contributions from aleatoric and epistemic) are output on the screen. On average, the computation takes ~20 seconds on CPU and ~10 seconds on GPU. Figure 7a pictorially illustrates the inputs and outputs for predicting the K_m value of a Hexokinase (from *Homo sapiens*) acting on its native substrate D-Glucose. The output value of 5.58 mM is within ~11% absolute error of the experimentally reported value of 6.3 mM[52]. In addition, the CatPred interface also checks if the given inputs already occur in the BRENDA and/or SABIO-RK databases to alert the user. If the check passes, then the database entries corresponding to the inputs are listed (Fig. 7b).

Discussion

Knowledge of enzyme kinetics is central to the understanding of individual enzymes, metabolic pathways, and the dynamic behavior of living cells[21,23,53]. However, experimental determination of enzyme kinetic parameters on a large-scale is an arduous and cost prohibitive task. Although several ML models have been developed before, there is no unified web resource for the prediction of k_{cat}, K_m and K_i parameters, using standardized training sets, with performance evaluated on out-of-distribution data, and with uncertainty prediction for individual queries. By leveraging deep learning on rich feature representations and training on expanded and standardized datasets, CatPred achieves competitive performance metrics compared to existing studies especially on out-of-distribution test examples. Currently, the prediction quality by CatPred is predominantly limited by the experimental uncertainty in the datasets (i.e., aleatoric uncertainty). We find that in ~74%-98% of predictions made by CatPred, aleatoric uncertainty is greater than the corresponding epistemic uncertainty (see Figure S3).

To investigate the generalizability of the CatPred framework, we trained and evaluated CatPred-k_{cat} on the test set using DLKcat's dataset curated by Li et al.[27] (see "Benchmarking UniKP and DLKcat on CatPred-DB" section of Methods). We found that CatPred achieves an R^2 of ~0.58 on a held-out test set performing favorably compared to DLKcat (R^2 ~ 0.50). When we trained and evaluated UniKP model on the same dataset, it achieved an R^2 of 0.609. On out-of-distribution evaluations, however, CatPred slightly outperformed UniKP. CatPred achieved R^2 values of 0.373, 0.295 and 0.218, compared to UniKP, which achieved 0.305, 0.227 and 0.147 on test sequences that were at most 99%, 80% and 60% identical to training sequences, respectively (see Supplementary Fig. 10). Notably, these metrics were achieved by CatPred-k_{cat} models without any additional hyperparameter optimization, indicating that CatPred's out-of-the-box architecture is well-engineered for kinetic parameter prediction. However, on test sequences that are significantly distant (i.e., less than 40% identity) from those of training samples, neither method achieved reasonable accuracies, with CatPred and UniKP yielding R^2 values of only 0.166 and 0.089, respectively. This suggests that both CatPred and UniKP display poor generalizability for k_{cat} predictions on unseen enzyme sequences. We also found that whenever CatPred models are trained on an existing smaller dataset for K_m prediction, there is little performance gain compared to what is reported in earlier studies. For example, CatPred when trained and evaluated on the dataset curated by Kroll et al.[28], performs similarly as previous reports[28,29]. In this evaluation, UniKP achieved an R^2 of 0.496 (compared to 0.528 for CatPred) on the held-out set (see Supplementary Fig. 11). Additionally, on out-of-distribution tests with 99% and 80% sequence identity cutoffs, both obtain the same R^2 values. And on out of distribution tests with 60% and 40% sequence identity cutoffs, UniKP achieves R^2 values of 0.372 and 0.350, slightly outperforming CatPred with R^2 values of 0.357 and 0.311 (see Supplementary Fig. 11). This alludes to the fact that prediction performance is becoming bottlenecked by the underlying noise in the data. In contrast, when trained and evaluated on CatPred-DB datasets, CatPred achieved better R^2 values of 0.365 and 0.475 for k_{cat} and K_m prediction respectively on out-of-distribution tests with 40% sequence identity cutoff (with corresponding R^2 values of 0.260 and 0.449 for UniKP). Taken together, our findings underscore two important conclusions. Firstly, there is a further need for larger, high-quality datasets with improved coverage. Secondly, training with the deep learning framework of CatPred is most advantageous when applied to large datasets. We discuss below on potential ways for improving prediction accuracy.



parameter: Km

uniprot_id: P35557

If you do not have a uniprot-id, enter some name (for eg: "enzyme1")

Hexokinase
sequence: MLDDRARMEAAKKEKVEQILAEFQLQEEDLKKVMRRMQKEMDRGLRLETHEEASVK

D-Glucose
SMILES: C([C@@H]1[C@H]([C@@H]([C@H](C(O1)O)O)O)O)O

Show code

```
Input success!
Enzyme sequence length: 465
Substrate structure:
```

$K_m = 5.57977 mM$

(Experimental K_m = 6.3 mM)

$log_{10}(K_m) = 0.74662$
$SD_{total} = 0.69207$
$SD_{aleatoric} = 0.64138$
$SD_{epistemic} = 0.25999$
SD: Standard Deviation

```
Predicting.. This will take a while..
```

Input occurence in BRENDA:

Show code

```
Found 1 occurences of input in BRENDA:
```

1 entry Filter

index	organism	ec	substrate	parameter	value	ph	temperature	comment
32699	Homo sapiens	2.7.1.1	D-glucose	km_value	6.3	8.6	33.0	#87# wild-type, K0.5 value, Hill coefficient 1.8, pH 7.6, 25 °C <240>

Show 25 per page

Input occurence in SABIO-RK:

Show code

```
Found 0 occurences of input in SABIO-RK:
```

Fig. 7 | Google Colab interface for making predictions using trained CatPred models. a The inputs are the 'amino acid' sequence of the enzyme and the 'SMILES' string of the substrate. The predicted output shows the kinetic parameter value (predicted K_m value of a Hexokinase enzyme with the D-Glucose substrate in the example shown) and the estimated uncertainty. The contributions to prediction uncertainty (in terms of Standard Deviation: SD in log10-scale) from both aleatoric and epistemic uncertainties are also shown. **b** The inputs entered are also searched against entries in the BRENDA and SABIO-RK databases. The example input matches with one entry in BRENDA, which is shown.

Environmental conditions such as temperature and pH can have substantial effects on enzyme kinetic parameters, and these are not explicitly accounted for by CatPred due to lack of proper annotations in parent databases[24]. Data uncertainty in kinetic parameter prediction could be ameliorated by directly incorporating these as features. Some previous studies have shown that incorporating these as additional input features for prediction can lead to better predictions[29,54,55]. Using UniKP framework, Yu et al.[29] trained an additional k_{cat} prediction ML model that explicitly considers pH and temperature as inputs and obtained a better accuracy of prediction compared to a baseline model. However, the datasets used for training using pH and temperature were quite small (~600 datapoints) indicating that these trained models may not be broadly applicable. At the same time, segregation of kinetic parameters of isoenzymes is currently not feasible due to inadequate annotations. Future work can focus on curating large datasets with high quality annotations of environmental conditions, isozymes and enzyme oligomeric states. Such limitations pertaining to datasets call for a systematic effort to generate (and open-source) high-quality measurements of enzyme kinetic parameters with complete annotations and broad coverage of enzyme functions. Training on high-quality datasets could give rise to model predictions with higher accuracies and lower uncertainties. Due to the interdependence between k_{cat} and K_m, models can also benefit from learning to predict both parameters simultaneously[54,56].

We did not find any improvements in prediction performance upon the addition of enzyme 3D-structural features extracted using the pretrained E-GNN on top of sequence attention and pLM features for k_{cat} and K_m prediction. This observation is unsurprising, given that the protein pLMs have previously been shown to encode not only sequence but also structural information[57]. Previous works also show that ML models using structural features in addition to pLM features show little improvement over those using pLM features alone[58]. Also, different graph neural network architectures can have significant impact on ML model performances. More detailed studies are needed to exhaustively explore these possibilities in context of improving enzyme kinetic parameter prediction. Instead of using entire 3D structures, a targeted description of enzyme-substrate binding regions with information of active-site amino acids could potentially be more informative[59]. CatPred-*DB* currently focuses on k_{cat}, K_m, and K_i values for wild-type enzymes. Future work could broaden its scope to include kinetic data for mutant enzymes. Incorporating detailed mechanistic features, such as active site and transition state modeling, could further enhance the ability to precisely capture the effects of mutations.

Finally, the effect of prediction inaccuracies is not sequestered to individual protein/enzymes, but often propagate in emergent system-wide properties of biological networks[60], or cells as a whole[61], (e.g., metabolic flux control, metabolic mode availability due to protein availability or expression and kinetic limitations, etc.). The prediction of kinetic parameters has found a natural niche in the parameterization of genome-scale metabolic models[62] and whole cell models[61]. Contrasting in vitro (e.g., using CatPred) vs. in vivo parameter values (e.g., by matching flux measurements) can provide clues to enzyme localization in the cell, the presence of metabolons and various post-translational regulatory programs[63].

Methods

Dataset curation

The BRENDA database version 2022_2 was downloaded in json format from their website. The SABIO-RK database was downloaded from their website in sbml format. The data from BRENDA are processed as follows: (1) The downloaded databases were processed using in-house Python scripts. All entries of the downloaded databases were parsed while discarding entries that do not have the essential annotations of (1) UniProt identifier for enzyme sequence (or) Organism name and EC number (2) Name of substrate(s) (3) Numerical value of a kinetic parameter (k_{cat}, K_m or K_i). For entries with a valid organism name and EC number but no Uniprot Id, Uniprot API search is used to find out all enzyme entries with the given Organism and EC combination. If the search returned a unique enzyme Uniprot-id, the entry was updated with the identified Uniprot-id (else, the corresponding entry is discarded). Entries belonging to engineered or mutated enzymes were discarded by parsing the comments attached to each entry.

The Uniprot identifiers of each entry were next used to obtain enzyme sequences and AlphaFold-2.0 predicted structures. In the absence of a 3D structure in the AlphaFold database, we used ESMFold[5] to carry out structure prediction. Substrate name-to-SMILES mappings for the entire databases were retrieved from BRENDA and SABIO-RK and used to populate the parsed entries with SMILES strings. For those substrates whose SMILES could not be found in BRENDA and SABIO-RK, we utilized PubChem's identifier exchange service [https://pubchem.ncbi.nlm.nih.gov/idexchange/] to obtain SMILES strings. Each SMILES string was canonicalized using the RDKit Python library. Because of the way the BRENDA database is structured, each k_{cat} entry is attached to a 'substrate'. However, to include a more detailed description, we preprocess these entries and map the substrates to their respective reactions using the reaction entries in BRENDA. Thereby, each k_{cat} entry in CatPred-*DB* originating from BRENDA is attached to the corresponding list of reactants (a concatenated SMILES string). On the contrary, k_{cat} entries originating from SABIO-RK are already attached to their reactions, making this preprocessing step unnecessary. Duplicate measurements (i.e., more than one measurement for the same pair of enzyme sequence and reactant/substrate SMILES) were processed by taking the geometric mean of measurements (for K_m and K_i) and the maximum value of measurements (for k_{cat}). This curation process yielded a total of 23,197 k_{cat}, 41,174 K_m and 11,929 K_i entries with enzyme sequence, enzyme structure, and substrate SMILES. Since the k_{cat}, K_m and K_i values span several orders of magnitude, the values were log10-transformed to obtain approximately normal distributions for each.

Dataset splitting

The curated *CatPred* datasets were split into training (80%), validation (10%), and held-out test sets (10%) using the scikit-learn Python package. The splitting ensures that entries in the test/validation splits do not have enzyme sequence and substrate SMILES pairs seen in training splits. The held-out sets are further filtered into subsets based on enzyme sequence identity cutoff to training sequences. Enzyme sequences within each dataset (k_{cat}, K_m or K_i) are clustered using identity cutoff values of 99%, 80%, 60% and 40% using the mmseqs2[64] library.

Calculation of enzyme sequence latent spaces

Enzyme sequences were converted into 1280-dimensional numerical representations using the mean features of the final layer of the pretrained ESM-2 model (650 million parameter version). The calculated representations were then clustered into k-nearest neighbor (kNN) graphs with the help of Approximate Nearest Neighbors algorithm[65] as implemented using the Annoy Python library. The cosine distance metric was used for clustering. A maximum of 50 kNN trees were built, with k value set to 10. Constructed trees were plotted using the TMAP[66] and Faerun[67] Python libraries. Two separate plots for CatPred-DB-k_{cat} and CatPred-DB-K_m were constructed. Within each plot, points were colored according to whether the enzyme sequences introduced in CatPred (i.e., were not present in the existing k_{cat} dataset[27] or K_m dataset.[28]) or not.

Deep learning architecture

The CatPred deep learning framework is built upon that used in ref. 68 and is written in the Python programming language. Each enzyme sequence is first transformed into numerical representation using a neural embedding layer. For CatPred models using Sequence Attention, the sequence embeddings are further enriched with the positional information using Rotary Positional Embeddings[47] and converted into key, query, values for input to attention layers as described in ref. 48. For CatPred models using pLM features, the ESM2 pretrained model (esm2_t33_650M_UR50D) developed in ref. 5 is utilized to extract 1280-dimensional features for each enzyme sequence. These features are concatenated with the sequence embedding and attention features. The concatenated features are pooled using an attentive pooling layer that learns a weight for each sequence position and performs a weighted averaging across the sequence length. These pooled features are the final enzyme representations. For each substrate, RDKit is used to generate an atom-bond connectivity graph using the Rdkit Python library. The atoms are converted into features using the corresponding atomic number, number of bonds, formal charge, hybridization, aromaticity, atomic mass, number of hydrogens bonded to the atom and chirality. Each feature is one-hot encoded and concatenated to form the atom feature vector. Similarly, the bonds are converted into features using the bond type (single, double, triple, or aromatic), bond conjugation, bond presence in a ring and bond chirality. These bond features are one-hot encoded and concatenated to form the bond feature vector. The atom and bond features are transformed into molecule features by utilizing the directed-message passing neural network (D-MPNN) as described in ref. 45. Using these, a directed edge feature is constructed for a pair of atoms connected by a bond by concatenating the first atom's feature with the bond's features. These edge features are iteratively updated using a learnable neural network with non-linear activation function to aggregate the features of neighborhood atoms[45]. The final molecular representation is obtained by summation of all atom features. The final enzyme and molecular representations are concatenated together and input to a fully connected neural network to output two real values representing the mean and the variance. The E-GNN pre-trained model and its pre-trained weights as described in ref. 49 are used without any modification to extract the structural features. For each enzyme 3D-structure, this yielded a 128-dimensional embedding.

Hyperparameter tuning and training

The hyperparameters for the enzyme feature learning modules include the dimension of embedding layer, the dimension of rotary positional embeddings, the number of attention layers and the number of layers in attentive pooling. All hyperparameters for the substrate feature learning module were set to the optimal values recommended in the reference[68]. The learning rate was set to 0.001, and the batch size was tuned accordingly. The number of models in the ensemble when training CatPred models was set to 10. The rectified linear unit (ReLU) activation function was used for all layers except for the output layers. All the models were trained in batches using the Adam optimizer and the training dataset was fed into the model for 20 epochs. We used minimization of the negative log-likelihood loss function as the objective function as described in ref. 43.

Different combinations of listed hyperparameters were tried to train models and optimal values are chosen by the performance of trained models on the validation dataset. The optimal values so obtained are used to train models on the training+validation and training+validation+test datasets for testing and production purposes respectively. The production models were trained for ten replicates (to form the ensemble) for 30 epochs. The list of tested hyperparameters and the obtained optimal values are listed in Supplementary Table 4.

Calculation of evaluation metrics

The coefficient of determination (R^2) is calculated using the Python library sklearn's method: sklearn.evaluation_metrics.r2_score. It is calculated as,

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{1}$$

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{2}$$

The Mean Absolute Error (MAE) is calculated using the Python library sklearn's method: sklearn.evaluation_metrics.mean_absolute_error. It is calculated as,

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{3}$$

The Percent of predictions within an order of magnitude error (p_{1mag}) is calculated using a custom Python function as,

$$p_{1mag} = \frac{100}{n} \times Count(|y_i - \hat{y}_i| < 1) \tag{4}$$

where y_i and $\hat{y}_i$ are the target value and the predicted value respectively for observation i and n is the total number of observations in a given dataset. Note that the target values and predicted value are in log10-base.

Uncertainty Quantification

The set of ten models forming an ensemble is used to compute the final prediction, epistemic and aleatoric uncertainties. Each member of the ensemble outputs a mean value prediction and a variance prediction. Specifically, the arithmetic average of the ten individual predictions forms the final prediction. Hence, the final prediction from ensemble is equal to $\frac{\sum \mu_i}{N}$, where μ_i is the mean value prediction from i^{th} member of the ensemble. The aleatoric uncertainty is estimated as, $\frac{\sum \sigma_i^2}{N}$, where σ_i^2 is the variance prediction from i^{th} member of the ensemble. The epistemic uncertainty is estimated as the variance within the individual mean value predictions of the ensemble, hence it equals $\frac{\sum \mu_i^2}{N} - \left(\frac{\sum \mu_i}{N}\right)^2$. Wherever uncertainty values are reported as standard deviation (SD), the square root of the corresponding variance is used.

Benchmarking UniKP and DLKcat on CatPred-DB

UniKP models were trained using CatPred-DB datasets using training scripts from Github [https://github.com/Luo-SynBioLab/UniKP.git]. Because UniKP originally was trained on a smaller dataset, we explored if increasing the number of parameters in the model can lead to a better accuracy. For this, we tuned the 'n_estimators' parameter and found that a value of '1000' gave an optimal performance (In the original work[29], this was set to 100). For DLKcat training, the codes from their original work on Github[https://github.com/SysBioChalmers/DLKcat] were utilized. We explored the same hyperparameter set as used in Li et al.[27] and found the optimal set of parameters. Because CatPred-DB-k_{cat} dataset contains concatenated SMILES strings of all reactants, we used a list of cofactors to form the corresponding datasets with substrate only datasets for training DLKcat. For training UniKP models, we used the concatenated SMILES strings of reactants similar to that used for training CatPred. This led to a better accuracy by UniKP compared to using substrate SMILES formed by filtering using the list of cofactors. Each evaluation includes training 10 replicate models for UniKP and DLKcat.

Benchmarking UniKP and CatPred on previously curated datasets

For each dataset, test and training splits are first formed by random splitting of the entire dataset in 10%-90% ratio. The test set is further split into out-of-distribution sets with decreasing sequence identity to training sequences by clustering using identity cutoff values of 99%, 80%, 60% and 40% using the mmseqs2[64] library. Five replicates each of CatPred-k_{cat} and CatPred-K_m models are trained using Substrate+Seq-Attn+pLM features with the same hyperparameters used for CatPred production models. Five replicates of UniKP models each for k_{cat} and K_m are trained using n_estimators = 1000.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

Unless otherwise stated, all data supporting the results of this study can be found in the article, supplementary, and source data files. The raw data of enzyme kinetic parameters used for curating CatPred-DB are available at BRENDA [https://www.brenda-enzymes.org/] and SABIO-RK [http://sabio.h-its.org/]. The enzyme sequence data and AlphaFold predicted enzyme structures are available at UniProt [https://www.uniprot.org/]. The processed CatPred-DB datasets are available at Zenodo [https://doi.org/10.5281/zenodo.14775076]. The processed datasets and pre-trained checkpoints are also available through at CodeOcean [https://codeocean.com/capsule/5255271/tree]. Source Data are available on Figshare [https://doi.org/10.6084/m9.figshare.27992060][69]. Source data are provided with this paper.

Code availability

Codes to use CatPred for training and inference (using pre-trained models) are available at Zenodo [https://doi.org/10.5281/zenodo.14775002] and Github [https://github.com/maranasgroup/CatPred/releases/tag/1.0.1]. A google colab notebook for using trained *CatPred* models is also available at Github [https://github.com/maranasgroup/CatPred/releases/tag/1.0.1]. The codes necessary for reproducing the results of the manuscript are also available at CodeOcean [https://codeocean.com/capsule/5255271/tree].

References

1. Bileschi, M. L. et al. Using deep learning to annotate the protein universe. *Nat. Biotechnol.* **40**, 932–937 (2022).
2. Sanderson, T., Bileschi, M. L., Belanger, D. & Colwell, L. J. ProteInfer, deep neural networks for protein functional inference. *eLife* **12**, e80942 (2023).
3. Yu, T. et al. Enzyme function prediction using contrastive learning. *Science (1979)* **379**, 1358–1363 (2023).
4. Kim, G. B. et al. Functional annotation of enzyme-encoding genes using deep learning with transformer layers. *Nat. Commun.* **14**, 7370 (2023).
5. Lin, Z. et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science (1979)* **379**, 1123–1130 (2023).
6. 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).
7. Bateman, A. et al. UniProt: the Universal Protein Knowledgebase in 2023. *Nucleic Acids Res.* **51**, D523–D531 (2023).
8. Markin, C. J. et al. Revealing enzyme functional architecture via high-throughput microfluidic enzyme kinetics. *Science (1979)* **373**, eabf8761 (2021).
9. Neun, S., Van Vliet, L., Hollfelder, F. & Gielen, F. High-Throughput Steady-State Enzyme Kinetics Measured in a Parallel Droplet Generation and Absorbance Detection Platform. *Anal. Chem.* **94**, 16701–16710 (2022).
10. Jumper, J. et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).
11. Sellés Vidal, L., Isalan, M., Heap, J. T. & Ledesma-Amaro, R. A primer to directed evolution: current methodologies and future directions. *RSC Chem. Biol.* **4**, 271–291 (2023).
12. Xiao, H., Bao, Z. & Zhao, H. High throughput screening and selection methods for directed enzyme evolution. *Ind. Eng. Chem. Res.* **54**, 4011–4020 (2015).
13. Carbonell, P. et al. Selenzyme: Enzyme selection tool for pathway design. *Bioinformatics* **34**, 2153–2154 (2018).
14. Upadhyay, V., Boorla, V. S. & Maranas, C. D. Rank-ordering of known enzymes as starting points for re-engineering novel substrate activity using a convolutional neural network. *Metab. Eng.* **78**, 171–182 (2023).
15. Islam, M. M., Schroeder, W. L. & Saha, R. Kinetic modeling of metabolism: Present and future. *Curr. Opin. Syst. Biol.* **26**, 72–78 (2021).
16. Kumar, A., Wang, L., Ng, C. Y. & Maranas, C. D. Pathway design using de novo steps through uncharted biochemical spaces. *Nat. Commun.* **9**, 184 (2018).
17. Domenzain, I. et al. Reconstruction of a catalogue of genome-scale metabolic models with enzymatic constraints using GECKO 2.0. *Nat. Commun.* **13**, 3766 (2022).
18. Hu, M. et al. Comparative study of two Saccharomyces cerevisiae strains with kinetic models at genome-scale. *Metab. Eng.* **76**, 1–17 (2023).
19. Foster, C. J., Wang, L., Dinh, H. V., Suthers, P. F. & Maranas, C. D. Building kinetic models for metabolic engineering. *Curr. Opin. Biotechnol.* **67**, 35–41 (2021).
20. Gopalakrishnan, S., Dash, S. & Maranas, C. K-FIT: An accelerated kinetic parameterization algorithm using steady-state fluxomic data. *Metab. Eng.* **61**, 197–205 (2020).
21. Choudhury, S. et al. Reconstructing Kinetic Models for Dynamical Studies of Metabolism using Generative Adversarial Networks. *Nat. Mach. Intell.* **4**, 710–719 (2022).
22. Srinivasan, B. A guide to the Michaelis–Menten equation: steady state and beyond. *FEBS J.* **289**, 6086–6098 (2022).
23. Robinson, P. K. Enzymes: principles and biotechnological applications. *Essays Biochem.* **59**, 1–41 (2015).
24. Chang, A. et al. BRENDA, the ELIXIR core data resource in 2021: New developments and updates. *Nucleic Acids Res.* **49**, D498–D508 (2021).
25. Wittig, U., Rey, M., Weidemann, A., Kania, R. & Müller, W. SABIO-RK: An updated resource for manually curated biochemical reaction kinetics. *Nucleic Acids Res.* **46**, D656–D660 (2018).
26. Kroll, A., Rousset, Y., Hu, X.-P., Liebrand, N. A. & Lercher, M. J. Turnover number predictions for kinetically uncharacterized enzymes using machine and deep learning. *Nat. Commun.* **14**, 4139 (2023).
27. Li, F. et al. Deep learning-based k cat prediction enables improved enzyme-constrained model reconstruction. *Nat. Catal.* **5**, (2022).
28. Kroll, A., Engqvist, M. K. M., Heckmann, D. & Lercher, M. J. Deep learning allows genome-scale prediction of Michaelis constants from structural features. *PLoS Biol.* **19**, e3001402 (2021).
29. Yu, H., Deng, H., He, J., Keasling, J. D. & Luo, X. UniKP: a unified framework for the prediction of enzyme kinetic parameters. *Nat. Commun.* **14**, 8211 (2023).
30. Gelman, S., Fahlberg, S. A., Heinzelman, P., Romero, P. A. & Gitter, A. Neural networks to learn protein sequence-function relationships from deep mutational scanning data. *Proc. Natl. Acad. Sci. USA* **118**, (2021).
31. Bar-Even, A. et al. The moderately efficient enzyme: Evolutionary and physicochemical trends shaping enzyme parameters. *Biochemistry* **50**, 4402–4410 (2011).
32. 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).
33. Ferruz, N., Schmidt, S. & Höcker, B. ProtGPT2 is a deep unsupervised language model for protein design. *Nat. Commun.* **13**, 4348 (2022).
34. 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).
35. Sugaya, N. Training based on ligand efficiency improves prediction of bioactivities of ligands and drug target proteins in a machine learning approach. *J. Chem. Inf. Model.* **53**, 2525–2537 (2013).
36. Badwan, B. A. et al. Machine learning approaches to predict drug efficacy and toxicity in oncology. *Cell Rep. Methods* **3**, 100413 (2023).
37. O'Boyle, N. M. Towards a Universal SMILES representation - A standard method to generate canonical SMILES based on the InChI. *J. Cheminform.* **4**, 22 (2012).
38. Kim, S. et al. PubChem 2019 update: Improved access to chemical data. *Nucleic Acids Res.* **47**, D1102–D1109 (2019).
39. Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y. & Morishima, K. KEGG: New perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res.* **45**, D353–D361 (2017).
40. Hastings, J. et al. ChEBI in 2016: Improved services and an expanding collection of metabolites. *Nucleic Acids Res.* **44**, D1214–D1219 (2016).

41. Pham, N. et al. Consistency, inconsistency, and ambiguity of metabolite names in biochemical databases used for genome-scale metabolic modelling. *Metabolites* **9**, 28 (2019).
42. Hie, B., Bryson, B. D. & Berger, B. Leveraging uncertainty in machine learning accelerates biological discovery and design. *Cell Syst.* **11**, 461–477.e9 (2020).
43. Hirschfeld, L., Swanson, K., Yang, K., Barzilay, R. & Coley, C. W. Uncertainty Quantification Using Neural Networks for Molecular Property Prediction. *J. Chem. Inf. Model*. **60**, 3780–3780 (2020).
44. Satorras, V. G., Hoogeboom, E. & Welling, M. E(n) Equivariant Graph Neural Networks. In *Proceedings of Machine Learning Research* vol. 139 (2021).
45. Yang, K. et al. Analyzing Learned Molecular Representations for Property Prediction. *J. Chem. Inf. Model*. **59**, 5304–5305 (2019).
46. Schoch, C. L. et al. NCBI Taxonomy: A comprehensive update on curation, resources and tools. *Database (Oxford)* **2020**, baaa062 (2020).
47. Su, J. et al. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing* **568**, 127063 (2024).
48. Vaswani, A. et al. Attention is all you need. In *Advances in Neural Information Processing Systems* vols 2017-December (2017).
49. Greener, J. G. & Jamali, K. Fast protein structure searching using structure graph embeddings. *bioRxiv* 2022.11.28.518224 https://doi.org/10.1101/2022.11.28.518224 (2022).
50. Nix, D. A. & Weigend, A. S. Estimating the mean and variance of the target probability distribution. In *IEEE International Conference on Neural Networks - Conference Proceedings* vol. 1 (1994).
51. Kroll, A. & Lercher, M. J. DLKcat cannot predict meaningful kcat values for mutants and unfamiliar enzymes. *Biol. Methods Protoc* bpae061 https://doi.org/10.1093/biomethods/bpae061 (2024).
52. Xu, L. Z., Harrison, R. W., Weber, I. T. & Pilkis, S. J. Human β-cell glucokinase: Dual role of Ser-151 in catalysis and hexose affinity. *J. Biol. Chem.* **270**, 9939–9946 (1995).
53. Nelsestuen, G. L. How Enzymes Work. *Princ. Med. Biol.* **4**, 25–44 (1995).
54. Wang, J. et al. MPEK: a multitask deep learning framework based on pretrained language models for enzymatic reaction kinetic parameters prediction. *Brief. Bioinform*. **25**, bbae387 (2024).
55. Qiu, S., Zhao, S. & Yang, A. DLTKcat: deep learning-based prediction of temperature-dependent enzyme turnover rates. *Brief. Bioinform*. **25**, bbad506 (2024).
56. Shen, X. et al. EITLEM-Kinetics: A deep-learning framework for kinetic parameter prediction of mutant enzymes. *Chem. Catal.* **4**, 101094 (2024).
57. Shen, J. et al. Unbiased organism-agnostic and highly sensitive signal peptide predictor with deep protein language model. *Nat. Comput. Sci.* **4**, 29–42 (2024).
58. Zhang, Z. et al. A Systematic Study of Joint Representation Learning on Protein Sequences and Structures. Preprint at arXiv [q-bio.QM (2023).
59. Goldman, S., Das, R., Yang, K. K. & Coley, C. W. Machine learning modeling of family wide enzyme-substrate specificity screens. *PLoS Comput. Biol.* **18**, e1009853 (2022).
60. Hu, M., Suthers, P. F. & Maranas, C. D. KETCHUP: Parameterizing of large-scale kinetic models using multiple datasets with different reference states. *Metab. Eng.* **82**, 123–133 (2024).
61. Elsemman, I. E. et al. Whole-cell modeling in yeast predicts compartment-specific proteome constraints that drive metabolic strategies. *Nat. Commun.* **13**, 801 (2022).
62. Maranas, C. D. & Zomorrodi, A. R. *Optimization Methods in Metabolic Networks. Optimization Methods in Metabolic Networks* https://doi.org/10.1002/9781119188902 (2016)
63. Dinh, H. V. & Maranas, C. D. Evaluating proteome allocation of Saccharomyces cerevisiae phenotypes with resource balance analysis. *Metab. Eng.* **77**, 242–255 (2023).
64. Steinegger, M. & Söding, J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat. Biotechnol.* **35**, 1026-1028 (2017).
65. Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R. & Wu, A. Y. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *J. ACM* **45**, (1998).
66. Probst, D. & Reymond, J. L. Visualization of very large high-dimensional data sets as minimum spanning trees. *J. Cheminform*. **12**, 12 (2020).
67. Probst, D. & Reymond, J. L. FUn: A framework for interactive visualizations of large, high-dimensional datasets on the web. *Bioinformatics* **34**, 1433–1435 (2018).
68. Heid, E. et al. Chemprop: A Machine Learning Package for Chemical Property Prediction. *J. Chem. Inf. Model*. **64**, 9–17 (2024).
69. Boorla, V. S. & Maranas, C. D. CatPred: A comprehensive framework for deep learning in vitro enzyme kinetic parameters kcat, Km and Ki. figshare. Dataset. https://doi.org/10.6084/m9.figshare.27992060 (2025).

Acknowledgements

This material is based upon work supported by the Center for Bioenergy Innovation (CBI), U.S. Department of Energy, Office of Science, Biological and Environmental Research Program under Award Number ERKP886. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the U.S. Department of Energy. This work was also supported by the U.S. National Science Foundation funded Molecule Maker Lab Institute (MMLI), award number 2019897 supported by National AI Research Institutes Program of the Directorate for Computer and Information Science and Engineering (CISE), in collaboration with the Division of Chemistry (CHE) and the Division of Chemical, Bioengineering, and Environmental Transport Systems (CBET) awarded to CDM. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. Computations for this research were performed on the Pennsylvania State University's Institute for Computational and Data Sciences' Roar supercomputer.

Author contributions

V.S.B. designed the methodology, developed the models, performed data analysis, and conducted evaluations. C.D.M. supervised the project and secured funding. Both authors contributed to project conception, manuscript writing, editing, and revision.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-025-57215-9.

Correspondence and requests for materials should be addressed to Costas D. Maranas.

Peer review information *Nature Communications* thanks Iván Domenzain, Bian Wu, and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. A peer review file is available.

Reprints and permissions information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.