Digital Discovery

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal

Cite this: DOI: 10.1039/d6dd00094k

Leveraging active site information for deep learning prediction of enzyme–substrate Michaelis constants

Daniil Lepikhov, †[a] Laura Sandner †[ab] and Ariane Nunes-Alves *[ac]

The Michaelis constant (K_M) is a key parameter in enzymology. Its experimental measurement is often low-throughput and costly, but machine learning can identify patterns to make predictions in a high-throughput way. In this work, we introduce a novel approach that explicitly incorporates enzyme–substrate interface information by encoding the enzyme's active site as a feature. Using a simple multilayer perceptron (MLP) with a gated layer, we demonstrate that this explicit active site information enables our model, Active Site for K_M (AS4Km), to achieve competitive performance on the independent GMKM test set, despite its relatively simple architecture. Ablation studies confirm that active site features significantly enhance generalization to unseen data and distant enzyme sequences. Furthermore, our analysis highlights a critical limitation in current enzymology databases: predictive performance is heavily reliant on substrate identity due to low substrate diversity and a bias towards active enzyme–substrate complexes. Our results show that AS4Km, a data-driven approach combined with explicit interaction interface features, displays competitive performance in the prediction of K_M values for enzyme–substrate complexes, and may be able to assist in the identification of novel substrates for known enzymes.

Received 24th February 2026
Accepted 30th May 2026

DOI: 10.1039/d6dd00094k

rsc.li/digitaldiscovery

1 Introduction

Enzymes are cell's biological catalysts and accelerate chemical reactions that are unduly lengthy by themselves or do not happen at all by lowering the activation energy. To initialize the whole process, they first bind to the substrate *via* non-covalent interactions such as hydrogen bonds or van der Waals interactions. Then, in order to form the product, the substrate passes through a high-energy, unstable transition state which is stabilized by the enzyme. The product is then rapidly released due to its low affinity to the enzyme, and the enzyme returns to its original state.[1]

The growing number of enzymes, either recombinant or designed using artificial intelligence,[2,3] and the need to identify new potential substrates for a given enzyme have created an urgent demand for efficient validation of enzyme–substrate candidates. To this end, different biocatalytic parameters can be measured *in vitro*, such as the Michaelis constant (K_M), the catalytic constant or the binding affinity. K_M, which is the focus of this study, represents the substrate concentration required to reach half the maximum reaction rate: the lower it is, the higher the enzyme's efficiency for a given substrate.

The investigation of a large and ever-growing number of enzyme–substrate complexes can be facilitated and sped up by high-throughput methods. Today, measuring K_M values experimentally is usually low-throughput and costly.[4] To bypass these limitations, K_M values can be computed using machine learning (ML) algorithms, which are able to extract relevant statistical features from enzyme–substrate complex information stored in databases such as BRENDA[5] and SABIO-RK.[6]

Kroll *et al.*[7] developed a gradient boosting (XGBoost) model to predict K_M values for enzyme–substrate pairs, using only wild-type enzymes to avoid biases from mutations and to capture evolutionarily optimized interactions. Substrates were encoded by a task-specific molecular fingerprint generated by a directed message-passing neural network (D-MPNN) that operates on atomic and bond features from the molecular graph; this fingerprint was extended with molecular weight and the octanol–water partition coefficient. Enzymes were represented by a 1900-dimensional UniRep vector, a deep representation pretrained on 24 million protein sequences. The model was trained on 11 737 entries from the BRENDA database. The authors evaluated several substrate representations (expert-crafted fingerprints and GNN-based fingerprints) and found that the GNN-based fingerprint gave the best substrate-only predictions ($R^2 = 0.42$ on a held-out test set). The final model, combining substrate and enzyme features, achieved $R^2 = 0.53$

[a]Institute of Chemistry, Technische Universität Berlin, Straße des 17. Juni 135, 10623 Berlin, Germany. E-mail: ferreira.nunes.alves@tu-berlin.de; ariane.alves@lnbio.cnpem.br

[b]Department of Oncology, Katholieke Universiteit Leuven, ON4 Herestraat 49, 3000 Leuven, Belgium

[c]Brazilian Biosciences National Laboratory, Brazilian Center for Research in Energy and Materials, Rua Giuseppe Máximo Scolfaro, 10,000, 13083-100 Campinas, Brazil

† These authors contributed equally to this work.

on the BRENDA test set and displayed similar performance ($R^2 = 0.49$) on an independent test set from the SABIO-RK database. Separate analyses showed that enzyme-sequence information alone yields $R^2 = 0.27$, indicating that the substrate carries the dominant predictive signal.

ML-Aided Global Optimization (MLAGO)[8] is a hybrid method, which has a ML component (random forest) for predicting an initial reasonable K_M value, and a genetic algorithm (REX^{star}/JGG[9]) to refine the initial prediction. As input features, it uses a minimalistic one-hot encoding for enzyme and substrate information: the enzyme is encoded using the Enzyme Commission (EC) number, the substrate using its KEGG ID[10] and the organism's KEGG ID. No explicit chemical features for the substrate and no sequence information for the enzyme were used. The data set was curated from the BRENDA database. Mutant enzymes and non-natural substrates were removed from the initial data set, which resulted in 17 151 entries. A R^2 value of 0.536 was obtained for the test set, and realistic values (78–89% of predicted values within 10-fold of experimental values) were predicted for two sets of metabolic pathways: carbon-based (32 K_M values) and nitrogen-based (18 K_M values). Overall, MLAGO outputs biologically plausible K_M values while leveraging a simple set of features, relying on enzyme and substrate identities alone.

He and Yan proposed the ML approach GraphKM,[11] which uses both wild type and mutant enzymes as training data. Combining entries from BRENDA and SABIO-RK, the entire data set comprised 19 754 entries. Additionally, an independent data set named HXKm, including 443 entries, was curated from the literature and used as an external test set. One of the components of the ML model was a GNN, which encoded substrate structural features. GAT-GCN, based on graph attention and graph convolution mechanism,[12,13] was used to encode substrate information. The enzyme sequence was encoded into a vector using the Evolutionary Scale Model version 2 (ESM2) from Facebook research (https://github.com/facebookresearch/esm). GraphKM performed with a Pearson's correlation coefficient of 0.589 for the external test set. The major contribution to the field was the incorporation of the mutation into the enzyme sequence extracted from the BRENDA database. This work was the first to propose a systematic comparison of ML models to predict K_M values using an independent, external test set, on which GraphKM performed the best by the time it was published.

More recently, the ML model CatPred[14] was developed, offering simultaneous prediction of K_i, K_M and k_{cat} for enzyme–substrate complexes. One significant contribution was the release of the data set used to train and test the model. While they used both BRENDA and SABIO-RK databases, their preprocessing pipeline greatly increased the number of enzyme–substrate complexes with K_M measurements available (41 174 *vs.* 11 722 in Kroll *et al.*[7]). The architecture of the model included an enzyme sequence attention module, an ESM model, and an enzyme structure module,[15] leveraging contrastive learning for latent representation of the enzyme structure. The substrate was encoded using directed message passing neural network (D-MPNN), taking as input the graph structure extracted from canonical SMILES representation. This is the first paper, to our knowledge, integrating enzyme structural information in the form of Cartesian coordinates for the task of K_M prediction using ML. Another novelty introduced by this work was the uncertainty quantification for the prediction of K_M values. Two uncertainties were defined, aleatoric uncertainty and epistemic uncertainty. The aleatoric uncertainty arises from inherent noise from the training data due to experimental conditions and measurement errors. The epistemic uncertainty arises from a lack of knowledge or insufficient training data. An ensemble of 10 models was trained to output 10 K_M values, and the average and variance of these values were used as final outputs. Using the negative log-likelihood loss function, the ensemble model learned to output a Gaussian distribution with a mean and a variance, effectively computing a K_M value with its associated uncertainty (represented by the variance). To evaluate the ensemble model, two types of test sets were constructed: a held-out test set, which contained 10% of the initial data set without overlap of enzyme–substrate complexes with the training set, and an out-of-distribution test set, where the sequences were at most identical given a threshold of 99% and 80% to any sequence in the training set. The R^2 values for K_M prediction were 0.648 and 0.536 (no significant impact on performances for different genetic cutoffs) for the held-out and out-of-distribution test sets, respectively, showing overall good performances.

In the last year, emerging approaches have begun to incorporate catalytically relevant or active-site-aware representations into K_M prediction models. For instance, OmniESI[16] leverages attention mechanisms to derive enzyme representations that are enriched in catalytic information, potentially capturing active-site characteristics implicitly through learned embeddings. Similarly, approaches such as SAKPE[17] explore the incorporation of structural or functional priors into enzyme representations. These methods highlight a growing recognition of the importance of enzyme–substrate interaction regions for kinetic prediction, although this information is typically encoded in latent or architecture-dependent forms rather than as explicit input features.

In this work, we developed Active Site for K_M (AS4Km), a ML model that uses enzyme–substrate interaction interface information in the prediction of K_M values, to investigate whether explicit sequence-level encoding of active-site information can improve K_M prediction within a lightweight and interpretable framework. Two different tools were tested to predict the interaction interface, or the enzyme's active site, the AI-based docking tools TankBind[18] and P2Rank.[19] Recently, AI-assisted docking[18,20–24] demonstrated great promise in high-throughput prediction of protein–ligand complex structures by enabling blind docking, the prediction of complex structures without prior knowledge of the binding site. Additionally, these ML-based tools facilitate modeling of protein conformational changes upon binding. While the effectiveness of these tools is under debate,[25,26] blind docking is a powerful contribution from the field, offering the opportunity to build a large-scale synthetic database of protein–ligand complex structures. Rather than proposing a more complex architecture, this work

investigates whether explicit incorporation of active-site information alone can improve K_M prediction within a simple and interpretable modeling framework. Our results show that models using active site information as a feature, while maintaining a lightweight and interpretable framework, have competitive performance compared to prior, more complex approaches.

2 Methods

2.1 Data collection, cleaning and pre-processing

To build the training set, data was retrieved from the BRENDA[5] database (accessed on 22.08.2024) and the SABIO-RK[6] database (accessed on 10.10.2025). BRENDA is an enzyme activity database, while SABIO-RK is a database of biochemical reactions. The original data contained the enzyme Uniprot ID, enzyme type (wild type or recombinant/mutant), enzyme PDB ID, substrate name, substrate SMILES string, and measured experimental K_M value. The data from SABIO-RK was cleaned using the SABIO-RK data gathering pipeline from He and Yan,[11] available at https://github.com/realHXiao/GraphKM, (accessed in October 2025). Only data points with a single substrate and wild-type enzyme were selected. Avoiding mutants and recombinant enzymes places focus on genetically conserved and optimized active sites. The K_M value units were converted to mM for compatibility with BRENDA. He and Yan scripts were modified in order to keep every K_M measurement, instead of the maximum value, as it was originally intended by the authors. BRENDA was preprocessed using an in-house script, applying the same criteria as for SABIO-RK. BRENDA and SABIO-RK were concatenated into one database and further data pre-processing was conducted on the merged database. Duplicates between BRENDA and SABIO-RK enzyme–substrate complexes and their associated K_M values were kept as is in order to account for uncertainty during training. The amino acid sequences of the enzymes were retrieved from the UniProt database *via* their UniProt ID. Data points without available amino acid sequences or without valid SMILES strings were removed. Additionally, the maximum sequence length was set to 1024 amino acids, and enzymes exceeding this length were removed from the data set. Data points with K_M values of 0 were removed from the data set. The K_M values were transformed into log 10 scale, and then normalized using min–max scaling. Additionally, the K_M values were clipped in the range from 10^{-5} mM to 10^{3} mM to account both for biologically plausible and implausible enzyme–substrate interactions while keeping these two classes balanced.

As an independent test set, the HXKm test set collected by He and Yan[11] was used. To avoid any redundancies between the training set and the HXKm database, redundant enzyme–substrate complexes were removed from the training set. The HXKm test set contains originally 443 values and, after pre-processing, 420 K_M values for different enzyme–substrate complexes were retained.

To assess the generalizability of the trained model, the external HXKm test set was partitioned into distinct evaluation clusters. Clustering was performed along two independent axes: (1) enzyme sequence similarity and (2) substrate similarity relative to the training data. Enzyme sequence similarity was calculated for each enzyme pair between the HXKm and training sets using MMseqs2.[27] Substrate similarity was quantified with the Tanimoto coefficient[28] and Morgan fingerprints generated with the RDKit package. For both dimensions, four similarity thresholds were applied: 40%, 60%, 80%, and 99%. At each threshold, a cluster was defined containing only HXKm entries whose enzyme or substrate similarity scores were strictly below the specified cutoff relative to all training set examples. Model performance was then evaluated separately on each resulting cluster.

2.2 Features

Enzymes and substrates were featurized separately (Fig. 1). For the enzyme, the following features were obtained from the primary sequence: residue identity, residue position in the primary sequence and one-hot encoding to indicate whether the residue belonged to the active site or not, which is the feature that carries information about the enzyme–substrate interaction interface. Residue identities were encoded by converting the one-letter code of the amino acid into ASCII encoding using the python built-in function ord(). These values were then normalized *via* min–max scaling. The residue positional encoding was normalized into values between 0 and 1 by dividing the position index of the residue by 1,024, which is the maximum number of enzyme residues.

To predict whether a residue was part of the active site or not, first it was required to predict the enzyme structure, since most of the enzymes in the data set did not have experimental structures available. Here, the enzyme structure was predicted *via* AlphaFold 2,[29] downloaded from the AlphaFold database.[30] Proteins were retained only if at least 70% of their residues had a pLDDT score above 70, ensuring that a few low-confidence, likely disordered residues would not lead to exclusion of an enzyme, despite high confidence in key regions, such as the active site.

Two different tools, P2Rank[19] and TankBind,[18] were used to predict whether a residue was part of the active site. P2Rank predicts plausible binding pockets given the structure of an enzyme and ranks them based on the pocket's structural features. Only the residues in the top ranked pocket were annotated as the active site. TankBind builds a graph from pocket and substrate structural features and uses it as input to predict the pocket's binding affinity to the substrate. The residues in the pocket with the lowest (most favorable) binding affinity to the substrate were annotated as the active site. Two sets of active site annotations were investigated in order to evaluate the impact of substrate information on the annotation. Each residue in the active site was attributed the value of 1, and residues which were not part of the active site (the rest of the enzyme amino acid sequence) were attributed the value 0 as their binding residue feature.

To assess the reliability of the predicted active-site annotations, we performed a sanity check using experimentally curated catalytic residues from the Mechanism and Catalytic Site Atlas (M-CSA).[30] We identified a subset of 82 enzymes in our

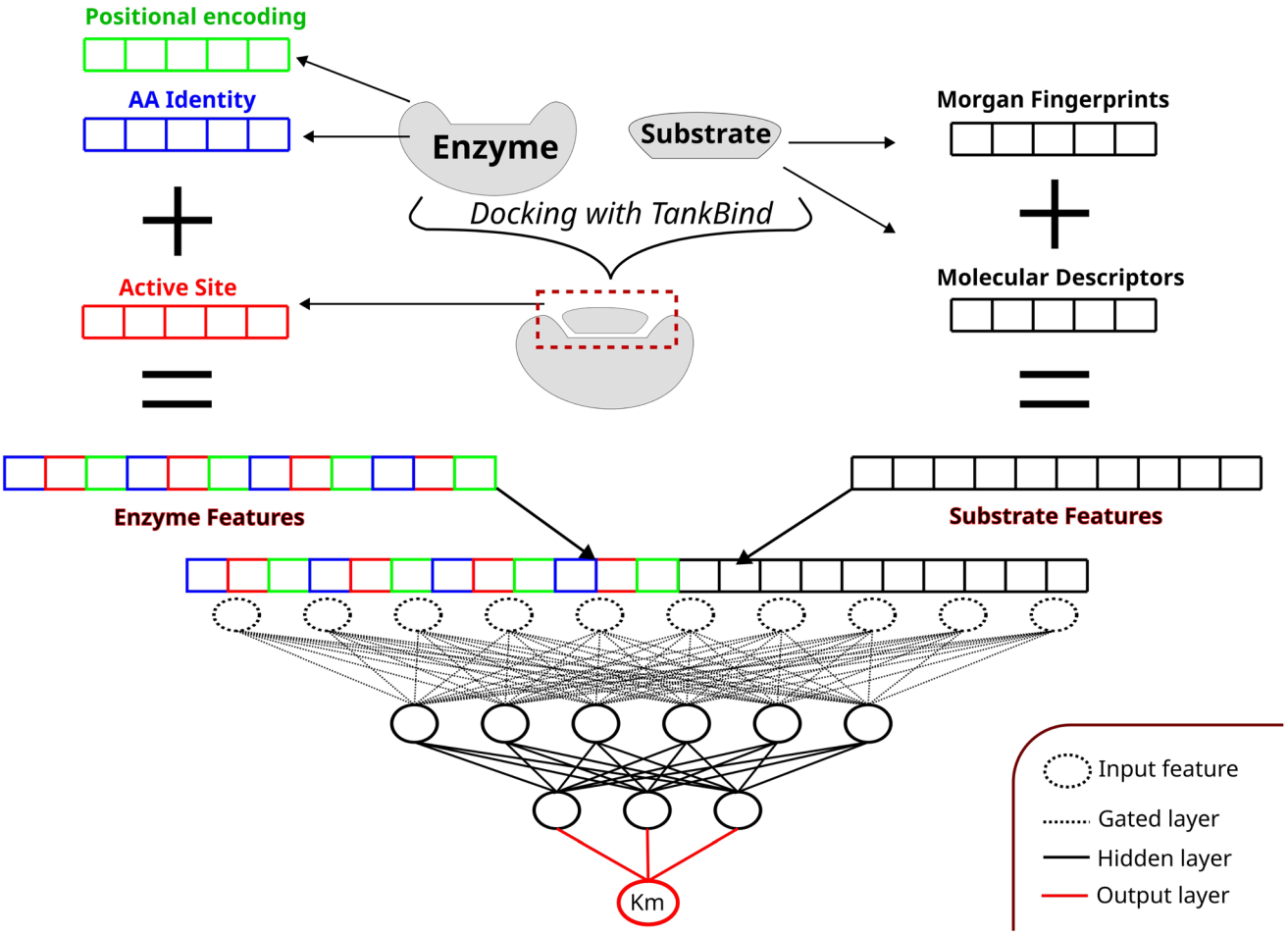

Fig. 1 Overview of features and deep learning model. The enzyme is represented as the concatenation of three features encoding each residue. First, the amino acid identity (AA Identity) feature, which encodes the residue identity as an ASCII character number; active site feature, which indicates if the residue is part of the enzyme–substrate interaction interface; the residue positional encoding as the position index of the residue in the enzyme sequence. Active site information is derived from TankBind.[18] The substrate is encoded by concatenating physicochemical molecular descriptors and Morgan fingerprints. The input for the model is the concatenation of the enzyme and substrate features. A multilayer perceptron (MLP) with one gated layer and two hidden layers with dropouts between each layer is fed with the input to compute the K_M value.

data set with available M-CSA annotations and compared the residues predicted as part of the active site by P2Rank and TankBind to the corresponding catalytic residues. For each enzyme, predicted and annotated residues were mapped to the same sequence coordinates, and overlap was quantified using precision, recall, and F1 score. The results obtained are shown in SI Fig. 1.

For the substrate, physicochemical molecular descriptors and molecular fingerprints obtained from SMILES strings were used as features. First, the SMILES strings were retrieved for the data set *via* the chemical identity resolver from the cactus group https://cactus.nci.nih.gov, (accessed in October 2025). The molecular descriptors and Morgan fingerprints were obtained from the SMILES strings using the python library RDKit (version 2024.3.5). All data points with exclusively zeros as molecular descriptor values were removed from the data set. The molecular descriptor "I_{pc}" had to be removed from the descriptors, as it contained values in the order of 10^{159}. The molecular descriptors were normalized with the PowerTransformer from the python library scikit-learn, then normalized between values of 0 and 1 using the min–max scaler. Morgan fingerprints were computed with 2048 bits and a radius of 2.

A total of 3072 enzyme features (3 features per residue and 1024 residues) and 2244 substrate features (196 molecular descriptors and 2048 bits for Morgan fingerprints) were collected for every point in the data set.

To test other features to represent enzymes, an additional set of models was trained where the enzyme and active site residues were encoded using Evolutionary Scale Models (ESM[31]). Following the methodology described in CatPred,[14] the enzyme was encoded using ESM2 [32] with a vector of length 1280 averaging the logits across all amino acid vectors. The active site was encoded in a similar fashion, where only the active site amino-acid vectors were averaged. The enzyme was encoded by concatenating the enzyme and the active site vectors into a vector of length 2560. The substrate features were not modified. To evaluate the active site information provided to the MLP, a set of models was trained using only ESM-encoded active-site residues.

2.3 Model architecture

A multi-layer perceptron (MLP) was used (Fig. 1). Every layer and model training was implemented using the PyTorch python framework. The first hidden layer was a gated layer:

$$y = \gamma(X) \odot \text{Sigmoid}(\psi(X)) \quad (1)$$

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

where X is the set of input features, y is the output of the gated layer, γ and ψ are two different sets of weights and biases made of 128 output neurons each. Here, the sigmoid function outputs values between 0 and 1, corresponding to learned importance scores for each feature. Recently, the attention mechanism[33] has proven very powerful and useful to learn statistical patterns through the weighing of each feature in relation to themselves. Here, an elemental implementation of this mechanism is used in the form of a gated layer. Conceptually, the gated layer (eqn (1)) attributes importance to each input by rescoring them using learned weight scores between 0 and 1, very similar to attention weights. The second and third layers were regular neural network hidden layers made of 64 and 16 neurons, respectively. Between each hidden layer, a dropout rate[34] of 20% and rectified linear unit (ReLU) as the activation function were implemented. Adam[35] was used as the optimizer, with a learning rate of 6.2×10^{-6}; a L2 regularization implemented in PyTorch as weight_decay parameter for the optimizer was set to 10^{-5}. The model optimized its parameters with batches of 32 samples randomly selected from the training data set. The loss function used was the mean squared error (MSE; eqn (3)).

Hyperparameter values for initial models (without active site information) were determined with a hyperparameter optimization performed using Optuna,[36] and served as basis to choose the hyperparameter values of the models presented. Two rounds of hyperparameter optimization were conducted: the first one with Optuna[36] grid search (SI Table 1) and the second one manually. The hyperparameters investigated were the number of neurons per layer, L2 regularization α term (weight decay), dropout and learning rates. The validation and training sets were randomly split into 5% and 95% portions of the data set, respectively. The initial MLP architecture contained 2048, 1024 and 512 neurons per layer after the first hyperparameter grid search. Analysis of the effect of each hyperparameter (SI Fig. 2) showed that the number of neurons per layer did not impact significantly performances. Thus, a lighter architecture was favored, and after manual hyperparameter finetuning, the MLP contained 128, 64 and 16 neurons per layer to focus on lower total parameter count and emphasize the data-driven approach.

For training, data was randomly split into 95% training and 5% validation set for a total of 10 models trained. The final prediction is the average value obtained from the 10 models to account for global uncertainty (see Section 2.9).

Several packages were used in this study including PyTorch (version 2.9.1), Pandas (version 2.3.3), RDKit (version 2024.3.5), matplotlib (version 3.10.8), among others, listed in the GitHub repository and executed on Python version 3.10.19.

2.4 Performance metrics

The performance of the models was tracked with three key metrics: MSE, R^2 and the Pearson's correlation coefficient (Pearson). The following equations describe these metrics:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (3)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}\left(y_i - \bar{\hat{y}}\right)^2} \quad (4)$$

$$\text{Pearson} = \frac{\sum_{i=1}^{n}(y_i - \bar{y})\left(\hat{y}_i - \bar{\hat{y}}\right)}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2 \sum_{i=1}^{n}\left(\hat{y}_i - \bar{\hat{y}}\right)^2}} \quad (5)$$

Here, y_i is the predicted K_M value and $\bar{y}$ its average, $\hat{y}_i$ the experimental K_M value and $\bar{\hat{y}}$ its average. n is the number of data points. All metrics were calculated with PyTorch functions (MSE: torch.nn.MSELoss, R^2: torcheval.metrics.R^2Score, Pearson: torchmetrics.RCorrCoef).

2.5 Ablation study

To investigate the impact of the different features in the model performance, the original data set containing the complete list of features was processed to build different data set versions, where each version omitted one specific feature or set of features. The omitted features were: residue identity, residue position in the primary sequence, residue in the active site or not, Morgan fingerprints, molecular descriptors, enzyme features (active site and residue identity), substrate features (Morgan fingerprints and molecular descriptors). For each version, the values of the omitted feature were set to 0. In the version with residue identity omitted, all the residues were replaced by alanine, at the exception of the residues which were part of the active site, effectively providing the active site identity while omitting the identity of other residues. Additionally, two sets of different model and feature configurations were investigated, where models were trained with and without a gated layer and with and without positional encoding as a feature.

2.6 Comparison with previous approaches

To provide a fair and transparent comparison with existing approaches, we evaluated AS4Km alongside the previously published models GraphKM,[11] MLAGO,[8] and CatPred[14] on the independent GMKM test set introduced by He and Yan.[11] This data set was originally constructed to avoid overlap with the GraphKM and MLAGO training sets and comprised 82 enzyme–substrate pairs, and has previously been used as a benchmark for model comparison in related work.

To ensure comparability across methods, we first removed any enzyme–substrate complexes overlapping with the GMKM data set from the AS4Km training set. For the benchmarking

against CatPred, we excluded GMKM entries that could potentially overlap with the CatPred training data, resulting in a final test set of 69 enzyme–substrate pairs. Importantly, this filtering was applied only to the test set, while AS4Km was trained on the non-overlapping data set, ensuring that no information leakage occurred from test to training data. We note that the final benchmark set is relatively small due to the cumulative overlap constraints across multiple independently trained models. This limitation should be taken into account when interpreting differences between model performances, particularly between closely performing methods.

All models were evaluated on this same filtered test set. Performance was assessed using R^2, RMSE, and Pearson correlation coefficient.

2.7 Baseline model

In addition to comparisons with learning-based models, we implemented a simple similarity-based baseline to contextualize performance gains. Specifically, we implemented a k-nearest-neighbor baseline based on substrate chemical similarity. For each test enzyme–substrate pair, we computed the Tanimoto similarity between the substrate Morgan fingerprints of the test sample and all substrate fingerprints in the training set. The $K = 10$ most similar training substrates were selected, and the predicted K_M value was defined as the median of their corresponding $\log_{10}$-transformed K_M values. If no nonzero similarity was observed between the test substrate and any training substrate, the global median $\log_{10} K_M$ value of the training set was used as a fallback prediction. The baseline was evaluated using the same pre-processing pipeline and evaluation metrics as AS4Km.

2.8 Explainable AI

Importance scores were computed from the weight matrices of the gated layer in order to investigate the impact of each feature in the neural network. Given W the matrix of the gated layer's weights, the importance score (IS) is defined as:

$$\mathrm{IS}_j = \mathrm{Softmax}\left(\sum_{i=1}^{m} |w_{ij}|\right) \tag{6}$$

where j is the feature's index in the MLP input vector and m represents the number of outputs from the gated layer.

The softmax function, also known as the normalized exponential function, is defined as:

$$\mathrm{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{7}$$

where z_i is the i-th element of the input vector and K is the vector length. It returns a vector where the sum of all its values is equal to 1.

2.9 Aleatoric and epistemic uncertainties

Two types of uncertainties arise from predicted values, as described earlier.[14] First, aleatoric uncertainty (AU): K_M measurements are subject to intrinsic data variance due to different experimental conditions, such as pH and temperature. For the epistemic uncertainty (EU), a predicted K_M value has a certain range of error due to the training data used, feature set, and choices in neural network architectures. AU creates a certain threshold on the predictive accuracy that cannot be improved, representing an intrinsic uncertainty always present in the prediction, while the EU can always be lowered by increasing the accuracy on unseen data. EU and AU are used in this work as an additional level of information by gauging each prediction's reliability.

To assess the variability in predictions, a repeated random sub-sampling validation procedure was used during the training process. Specifically, the data set was randomly split 10 times into 95% training and 5% validation sets. A separate model was trained for each split. During training, if the R^2 score on the validation set did not improve for 5 epochs, the training stopped and the best model was saved. Final predictions were obtained by averaging the outputs of the 10 independently trained models, which allowed for estimation of predictive uncertainty.

Both types of uncertainties were implemented in an approximate way: to account for AU, different experimental K_M measurements for the same enzyme–substrate complex were provided to the model without taking its geometric mean[7] nor maximum value,[11] effectively incorporating target variation during training. The EU was approximated through several outputs provided by different models. The global uncertainty (GU = AU + EU) was calculated through the standard deviations of the predictions from 10 different models.

3 Results

3.1 Data set and model architecture

The training set was obtained from the BRENDA and SABIO-RK databases. The preprocessing of the data was comparable to other studies presented earlier.[11] Preprocessing of data from the SABIO-RK database followed the same steps as shown in He and Yan.[11] The impact of each filtering step on the total entry count has been described in SI Table 2. Data from the BRENDA database was preprocessed using an in-house script. To evaluate models' performances, the HXKm data set was used as an independent test set. To shed light on the advantages and limitations of the data set, data distribution regarding K_M values, EC number and enzyme mutants was investigated (Fig. 2). The training set had a strong K_M value bias towards strong binders (values between 0 and 1 mM, Fig. 2A). Indeed, out of the 18 541 K_M values available after preprocessing, around 10 000 were in this range. The HXKm test set displayed the same pattern observed in the training set distribution, but on a smaller scale, given that only 420 values were available. Analysis of the enzyme class distribution revealed that a majority of the data pertained to EC1 (oxidoreductases) and EC3 (hydrolases) classes, which together accounted for over half of the unique substrates in the data set (Fig. 2C). This indicates that the model's predictive performance may be best for these well-represented classes. Furthermore, the composition of the

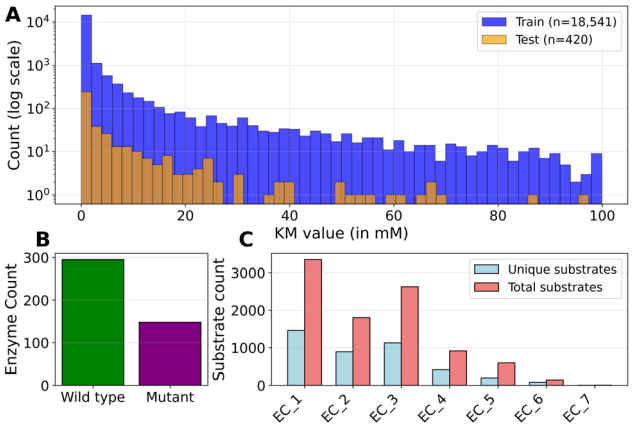

Fig. 2 Distribution of data points in the training and test sets. (A) Distribution of K_M values for the training and test sets. (B) Distribution of wild type and mutant enzymes in the HXKm[11] test set. (C) Distribution of total substrates and substrates unique to an enzyme class in the training set. EC: enzyme commission number.

test set was examined to assess its applicability (Fig. 2B). It contained a substantial proportion of wild type enzymes, providing a robust benchmark for evaluating the model's ability to generalize and predict the effects of genetically optimized, conserved sequences on enzyme kinetics.

In order to highlight the impact of the interaction interface information on predicting K_M values, a neural network with one gated layer and two hidden layers was used. Simple enzyme features such as residue identity and residue position in the primary sequence, combined with chemically-rich substrate features, formed an intricate, multidimensional representation encoding both structural and sequential information of the enzyme–substrate complex (Fig. 1). In order to represent the interaction interface, we introduced the residue's involvement in the active site as an additional feature in the model. One of the main challenges was the balance between the total number of features (5316), the number of data points (18 541) and the number of parameters in the model. As the model contained 1.370 million parameters (SI Fig. 3), which greatly exceeds the number of data points, minimizing overfitting was an important aspect during the model's weight optimization. By implementing a gated layer as a simplified version of the attention mechanism, together with L2 regularization and dropout, we sought to mitigate overfitting.

3.2 Performance of machine learning models

Explicit interaction interface information, encoded as a binary feature indicating if the residue is part of the active site, was investigated for ML prediction of K_M values. This representation enables a direct and interpretable encoding of enzyme–substrate interfaces without requiring complex latent embeddings or graph-based interaction modeling. The active site was predicted in two possible ways, as unconditioned on the substrate (using P2Rank, referred here as "Unconditioned AS"), or as conditioned on the substrate (using TankBind, referred here as "Conditioned AS"). The model with all features was trained using the "Conditioned AS" (SI Fig. 4 displays the performance for training and validation sets *vs.* number of epochs), and predictions were made using either the "Conditioned AS" or the "Unconditioned AS" as features. The model using "Unconditioned AS" as a feature performed slightly better than the model using "Conditioned AS" (as demonstrated in Section 3.3, Ablation study), likely due to predictions consistent with the most plausible active site. Indeed, P2Rank accounts for genetically optimized binding pockets, interpolated from the training set of a wide variety of protein–ligand data sets including natural and artificial ligands, single and multi-chain structures. Since the model using "Unconditioned AS" as a feature had the best performance, we adopted it as the main model in this work, and called it Active Site for K_M (AS4Km).

As an additional level of information, the correlation between predicted and experimental K_M values was visualized as a scatter plot (Fig. 3A) for the best model, AS4Km. As described in the methods, the predicted K_M value is an average of 10 different models. The standard deviation is the global uncertainty, accounting for aleatoric and epistemic uncertainties. Fig. 3A shows the average and standard deviation for each prediction, normalized between 0 and 1. When preprocessed, the distribution of K_M values changed, going from an exponential shape (Fig. 2A) to resembling a Gaussian distribution (Fig. 3B). Visually, predictions have varying uncertainties for the same K_M range. For instance, in the experimental K_M range of 0.2 to 0.4, standard deviations can vary from 0.05 to 0.1. For extremely low and high values of experimental K_M (close to 0 or 1), the standard deviations are in a small range, indicating overall a lower uncertainty. The distribution of experimental and predicted K_M values is similar (Fig. 3B).

In this work, only wild-type enzymes were kept in the training set. However, the HXKm test set includes mutant and wild-type enzymes (Fig. 2B). The performance of the ML model for both enzyme groups was compared (Fig. 3C), and a paired *t*-test demonstrated that the model's performance for the wild-type enzymes was significantly higher than for mutants, with a Pearson value of 0.609 for wild-type enzymes and 0.529 for

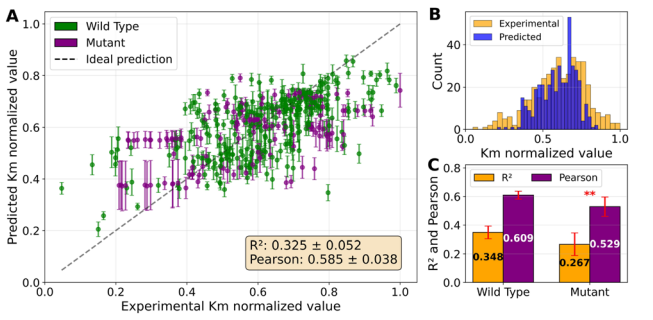

Fig. 3 Performance of AS4Km on the HXKm test set.[11] (A) Scatter plot of experimental *vs.* predicted K_M values, normalized in a range between 0 and 1. Each point and bar represents, respectively, the average and standard deviation for the predictions from 10 different models. (B) Distribution of predicted and experimental K_M values, normalized on a range between 0 and 1. (C) Comparison of performances for wild-type and mutant enzymes. Paired *t*-test shows a significant difference in performances (**: *p*-value < 0.01).

mutants. Furthermore, the standard deviations of predicted K_M values for mutants were usually higher than for wild-type enzymes (Fig. 3A), indicating more uncertainty in the predictions for mutants. No mutant enzymes were used during training of the model, which explains the worse performance for mutant enzymes. Training the model only on wild-type enzymes constraints predictions on evolutionary-conserved, functional enzymes. Antagonistically, mutants usually include enzymes that are the product of functional mutagenesis aimed at increasing the enzymatic activity or affinity for the substrate (which is mostly the case for the mutants in BRENDA, as shown in SI Fig. 5), or enzymes rendered inactive. Therefore, predicting K_M values for mutant enzymes using a model trained on wild-type enzymes is an extrapolation to another enzymatic context. Accordingly, when taking into account only the wild-type enzymes as the test set (interpolating rather than extrapolating), the model's predictions were more reliable and overall performances were better compared to predictions for the entire HXKm test set.

While the HXKm test set does not contain enzyme–substrate entries overlapping with the training set, the generalization capabilities to relatively distant entries are not clear. To address generalization to distant sequences and substrates, two sets of results were obtained based on HXKm's enzymes and substrates clustered at different similarity cutoff values, as presented in Fig. 4. A decreasing threshold value indicates an increasing dissimilarity between HXKm and the training set based on the enzyme or substrate. To evaluate generalization capability to distant enzyme sequences, Fig. 4A indicates at 99% similarity a Pearson of 0.57. At 80% similarity the Pearson is 0.59, staying around the same level of performance with a Pearson of 0.59 at the threshold of 60%. At a threshold of 40%, with entries whose enzyme sequences are the most dissimilar to the training set, the R^2 and Pearson reach 0.18 and 0.56. As a result, Pearson scores are without significant difference between 80% and 60% similarity, or with an increasing dissimilarity between 60% and 40%. Only a few enzymes satisfy the stringent cutoff of 40% similarity, showing the highest degree of variability and a significant impact on performance. When compared to the enzyme-clustered results using the AS4Km model trained without active site annotation (SI Fig. 6), the Pearson scores are overall lower (spanning from 0.06 to 0.55 for Pearson) between the threshold values of 40% and 99% similarity. Additionally, while the stringiest 40% cutoff is significant for models trained with and without the active site feature, the decrease of performances is stronger without the active site feature: there is roughly a 10 fold Pearson score decrease without the active site, while the Pearson decreases not even by a fold when active site information is provided. Overall, an increased generalization of AS4Km to distant enzyme sequences when active site information is included can be observed. Fig. 4B, compared to Fig. 4A, has a larger count decrease going from a threshold of 100 to 99% (420 to 55 data points for Fig. 4B and 420 to 391 data points for Fig. 4A), indicating that a vast majority of HXKm substrates has at least 99% similarity to at least one substrate in the training set. Furthermore, the decrease in performances is significant for all thresholds compared to the reference of

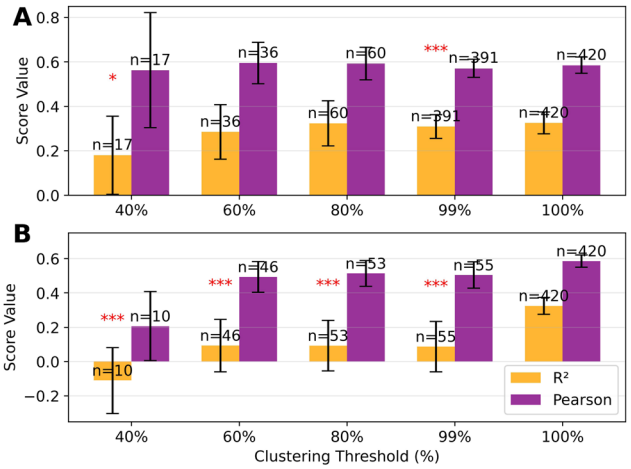

Fig. 4 Performance of AS4Km on the HXKm test set,[11] clustered using different thresholds of similarity. (A) Enzyme-based clustering performances. Scores computed based on different thresholds of enzyme sequence identity between HXKm and the training set using the MMseqs2[27] software. (B) Substrate-based clustering performances. Scores computed based on different thresholds of substrate similarity using the Tanimoto[28] score and Morgan fingerprints. After clustering the HXKm database entries whose similarity scores were lower than the given threshold, R^2 and Pearson scores were computed, depicting the generalization capability in function of enzyme sequence similarity or substrate similarity to the training set. A paired t-test was performed using as a reference the 100% threshold R^2 values. *** indicates p-value < 0.001; ** indicates p-value < 0.01 and * indicates p-values < 0.05.

100%, with a sharp Pearson decrease from 100% to 99% (going from 0.59 to 0.51). This pattern is also highlighted when no active site information is used, suggesting that this feature does not increase generalization to distant substrates.

3.3 Ablation study

Enzyme and substrate features were ablated in order to investigate the impact of each feature and the importance of the interaction interface information (Fig. 5). Performances after the ablation of each feature were computed using 10 different models (10 validation folds), which were trained using all features and the "conditioned AS" feature, and later modified to remove the ablated feature and make predictions. Statistical significance of the difference between conditions was computed using paired t-test. The model with the "conditioned AS" feature was used as a reference for the statistical tests. SI Fig. 7 shows metrics obtained with models trained using different sets of features, or trained without the gated layer.

To investigate the impact of the active site information, several conditions were constructed in the ablation study (Fig. 5; detailed metrics can be found in SI Table 3). First, "unconditioned AS" harbored predictions of active site residues by P2Rank, instead of predictions from TankBind, ablating the active site conditioned on the substrate in favor of the unconditioned one. Second, "AA identity free" harbored the ASCII encoding of amino acids involved in the active site predicted by TankBind alone, ablating the residue identity of the rest of the

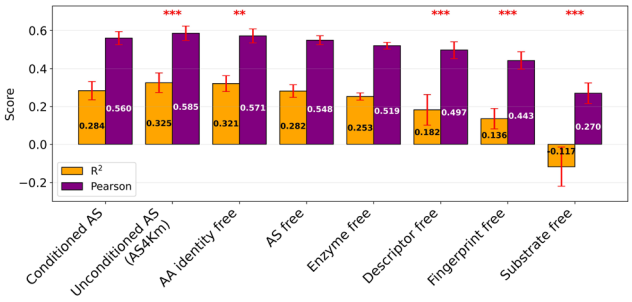

Fig. 5 Model performances for the HXKm test set[11] for the full model and after feature ablation. In "conditioned active site" ("Conditioned AS"), the full set of features is used, and TankBind[18] is used to predict active sites. In "unconditioned active site" ("Unconditioned AS"), P2Rank[19] is used to predict active sites. In "active site free" ("AS free"), no active site information is used. In "amino acid identity free" ("AA identity free"), the model preserves the identity of residues in the active site, while excluding the identity of the rest of the residues in the enzyme sequence. In "enzyme free", only residue positional encoding features are used as enzyme information. In "substrate free", all substrate features (the molecular descriptors and the Morgan fingerprints) are excluded. In "descriptor free", only Morgan fingerprints are kept as substrate information, and in "fingerprint free", only the molecular descriptors are kept as substrate information. Each bar represents the average and standard deviation for the predictions from 10 different models, which differed based on validation folds. The model with all features and conditioned active site feature is on the left-end ("Conditioned AS"), followed by models ordered by decreasing R^2 value. A paired t-test was performed for R^2 values for different conditions using "Conditioned AS" as the reference. *** for p-value < 0.001; ** for p-value < 0.01.

enzyme sequence. Lastly, the active site information was ablated completely ("AS free"). The inclusion of active site information alone ("AA identity free") enhanced generalization to unseen data by a significant margin (p-value < 0.01), going from average Pearson of 0.560 to 0.571. This suggests that active site information can be lost in a very sparse feature space, which includes information about the rest of the enzyme sequence. Therefore, sequence identity can be considered as "noise" that prevents enzyme–substrate interface information from being fully incorporated in the K_M value prediction. The best R^2 and Pearson scores were obtained for AS4Km, or the "unconditioned AS" model. Indeed, AS4Km demonstrated significantly higher performances (p-value < 0.001) than the model based on the conditioned active site alone. While unexpected, it is coherent, since the unconditioned annotation exhibits less variance, as it is determined exclusively by highly conserved, evolutionary optimized structural elements of the enzyme.[37] Meanwhile, conditioned active site prediction takes into account enzyme and substrate information, which may lead to reduced performances due to predictions not being fully consistent with conserved sequence patterns. In conclusion, active site information improved generalization to unseen data when combined with genetically optimized, highly conserved sequence patterns.

Apart from the active site information, the ablation of other enzyme features had a lower impact on performances (Fig. 5). First, there was not a notable impact in model performance from removing all enzyme features ("enzyme free" model). In SI Fig. 7A it can be observed that the sequence positional encoding feature did not impact generalization. Given the simplistic ASCII enzyme encoding scheme, which harbors no explicit physicochemical nor evolutionary information, the observed difference between a set of features with and without active site information could arise due to an inefficient set of features encoding amino acid identity only. To address this, ESM2 [32] was used as an alternative, evolutionary-informed, ML-derived enzyme embedding. As shown on SI Fig. 8, depicting a comparison of different sets of ESM features (enzyme and active site, enzyme only, active site only), the ML model showed significantly decreased performance when using enzyme and active site information encoded using ESM embeddings, compared to ASCII-based embeddings (going from Pearson = 0.56 to Pearson = 0.49). Additionally, the ML model using only ESM-encoded active site showed significantly decreased performances (going from Pearson = 0.56 to Pearson = 0.49).

Fig. 5 shows the significant impact of substrate features on K_M value prediction, compared to enzyme features, which is a result consistent with previous work.[7] Indeed, generalization dropped drastically when all substrate features were removed (from R^2 = 0.284 to negative R^2 value for the "substrate free" model). Several observations from this study shed light on the causes behind the importance of the substrate information for K_M prediction. One important aspect is the substrate's chemical space explored by the ML model, illustrated by substrate diversity in the training set. Fig. 2C shows the count of unique substrates per enzyme class, which indicates a low substrate diversity per enzyme class. Namely, given a class, roughly one half of its substrates interacted only with that enzyme class, while the other half of substrates was found to interact with other enzyme classes. This indicates a 50% chance that a substrate interacts only with an enzyme of a certain group. Further demonstrated by the feature space, when Morgan fingerprints ("fingerprint free") and molecular descriptors ("descriptor free") were ablated, the impact on performances was not as drastic as for the "substrate free" model, suggesting that neither of these chemical features are necessary for generalization on unseen complexes. Similarly, in a previous work aiming at predicting K_M values, MLAGO[8] demonstrated the high predictive performances on unseen data and the ability to predict K_M values using the substrate KEGG ID, without explicit chemical features. Coherent with our results, the "enzyme free" model (only substrate features used) learning curves demonstrated a pattern of enhanced generalization on the validation set, compared to training on the full feature set (SI Fig. 9), suggesting that substrate information alone enables generalization.

Feature Importance Scores (IS) were computed (Fig. 6) as described in the Explainable AI subsection of the methods. In theory, the gated layer puts more weight on certain features. IS values are derived from the gated layer's learned importance attribution to each feature. To investigate feature importance further, Fig. 6B groups the IS values based on feature type. Since the sum of all IS values is equal to 1, contributions can be compared relative to each other. Fig. 5B shows that enzyme

features had the highest contribution to model performance. Among substrate features, Morgan fingerprints had the highest IS value (SI Fig. 10). As a possible explanation, the overall contribution of each feature type can be influenced by the number of features that constitute a feature type: molecular descriptors had 196 features, while Morgan fingerprints had 2048 features. Taking into account the length of each feature type, the IS values plotted in Fig. 6 are in line with the results of the ablation study. Namely, that the substrate features carry most of the information useful for K_M prediction. From Fig. 6B, the divergence between enzyme and substrate weights, in comparison to Fig. 5, is explained by the number of enzyme features being higher than the number of substrate features.

3.4 Comparative benchmarking on the GMKM test set

Fig. 7 summarizes the performance of AS4Km, CatPred, GraphKM, MLAGO, and the similarity-based baseline on the independent GMKM test set.

The similarity-based baseline captures trends driven by substrate chemistry but performs substantially worse than all learning-based models (Pearson = 0.43), indicating that substrate similarity alone is insufficient for accurate K_M prediction. This underscores the importance of modeling enzyme information and enzyme–substrate interactions explicitly. Among previously published models, GraphKM achieved moderate performance on this benchmark (Pearson = 0.62), followed by MLAGO (Pearson = 0.53). AS4Km performed consistently stronger than GraphKM and MLAGO across all evaluated metrics (Pearson = 0.686 ± 0.016), despite relying on a comparatively simple, shallow neural architecture rather than a complex graph-based design. These results suggest that incorporating active-site information provides a strong inductive bias that can compensate for architectural simplicity, while maintaining competitive performance. Importantly, AS4Km achieved these results using a comparatively lightweight and interpretable representation strategy based on explicit active-site annotations, rather than highly parameterized latent interaction modeling. CatPred achieved the strongest overall performance on this test set (Pearson = 0.77). CatPred appears to be trained on a substantially larger data set (41 174 samples) compared to GraphKM (19 754 samples) and AS4Km (18 541 samples), despite using data from the same primary data sources. As the precise preprocessing and filtering steps for these additional training instances are not fully specified, strict comparability across models cannot be guaranteed. This difference in training set size and composition may have contributed to the observed performance gap.

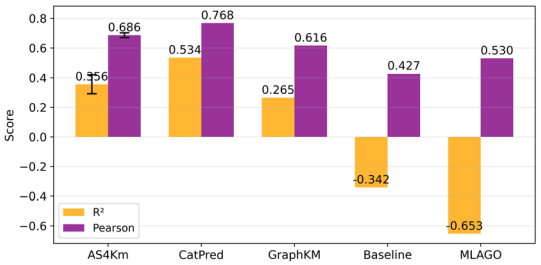

Fig. 7 Comparison of the performance of AS4Km to the performance of CatPred, GraphKM, MLAGO, and the similarity-based baseline on the independent GMKM test set. Performance was measured in R^2 (yellow bars) and Pearson correlation coefficient (purple bars).

Overall, AS4Km demonstrated competitive performance relative to more complex graph-based architectures and highly outperforms a similarity-based baseline, which highlights the effectiveness of incorporating active-site information within a lightweight and computationally efficient framework.

4 Discussion

Including enzyme–substrate interaction interface information in models to predict K_M values by incorporating active site information in the feature space allows better generalization to unseen data. Here, we present Active Site for K_M, AS4Km, an ML model that leverages interaction interface information to predict K_M values and presents competitive performance relative to prior approaches. Overall, the main contribution of this work is to demonstrate that explicit, sequence-level encoding of active-site information provides a useful inductive bias for K_M prediction, particularly for improving generalization to unseen or distant enzyme sequences.

The optimal setup for improved performances was achieved by training on all features and using active site annotation conditioned on the substrate, as provided by TankBind, then making predictions by relying on genetically optimized, substrate-unconditioned active site annotation, as provided by P2Rank. While these two methods were initially optimized for binding-site annotation, when applied to enzymes, they were able to identify catalytically active site residues (SI Fig. 1). Hence, genetically optimized, highly conserved sequence patterns combined to enzyme–substrate interface information were leveraged in order to achieve competitive performance, while maintaining a lightweight and interpretable framework.

TankBind and P2Rank were optimized for high-throughput binding site prediction, in accordance with their provided results. As described in the tool paper[19] P2Rank, a deep-learning based approach, outperforms energetic (SiteHound[38]) and geometrical tools (Fpocket[39]), as well as other deep-learning

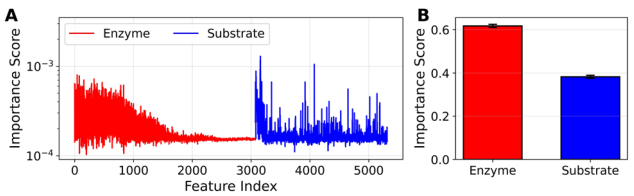

Fig. 6 Feature importance scores (IS) obtained using the conditioned active site model. (A) Histogram of the IS values (represented by the gated layer's weight) for each feature. Only one model was plotted for visualization. (B) Bar plot of the sum of IS values over indices corresponding to a given feature from either the enzyme or the substrate. The averages and standard deviations from 10 different models trained using all the features and the conditioned active site are shown. The sum of all IS values in panel A or in panel B is equal to 1.

tools (DeepSite[40]) in predicting the binding site location (72% of predicted binding sites were in a 4 Ångstrom distance away from the true binding site), while being very rapid (seconds timescale for a prediction, compared to minutes with other tools). Similarly, TankBind reported competitive results[18] in docking a given ligand to a protein (75.4% of ligand positions were below a 5 Ångstrom distance threshold from the true ligand position), with very fast computation time, offering in addition the ligand position and conformation inside the binding site (information not used in this study). Specifically, in the context of enzyme–substrate complexes, annotated binding site residues correspond to the active site, that is the residues which interact with the substrate and those that take part in the catalytic reaction.

Recent work has begun to incorporate catalytically relevant or active-site-aware representations for K_M prediction, including approaches such as SAKPE[17] and OmniESI,[16] which leverage structural priors or attention-based embeddings. OmniESI,[16] for instance, uses an attention mechanism to derive catalytically informed embeddings that may capture active-site information implicitly, while approaches such as SAKPE[17] similarly explore incorporating structural or functional priors into enzyme representations. Together, these methods reflect a growing recognition of the importance of enzyme–substrate interaction regions for kinetic prediction. In contrast, the approach presented here focuses on the explicit incorporation of active-site annotations as simple, sequence-level features, enabling a direct and interpretable representation of enzyme–substrate interaction interfaces without relying on complex architectures or latent embeddings. While the effectiveness of correctly identifying active-site residues was not directly assessed in this work, comparing feature spaces with and without active-site annotation demonstrated improved performance when such information is included.

Concerning the latest developments in K_M prediction, KmPred (LSTM/Transformer + XGBoost),[41] iESC,[42] and CPI-Pred[43] (message-passing neural network with cross-attention) achieve strong K_M prediction performance by combining protein language model embeddings with substrate features. They encode enzyme–substrate interactions implicitly through attention or deep architectures. In contrast, AS4Km demonstrates that explicitly providing binary active-site annotations as input features enables competitive performance with a simpler, more interpretable multilayer perceptron. This comparison suggests that, for K_M prediction, a direct inductive bias about the interaction interface can partially substitute for architectural complexity, and future models may benefit from combining explicit active-site information with more advanced deep learning frameworks.

Notably, no Cartesian coordinates were used here. While physically plausible positioning of atoms at the enzyme–substrate interface may give meaningful insights, primary sequence residue annotation of active sites enables an alternative, easy to compute representation of the interaction interface. Therefore, high-quality, high-throughput enzyme–substrate active site annotation has the potential to assist big-data ML applications for enzymology. Indeed, by prioritizing active site annotation instead of precise atomic position computation for enzyme–substrate complexes, docking may be replaced without a large information loss.

In this work, we did not include mutant enzymes in the training set. Accounting for mutant enzymes can be challenging, since a single residue mutation can have a strong impact in catalysis and substrate binding, which needs to be properly weighted during model training. To account for such subtle modifications, active site annotation needs to be highly sensitive to single-point mutations. However, the ability of TankBind and P2Rank to predict active sites in mutant enzymes was not investigated by us or in the respective publications.[18,19] Moreover, experimental measurements acquired using mutated enzymes, present in databases such as BRENDA[5] and SABIO,[6] tend to include positive data, with results showing increased activity for mutant enzymes.[41,42] As a result, mutant enzymes increase the bias of the data set towards positive data, as observed in the distribution of K_M values for BRENDA according to enzyme type (SI Fig. 5). Consequently, if no negative data are available in the training set, ML models trained on these data sets have a high risk of "hallucinating" a false positive. One possible solution is augmenting catalytic data sets with negative data. Furthermore, the functional impact of a mutated enzyme can be subtle relative to the range of all possible K_M values. As shown in kinetic ML models focusing on mutation effects,[43,44] an absolute R^2 score may be biased, and is not necessarily indicative of the model's ability to predict the effect of mutations. As previously suggested,[43,44] a better approach to evaluate the impact of a single residue modification is a relative comparison between the K_M values of matched wild-type and mutated enzyme pairs.

The impact of substrate features on predictive performances was investigated in previous works[7,11] and highlighted in the ablation study presented here. Given how the BRENDA and SABIO-RK databases were built, they contain functional enzyme–substrate interactions, or in the case of ML application, positive data. Since ML predictions are usually interpolated from training data, performances on negative data, that is non-active enzyme–substrate complexes, were not investigated in this study, given that mostly positive data are available in the training set. Furthermore, ML algorithms extract statistical patterns from a large amount of data, and the path of least resistance is to associate a K_M measurement with a substrate. Finally, when clustering test substrates based on their similarity to the training set (Fig. 4B), a large proportion of data is lost going from 100 to 99%, showing that the majority of test substrates are at least 99% similar to at least one of the substrates found in the training set. In practice, many organic molecules metabolized by enzymes share a lot of substructures, such as aromatic rings. As of now, active site information improves generalization to distant enzyme sequences, but in order to improve generalization to new and distant substrates, trained models should be sensitive to low substrate structural variation. By increasing the number of negative data in such databases, the model is forced to recognize non-active enzyme–substrate complexes. In such contexts, predictive performances would rely less on the identity of the substrate, but rather on

physico–chemical parameters of the interaction interface, given that one substrate could be active with one enzyme and inactive with another.

Active sites are highly conserved in protein primary sequences,[37,45,46] especially in enzymes, where active sites are functionally optimized through evolution.[47] One such example is the catalytic triad[48] composed of serine, histidine and aspartic acid, a common motif for generating a nucleophilic residue for catalysis across various hydrolase and transferase enzymes, despite differences in their overall protein folds and evolutionary origins. Additionally, binding pocket residue sequences are reinforced in *de novo* generated protein structures by repeating genetically-optimized patterns,[49] reducing active site sequence diversity in newly obtained enzyme structures. While this high degree of conservation is a powerful signal for identifying potential active sites, it also presents a fundamental challenge for machine learning models: the feature space of known enzyme–substrate interfaces is inherently limited and biased toward well-studied, conserved motifs. Consequently, models trained on these data may lack the generalizability to recognize novel or divergent enzymatic functions. To overcome this limitation and truly explore the full functional landscape of enzymes, data sets need to be augmented with a much broader and more diverse set of enzyme–substrate interaction data, capturing a wider spectrum of structural and chemical features at the complex interaction interface.

Compared to other tools, AS4Km outperformed the Kroll *et al.*[7] model and GraphKM[11] when evaluated on the GMKM test set. The active site feature improved the K_M prediction for unseen enzyme sequences thanks to the overall conserved active site residue patterns. Indeed, results for the test set, when clustered by enzyme sequence similarity, demonstrated that the active site feature improved generalization, and helped the model to keep a good performance even at a stringent 40% cutoff similarity for enzyme sequences. Additionally, when comparing our simplistic set of ASCII features to ESM embeddings, we show that ASCII features perform better than ML-derived, abstract features. Furthermore, binary annotation of the active site was shown to be more efficient and generalize better compared to ESM-encoded active residues, emphasizing the importance of a data-driven approach for catalytic ML tasks. As shown in previous work addressing the effectiveness of ESM embeddings for different tasks,[31] while providing information-rich representations, its effectiveness varies based on the scarcity of training data and the applicability domain. Simply put, if not enough data is available, ESM models can show overfitting patterns, especially for K_M prediction, as mentioned in the CatPred[14] paper and other works where simpler sets of features or architectures perform better than ESM-based predictions.[50,51] Taken together, residue-level encoding using ASCII embedding combined with binary active site information is sufficient to interpolate genetic and physicochemical parameters for K_M prediction. In comparison, GraphKM takes as input ESM models to encode protein features and incorporates substrate structure information through graph neural networks. Thanks to a sophisticated ML architecture, separate enzyme and substrate features may be combined into abstract enzyme–substrate interface information deep into the layers. Here, AS4Km instead relies on explicit active-site encoding to directly represent enzyme–substrate interaction interfaces using simple sequence-level features, which enabled competitive generalization performance without requiring highly complex latent architectures. Importantly, while the incorporation of explicit active-site information improves performance within the present lightweight framework, it is not guaranteed that the same type of feature integration would produce similar improvements in more complex, highly optimized architectures such as CatPred[14] or GraphKM.[11] Rather, this work demonstrates that explicit active-site encoding provides a useful and interpretable inductive bias in models where such interaction information is otherwise not directly represented.

Given the number of parameters of the model (1.370 million parameters, SI Fig. 3), the size of the training set (18 541 entries) and the number of input features (5316 features), overfitting was a concern, and here it was mitigated with dropout and gated layers, as well as regularization. Ideally, instead of fully connected layers, an architecture leveraging attention mechanism[33] would be more fitted in this context. Specifically, by tokenizing each set of residue features, absolute residue positional bias would be limited by addressing each amino acid to any other in the sequence during key-query (KQ) matrix multiplication. As a result, even distant sequence patterns would be better leveraged compared to an MLP. While true attention mechanism has its advantages in this context, we opted for a fully connected neural network, given that it requires less parameters. Additionally, given the exploratory nature of this study, a robust explainable AI approach leveraging importance scores was imperative to evaluate the impact of active site information, which is less evident to compute using attention scores derived from K, Q, V abstract projections. Consequently, a MLP architecture leveraging gating mechanism was preferred in this context, but further architectures leveraging active site information should incorporate attention mechanism for better performances. The impact of the active site feature could be further improved in the ML models by increasing the number of entries drastically to further mitigate overfitting. In the present work, active site annotation is represented as a binary embedding, encoding 3D information of neighboring substrate–residue interface in a 1D vector feature. Future efforts could focus on physics and evolutionary more informed active site information through explicit structural features: distance-weighted contributions such as radial basis functions,[52] or residue-contact frequency tables similar to PSSM encodings.[53]

5 Conclusions

In this work, we present Active Site for K_M, AS4Km, a deep learning model that leverages interaction interface information of enzyme–substrate complexes to predict K_M values within a lightweight framework. The results show that interface information in the form of active site annotation improves generalization of the lightweight framework to unseen data, especially to more distant enzyme sequences. As indicated in the comparative ablation study, enzyme–substrate interface

featurization consistent with genetically optimized active sites offers better performances compared to variable active site prediction, conditioned on the substrate. Our results also suggest that locally resolved chemical features of the substrate are not leveraged to generalize on unseen data, but rather the identity of the substrate is sufficient, in agreement with previous studies.[7,8] Enriching the current databases with non-active enzyme–substrate complexes could help to make models less dependent on substrate features.

Author contributions

D. L.: conceptualization, data curation, formal analysis, investigation, methodology, visualization, validation, writing – original draft, software, supervision. L. S.: data curation, formal analysis, investigation, methodology, software, validation, writing – review & editing. A. N. A.: conceptualization, writing – review & editing, visualization, supervision, funding acquisition, resources.

Conflicts of interest

There are no conflicts to declare.

Data availability

All data supporting the findings of this study are openly available at: https://doi.org/10.5281/zenodo.20392157 and the code saved on GitHub and available at: https://doi.org/10.5281/zenodo.20393126.

Supplementary information (SI) is available. See DOI: https://doi.org/10.1039/d6dd00094k.

Acknowledgements

D. L. and A. N. A. thank funding from DFG under Germany's Excellence Strategy – EXC 2008/1-390540038 – UniSysCat. D. L. was funded by the Einstein Foundation Berlin. A. N. A. thanks funding from the the Brazilian Biosciences National Laboratory (LNBio), part of the Brazilian Center for Research in Energy and Materials (CNPEM), a private non-profit organization under the supervision of the Brazilian Ministry of Science, Technology, and Innovations (MCTI). The authors thank prof. Silke Leimkühler (University of Potsdam) and prof. Peter Neubauer (Technical University of Berlin) for helpful discussions. We thank the developers and maintainers of PyTorch, NumPy, Pandas, RDKit and of python packages used in this study.

Notes and references

1 P. K. Robinson, *Essays Biochem.*, 2015, **59**, 1–41.
2 G. M. Landwehr, J. W. Bogart, C. Magalhaes, E. G. Hammarlund, A. S. Karim and M. C. Jewett, *Nat. Commun.*, 2025, **16**, 1–13.
3 L. P. Merlicek, J. Neumann, A. Lear, V. Degiorgi, M. M. de Waal, T.-S. Cotet, A. J. Mulholland and H. A. Bunzel, *Angew. Chem., Int. Ed.*, 2025, **64**, e202507031.
4 N. Takaoka, S. Sanoh, S. Ohta, M. Esmaeeli, S. Leimkühler, M. Kurosaki, M. Terao, E. Garattini and Y. Kotake, *Arch. Biochem. Biophys.*, 2022, **715**, 109099.
5 I. Schomburg, L. Jeske, M. Ulbrich, S. Placzek, A. Chang and D. Schomburg, *J. Biotechnol.*, 2017, **261**, 194–206.
6 U. Wittig, M. Rey, A. Weidemann, R. Kania and W. Müller, *Nucleic Acids Res.*, 2017, **46**, D656–D660.
7 A. Kroll, M. K. M. Engqvist, D. Heckmann and M. J. Lercher, *PLoS Biol.*, 2021, **19**, e3001402.
8 K. Maeda, A. Hatae, Y. Sakai, F. C. Boogerd and H. Kurata, *BMC Bioinf.*, 2022, **23**, 1–17.
9 S. Kobayashi, *Trans. Jpn. Soc. Artif. Intell.*, 2009, **24**, 147–162.
10 M. Kanehisa, M. Furumichi, M. Tanabe, Y. Sato and K. Morishima, *Nucleic Acids Res.*, 2017, **45**, D353–D361.
11 X. He and M. Yan, *BMC Bioinf.*, 2024, **25**, 135.
12 T. N. Kipf and M. Welling, Semi-Supervised classification with graph convolutional networks, https://arxiv.org/abs/1609.02907, 2016.
13 P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò and Y. Bengio, Graph Attention Networks, https://arxiv.org/abs/1710.10903, 2017.
14 V. S. Boorla and C. D. Maranas, *Nat. Commun.*, 2025, **16**, 1–17.
15 J. G. Greener and K. Jamali, *Fast protein structure searching using structure graph embeddings*, Cold spring harbor laboratory technical report, 2022.
16 Z. Nie, H. Zhang, H. Jiang, Y. Liu, X. Huang, F. Xu, J. Fu, Z. Ren, Y. Tian, W.-B. Zhang and J. Chen, OmniESI: A unified framework for enzyme-substrate interaction prediction with progressive conditional deep learning, *arXiv*, 2025, preprint, arXiv:2506.17963, DOI: 10.48550/arXiv.2506.17963.
17 J.-H. Qiu, Z. Lin, K.-W. Chen, T.-Y. Sun, X. Zhang, L. Yuan, Y. Tian and Y.-D. Wu, *SAKPE: A Site Attention Kinetic Parameters Prediction Method for Enzyme Engineering*, 2025.
18 W. Lu, Q. Wu, J. Zhang, J. Rao, C. Li and S. Zheng, *NeurIPS*, 2022.
19 R. Krivák and D. Hoksza, *J. Cheminf.*, 2018, **10**, 39.
20 J. Sim, D. Kim, B. Kim, J. Choi and J. Lee, *Curr. Opin. Struct. Biol.*, 2025, **92**, 103020.
21 J. Abramson, J. Adler, J. Dunger, R. Evans, T. Green, A. Pritzel, O. Ronneberger, L. Willmore, A. J. Ballard, J. Bambrick, S. W. Bodenstein, D. A. Evans, C.-C. Hung, M. O'Neill, D. Reiman, K. Tunyasuvunakool, Z. Wu, A. Žemgulytė, E. Arvaniti, C. Beattie, O. Bertolli, A. Bridgland, A. Cherepanov, M. Congreve, A. I. Cowen-Rivers, A. Cowie, M. Figurnov, F. B. Fuchs, H. Gladman, R. Jain, Y. A. Khan, C. M. R. Low, K. Perlin, A. Potapenko, P. Savy, S. Singh, A. Stecula, A. Thillaisundaram, C. Tong, S. Yakneen, E. D. Zhong, M. Zielinski, A. Žídek, V. Bapst, P. Kohli, M. Jaderberg, D. Hassabis and J. M. Jumper, *Nature*, 2024, **630**, 493–500.
22 J. Wohlwend, G. Corso, S. Passaro, N. Getz, M. Reveiz, K. Leidal, W. Swiderski, L. Atkinson, T. Portnoi, I. Chinn, J. Silterra, T. Jaakkola and R. Barzilay, *Boltz-1 Democratizing Biomolecular Interaction Modeling*, Cold spring harbor laboratory technical report, 2024.

23 H. Stärk, O.-E. Ganea, L. Pattanaik, R. Barzilay and T. Jaakkola, EquiBind: Geometric deep learning for drug binding structure prediction, https://arxiv.org/abs/2202.05146, 2022.
24 G. Corso, H. Stärk, B. Jing, R. Barzilay and T. Jaakkola, DiffDock: Diffusion steps, twists, and turns for molecular docking, https://arxiv.org/abs/2210.01776, 2022.
25 E. Nittinger, Ö. Yoluk, A. Tibo, G. Olanders and C. Tyrchan, *Artif. Intell. Life Sci.*, 2025, **8**, 100136.
26 M. Buttenschoen, G. M. Morris and C. M. Deane, PoseBusters: AI-based docking methods fail to generate physically valid poses or generalise to novel sequences, https://arxiv.org/abs/2308.05777, 2023.
27 F. Kallenborn, A. Chacon, C. Hundt, H. Sirelkhatim, K. Didi, S. Cha, C. Dallago, M. Mirdita, B. Schmidt and M. Steinegger, *Nat. Methods*, 2025, **22**, 2024–2027.
28 D. Bajusz, A. Rácz and K. Héberger, *J. Cheminf.*, 2015, **7**, 20.
29 J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli and D. Hassabis, *Nature*, 2021, **596**, 583–589.
30 J. Fleming, P. Magana, S. Nair, M. Tsenkov, D. Bertoni, I. Pidruchna, M. Q. Lima Afonso, A. Midlik, U. Paramval, A. Žídek, A. Laydon, O. Kovalevskiy, J. Pan, J. Cheng, Ž. Avsec, C. Bycroft, L. H. Wong, M. Last, M. Mirdita, M. Steinegger, P. Kohli, M. Váradi and S. Velankar, *J. Mol. Biol.*, 2025, **437**, 168967.
31 Q. Yang, J. Yu and J. Zheng, *Quant. Biol.*, 2026, **14**, e70013.
32 Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, A. Dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido and A. Rives, *Science*, 2023, **379**, 1123–1130.
33 A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, Attention is all you need, https://arxiv.org/abs/1706.03762, 2017.
34 S. Nitish, Dropout: A simple way to prevent neural networks from overfitting, *JMLR*, 2014, **15**(No 1).
35 D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, https://arxiv.org/abs/1412.6980, 2014.
36 T. Akiba, S. Sano, T. Yanase, T. Ohta and M. Koyama, *The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2623–2631.
37 O. Lemke, B. M. Heineike, S. Viknander, N. Cohen, F. Li, J. L. Steenwyk, L. Spranger, F. Agostini, C. T. Lee, S. K. Aulakh, J. Berman, A. Rokas, J. Nielsen, T. I. Gossmann, A. Zelezniak and M. Ralser, *Nature*, 2025, **644**, 280–289.
38 M. Hernandez, D. Ghersi and R. Sanchez, *Nucleic Acids Res.*, 2009, **37**, W413–W416.
39 V. Le Guilloux, P. Schmidtke and P. Tuffery, *BMC Bioinf.*, 2009, **10**, 168.
40 J. Jiménez, S. Doerr, G. Martínez-Rosell, A. S. Rose and D. Fabritiis, *Bioinformatics*, 2017, **33**, 3036–3042.
41 M. Rakonjac Ryge, M. Tanabe, P. Provost, B. Persson, X. Chen, C. D. Funk, A. Rinaldo-Matthis, B. Hofmann, D. Steinhilber, T. Watanabe, B. Samuelsson and O. Rådmark, *Arch. Biochem. Biophys.*, 2014, **545**, 179–185.
42 L. Llanos, R. Briones, A. Yévenes, F. D. González-Nilo, P. A. Frey and E. Cardemil, *FEBS Lett.*, 2001, **493**, 1–5.
43 A. Kroll and M. J. Lercher, *Biol. Methods Protoc.*, 2024, **9**, bpae061.
44 Y. Rousset, A. Kroll and M. J. Lercher, Overcoming systematic data biases enables accurate prediction of enzyme kcat fold-changes for computational protein design, *BioRxiv*, 2026, preprint, DOI: 10.64898/2026.01.23.701068.
45 S. B. King and M. Singh, *PLoS Comput. Biol.*, 2023, **19**, e1010966.
46 G. Abrusán and J. A. Marsh, *Cell Rep.*, 2018, **22**, 3265–3276.
47 B. R. Jack, A. G. Meyer, J. Echave and C. O. Wilke, *PLoS Biol.*, 2016, **14**, e1002452.
48 G. Dodson, *Trends Biochem. Sci.*, 1998, **23**, 347–352.
49 A. A. Bekar-Cesaretli, O. Khan, T. Nguyen, D. Kozakov, D. Joseph-Mccarthy and S. Vajda, *J. Chem. Inf. Model.*, 2024, **64**, 960–973.
50 A. Harmalkar, R. Rao, Y. Richard Xie, J. Honer, W. Deisting, J. Anlahr, A. Hoenig, J. Czwikla, E. Sienz-Widmann, D. Rau, A. J. Rice, T. P. Riley, D. Li, H. B. Catterall, C. E. Tinberg, J. J. Gray and K. Y. Wei, *mAbs*, 2023, **15**, 2163584.
51 K. K. Yang, N. Fusi and A. X. Lu, *Cell Systems*, 2024, **15**, 286–294e2.
52 K. T. Schütt, P.-J. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko and K.-R. Müller, *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2017, pp. 992–1002.
53 D. F. Marzella, G. Crocioni, T. Radusinović, D. Lepikhov, H. Severin, D. L. Bodor, D. T. Rademaker, C. Lin, S. Georgievska, N. Renaud, A. L. Kessler, P. Lopez-Tarifa, S. I. Buschow, E. Bekkers and L. C. Xue, *Commun. Biol.*, 2024, **7**, 1661.