Towards Deep Learning Models of Metabolism

by

Itamar Chinn

S.B., Massachusetts Institute of Technology (2022)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2024

Authored by: Itamar Chinn
Department of Electrical Engineering and Computer Science
May 16, 2024

Certified by: Regina Barzilay
Distinguished Professor for AI and Health
Thesis Supervisor

Accepted by: Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Towards Deep Learning Models of Metabolism

by

Itamar Chinn

Submitted to the Department of Electrical Engineering and Computer Science
on May 16, 2024 in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

ABSTRACT

Enzymes play a critical role in catalyzing the chemical reactions that underpin metabolic processes in living organisms. Despite their importance, a vast majority of enzymes remain uncharacterized, limiting our understanding of their potential roles in metabolism and disease. This thesis aims to address this gap by leveraging recent advancements in protein and molecular modeling to predict the outcomes of enzymatic reactions and identify functions of unannotated enzymes. Two key contributions are highlighted. Firstly, a graph-based forward synthesis prediction model is introduced, which relies only on the molecular structure of the substrates and the enzyme's primary sequence. By capturing the biochemical interaction between enzyme residues and substrate atoms, the model achieves better generalization to new chemistry, demonstrating significant improvements in predicting unseen products and showcasing its potential for drug metabolism prediction. The second contribution is CLIPZyme, a contrastive learning method for virtual enzyme screening that frames the task of identifying enzymes catalyzing a reaction of interest as a retrieval problem. CLIPZyme outperforms the baseline approach of screening enzymes via their enzyme commission (EC) number. The combination of CLIPZyme with EC prediction consistently yields improved results over either method alone. Both of these contributions aim to provide the initial building blocks to model entire complex metabolic networks with downstream applications including metabolic engineering and drug discovery.

Thesis supervisor: Regina Barzilay
Title: Distinguished Professor for AI and Health

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my wife, Maayan. She is my inspiration, my guiding light, and the reason I persevere each day – without her unwavering love and support, this work would never have come to fruition.

I am incredibly grateful for my advisor Regina, who has believed in me since I was an undergraduate. Thank you, Regina, for instilling in me the courage and self-confidence to tackle the world's greatest challenges, even when they seem insurmountable. Your pursuit of perfection through meticulous attention to detail is inspiring, and has shaped me as a student and researcher.

The completion of this thesis would not have been possible without Peter, my dear friend and research partner. Your dedication and drive are truly inspiring, and I am privileged to have had the opportunity to work so closely with you. I am amazed by your exceptional work ethic, and deeply grateful for our countless hours of collaboration – I continue to learn from you every day.

I also want to extend my heartfelt thanks to my lab mates who have been a constant source of support. In particular, I thank my close friend Jeremy: Your creativity and intellect are matched only by your genuine kindness. Your guidance was key to shaping this project.

Finally, I am so grateful for the support of my family. To my parents, David and Nili: thank you for instilling in me a love of learning from an early age and for your unwavering belief in my potential. To my sisters, Nurit and Michal: I could not ask for more supportive, loving, and inspiring sisters. You are both truly role models to me, and I feel that I have become the person I am from being your brother.

Contents

Chapter 1

Introduction

The intricate web of metabolic processes forms the foundation of biological function, underpinning everything from cellular energy production to the synthesis of essential biomolecules. At the heart of metabolism lie enzymes, the catalysts driving these chemical reactions. Their role extends beyond mere facilitation of biochemical transformations; they are the linchpins in the regulation and efficiency of metabolic pathways. Despite their critical importance, a vast majority of enzymes remain uncharacterized, their potential roles in metabolism and disease largely unexplored. This gap in our understanding represents a significant barrier to both fundamental biological insights and the practical application of enzymatic reactions in industry.

Traditionally, *in silico* modeling of metabolism has relied on genome-scale metabolic models (GEMs). These models, built upon years of research, provided valuable insights through methods like flux balance analysis (FBA) [1]. However, FBA often falls short in capturing the nuances of protein function and changes under perturbations which lead to effects on downstream metabolic fluxes. Recent advancements in protein and molecular modeling offer a promising avenue to bridge this gap. We now have effective and accurate methods to model protein-ligand binding [2], protein structure [3, 4] and protein localization [5, 6]. By accurately simulating enzymatic reactions, and capturing the impact of mutations or

molecular perturbations, we unlock the potential for many impactful downstream applications. Such applications include metabolic engineering, for example via novel biosynthesis routes, and improved target identification and validation, for example through enzyme deorphaning.

Despite huge advances in protein and molecular modeling, challenges remain, particularly in predicting enzymatic reaction outcomes and functions of unannotated enzymes – both active areas of research. Forward and retro-synthesis models have focused on general chemical reactions, where there are rich sources of data such as the USPTO reaction dataset [7, 8, 9]. Attempts to transfer these methods to enzymatic reactions have lacked success. Similarly, attempts to computationally categorize enzymes have focused on predicting Enzyme Commission (EC) numbers, which provides only a partial solution. Even for highly documented reaction classes EC prediction models lack useful performance. For novel enzymes and reactions, no method has found to be effective as EC numbers for these reactions do not exist. As such, metabolic engineering continues to rely heavily on experimental methods to optimize pathways [10] and enzyme function elucidation remains time, cost and labor intensive, limiting the number of enzymes that can be reasonably screened. Recent releases of highly curated datasets such as ECReact [11] and EnzymeMap [12] provide an opportunity to significantly improve these methods to a point where such applications become realistic.

The broader objective of this work is to lay the groundwork for modeling entire metabolic networks, addressing the critical gap posed by uncharacterized enzymes. This thesis presents two building blocks towards this goal: a novel method for predicting the products of biocatalyzed reactions and a new method and approach to virtually screening enzymes and thereby characterizing their function.

This thesis is based on the following works:

> **Graph-Based Forward Synthesis Prediction of Biocatalyzed Reaction.** Peter Mikhael*, Itamar Chinn*, and Regina Barzilay. Generative and Experimental Perspectives for Biomolecular Design Workshop at the 12th International Conference on Learning Representations (GEM Workshop,

ICLR 2024). [13]

CLIPZyme: Reaction-Conditioned Virtual Screening of Enzymes. Peter Mikhael*, Itamar Chinn*, and Regina Barzilay. Forty-first International Conference on Machine Learning (ICML 2024). [14]

Chapter 2

Graph-Based Forward Synthesis Prediction of Biocatalyzed Reaction

2.1 Motivation

A key computational task in biocatalysis is predicting the products of a reaction from an enzyme and its substrates. *In silico* methods for this task enable new opportunities in enzyme discovery, therapeutic development, and metabolic engineering. Current machine learning models have shown initial feasibility at automating this process; however, thus far they rely on information that may not be available for novel chemistry (e.g. Enzyme Commission number). As a result, this limits their practical use as an alternative to experimental methods.

The goal of our work is to improve the generalization capacity of these models to new chemistry. To this end, we assume access to only the molecular structure of the substrates and the enzyme primary sequence, without any additional information. In predicting the products of spontaneous chemical reactions, graph-based methods have outperformed both language model and rule-based approaches. These methods, however, fail to take into consideration the enzyme and therefore experience a significant drop in performance on biocatalyzed reactions. We hypothesized that better generalization can be achieved by a mechanistically-inspired

model that captures the biochemical interaction between the enzyme residues and substrate atoms. We demonstrate that these interactions can be learned through a multi-headed cross attention using graph convolutions to encode the substrates as 2D molecular graphs and a protein language model to encode the enzyme's amino acid sequence.

In the context of drug design, the metabolism of small molecule drugs impacts their efficacy, toxicity, and mechanism of action. For example, Fenofibrate, which is used to treat high cholesterol, must first be metabolized into fenofibric acid by liver carboxylesterase 1 in order to become active. Therefore, we consider phase II metabolism of small molecule drugs as a potential real-world application and a prime example of generalization to a novel chemical space. Specifically, we use a dataset of drug reactions from DrugBank to predict the products generated by the reaction of a drug with its target enzyme. This is an interesting and challenging generalization scenario since the chemical distribution of therapeutics differs significantly to that of metabolites on which these models are trained.

We develop our model using the EnzymeMap Version 1 dataset, consisting of 103,120 pairs of atom-mapped reactions and UniProt-SwissProt proteins. We demonstrate a significant improvement in predicting unseen products on a standard product split. For instance, we obtain 89% accuracy in generating correct products when evaluating the top 10 predictions and outperform current methods that range between 50%-70%. The comparison between our method and previous methods highlights the importance of adequate enzyme encoding. Ignoring the enzyme altogether or utilizing the protein EC numbers leads to significantly worse performance [11, 15]. Finally we show comparable improvements using a dataset from DrugBank.

2.2 Background

Enzyme Modeling Central to correctly predicting the product of a biochemical reaction is learning the function of the enzyme. In fact, depending on the enzyme identity, the same

substrates can undergo different chemical transformations [16]. Prior research on biocatalyzed reaction prediction considers two alternative methods for incorporating enzyme information: using enzyme nomenclature [15] or EC number [11]. The former method encodes the scientific name of the enzyme using a language model, while the latter relies on expert defined enzyme classes (i.e., their EC numbers). In both cases, enzymes with similar characteristics are likely to exhibit similarity in their encoding. However, both methods only provide limited generalization capability especially for unseen enzymes where categorization information may not be available. Moreover, these methods ignore the rich biological information embedded in protein sequences. In operating on enzyme classes, previous research also disregards the specificity of proteins and treats all enzymes of a particular class as capable of catalyzing the same substrates. In contrast, utilizing sequence information, our method can be applied to unseen enzymes, without relying on functional annotations.

Chemical Reaction Prediction The field of biocatalyzed reaction prediction is still relatively nascent, and prior methods frame the task as a machine translation problem using language models. However, language-based generation does not make use of the fact that the atoms of the reactants are conserved, and small mistakes in generation can lead to widely different molecules. Our work most closely follows graph-based approaches developed for the small molecule, general chemistry, space. These methods leverage this inductive bias and learn the graph edits to apply on the molecular graph encoding of the reactants, and recently demonstrated better generalization than language model based approaches [9, 17, 18, 19, 20, 21, 22]. Our approach builds on the success of these graph based methods and develops it further to include enzyme sequences and exploit the interactions between the sequences and substrates.

Drug Metabolism Prediction In the pharmaceutical industry, drug metabolism screening is typically done through experimental assays. Existing analytical methods largely rely on rule-based approaches [23, 24, 25, 26, 27, 28, 29, 30]. For example, [31] used a template-based

search to predict that the anti-cancer drug 5-fluorouracil can be metabolized into competitive inhibitors of native enzymes, which can help explain the observed toxicity of the drug. As an alternative to rule-based approaches, several machine learning methods have emerged. However, these approaches are limited in their reach as they are typically trained on a specific class of enzymes (e.g., cytochrome P450s) [32, 33, 34, 35]. In contrast, our method provides a more general framework that can be applied to any chemical matter while delivering strong performance on a curated dataset from DrugBank [36].

2.3 Method

2.3.1 Biocatalyzed Product Generation

We present here an overview of the method. We first predict whether and how the reactant bonds change conditioned on both the set of reactants and the protein sequence of the associated enzyme (Section 2.3.1). We deterministically perform chemically valid graph edits to obtain all products that can be generated with up to k of the most likely predicted bond changes. We train a second model to retrieve the correct product given the full reaction and the protein sequence (Section 2.3.1).

Reaction Center Prediction

Reactants and products are constructed as 2D graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with node features $v_i \in \mathcal{V}$ and edges $e_{ij} \in \mathcal{E}$. While the bonds we predict correspond to the overall net change between the atom-mapped reactants and products, they are nonetheless dependent on chemical interactions between atoms in the same reactants, atoms in different reactants, and the enzyme amino acid residues. We model each type of interaction and use them together to predict all bond changes.

First, a Graph Attention Network [37] f_{local} is used to encode each reactant separately

and obtain node embeddings for each atom:

$$A = f_{\text{local}}(\mathcal{V}, \mathcal{E})$$

where $A = \{a_1, a_2, \ldots, a_n\}$ is the set of reactant node features after applying the GNN and $a_i \in \mathbb{R}^d$.

Similarly to [9], a second model, f_{global}, is then used to encode the interaction between atoms in different molecules by constructing a complete graph from the reactants. Specifically we add an edge between every pair of nodes: $e'_{ij} = [1_{\text{same}} \parallel 1_{\text{diff}} \parallel e_{ij}]$, where 1_{same} indicates whether the atoms are in the same molecule, 1_{diff} indicates whether the atoms are in different molecules, and e_{ij} are the bond features. We set $e_{ij} = \mathbf{0}$ when the atoms are not connected by a chemical bond. We compute a pairwise attention with every atom in the complete graph and obtain the global node embeddings $a'_i \in \mathbb{R}^d$ as a weighted sum:

$$\begin{aligned}
\alpha_{ij} &= \sigma(\mathbf{u}^\top \text{ReLU}\left(P_a(a_i + a_j) + P_b e_{ij}\right)) \\
a'_i &= \sum_j \alpha_{ij} a_j \\
A' &= \{a'_1, a'_2, \ldots, a'_n\}
\end{aligned}$$

Third, we use ESM-2 [4] as the protein encoder f_p to obtain residue-level representations $P = \{r_1, r_2, \ldots, r_m\}$ and perform a multi-headed cross-attention [38] between the residues and the node embeddings of the reactant graphs a_i:

$$A'' = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) V$$

where

$$Q = W^Q A; \qquad K = W^K P; \qquad V = W^V P$$

Finally, for each atom pair (i, j), we compute the probability that a particular bond

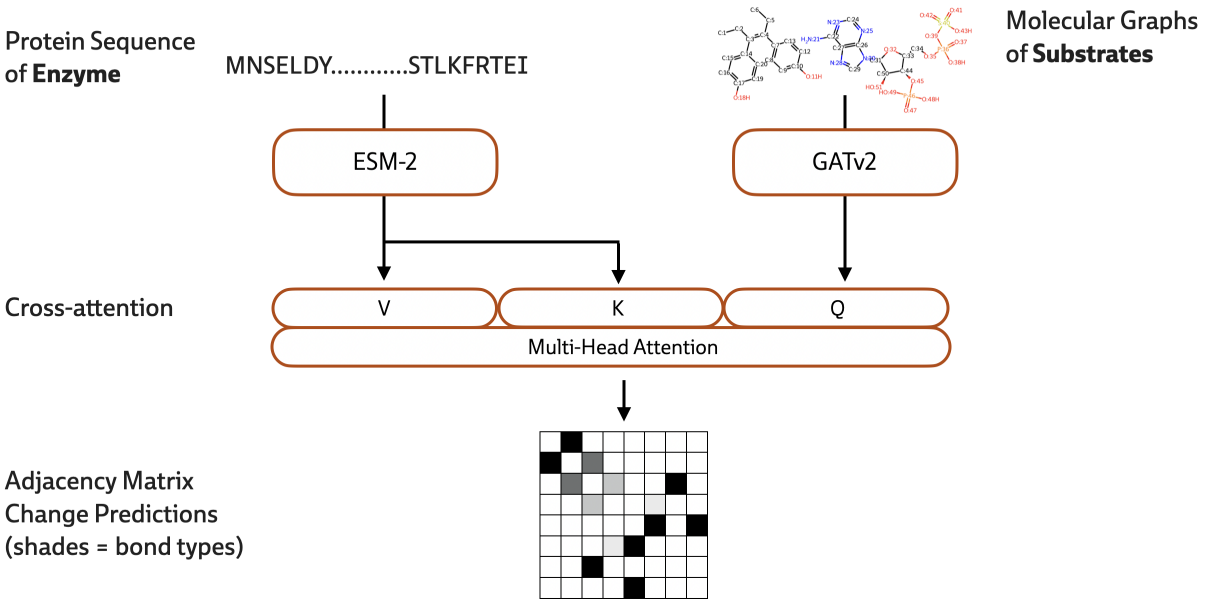

Figure 2.1: Schematic of the model architecture for predicting the bond changes associated with a given an enzyme and its substrates.

change k occurs between them, which consists of either the loss of a bond or the formation of a single, double, triple, or aromatic bond:

$$c_i = W_a[a_i \parallel a_i' \parallel a_i'']$$

$$b_{ij} = W_b e_{ij}$$

$$s_{ijk} = W_k \text{ReLU}\left([c_i + c_j \parallel b_{ij}]\right)$$

To force the model to focus on bond changes associated with substrate, we do no compute the loss over bond changes associated with common co-factors and co-enzymes like ATP, which often comprise most of the bond changes associated with the reaction.

Candidate Product Ranking

Given the predicted bond changes above, we select the top k predictions. We empirically predefine a k' as the maximum number of changes that could occur within a biochemical

reaction and construct all sets of size at most k' consisting of chemically valid changes. Each set of bond changes is applied as graph edits on the original reactant graphs to obtain candidate products. We then train a classifier to retrieve the products associated with the ground truth set of changes from the list of all candidate products.

The identity of the correct product depends on the reactants and enzyme, and the most likely products are those whose transition state is stabilized by the enzyme [39, 40]. As a result, we represent a pseudo-transition state using the condensed reaction graph [41, 42] for each prediction by superimposing the reactants and generated products and concatenating their node and edge features. This aims to incorporate all representations of the predicted reaction and the enzyme together. We then encode the graph structure with a directed message passing neural network f_{rxn} [43] to obtain atom-level features a_i and obtain residue-level features r_i of the enzyme using ESM-2.

$$
\begin{aligned}
a_i &= f_{\text{rxn}}\left(\left[v_i^{(\text{reactants})} \parallel v_i^{(\text{products})}\right], \left[e_{ij}^{(\text{reactants})} \parallel e_{ij}^{(\text{products})}\right]\right) \\
a &= \sum_i a_i; \qquad p = \frac{1}{|P|}\sum_i r_i \\
g &= f_{\text{rank}}([a \parallel p])
\end{aligned}
$$

Finally, we aggregate both the reaction graph representations and the protein representations, and pass them together through a small feed-forward network, f_{rank}, to score each proposed reaction.

2.3.2 Model Training

Reaction center prediction We use pre-trained ESM-2 with 35M parameters (`esm2_t12_35M_UR50D`) to encode the enzyme sequences. We use Graph Attention Networks [37] for f_{local} with 3 layers, 16 attention heads, and a hidden dimension of 480. We construct a complete graph of the reactants to compute the pairwise attentions across all atom pairs in

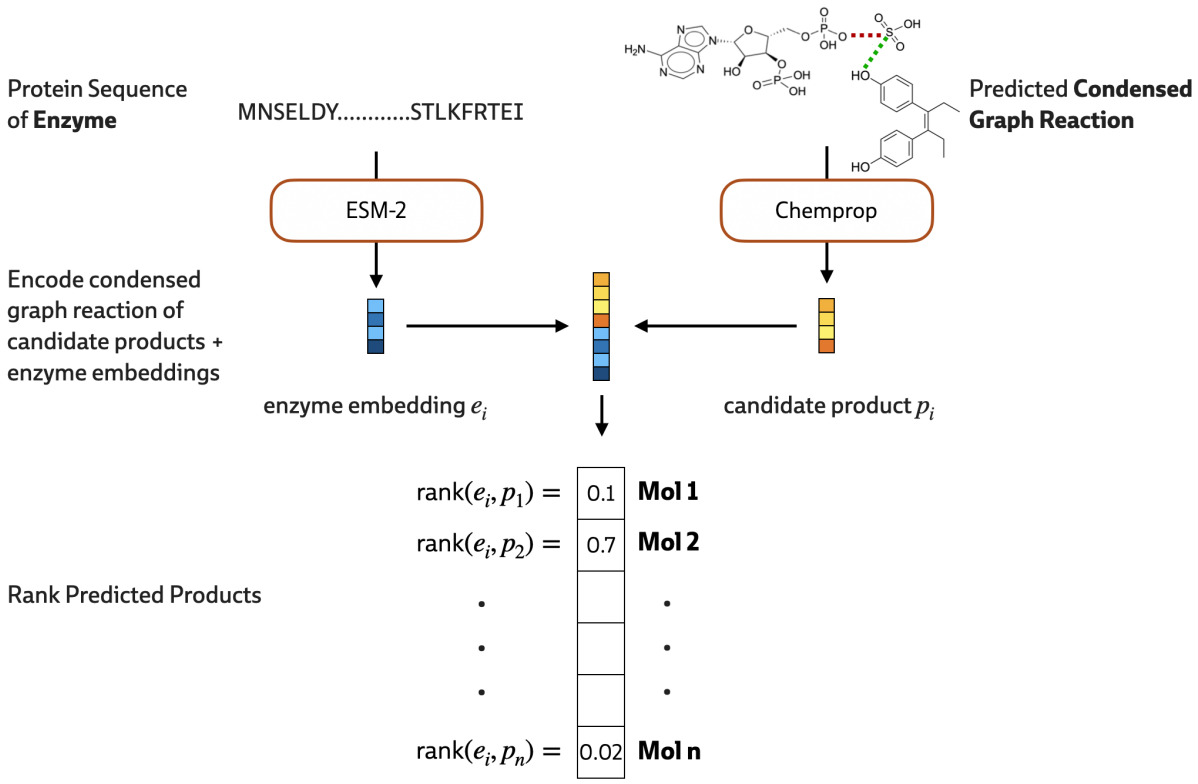

Figure 2.2: Schematic of the ranking model used to select the correct product from a list of candidates by considering the enzyme and the full predicted reaction. The red and green dashed edges represent bonds that are predicted to be deleted and created, respectively.

the f_{global} model. The multi-head cross-attention between the protein residues and reactant atoms is implemented with 4 attention heads. Individual atom representations from each are concatenated and passed through a linear projection layer (3×480 to 480) before predicting pair-wise bond changes.

Candidate product ranking We apply chemprop [43], a directed message passing network, on the condensed graph reaction representation of the reactants and candidate products with 5 layers and a hidden dimension of 480. We obtain the mean protein embeddings from ESM-2 (35M parameters) and concatenate them with the graph-level feature representations of the reaction. The final ranking is done with a 2-layer feed-forward network with layer norm [44].

Training parameters We use a batch size of 16, learning rate of $1e^{-4}$, learning rate decay of 0.1, and the Adam optimizer [45]. Training is done with half precision training with bfloat16 [46], and we train the reaction center for 20 epochs and the ranker for 5 epochs.

2.4 Experimental Setup

2.4.1 EnzymeMap Dataset

We train all models on data derived from EnzymeMap [12], which consists of biocatalyzed reactions paired with protein UniProt identifiers and their EC numbers. All reactions are fully atom-mapped, meaning that every atom in the products can be traced back to an atom in the reactants of the reaction. To obtain protein sequences, we consider only reactions associated with UniProt or SwissProt identifiers and pull their sequences from their respective databases. As is standard in the literature, we remove products that occur as reactants in the same reaction, common byproducts, and products with fewer than 4 heavy atoms. We follow [11] and split reactions with multiple products and exclude reactions with large molecules (> 100 heavy atoms). To control for the size of the proteins, we only consider sequences that

are no more than 800-amino acids long. This yields 103,120 enzyme-catalyzed reactions with 20,385 unique chemical reactions, 12,541 enzymes, covering 2743 EC numbers.

We consider several splits of the dataset. In keeping with previous work, our primary test set is constructed using a product split, where no product in the test set is seen in the training set. Additionally, we explore a structure similarity split and an EC split in the appendix. In particular, enzymes are clustered using Foldseek [47] with a 90% structure overlap and a sequence identity of 0, and enzymes in the same cluster are assigned to the same split. We split the data into train, development, and test datasets with a ratio of $8:1:1$. For the EC split, we held-out reactions in an EC for the test set and split the remaining reactions into ($\sim$89%) train and ($\sim$11%) development. Details on data processing are provided in Section 2.4.4.

2.4.2 DrugBank Dataset

We consider the out-of-distribution chemical domain of drug metabolism and showcase the improved performance of our model as compared to other models on this task. We obtain drug reactions from DrugBank for which a UniProt ID is available. Since our graph-editing procedure requires all reactant molecules present, including co-factors, we obtain from UniProt all reactions annotated for each protein entry and extract the substrates that are common among all reactions of an entry and add them to the corresponding DrugBank data samples. We further focus our analysis on reactions from phase II metabolism and exclude reactions catalyzed by cytochromes. The cytochrome P450 superfamily is known to perform a wide range of chemical transformations and is often non-specific to location or to substrate such as hydroxylation of unactivated C-H bonds, C-C or C-N bond formation, heteroatom oxidation, oxidative C-C bond cleavages, and nitrene transfer [48]. Since these chemical transformations can be stochastic in their location, annotated datasets represent only a small subset of possible products making it hard to evaluate predictions using the same method and so we exclude them. This curated dataset yields 804 reaction-enzyme pairs, with 160 unique proteins and

342 drugs.

2.4.3 Baselines

We consider the two prior works on biocatalysis prediction as baselines [15, 11]. We retrain the transformer models on the USPTO and EnzymeMap training sets following the paradigm reported in the publications and detailed in Section 2.8.1. [15] uses the enzyme names to encode the protein, therefore we map protein identifiers from EnzymeMap to their annotated name in UniProt. In cases where annotated names are missing we mark the name as "unknown" in order to avoid skipping many samples. Providing the protein name as input to the model suggests that the protein's function has already been studied and its function characterized, thus defining the name of the protein. In cases where the name does not provide any indication of the function, it should not provide useful information to the model (e.g. "unknown"). In these cases the model must rely on the substrates alone. On the other hand, [11] encodes the first three levels of the EC number of the reaction; similarly, knowing the associated EC number suggests that much of the biochemical reaction is already characterized and provides the model with information that is beyond what we assume to be available at inference time.

2.4.4 Data Splits

Product Split We follow prior work and split the data such that there is no overlap of products between the three data splits. We use the `rxn4chemistry` tools (https://github.com/rxn4chemistry/biocatalysis-model, [11]) to pre-process our data and exclude reactions with large molecules ($>$ 100 heavy atoms), those with products with fewer than 4 heavy atom, and those with proteins that are more than 800-amino acids long. This yields 95,318 training samples, 5,037 development samples, and 2,765 test samples.

Structure Split We download predicted protein structures from AlphaFold [3] as `.cif` files. We use Foldseek [47] to cluster our database of structures using `easy-cluster` with `-min-seq-id` = 0.0 and `-c` = 0.9. We obtained 13,866 clusters which were split into 80% train, 10% development, and 10% test. Samples were placed in a split according to the enzyme's cluster identity. This yields 79,443 training samples, 11,208 development samples, and 10,781 test samples.

EC Split For each EC, $ec \in \{1, 2, 3, 4, 5, 6\}$, we held out all reactions with that specific ec number, considering only the top level class. The remaining reactions were split according to product-based split into $\frac{8}{9}$ training and $\frac{1}{9}$ development sets (maintaining the ratio of 8:1:1). The number of samples in each split are provided in Table 2.1.

Table 2.1: Number of reactions in each data split when using the top-level EC number to construct the test sets.

Held-out EC	Training Split	Development Split	Test Split
1	64,065	8,159	30,896
2	65,652	8,247	29,221
3	68,462	7,335	27,323
4	82,669	10,809	9,642
5	88,352	10,751	4,017
6	89,907	11,192	2,021

2.5 Results

2.5.1 EnzymeMap

While prior deep learning methods developed specifically for this task use more detailed data on the enzyme identity (either the EC number or the enzyme nomenclature) our method assumes that only the amino-acid sequence of the enzyme and the substrate molecules are known. However, we compare against these methods for completeness. Additionally, we

Table 2.2: Top-k accuracy of our graph-based method compared to existing approaches for biocatalyzed forward synthesis. Published methods are trained as detailed in their respective GitHub codebases (Section 2.8.1). Performance is evaluated on EnzymeMap using a product split.

Model	Top 1	Top 3	Top 5	Top 10
[15]	35.3%	43.6%	46.0%	47.8%
[11]	50.5%	61.7%	65.4%	68.8%
Ours	72.5%	84.3%	87.3%	89.4%

impose a conservation of mass constraint and only generate bond changes whereas existing baselines use free generation to decode the product SMILES.

We test the hypothesis that encoding the protein and molecular structure leads to better generalizability in predicting unseen products. We find that our model is able to generalize better to unseen reaction products and surpasses other models by a considerable margin with a top-1 accuracy of 72.5% relative to 35.5% and 50% using enzyme name and EC, respectively (Table 2.4). Since reactions can have multiple possible products, we expect that not all products can be recovered within the first prediction. Considering the top $k > 1$ predictions, we observe sustained performance gains in recovering all products, approaching 90% accuracy with $k = 10$. We also consider other biologically relevant splits based on protein structure similarity and the reaction classes defined by EC numbers, and observe that our model exhibits comparable on these harder splits, albeit without assuming any additional protein annotations (Section 2.6).

2.5.2 Impact of Protein Sequence

Enzymes play an important role in biocatalysis. However, since the molecular structure of the substrates alone provides some information about the potential sites of metabolism [49], we sought to evaluate the extent to which these models simply memorize reaction rules versus take into account the impact of the enzyme itself. Here, we show how well each model predicts the products of enzymatic reactions from the reactants alone without enzyme

information. We train both the Molecular Transformer architecture [50] and WLN [17] without incorporating any protein information. Since our primary task is to generalize to new products, we focus our analysis on the product split. We observe that both models achieve improved performance when the protein sequence is included (Table 2.3), with the top 1 accuracy of our method obtaining a 14% gain in performance relative to the WLN model (no protein sequence). However, this gap decreases to 4% as more candidates are considered (top k=10). We also find that both graph-based models perform better over sequence-to-sequence models.

Table 2.3: Top-k accuracy of a transformer and graph model that exclude protein information compared to our full model. Performance is evaluated on EnzymeMap using data splits based on a product split.

Model	Top 1	Top 3	Top 5	Top 10
[50]	35.0%	50.6%	55.5%	58.9%
[17]	58.3%	75.9%	81.8%	85.2%
Ours	72.5%	84.3%	87.3%	89.4%

2.6 Performance Along Additional Splits

2.6.1 Performance on Structure Splits

We assign proteins to the training and testing splits based on their Foldseek [47] cluster identity. We observe that all models achieve similar performance ranging from 60% top-1 accuracy to 80% top-10 (Table 2.4). While the proteins in the test are expected to assume different 3D folded structures, they may still share catalytic activities with proteins seen during training [51]. For instance, convergent evolution can result in significantly different proteins that catalyze the same reaction. As a result, this can result in data splits where the encoding used in our model does not provide an advantage.

Table 2.4: Top-k accuracy of our graph-based method compared to existing approaches for biocatalyzed forward synthesis. Published methods are trained as detailed in their respective GitHub codebases (Section 2.8.1). Performance is evaluated on EnzymeMap using data splits based on protein structure similarity using FoldSeek [47].

Model	FoldSeek 90% Split			
	Top 1	Top 3	Top 5	Top 10
[15]	64.5%	77.5%	79.8%	81.2%
[11]	60.2%	75.0%	77.9%	80.3%
Ours	60.4%	71.7%	75.9%	78%

2.6.2 Performance on EC Splits

The EC system defines seven large classes of biochemical transformations. To measure the generalization across enzyme families and types of chemical transformations, we trained six models separately, holding out each time all reactions with a specific EC number (only six classes are contained in the EnzymeMap dataset). This constituted the hardest setting among the three splits. Since [11] utilizes the EC number as an input, we omitted it from this experiment since it would never see the test-set EC during training. We observe that our method is comparable to [15] on ECs 2,3, and 5, better on ECs 1 and 4, and significantly worse on EC 6 (Table 2.5) . Across ECs, however, both models achieve poor performance in terms of absolute accurate generalization, demonstrating the challenge of truly learning the chemistry underlying enzymatic catalysis.

2.6.3 Attention Analysis

While the results of Table 2.3 suggest that the model is utilizing the protein sequence in improving its final prediction, they provide no indication whether it learns any biologically meaningful properties regarding the protein's catalytic function. Our architecture, however, learns a multi-head cross-attention between the full protein sequence and the latent atom representations of the substrates, yielding attention scores for every residue-atom pair. By

Table 2.5: Top-k accuracy of our graph-based method compared to existing approaches for biocatalyzed forward synthesis on different EC-based splits. Each model is trained on all other ECs and tested on the held-out EC.

Model	Held Out EC	Top 1	Top 3	Top 5	Top 10
[15]	EC 1 (n=30,896)	8.6%	17.3%	21.9%	26.3%
Ours		9.4%	22.2%	27.2 %	34.0 %
[15]	EC 2 (n=29,221)	9.1%	16.6%	20.9%	26.2%
Ours		8.0%	15.9%	20.7%	25.8%
[15]	EC 3 (n=27,323)	26.8%	47.7%	55.2%	61.7%
Ours		32.1%	45.9%	52.9%	60.7%
[15]	EC 4 (n=9,642)	13.9%	19.8%	23.0%	28.6%
Ours		7.4%	20.1%	29.0%	35.1%
[15]	EC 5 (n=4,017)	2.4%	6.4%	7.1%	11.9%
Ours		1.7%	9.0&	10.9%	12.0%
[15]	EC 6 (n=2,021)	20.6%	41.5%	47.0%	48.3%
Ours		4.1%	15.9%	21.4%	26.1%
[15]	Mean	13.5%	24.9%	29.2%	33.8%
Ours		10.5%	24.0%	27.0%	32.3%

summing over the attentions scores across all atoms, we obtain a weighting per residue. We extract active site annotations from the Mechanism and Catalytic Site Atlas [52] for both the reference sequences as well their homologs, which are assumed to have identical active sites, and we compare them with the top-scoring residues according to the learned attention scores. We take the residues with the top q-th quantile of attention scores and compute the fraction of annotated active site residues included in that predicted set. We find that our learned attention has a consistently better correspondence with the active site than an equivalent random guess (Figure 2.3). This suggests that our model is able to learn a functionally meaningful association between the protein sequence and the substrates.

Our multi-head cross-attention in the reaction center prediction model is performed between the full protein sequence embedding and the latent atom representations of the substrates, yielding attention scores for every residue-atom pair, $\boldsymbol{A} \in \mathbb{R}^{|\mathcal{V}|\times|P|}$, with $\boldsymbol{A}_{ij} \in [0,1]$. For every residue, we sum the attentions scores across all reactant atoms and obtain a weight per residue r_i: $\boldsymbol{a}_i = \sum_v \boldsymbol{A}_{vi}$.

Where available, we collect a set of indices, $\mathcal{R}_{AS}$ for each sample in the test set corresponding to the location of annotated active sites from the Mechanism and Catalytic Site Atlas. For each protein p, We take the residues in the top q-th quantile of attention scores and compute the fraction of annotated active site residues included in that predicted set:

$$\hat{\mathcal{R}} = \{i | r_i > k, k = \text{j}^{\text{th}} \text{ quantile of } \boldsymbol{a}\} \tag{2.1}$$

$$s_p^{(k)} = \frac{|\mathcal{R}_{AS} \cap \hat{\mathcal{R}}|}{|\mathcal{R}_{AS}|} \tag{2.2}$$

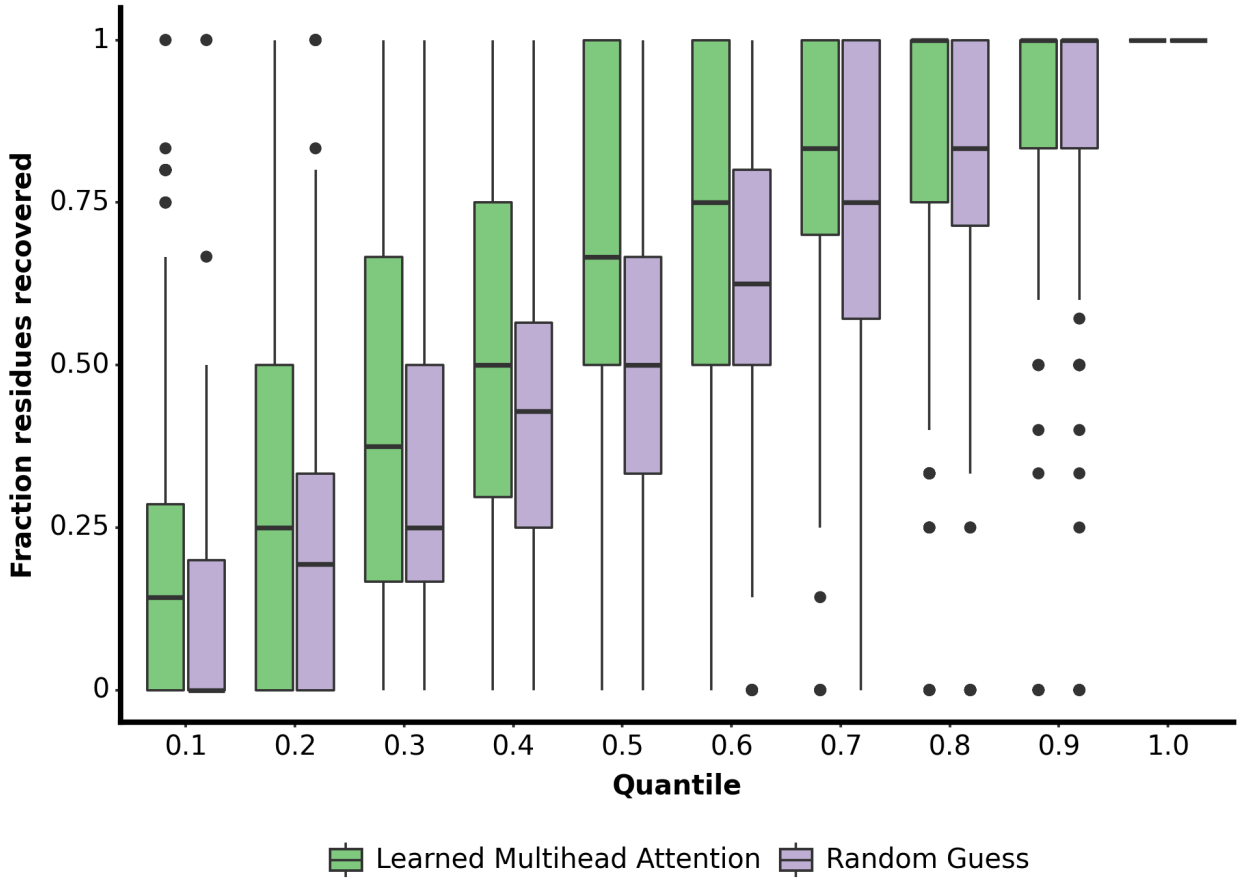

Figure 2.3: Fraction of true active site residues included in the top q quantile of attention scores extracted from the multi-head cross-attention layer used to predict the bond changes in each reaction. For every quantile, we take a random permutation over all residue indices and select the same number of predictions as in that quantile to obtain a random guess baseline.

We plot $s_p^{(k)}$ for all test sample proteins with annotations at 10 equally spaced quantile levels. As a control, we compare these scores with those obtained by randomly selecting an equivalent number of indices spanning the length of the protein.

2.6.4 Predicting Drug Metabolism

Here, we take the metabolism of small molecule drugs as a potential real-world application of our model and showcase the improved performance of our model as compared to others on this task. How a drug is metabolized has important implications for its efficacy, toxicity, and mechanism of action. While some experimental approaches exist to study drug-metabolizing enzymes, there remains a critical need for *in-silico* drug metabolism models to address the cost, time, and human expertise required by *in-vitro* and *in-vivo* methods. We evaluate our model on drug reactions from DrugBank for which a UniProt ID is available and focus on non-cytochrome-catalyzed biotransformations [36]. We find that our model is able to predict the correct drug metabolite with a 60.1% top-10 accuracy and outperform other deep learning models (Table 2.6).

Table 2.6: Performance on the DrugBank drug reactions.

Model	**Top 1**	**Top 3**	**Top 5**	**Top 10**
[15]	28.6%	37.2%	40.8%	43.8%
[11]	25.7%	33.0%	38.1%	42.8%
Ours	40.7%	56.0%	58.0%	60.1%

To better understand the errors observed on drug reactions, we manually inspect cases where the model fails to find an exact match to the annotated product within the top ten predictions. In many cases, we find that the model comes close in identifying the reaction type but focuses on incorrect, yet similar, sites of metabolism. For example, the model correctly predicts the reaction in Figure 2.4(a) to be a hydroxylation but predicts the wrong methyl group to which to add the OH group, though it is near the true site. We also identify cases where the model predictions are considered wrong as a result of

inconsistencies in the databases. Raloxifene (DB00481) is reported to be metabolized by a UDP-glucuronosyltransferase (Q9HAW8) (Figure 2.4(b)). Since we obtain enzyme co-factors from UniProt, we utilize UDP-α-D-glucuronate as the other substrate in the reaction. Our prediction matches exactly the chemical pattern annotated in UniProt and provided by the Rhea database. However, this appears to be inconsistent with the metabolic reaction of raloxifene in DrugBank and results in our prediction to be considered incorrect. In some cases, the model is not able to fully capture the complexity of the biochemical reaction. The metabolism of morphine (DB00295) consists of the transfer of glucuronic acid and ring breaking (Figure 2.4(c)). The model is found to be partially correct as it predicts the right glucuronidation site but is unable to identify the bond changes to the ring in any of its top-k predictions.

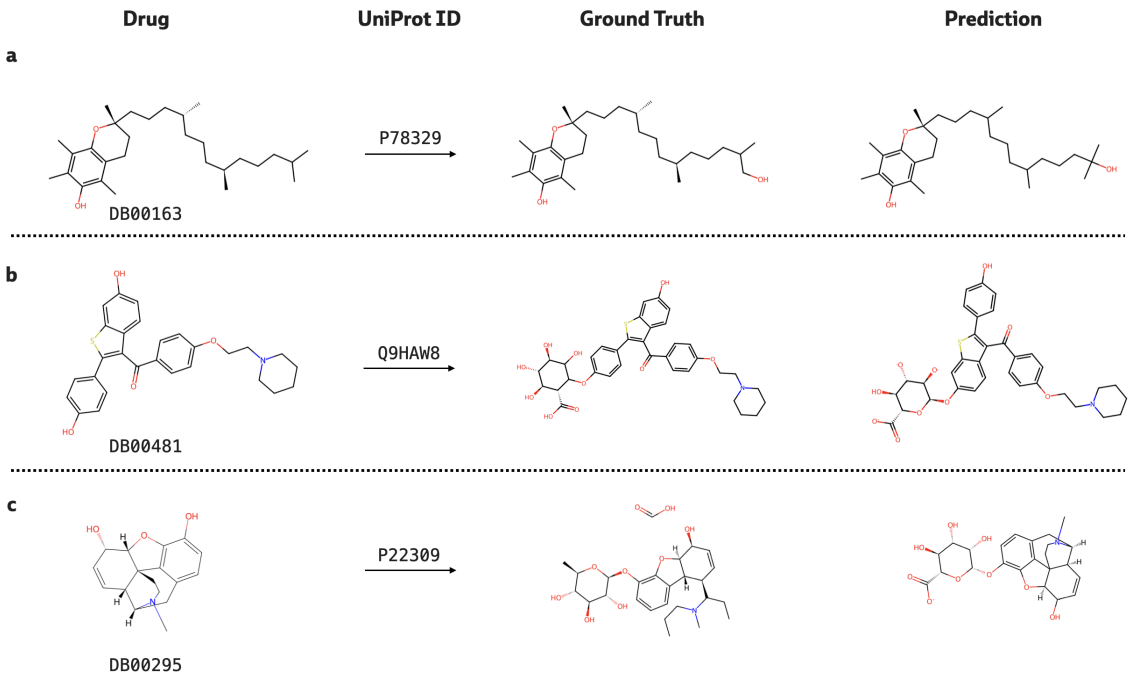

Figure 2.4: Illustrative examples of errors made by our model, where (a) the predicted reaction type is correct but the reactive site is misclassified; (b) the mistake is possibly due to inconsistencies between databases; and (c) the reaction consists of several changes that the model is unable to fully recover.

2.7 Conclusion

This paper presents a novel graph-based method for predicting the products of biocatalyzed reactions given a set of substrates and an enzyme sequence. We show that incorporating the enzyme sequence in the input improves performance compared to other methods that include alternative representations of enzymes, namely EC numbers and enzyme names. We report an improvement of 37.2 points in top-1 accuracy against preceding state-of-the-art methods on the EnzymeMap dataset. Lastly, we note that by relying on enzyme sequence, we widen the utility of our model compared to previous models to encompass unannotated and orphan enzymes.

The results presented also exhibit a number of limitations. While we show that the model has a capacity to generalize to out-of-distribution molecules, like small molecule drugs, there still remains room for improvement especially for completely new chemical transformations. Furthermore, enzymes are critically stereo-selective, and the current method is not capable of distinguishing between stereo-isomers in its predictions. Moreoever, biochemical transformations in the active site pockets of enzymes occur in a three-dimensional space, often accompanied by conformational changes in the protein structure. None of these aspects are modeled here, but they provide ample opportunity for future work.

2.8 Additional Implementation Details

2.8.1 Training of transformer-based models

We train existing deep learning model for biocatalysis [15] and [11] according to the code-bases associated with their respective publications: https://github.com/rxn4chemistry/OpenNMT-py/tree/carbohydrate_transformer and https://github.com/rxn4chemistry/biocatalysis-model. Specifically, we use the same tokenization scheme for the enzyme names with either byte pair encoding or the EC numbers. We pre-process (`onmt_preprocess`) the data with the default

parameters of sequence source and target lengths of 3000 and a shared vocabulary, and we train the models (`onmt_train`) simultaneously on the USPTO dataset [7, 8] and the same splits of EnzymeMap that we use for our model. We use the default hyper-parameters for training (Section 2.8.1).

ll

Hyper-parameter	Value
`data_weights`	(9,1) (for USPTO and EnzymeMap, respecitvely)
`seed`	42
`gpu_ranks`	0
`world_size`	1
`train_steps`	250,000
`param_init`	0
`param_init_glorot`	true
`max_generator_batches`	32
`batch_size`	32768
`batch_type`	tokens
`normalization`	tokens
`max_grad_norm`	0
`accum_count`	1
`optim`	adam
`adam_beta1`	0.9
`adam_beta2`	0.998
`decay_method`	noam
`warmup_steps`	8,000
`learning_rate`	2
`label_smoothing`	0.1

`layers` 6

`rnn_size` 512

`word_vec_size` 512

`encoder_type` transformer

`decoder_type` transformer

`dropout` 0.1

`position_encoding` true

`share_embeddings` true

`global_attention` general

`global_attention_function` softmax

`self_attn_type` scaled-dot

`heads` 8

`transformer_ff` 2048

Chapter 3

CLIPZyme: Reaction-Conditioned Virtual Screening of Enzymes

3.1 Motivation

Biosynthesis is the method of choice for the production of small molecules due to the cost effectiveness, scalability and sustainability of enzymes [53, 54]. To find enzymes that can catalyze reactions of interest, practitioners often begin by identifying naturally occurring enzymes to repurpose or optimize. Only 0.23% of UniProt is well studied and efficient enzymes likely lie among the hundreds of millions of sequences that are yet to be explored [55]. As a result, the ability to computationally identify naturally occurring enzymes for any reaction can provide high quality starting points for enzyme optimization and has the potential to unlock a tremendous number of biosynthesis applications that may otherwise be inaccessible.

In this work, we propose CLIPZyme, a novel method to address the task of virtual enzyme screening by framing it as a retrieval task. Specifically, given a chemical reaction of interest, the aim is to obtain a list of enzyme sequences ranked according to their predicted catalytic activity. In order to identify reaction-enzyme pairs, methods must contend with several

unique challenges. First, in some cases, small changes in enzyme structures can lead to a large impact on its activity. Yet in other cases, multiple enzymes with completely different structural domains catalyze the same exact reaction [55]. Similar principles hold for changes to the molecular structures of the reactants (substrates). This makes the task particularly challenging as methods must capture both extremes. Second, the efficacy of an enzyme is intricately linked to its interaction with the reaction's transition states [39, 56], which are difficult to model. Finally, in addressing the challenge of screening extensive datasets of uncharacterized enzymes, the scalability of computational methods becomes a critical factor.

CLIPZyme is a contrastive learning method for virtual enzyme screening. Originally developed to align between image-caption pairs, CLIP-style training has been successfully extended to model the binding of drugs and peptides to their target protein [57, 58]. Unlike binding, however, the need to achieve transition state stabilization makes enzymatic catalysis a more nuanced process (in fact, very strong binding may inhibit an enzyme). Therefore, in order to represent the transition state, we develop a novel encoding scheme that first models the molecular structures of both substrates and products then simulates a pseudo-transition state using the bond changes of the reaction. To leverage the 3D organization of evolutionarily conserved enzyme domains, we encode AlphaFold-predicted structures [3, 59]. Since enzyme embeddings can be precomputed efficiently, screening large sets of proteins sequences for a new query reaction is computationally feasible.

Since no standard method currently exists for virtual enzyme screening, we utilize enzyme commission (EC) number prediction as a baseline. Specifically, the EC number is an expert-defined classification system that categorizes enzymes according to the reactions they catalyze. Each EC number is a four-level code where each level provides progressively finer detail on the catalyzed reaction. For this reason, if a novel reaction is associated with an EC class, EC predictors can be used to identify candidate enzymes matching that EC class.

We establish a screening set of 260,197 enzymes curated from BRENDA, EnzymeMap and CLEAN [60, 12, 61]. In our evaluation, we adopt the BEDROC metric, as is standard

for virtual screening, and set its parameter $\alpha = 85$. This places the most importance on the first ∼10,000 ranked enzymes, which constitutes a reasonable experimental screening capacity. We compare CLIPZyme to CLEAN, a state-of-the-art EC prediction model, on the virtual screening task and showcase its superior performance. While CLIPZyme can perform virtual screening without any expert annotations of reactions, methods like CLEAN cannot. We show that even when given some knowledge of a novel reaction's EC class, CLIPZyme is still superior to EC prediction for virtual screening ($BEDROC_{85}$ of 44.69% compared to 25.86%). Additionally, we show that combining CLIPZyme with EC prediction consistently achieves improved results. We also demonstrate that our reaction encoding outperforms alternative encoding schemes. Finally, we test our method on both unannotated reactions in EnzymeMap and a dataset of more challenging reactions involving terpene synthases [62].

3.2 Background

Reaction representation learning Methods to encode chemical reactions have been developed for a range of different computational tasks. This includes language models operating on reaction SMILES strings [63, 64] and graph-based methods operating on the individual molecular structures of a reaction or on the condensed graph representations [9, 65, 41]. These have shown strong performance on tasks like reaction rate prediction and forward synthesis [66, 42], but fail to take advantage of the data to effectively learn transition state representations. Models developed explicitly for transition state prediction are trained on simulations of very small molecules and are not scalable to enzymatic reactions [67, 68]. In contrast to existing approaches that deterministically featurize bond changes, our method learns the features of these transition states directly from the data.

Catalysis of novel reactions Successful design of enzymes most often begins with finding natural proteins that can subsequently be repurposed or optimized [69, 70]. One option is to use EC prediction to filter enzyme screening sets. However, EC numbers are predefined

by experts and provide a relatively coarse characterization of enzymes. As a result, one EC can capture many different reactions, while none may be able to capture a completely novel reaction. Therefore, filtering large libraries of enzymes by EC may yield impractically large sets of enzymes or none at all. Lastly, state-of-the-art EC predictors still show limited success (top F1 scores of 0.5-0.6) [71, 61, 72, 73]. In this work, we move away from human-crafted enzyme classes and instead operate directly on molecular and protein structures.

Alternatively, the rational design of a new enzyme or active site requires a thorough understanding of the underlying mechanism [74, 75, 76, 77, 78]. While methods for protein sequence and structure generation have shown promise in creating custom folds and strong binders [79, 80, 81], unnatural enzymes still suffer from low activity relative to naturally occurring ones [54]. Instead, we focus on identifying natural protein leads that can be optimized further either computationally or experimentally [69, 53, 70].

3.3 Method

We formulate enzyme screening as a retrieval task, where we have access to a predefined list of proteins and are asked to order them according to their ability to catalyze a specific chemical reaction. The representation of a protein P is denoted by $p \in \mathbb{R}^d$ and the query reaction R by $r \in \mathbb{R}^d$. We aim to learn a scoring function $s(r, p)$ such that a higher score corresponds to a higher likelihood that P catalyzes R. We jointly learn a reaction encoder, f_{rxn}, and a protein encoder, f_p, to compute r and p (Figure 3.1). We adopt a contrastive learning objective [82, 83] to maximize the cosine similarity between the embeddings of biochemical reactions and their associated enzymes (Equations (3.1) and (3.2)). We treat all enzymes in a training batch that are not annotated to catalyze a reaction as negative samples. Implementations

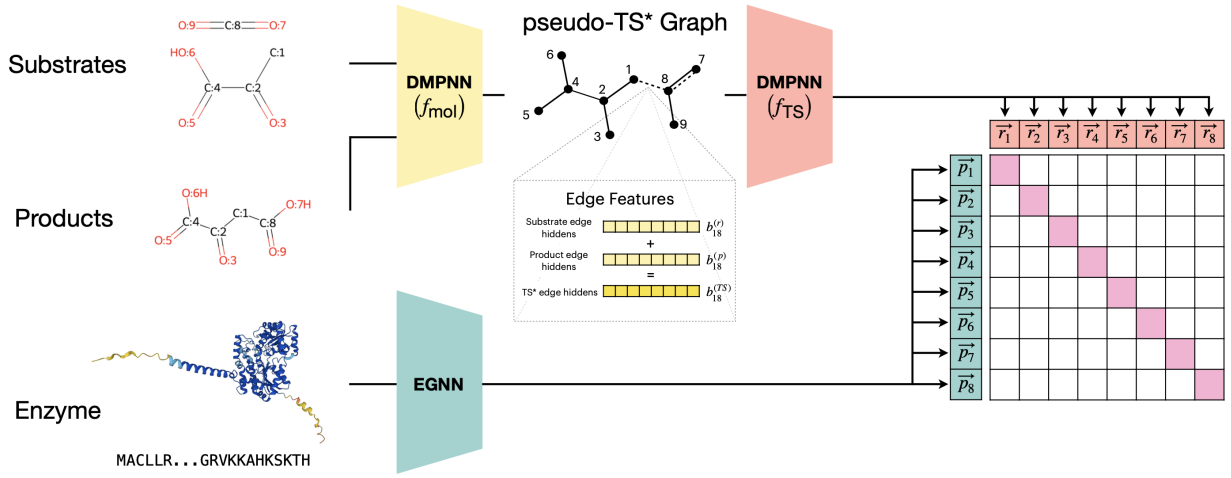

Figure 3.1: Overview of our approach. We encode atom-mapped chemical reactions using a DMPNN. We combine the substrate and product graphs by adding the hidden embeddings of their corresponding bonds to obtain an intermediate graph representing a pseudo transition state. A second DMPNN computes an embedding for the entire reaction. Enzymes are encoded with an EGNN using their predicted crystal structure and ESM-2 sequence embeddings. The reaction and enzyme representations are aligned with a CLIP objective.

details are provided in Sections 3.3.3 and 3.5.3.

$$s_{ij} = s(r_i, p_j) = \frac{r_i}{r_i} \cdot \frac{p_j}{p_j} \tag{3.1}$$

$$\mathcal{L}_{ij} = -\frac{1}{2N}\left(\log \frac{e^{s_{ij}/\tau}}{\sum_i e^{s_{ij}/\tau}} + \log \frac{e^{s_{ij}/\tau}}{\sum_j e^{s_{ij}/\tau}}\right) \tag{3.2}$$

3.3.1 Chemical Reaction Representation

To obtain a functionally meaningful representation of the reaction, we leverage the key insight that the active sites of enzymes have evolved to stabilize the transition state(s) of their corresponding reactions [84]. As a result, there is a geometric complementarity between the 3D shape of the protein active site and the molecular structure of the transition state. This complementarity determines to a large extent the catalytic activity of enzymes [39, 56]. While we do not have access to ground truth or predicted transition states, we use the atom-mapping available in the dataset to learn a superposition of the reactant and product molecular graphs

and obtain the reaction embedding.

Specifically, reactants and products are constructed as 2D graphs, where each molecular graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ has atom (node) features $v_i \in \mathcal{V}$ and bond (edge) features $e_{ij} \in \mathcal{E}$. A directed message-passing neural network (DMPNN) [43], f_{mol}, is used to separately encode the graph of the reactants G_x and that of the products G_y. This results in learned atom and bond features $a_i, b_{ij} \in \mathbb{R}^d$. To simulate the transition state, we construct a pseudo-transition state graph, $G_{TS} = (\mathcal{V}_{TS}, \mathcal{E}_{TS})$, by adding the bond features for edges connecting the same pairs of nodes in the reactants and the products. Edges between atom pairs that are not connected have bond features set to zero. We use the original atom features v_i as the nodes of $\mathcal{G}_{TS}$ to preserve the atom identities.

$$a_i, b_{ij} = f_{\text{mol}}(\mathcal{G}_x, \mathcal{G}_y) \tag{3.3}$$

$$v_i^{(TS)} := v_i^{(x)} \quad (= v_i^{(y)}) \tag{3.4}$$

$$e_{ij}^{(TS)} := b_{ij}^{(x)} + b_{ij}^{(y)} \tag{3.5}$$

We jointly train a second DMPNN, f_{TS} to encode G_{TS} and obtain the reaction embedding r by aggregating the learned node features.

$$a'_i, b'_{ij} = f_{\text{TS}}(\mathcal{G}_{TS}) \tag{3.6}$$

$$r = \sum_i a'_i \tag{3.7}$$

3.3.2 Protein Representation

Enzyme representation plays a pivotal role in modeling their function and interaction with substrates. To this end, we leverage advancements in both protein language models and graph neural networks.

Each protein is represented as a 3D graph $\mathcal{G}_p = (\mathcal{V}, \mathcal{E})$, with residue (node) features $h_i \in \mathcal{V}$ and bond (edge) features $e_{ij} \in \mathcal{E}$. Additionally each node i has coordinates $c_i \in \mathbb{R}^3$. The

node features of $\mathcal{G}_p$ are initialized using embeddings from the ESM-2 model with 650 million parameters (`esm2_t33_650M_UR50D`) [85], which has demonstrated success in capturing many relevant protein features for a range of downstream tasks. The ESM model produces a feature vector for each residue denoted as $h \in \mathbb{R}^{1280}$.

To encode the protein graphs, we utilize an $E(n)$-Equivariant Graph Neural Network (EGNN) with coordinate updates [86]. This network is particularly suited for our purpose as it preserves translation, rotation and reflection equivariant graph features but is computationally inexpensive. Alternative methods preserve additional symmetries that are relevant to proteins such as SE(3) equivariance but are much more computationally expensive. We follow the implementation outlined in [86] except that the relative distances between nodes are encoded using a sinusoidal function (Section 3.3.3), as is common in protein structure modeling [87, 88, 38].

3.3.3 Implementation Details

All models are developed in PyTorch v2.0.1 [89] and trained using PyTorch Lightning v2.0.9 [90].

f_{mol} and f_{TS} We implement our reaction encoder (3.3.1) as two DMPNNs [43]. We use standard node and edge features (Table 3.1) to initialize the reactant and product graphs, with input node dimensions of 9 and input edge dimensions of 3. The first encoder, f_{mol} has 5 layers and a hidden dimension of 1,280. The node features for the second encoder, f_{TS} are unchanged, while edges are obtained from taking the sum of the hidden edge representations from f_{mol}. Hence the node dimensions are still 9, while the input edge features have dimensions 1,280. The model also consists of 5 layers and a hidden size of 1,280. We aggregate the graph as a sum over the node features.

Condensed Graph Reaction We construct the condensed graph reaction as described in [42]. Specifically, the atom and edge features for the reactants and products are created as

Table 3.1: Chemical properties used as node and edge features in constructing molecular graphs.

Entity	Features
atom (node) features	atomic number, chirality, degree, formal charge, number of hydrogens, number of radical electrons, hybridization, aromaticity, belonging to a ring
bond (edge) features	bond type, stereochemistry, conjugation

binary vectors for the properties detailed in Table 3.1. For node features $x_i^{(r)}, x_i^{(p)}$ and edge features $e_{ij}^{(r)}, e_{ij}^{(p)}$, we compute $x' = x_i^{(r)} - x_i^{(p)}$ and $e'_{ij} = e_{ij}^{(r)} - e_{ij}^{(p)}$. We do not use the atomic number in calculating x'. Concatenating these with our reactants' features, our final CGR graph consists of 225 atom and 26 edge features, $x_i^{CGR} = [x_i^{(r)} \parallel x'_i]$ and $e_{ij}^{CGR} = [e_{ij}^{(r)} \parallel e'_{ij}]$, respectively.

Reaction SMILES The reaction SMILES is first canonicalized then tokenized according to [50] without atom-mapping. We create a vocabulary based on this tokenization scheme and use a tranformer architecture [38] as implemented by the Hugging Face library (we use the BertModel) [91]. The transformer is initialized with 4 layers, a hidden and intermediate size of 1,280, and 16 attention heads. An absolute positional encoding is used over a maximum sequence length of 1,000. We prepend the reaction with a `[CLS]` token and use its hidden representation as the reaction embedding.

WLDN We implement WLDN as originally described in [9] and initialize it with 5 layers and a hidden dimension of 1,280. The difference graph is calculated as the difference between atom-mapped node embeddings of the substrate and product graph. We apply a separate 1-layer WLN to obtain the final graph-level representation.

EGNN Node features are initialized with residue-level embeddings from ESM-2 (the 650M parameter variant with 33 layers) [85]. We use a hidden size of 1,280, 6 layers, and a message

dimension of 24. Both features and coordinates are normalized and updated at each step. Neighborhood aggregation is done as an average, and protein-level features are taken as a sum over the final node embeddings. Repurposing the positional encodings used in [38], pairwise distances are transformed with sinusoidal embeddings. For a given relative distance d_{ij} between nodes i and j, the encoding function $f : \mathbb{N} \to \mathbb{R}^d$ transforms this distance into a d-dimensional sinusoidal embedding. The encoding is defined as follows:

$$f(d_{ij})^{(k)} = \begin{cases} \sin\left(\frac{1}{\theta^{k/2}} \cdot d_{ij}\right), k < \frac{d}{2}, \\ \cos\left(\frac{1}{\theta^{\frac{k-\frac{d}{2}}{2}}} \cdot d_{ij}\right), k \geq \frac{d}{2}. \end{cases} \tag{3.8}$$

where k is the index of the dimension of the distance vector, θ is a hyperparameter that controls the frequency of the sinusoids, which in our case is set to 10,000. The resulting embedding for a particular relative distance d_{ij} is constructed by concatenating the sine-encoded and cosine-encoded vectors, thus interleaving sinusoidal functions along the dimensionality of the embedding space.

CLEAN We train CLEAN with the supervised contrastive ("Supcon-Hard") loss following the training protocol and parameters loss described in the project's repository (https://github.com/tttianhao/CLEAN). Specifically, we use the supervised contrastive loss and the data split in which none of the test enzymes share $> 50\%$ sequence identity with those in the training set. At inference, we use the same approach described in [61] to compute the EC anchors. We obtain the predicted distance between each enzyme in our screening set and each EC anchor. We extend this to parent classes of the ECs. For instance, the representation for EC 1.2.3.x is the mean embedding of all CLEAN proteins in that class. We also predict the EC numbers for all of the enzyme sequences in our screening set using the "max-separation" algorithm.

3.3.4 Training Details

All models are trained with a batch size of 64 with bfloat16 precision and trained until convergence (approximately 30 epochs). We use a learning rate of $1e^{-4}$ with a cosine learning rate schedule and 100 steps of linear warm-up. Warm-up starts with a learning rate of $1e^{-6}$, and the minimum learning rate after warm-up is set to $1e^{-5}$. We use the AdamW optimizer [92] with a weight decay of 0.05 and $(\beta_1, \beta_2) = (0.9, 0.999)$. When training the ESM model, we initialize with the pretrained weights of `esm2_t33_650M_UR50D` and use a mean of the residue embeddings for the sequence representation. We train all models on 8 NVIDIA A6000 GPUs.

3.4 Experimental Setup

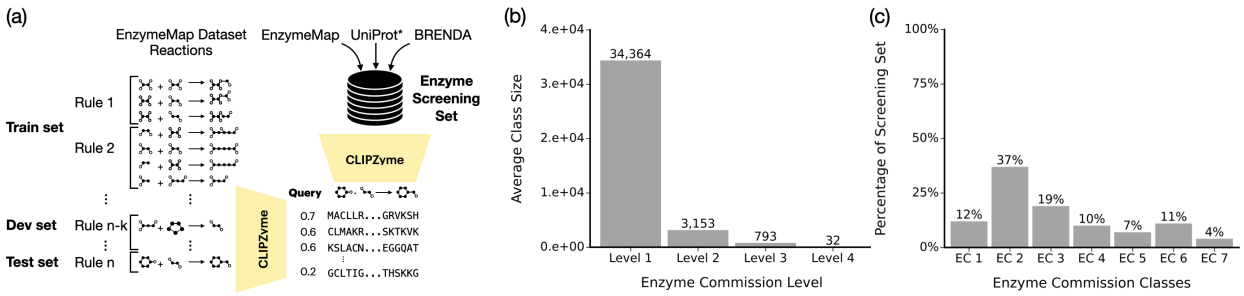

Figure 3.2: Overview of dataset construction and statistics. **(a)** Reaction-enzyme pairs are obtained from the EnzymeMap dataset [12] and split based on their reaction rules. At test time, a reaction is queried and enzymes are ranked from a screening set composed of sequences from EnzymeMap, UniProt*, and BRENDA. **(b)** Averge number of sequences in each EC class when considering different levels of the EC hierarchy. **(c)** Distribution of sequences in the screening set according to their first EC level.
*The UniProt dataset is obtained from [61].

3.4.1 EnzymeMap Dataset

Similarly to Section 2.5.1, our method is developed on the EnzymeMap Version 2 dataset [12], which includes biochemical reactions linked with associated UniProt IDs and their respective EC numbers. Each reaction is atom-mapped, allowing every atom in the product to be traced back to a corresponding atom in the reactants. To acquire the corresponding protein sequences, we select reactions linked to UniProt or SwissProt IDs and retrieve their sequences from their respective databases [93]. Additionally, we retrieve the predicted enzyme structures from the AlphaFold Protein Structure Database [3, 59]. We filter samples to include protein sequences up to 650 amino acids in length only. EnzymeMap provides a reaction rule for each reaction, which captures the bio-transformation in a reaction and can be applied to recreate the products of a reaction from its substrates [94]. With the goal of extending our model to unfamiliar chemical reactions, we divide our dataset into training, development, and testing groups based on these reaction rules (Figure 3.2). This results in a total of 46,356 enzyme-driven reactions, encompassing 16,776 distinct chemical reactions, 12,749 enzymes, across 2,841 EC numbers and 394 reaction rules.

EnzymeMap includes additional reactions that are associated with an EC number but lack an annotated protein sequence. We identify 7,967 of these unannotated reactions involving 1,101 EC numbers, distinct from our training data in terms of reaction rules. This subset serves as an additional validation set, allowing us to evaluate how our method ranks enzymes in relation to the EC number for each reaction. More information on how the data was processed can be found in **??**.

Data Processing

We obtain version 2 of the EnzymeMap dataset [12] and use only the reactions with assigned protein references from either SwissProt or UniProt. Our method requires that the same atoms appear on both sides of the reaction, so we exclude samples where this is not the case.

We also filter reactions where the EC number is not fully specified, the sequence could not be retrieved from UniProt, or there wasn't a computable bond change. We restrict our data to proteins of sequence length no more than 650 (maintaining 90% of the sequences) and those with a predicted structure in the AlphaFold database. We remove duplicate reactions, where the same reaction and sequence appear for multiple organisms. We split reactions into train/development/test splits with a ratio of 0.8/0.1/0.1 based on the reaction rule IDs assigned in the dataset. The statistics for the final dataset are shown in Table 3.2.

Table 3.2: Statistics of the EnzymeMap dataset used to develop CLIPZyme after preprocessing.

	Training Split	Development Split	Test Split
Number of Samples	34,427	7,287	4,642
Number of Reactions	12,629	2,669	1,554
Number of proteins	9,794	1,964	1,407
Number of ECs	2,251	465	319

3.4.2 Terpene Synthase Dataset

Terpenoids are a large and diverse family of biomolecules with wide applications to medicine and consumer goods. The reactions generating these natural compounds involve particularly complex chemical transformations that are typically catalyzed by a class of enzymes called terpene synthases [62]. This enzyme class is noteworthy for utilizing a relatively small number of substrates ($\sim$11) but is capable of generating thousands of distinct products. This presents a significant challenge with substantial implications. To further evaluate our method's performance on reactions known to involve challenging chemistry, we use a dataset of terpenoid reactions made available by recent work in detecting novel terpene synthases [62]. We exclude reactions that are themselves or their enzyme included in our training set, obtaining 110 unique reactions and 99 enzymes.

3.4.3 Enzyme Screening Set

To construct our screening set of enzymes, we include sequences annotated in the EnzymeMap dataset [12], Brenda release `2022_2` [60], and those used in developing CLEAN (UniProt release `2022_01`) [61]. We filter our set to those of sequence length < 650 with available AlphaFold predicted structures [3, 59] and obtain a final list of 260,197 sequences.

3.4.4 Protein Structures

We obtain all protein structures as CIF files from the AlphaFold Protein Structure Database [3, 59]. We parse these files using the BioPython MMCIFParser. We then construct graphs for use in the PyTorch Geometric library [95]. First we filter out the atoms from the CIF file to only include the C_α atoms of the protein. Each graph node as a result represents a residue and the associated coordinates from the CIF file. The edges are determined using the k-nearest neighbors (kNN) method, creating a connected graph that reflects the chemical interactions within the protein. We use a distance of 10 angstroms as a cutoff for the edges.

3.4.5 MSA Embeddings

We explore using the hidden representations from the MSA Transformer [96] as node embeddings of the enzyme 3D structure. Rather than using HHblits [97], we opt for MMSeqs2 [98] because of its speed and efficient search. We follow the pipeline employed by ColabFold [99] but use only the UniRef30 (`uniref30_2302`) database and do not use an expanded search [100, 101]. We sample 128 sequences for each MSA using a greedy search (maximum similarity) to obtain the input for the MSA-Transformer. We keep only the hidden representations of the query enzyme sequence and discard those from the MSA search. For an enzyme of length n, this yields sequence embeddings $h \in \mathbb{R}^{n \times 768}$.

3.4.6 Computing Screening Set Enzyme Clusters

To exclude from our enzyme screening set those proteins that are similar to sequences used in our training dataset, we compute protein clusters using MMSeqs2 [98] and Foldseek [47]. For MMSeqs2, we use the default parameters with `--min-seq-id`= 0.3 and `--similarity`= 0.8. For Foldseek, we use the default parameters with `--min-seq-id`= 0 and `--c`= 0.3.

3.4.7 Baselines

Ranking Enzymes via EC Prediction

The ultimate goal of enzyme screening is to identify candidate proteins from large protein databases, including the hundreds of millions of unannotated sequences. Since no standard computational procedure for enzyme screening has emerged, a reasonable approach is to first assign an EC number to the query reaction and then select all enzymes that share that EC class. To identify the EC classes of the enzymes in the screening set, one can use an EC predictor. On the other hand, assigning the full EC number of a reaction is not always straightforward or possible. For this reason, we consider baselines where between 1 to 4 levels of a query reaction's EC number are assignable (e.g., 1 level: 1.x.x.x to 4 levels: 1.2.3.4). We evaluate EC prediction and CLIPZyme on ranking the enzymes screening set for each reaction in the EnzymeMap test set.

We use CLEAN, a state-of-the-art EC predictor, to obtain a ranked list of enzymes for each EC [61]. CLEAN computes a single representation for every EC in its dataset as the mean embedding of sequences in that class and uses these as test-time anchors. The predicted EC class of a new sequence is then determined by the Euclidean distance to each EC anchor. Accordingly, given a reaction's assigned EC number, we rank our screening set enzymes by their distances to the reaction's EC anchor (Figure 3.3). If a reaction's EC class does not exist in the CLEAN dataset, we broaden the search to one level higher. As an example, for a reaction with EC 1.2.3.4, if this EC is not in the CLEAN dataset, we rank enzymes according

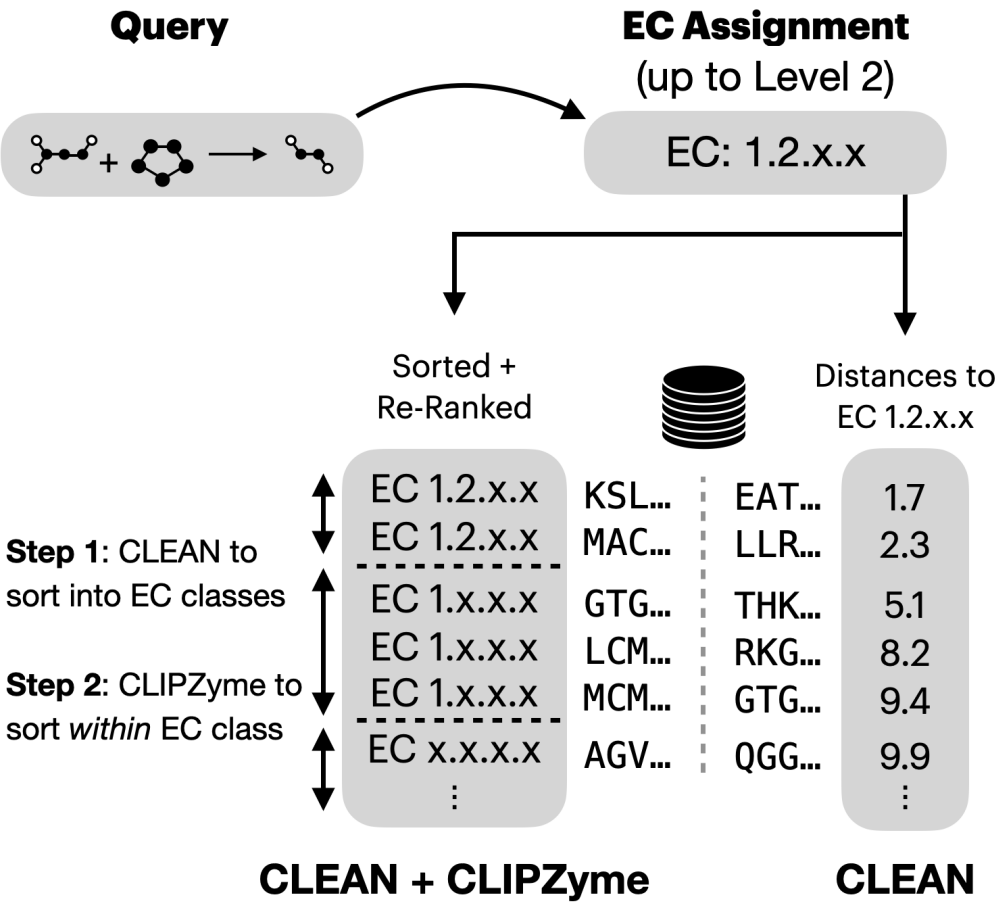

Figure 3.3: Approaches for adapting EC prediction to virtual enzyme screening. We first assign a reaction an EC up to some level of specificity (here, level 2). To obtain rankings based on CLEAN, we use each sequence's distance to the EC class. To combine CLEAN and CLIPZyme, enzymes are first sorted according to their predicted EC class. Then they are ranked within each class using CLIPZyme.

to their distances to the mean representation of EC 1.2.3 (and so on). For consistency with previous work, we use the CLEAN model trained on a split where none of the test enzymes share more than 50% sequence identity with those in the training set [61].

We hypothesize that combining CLEAN to obtain EC predictions and CLIPZyme to rank them presents an opportunity for improved performance. Specifically, we predict the EC numbers for all of the enzyme sequences in our screening set using CLEAN. Given the reaction's assigned EC number, we first filter our screening set to those enzymes with the

same exact predicted EC and rank this list using CLIPZyme (Figure 3.3). We then filter all remaining enzymes to those that belong to one EC level above and again rank that list using CLIPZyme. As an example, given an input reaction with assigned EC of 1.2.3.4, we identify all enzymes predicted for that EC and rank them with CLIPZyme. We then rank all remaining enzymes with predicted EC 1.2.3. This process is repeated until all enzymes are ranked.

Reaction Representation

We explore three alternative methods for encoding the reaction and compare against these in our results. The first uses the condensed graph reaction (CGR) representation [41] by overlaying the reactants and products and concatenating the edge features. A DMPNN encodes the CGR to obtain a hidden representation of the reaction. The second approach is to use the full reaction SMILES [63] as an input to a language model and obtain a final representation of the reaction. We follow the tokenization scheme for SMILES introduced by the Molecular Transformer [50] and train a transformer model as our encoder [38]. We also consider the Weisfeiler-Lehman Difference Network (WLDN) architecture and implement it as described in [9]. We train all models until convergence, using the same data splits and hyper-parameters (Section 3.5.3).

Protein Representation

We focus on achieving a balance between efficiency and the ability to process extensive enzyme datasets. To this end, we explore both sequence-based and structure-based approaches, acknowledging the critical influence of structure on enzymatic activity despite its inherent computational demands. We train ESM-2 [85] as a sequence-based baseline for protein encoding. We also encode the structure with an EGNN [86] and compare initializing node embeddings from either the MSA-transformer [96] or ESM-2, to identify the best method in terms of both performance and speed.

3.4.8 Evaluation Setup

We aim to simulate the scenario where an enzyme is desired to catalyze a novel reaction, and it exists in nature but is not annotated. We compare different approaches to encoding the reactions and their enzymes, and compare our method to an alternative approach using EC prediction.

As our main aim is to generalize to novel chemical transformations, our test set consists of reactions with reaction rules that are unseen during training, queried against all 260,197 sequences. However, this means our screening set does include proteins used in training the model. Therefore, we also evaluate model performance when excluding proteins used in training. Additionally, we use MMSeqs2 [98] and Foldseek [47] to exclude proteins based on their similarity to the training set proteins in terms of sequence identity and protein fold, respectively. If the exclusion of a protein results in a test reaction having no actives in the screening set, we exclude the entire reaction.

Throughout our evaluations, we take the BEDROC score as our primary metric [102]. We focus on the case $\alpha = 85$, where the top 3.5% of predictions contribute to 95% of the score, and as suggested in [102], we also calculate the BEDROC score for $\alpha = 20$. We also report the enrichment factor (EF) when taking the top 0.5% and 1% of predictions. This estimates the fraction of catalyzing enzymes found in our top predictions relative to random selection.

3.5 Results

We present here an overview of our key results. In Table 3.3, we compare CLIPZyme's performance to that of EC prediction with CLEAN and show the benefit of combining methods. CLIPZyme shows improved performance in all comparisons. Table 3.6 shows the impact of different protein and reaction representations and highlights the superior performance of our novel reaction encoding. In Table 3.7, we show that CLIPZyme's performance extends to a challenging dataset of terpene synthase reactions and unannotated

reactions. Lastly, we show in Table 3.8 that CLIPZyme's performance drops when screening enzymes that significantly differ from those it was trained on, but still maintains useful predictive value. Additional analysis is provided in **??**.

3.5.1 Enzyme Screening Evaluation on EnzymeMap

Table 3.3: Enzyme virtual screening performance compared to using EC prediction alone and together with CLIPZyme. For a given reaction EC level, enzymes are ranked according to their Euclidean distance to EC class anchors when using CLEAN [61]. Alternatively, CLEAN is first used to place enzymes into successively broader EC levels matching that of the reaction, and CLIPZyme is used to reorder the enzymes within each level. BEDROC: Boltzmann-enhanced discrimination of receiver operating characteristic; EF: enrichment factor.

EC Level Assumed Available	Method	$BEDROC_{85}$ (%)	$BEDROC_{20}$ (%)	$EF_{0.05}$	$EF_{0.1}$
-	CLIPZyme	44.69	62.98	14.09	8.06
Level 1 (x.-.-.-)	CLEAN	0.96	6.53	1.22	1.72
	CLIPZyme + CLEAN	57.03	78.50	17.84	9.56
Level 2 (x.x.-.-)	CLEAN	4.86	14.10	3.23	2.49
	CLIPZyme + CLEAN	75.57	90.20	19.40	9.84
Level 3 (x.x.x.-)	CLEAN	25.86	36.75	8.03	4.81
	CLIPZyme + CLEAN	82.69	93.23	19.43	9.84
Level 4 (x.x.x.x)	CLEAN	89.74	93.42	18.97	9.60
	CLIPZyme + CLEAN	89.57	95.24	19.43	9.84

CLIPZyme effectively ranks the screening set against reactions in the EnzymeMap test set with an average $BEDROC_{85}$ of 44.69% and an enrichment factor of 14.09 when choosing the top 5% (Table 3.3). We compare its performance to ranking using EC prediction with CLEAN. Since it is not always possible to assign all 4 levels of an EC to a chemical reaction, we examine scenarios where different EC levels are assumed to be known for query reactions in the test set.

For example, with only the first EC level known, using EC prediction alone obtains a $BEDROC_{85}$ score of 0.96% (Table 3.3). This improves to 25.86% when we are able to specify

a reaction up to the third EC level. With four EC levels known, the CLEAN method becomes more effective than CLIPZyme alone. However, being able to assign all four EC levels for a reaction may not be always feasible in real-world applications.

Combining the CLEAN method with CLIPZyme achieves improved performance regardless of how many EC levels we assume to be known for reactions. Here, CLEAN is first used to predict the EC classes of enzymes in the screening set. Enzymes within the predicted EC class are re-ranked using CLIPZyme (Figure 3.3). Even basic knowledge of the first EC level of a chemical reaction enhances CLIPZyme's performance from a $BEDROC_{85}$ of 44.69% to 57.03%. With the first two levels assumed to be known, performance also improves to 75.57%.

We note that EC classification may be insufficient for categorizing chemical reactions that do not fit in existing EC classes. As a result, any EC prediction method is not applicable in that setting, while CLIPZyme is as it operates directly on the reaction.

3.5.2 Enzyme Screening Within EC Classes

Table 3.4: Performance of CLIPZyme when limiting the screening set to enzymes belonging to the query reaction's top EC level.

$BEDROC_{85}$ (%)	$BEDROC_{20}$ (%)	$EF_{0.05}$	$EF_{0.1}$
36.25	51.61	11.30	6.83

We also explore CLIPZyme's ability to discriminate between enzymes within the same EC class, where enzymes are more likely to share function and physical-chemical features. To do so, for each query reaction in the test set, we adjust the screening set to include only those enzymes belonging to its EC class. The number of enzymes quickly diminishes when considering EC subclasses to the extent that the EC-based screening sets become too small for virtual screening (Figure 3.2b) – for example, the BEDROC metric is only valid only when ($\alpha \times$ proportion of actives) $\ll 1$. For this reason, we consider only the top EC level in

this analysis. We observe that it is more difficult to rank the correct enzymes higher when only considering sequences in the same EC class but that the top predictions are still enriched for the active enzymes (Table 3.4).

3.5.3 Adapting CLEAN for Ranking Enzymes

We consider using both CLEAN EC predictions and computed distances to perform virtual screening similar to Section 3.4.7. Here we present an alternative reranking approach than that in Section 3.5.1. We follow the exact same setup as reranking EC predictions using CLIPZyme but instead rerank using the distance to the EC anchors. For example, given a query reaction with EC 1.2.3.4, we first predict the EC numbers for all of the enzymes in the screening set using CLEAN. We then rank the enzymes with predicted EC of 1.2.3.4 by the distance from the anchor with EC 1.2.3.4 (computed as the mean embedding of all ECs in the CLEAN training set with EC 1.2.3.4). We then rank all remaining enzymes with predicted EC of 1.2.3.x by their distance to the anchor embeddings of EC 1.2.3.4 (this is the same anchor). This differs from the Section 3.5.1 approach since CLEAN assigns EC numbers based on a varying threshold (i.e., max-separation) for each embedding. By first ordering by EC and then reranking within each EC we achieve different results than by ranking all at once by distance to the 1.2.3.4 anchor.

Table 3.5: Enzyme virtual screening performance when using CLEAN to first place enzymes into successively broader EC levels matching that of the reaction, then re-ranking them according to their Euclidean distance to the reaction's EC.

EC Level	$BEDROC_{85}$ (%)	$BEDROC_{20}$ (%)	$EF_{0.05}$	$EF_{0.1}$
Level 1	5.43	26.94	5.55	6.33
Level 2	35.56	71.10	18.95	9.72
Level 3	63.40	85.61	19.35	9.74
Level 4	92.65	96.16	19.48	9.80

3.5.4 Impact of Reaction and Protein Representation

Table 3.6: Performance of various protein and reaction encoding schemes on virtual screening for reactions in the EnzymeMap test set. The symbol ❄ denotes models where the weights are kept unchanged during training.

Protein Encoder	Reaction Encoder	$BEDROC_{85}$ (%)	$BEDROC_{20}$ (%)	$EF_{0.05}$	$EF_{0.1}$
ESM❄	Ours (Section 3.3.1)	17.84	29.39	6.61	4.17
ESM	Ours (Section 3.3.1)	36.91	53.04	11.93	6.84
MSA-Transformer❄ + EGNN	Ours (Section 3.3.1)	28.76	46.53	10.34	6.67
ESM❄ + EGNN	CGR [41]	38.91	57.58	13.16	7.73
ESM❄ + EGNN	Reaction SMILES	29.94	46.01	10.34	6.32
ESM❄ + EGNN	WLDN [9]	29.84	46.70	10.71	6.41
ESM❄ + EGNN	Ours (Section 3.3.1)	44.69	62.98	14.09	8.06

We explore a number of different encoding methods for both reaction and protein representations and find that the model is highly sensitive to changes in both (Table 3.6). Using the molecular structures of the reaction obtains better performance than language-based methods operating over the reaction SMILES, with the former achieving a $BEDROC_{85}$ of 44.69% compared to 29.94%. This suggests that structural representations may capture chemical transformations that correspond to enzyme activity more explicitly than language based ones. The patterns observed in structures may be more difficult for language models to capture without additional features or data. Employing a more expressive model also improves performance when compared to using WLDN as the reaction encoder. While all reaction representation methods include the full reaction, they differ in how the bond changes are encoded. Methods that explicitly delineate chemical transformations between substrates and products appear to obtain generally better performance.

We find a similar sensitivity to enzyme encoding. We compare using ESM embeddings alone and using ESM embeddings together as node features for EGNN. We find that using an EGNN to capture the structural components of the enzyme improves performance compared to training a sequence-based model alone (44.69% compared to 36.91%), which indicates that

enzyme structure is important for achieving good performance on this task. We also explore initializing the EGNN node features with embeddings from the pre-trained MSA-Transformer [96]. These embeddings do not appear to improve performance, although they capture evolutionary information of the sequence. This, however, may be due to differences in quality of representations learned by ESM and MSA-Transformer in which ESM-2 was trained on much larger set of sequences.

3.5.5 Evaluation on Reaction-Specific Datasets

Table 3.7: Performance of CLIPZyme on additional biochemical reactions. The terpene synthase dataset is obtained from [62] and includes reactions considered to involve more complex biotransformations. The unannotated subset of EnzymeMap consists of reactions in the dataset that are not assigned a UniProt or SwissProt identifier. In this case, virtual screening is evaluated as the ability to highly rank proteins with the correct EC class.

Dataset	$BEDROC_{85}$ (%)	$BEDROC_{20}$ (%)	$EF_{0.05}$	$EF_{0.1}$
Terpene Synthases	72.46	85.89	18.29	9.42
Unannotated EnzymeMap	42.94	61.39	13.92	7.73

We extend our evaluation to two additional datasets to further assess CLIPZyme's utility in practical applications in Table 3.7. The first dataset encompasses reactions catalyzed by terpene synthases. We evaluated CLIPZyme using the same screening set and observed robust performance, evidenced by a $BEDROC_{85}$ score of 72.45%. Due to the small and uniform substrate pool, the model might be preferentially ranking terpene synthases as a whole, rather than effectively distinguishing between specific reactions.

Additionally, we present an evaluation using unannotated reactions from EnzymeMap. For the sake of evaluation, we assume the true enzymes in the screening set for a given reaction are those with EC classes matching that of the reaction. Under this setup, CLIPZyme achieves a $BEDROC_{85}$ of 42.94%, which aligns closely with the results from the annotated subset of EnzymeMap. Because the metrics are calculated relative to the EC classes of each protein, this result suggests that the CLIPZyme rankings correspond with the proteins' EC numbers.

3.5.6 Generalization to Novel Proteins

Table 3.8: Performance when excluding sequences from the screening set with various levels of similarity to training set enzymes.

Exclusion criteria	$BEDROC_{85}$ (%)	$BEDROC_{20}$ (%)	$EF_{0.05}$	$EF_{0.1}$
Exact Match	39.13	58.86	13.40	7.81
MMSeqs 30% Similarity	35.32	54.86	12.43	7.30
Foldseek 30% Similarity	21.44	35.39	7.93	4.93

Our primary focus has been on evaluating the generalization of CLIPZyme on reactions unseen during training. However, given the ultimate goal of screening a wide array of both annotated and unannotated enzymes, it's crucial to understand the model's efficacy in ranking proteins dissimilar to those in the training set.

To do so, we exclude proteins that are similar to our training set according to three similarity metrics. We first exclude training set enzymes. Second, we apply MMSeqs2 [98] to remove enzymes with 30% or greater sequence similarity. Lastly, we exclude enzymes with 30% fold similarity as determined by Foldseek [47]. By measuring performance on these three screening subsets, we demonstrate CLIPZyme's generalizability across both reactions and enzymes.

Each exclusion criteria led to a reduction in performance. For example, CLIPZyme's performance decreases by approximately 5 percentage points on both BEDROC metrics when excluding training set enzymes Table 3.8. The most marked impact was observed with Foldseek-based filtering, showing a 23.25 point decrease in $BEDROC_{85}$ scores. This aligns with our previous findings that protein structural features play a critical role in effective screening. Despite this, the model still demonstrated a notable ability to rank enzymes effectively as the top-ranked candidates consistently showed enrichment for active enzymes.

3.6 Conclusion

We present here the task of virtual enzyme screening and a contrastive method, CLIPZyme, to address it. We show that our method can preferentially rank catalytically active enzymes against reactions across multiple datasets. Without a standard baseline, we examine enzyme screening through EC prediction and highlight CLIPZyme's competitive ability. We furthermore show that combining EC prediction with CLIPZyme achieves significantly improved performance. Lastly, we evaluate CLIPZyme's capacity to generalize by evaluating it on additional challenging reaction datasets and on unseen protein clusters. In practical scenarios, where millions or even hundreds of millions of enzymes need screening, we foresee the necessity of methods like CLIPZyme with even higher sensitivity for effective enzyme design at scale.

Among its limitations, the current approach does not model the physical interactions between reactants and enzymes, and it is unable to capture the mechanisms that give rise to the observed reaction. Moreover, the available data covers a relatively small chemical space and includes a restricted set of reactions and enzyme sequences (e.g., EC class 7 is completely unrepresented). We also note that our approach of random negative sampling may give rise to false negatives due to the promiscuity of many enzymes and the method may benefit from alternative sampling techniques. Directions for future work include modeling the 3D interactions characterizing biochemical reactions (e.g., through docking) and leveraging transition state sampling through quantum chemical simulations.

Bibliography

[1] Ankur Sahu, Mary-Ann Blätke, Jędrzej Jakub Szymański, and Nadine Töpfer. Advances in flux balance analysis by integrating machine learning and mechanism-based models. *Computational and Structural Biotechnology Journal*, 19:4626–4640, 2021.

[2] Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. *arXiv preprint arXiv:2210.01776*, 2022.

[3] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873): 583–589, 2021.

[4] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637): 1123–1130, 2023.

[5] Yuexu Jiang, Duolin Wang, Yifu Yao, Holger Eubel, Patrick Künzler, Ian Max Møller, and Dong Xu. Mulocdeep: a deep-learning framework for protein subcellular and suborganellar localization prediction with residue-level interpretation. *Computational and structural biotechnology journal*, 19:4825–4839, 2021.

[6] Henry Kilgore, Itamar Chinn, Peter Mikhael, Ilan Mitnikov, Catherine Van Dongen, Guy Zylberberg, Lena Afeyan, Salman Banani, Susana Wilson-Hawken, Tony Lee, et al. Chemical codes promote selective compartmentalization of proteins. *bioRxiv*, pages 2024–04, 2024.

[7] Daniel Mark Lowe. *Extraction of chemical structures and reactions from the literature*. PhD thesis, University of Cambridge, 2012.

[8] Daniel Lowe. Chemical reactions from us patents (1976-sep2016), 2017. *DOI*, 10:m9, 1976.

[9] Wengong Jin, Connor Coley, Regina Barzilay, and Tommi Jaakkola. Predicting organic reaction outcomes with weisfeiler-lehman network. *Advances in neural information processing systems*, 30, 2017.

[10] M. Jeschek, Daniel Gerngross, and S. Panke. Combinatorial pathway optimization for streamlined metabolic engineering. *Current opinion in biotechnology*, 47:142–151, 2017. doi:10.1016/j.copbio.2017.06.014.

[11] Daniel Probst, Matteo Manica, Yves Gaetan Nana Teukam, Alessandro Castrogiovanni, Federico Paratore, and Teodoro Laino. Biocatalysed synthesis planning using data-driven learning. *Nature communications*, 13(1):964, 2022.

[12] Esther Heid, Daniel Probst, William H Green, and Georg KH Madsen. Enzymemap: Curation, validation and data-driven prediction of enzymatic reactions. 2023.

[13] Peter Mikhael, Itamar Chinn, and Regina Barzilay. Graph-based forward synthesis prediction of biocatalyzed reactions. In *ICLR 2024 Workshop on Generative and Experimental Perspectives for Biomolecular Design*.

[14] Peter G Mikhael, Itamar Chinn, and Regina Barzilay. Clipzyme: Reaction-conditioned virtual screening of enzymes. 2024.

[15] D Kreutter, P Schwaller, and JL Reymond. Predicting enzymatic reactions with a molecular transformer, chem. *Sci*, 12(25):8648–8659, 2021.

[16] Janani Durairaj, Alice Di Girolamo, Harro J Bouwmeester, Dick de Ridder, Jules Beekwilder, and Aalt DJ van Dijk. An analysis of characterized plant sesquiterpene synthases. *Phytochemistry*, 158:157–165, 2019.

[17] Connor W Coley, Regina Barzilay, Tommi S Jaakkola, William H Green, and Klavs F Jensen. Prediction of organic reaction outcomes using machine learning. *ACS central science*, 3(5):434–443, 2017.

[18] Marwin HS Segler and Mark P Waller. Neural-symbolic machine learning for retrosynthesis and reaction prediction. *Chemistry–A European Journal*, 23(25):5966–5971, 2017.

[19] John Bradshaw, Matt J Kusner, Brooks Paige, Marwin HS Segler, and José Miguel Hernández-Lobato. A generative model for electron paths. *arXiv preprint arXiv:1805.10970*, 2018.

[20] Hangrui Bi, Hengyi Wang, Chence Shi, Connor Coley, Jian Tang, and Hongyu Guo. Non-autoregressive electron redistribution modeling for reaction prediction. In *International Conference on Machine Learning*, pages 904–913. PMLR, 2021.

[21] Mikołaj Sacha, Mikołaj Błaz, Piotr Byrski, Paweł Dabrowski-Tumanski, Mikołaj Chrominski, Rafał Loska, Paweł Włodarczyk-Pruszynski, and Stanisław Jastrzebski. Molecule edit graph attention network: modeling chemical reactions as sequences of graph edits. *Journal of Chemical Information and Modeling*, 61(7):3273–3284, 2021.

[22] Shuan Chen and Yousung Jung. A generalized-template-based graph neural network for accurate organic reactivity prediction. *Nature Machine Intelligence*, 4(9):772–780, 2022.

[23] Gabriele Cruciani, Emanuele Carosati, Benoit De Boeck, Kantharaj Ethirajulu, Claire Mackie, Trevor Howe, and Riccardo Vianello. Metasite: understanding metabolism in human cytochromes from the perspective of the chemist. *Journal of medicinal chemistry*, 48(22):6970–6979, 2005.

[24] Lars Ridder and Markus Wagener. Sygma: combining expert knowledge and empirical scoring in the prediction of metabolites. *ChemMedChem: Chemistry Enabling Drug Discovery*, 3(5):821–832, 2008.

[25] Yannick Djoumbou-Feunang, Jarlei Fiamoncini, Alberto Gil-de-la Fuente, Russell Greiner, Claudine Manach, and David S Wishart. Biotransformer: a comprehensive computational tool for small molecule metabolism prediction and metabolite identification. *Journal of cheminformatics*, 11(1):1–25, 2019.

[26] Arndt R Finkelmann, Daria Goldmann, Gisbert Schneider, and Andreas H Göller. Metscore: site of metabolism prediction beyond cytochrome p450 enzymes. *ChemMedChem*, 13(21):2281–2289, 2018.

[27] Samuel E Adams. *Molecular similarity and xenobiotic metabolism*. PhD thesis, University of Cambridge, 2010.

[28] Johannes Kirchmair, Mark J Williamson, Avid M Afzal, Jonathan D Tyzack, Alison PK Choy, Andrew Howlett, Patrik Rydberg, and Robert C Glen. Fast metabolizer (fame): A rapid and accurate predictor of sites of metabolism in multiple species by endogenous enzymes. *Journal of chemical information and modeling*, 53(11):2896–2907, 2013.

[29] Ferenc Darvas. Metabolexpert: an expert system for predicting metabolism of substances. In *QSAR in environmental toxicology-II*, pages 71–81. Springer, 1987.

[30] Anastasia V Rudik, Alexander V Dmitriev, Alexey A Lagunin, Dmitry A Filimonov, and Vladimir V Poroikov. Metatox 2.0: Estimating the biological activity spectra of

drug-like compounds taking into account probable biotransformations. *ACS omega*, 2023.

[31] Homa Mohammadi Peyhani, Anush Chiappino-Pepe, Kiandokht Haddadi, Jasmin Hafner, Noushin Hadadi, and Vassily Hatzimanikatis. Database for drug metabolism and comparisons, nicedrug. ch, aids discovery and design. *bioRxiv*, pages 2020–05, 2020.

[32] Christina de Bruyn Kops, Martin Šícho, Angelica Mazzolari, and Johannes Kirchmair. Gloryx: prediction of the metabolites resulting from phase 1 and phase 2 biotransformations of xenobiotics. *Chemical research in toxicology*, 34(2):286–299, 2020.

[33] Tyler B Hughes and S Joshua Swamidass. Deep learning to predict the formation of quinone species in drug metabolism. *Chemical research in toxicology*, 30(2):642–656, 2017.

[34] Lars Olsen, Marco Montefiori, Khanhvi Phuc Tran, and Flemming Steen Jørgensen. Smartcyp 3.0: enhanced cytochrome p450 site-of-metabolism prediction server. *Bioinformatics*, 35(17):3174–3175, 2019.

[35] Matthias Hennemann, Arno Friedl, Mario Lobell, Jörg Keldenich, Alexander Hillisch, Timothy Clark, and Andreas H Göller. Cypscore: Quantitative prediction of reactivity toward cytochromes p450 based on semiempirical molecular orbital theory. *ChemMedChem: Chemistry Enabling Drug Discovery*, 4(4):657–669, 2009.

[36] David S Wishart, Yannick D Feunang, An C Guo, Elvis J Lo, Ana Marcu, Jason R Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, et al. Drugbank 5.0: a major update to the drugbank database for 2018. *Nucleic acids research*, 46(D1): D1074–D1082, 2018.

[37] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? 2022. URL https://openreview.net/forum?id=link-to-the-paper-if-available.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[39] Sergio Martí, Maite Roca, Juan Andrés, Vicent Moliner, Estanislao Silla, Iñaki Tuñón, and Juan Bertrán. Theoretical insights in enzyme catalysis. *Chemical Society Reviews*, 33(2):98–107, 2004.

[40] Arieh Warshel, Pankaz K Sharma, Mitsunori Kato, Yun Xiang, Hanbin Liu, and Mats HM Olsson. Electrostatic basis for enzyme catalysis. *Chemical reviews*, 106(8): 3210–3235, 2006.

[41] Frank Hoonakker, Nicolas Lachiche, Alexandre Varnek, and Alain Wagner. Condensed graph of reaction: considering a chemical reaction as one single pseudo molecule. *Int. J. Artif. Intell. Tools*, 20(2):253–270, 2011.

[42] Esther Heid and William H Green. Machine learning of reaction properties via learned representations of the condensed graph of reaction. *Journal of Chemical Information and Modeling*, 62(9):2101–2110, 2021.

[43] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019.

[44] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[45] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[46] Dhiraj Kalamkar, Dheevatsa Mudigere, Naveen Mellempudi, Dipankar Das, Kunal Banerjee, Sasikanth Avancha, Dharma Teja Vooturi, Nataraj Jammalamadaka, Jianyu Huang, Hector Yuen, et al. A study of bfloat16 for deep learning training. *arXiv preprint arXiv:1905.12322*, 2019.

[47] Michel van Kempen, Stephanie S Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron LM Gilchrist, Johannes Söding, and Martin Steinegger. Fast and accurate protein structure search with foldseek. *Nature Biotechnology*, pages 1–4, 2023.

[48] Anja Greule, Jeanette E. Stok, James J. De Voss, and Max J. Cryle. Unrivalled diversity: the many roles and reactions of bacterial cytochromes p450 in secondary metabolism. *Nat. Prod. Rep.*, 35:757–791, 2018. doi:10.1039/C7NP00063D. URL http://dx.doi.org/10.1039/C7NP00063D.

[49] Johannes Kirchmair, Andreas H Göller, Dieter Lang, Jens Kunze, Bernard Testa, Ian D Wilson, Robert C Glen, and Gisbert Schneider. Predicting drug metabolism: experiment and/or computation? *Nature reviews Drug discovery*, 14(6):387–404, 2015.

[50] Philippe Schwaller, Teodoro Laino, Théophile Gaudin, Peter Bolgar, Christopher A Hunter, Costas Bekas, and Alpha A Lee. Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction. *ACS central science*, 5(9):1572–1583, 2019.

[51] Marina V Omelchenko, Michael Y Galperin, Yuri I Wolf, and Eugene V Koonin. Non-homologous isofunctional enzymes: a systematic analysis of alternative solutions in enzyme evolution. *Biology direct*, 5:1–20, 2010.

[52] António J M Ribeiro, Gemma L Holliday, Nicholas Furnham, Jonathan D Tyzack, Katherine Ferris, and Janet M Thornton. Mechanism and catalytic site atlas (m-csa): a database of enzyme reaction mechanisms and active sites. *Nucleic acids research*, 46 (D1):D618–D623, 2018.

[53] Uwe T Bornscheuer, GW Huisman, RJ Kazlauskas, S Lutz, JC Moore, and K Robins. Engineering the third wave of biocatalysis. *Nature*, 485(7397):185–194, 2012.

[54] Euan J Hossack, Florence J Hardy, and Anthony P Green. Building enzymes through design and evolution. *ACS Catalysis*, 13(19):12436–12444, 2023.

[55] Antonio JM Ribeiro, Ioannis G Riziotis, Neera Borkakoti, and Janet M Thornton. Enzyme function and evolution through the lens of bioinformatics. *Biochemical Journal*, 480(22):1845–1863, 2023.

[56] Mingjie Liu, Azadeh Nazemi, Michael G Taylor, Aditya Nandy, Chenru Duan, Adam H Steeves, and Heather J Kulik. Large-scale screening reveals that geometric structure matters more than electronic structure in the bioinspired catalyst design of formate dehydrogenase mimics. *ACS Catalysis*, 12(1):383–396, 2021.

[57] Rohit Singh, Samuel Sledzieski, Bryan Bryson, Lenore Cowen, and Bonnie Berger. Contrastive learning in protein language space predicts interactions between drugs and protein targets. *Proceedings of the National Academy of Sciences*, 120(24):e2220778120, 2023.

[58] K Palepu, M Ponnapati, S Bhat, E Tysinger, T Stan, G Brixi, SR Koseki, and P Chatterjee. Design of peptide-based protein degraders via contrastive deep learning. biorxiv 2022. *Google Scholar*.

[59] Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, et al. Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research*, 50(D1): D439–D444, 2022.

[60] Antje Chang, Lisa Jeske, Sandra Ulbrich, Julia Hofmann, Julia Koblitz, Ida Schomburg, Meina Neumann-Schaal, Dieter Jahn, and Dietmar Schomburg. Brenda, the elixir core

data resource in 2021: new developments and updates. *Nucleic acids research*, 49(D1): D498–D508, 2021.

[61] Tianhao Yu, Haiyang Cui, Jianan Canal Li, Yunan Luo, Guangde Jiang, and Huimin Zhao. Enzyme function prediction using contrastive learning. *Science*, 379(6639): 1358–1363, 2023.

[62] Raman Samusevich, Teo Hebra, Roman Bushuiev, Anton Bushuiev, Ratthachat Chatpatanasiri, Jonáš Kulhánek, Tereza Čalounová, Milana Perković, Martin Engst, Adéla Tajovská, Josef Sivic, and Tomáš Pluskal. Discovery and characterization of terpene synthases powered by machine learning. *bioRxiv*, 2024. doi:10.1101/2024.01.29.577750. URL https://www.biorxiv.org/content/early/2024/01/31/2024.01.29.577750.

[63] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

[64] Philippe Schwaller, Benjamin Hoover, Jean-Louis Reymond, Hendrik Strobelt, and Teodoro Laino. Extraction of organic chemistry grammar from unsupervised learning of chemical reactions. *Science Advances*, 7(15):eabe4166, 2021.

[65] Shinsaku Fujita. Description of organic reactions based on imaginary transition structures. 1. introduction of new concepts. *Journal of Chemical Information and Computer Sciences*, 26(4):205–212, 1986.

[66] TI Madzhidov, PG Polishchuk, RI Nugmanov, AV Bodrov, AI Lin, II Baskin, AA Varnek, and IS Antipin. Structure-reactivity relationships in terms of the condensed graphs of reactions. *Russian Journal of Organic Chemistry*, 50:459–463, 2014.

[67] Chenru Duan, Yuanqi Du, Haojun Jia, and Heather J Kulik. Accurate transition state generation with an object-aware equivariant elementary reaction diffusion model. *arXiv preprint arXiv:2304.06174*, 2023.

[68] Puck van Gerwen, Ksenia R Briling, Charlotte Bunne, Vignesh Ram Somnath, Ruben Laplaza, Andreas Krause, and Clemence Corminboeuf. Equireact: An equivariant neural network for chemical reactions. *arXiv preprint arXiv:2312.08307*, 2023.

[69] Burckhard Seelig and Jack W Szostak. Selection and evolution of enzymes from a partially randomized non-catalytic scaffold. *Nature*, 448(7155):828–831, 2007.

[70] Nicholas S. Sarai, Tyler J. Fulton, Ryen L. O'Meara, Kadina E. Johnston, Sabine Brinkmann-Chen, Ryan R. Maar, Ron E. Tecklenburg, John M. Roberts, Jordan C. T. Reddel, Dimitris E. Katsoulis, and Frances H. Arnold. Directed evolution of enzymatic silicon-carbon bond cleavage in siloxanes. *Science*, 383(6681):438–443, 2024. doi:10.1126/science.adi5554. URL https://www.science.org/doi/abs/10.1126/science.adi5554.

[71] Gavin Ayres, Geraldene Munsamy, Michael Heinzinger, Noelia Ferruz, Kevin Yang, and Philipp Lorenz. Hifi-nn annotates the microbial dark matter with enzyme commission numbers.

[72] Jae Yong Ryu, Hyun Uk Kim, and Sang Yup Lee. Deep learning enables high-quality and high-throughput prediction of enzyme commission numbers. *Proceedings of the National Academy of Sciences*, 116(28):13996–14001, 2019.

[73] Theo Sanderson, Maxwell L Bileschi, David Belanger, and Lucy J Colwell. Proteinfer, deep neural networks for protein functional inference. *Elife*, 12:e80942, 2023.

[74] Daniela Röthlisberger, Olga Khersonsky, Andrew M Wollacott, Lin Jiang, Jason DeChancie, Jamie Betker, Jasmine L Gallaher, Eric A Althoff, Alexandre Zanghellini, Orly Dym, et al. Kemp elimination catalysts by computational enzyme design. *Nature*, 453(7192):190–195, 2008.

[75] Lin Jiang, Eric A Althoff, Fernando R Clemente, Lindsey Doyle, Daniela Rothlisberger, Alexandre Zanghellini, Jasmine L Gallaher, Jamie L Betker, Fujie Tanaka, Carlos F

Barbas III, et al. De novo computational design of retro-aldol enzymes. *science*, 319 (5868):1387–1391, 2008.

[76] Andy Hsien-Wei Yeh, Christoffer Norn, Yakov Kipnis, Doug Tischer, Samuel J Pellock, Declan Evans, Pengchen Ma, Gyu Rie Lee, Jason Z Zhang, Ivan Anishchenko, et al. De novo design of luciferases using deep learning. *Nature*, 614(7949):774–780, 2023.

[77] Ryan Feehan, Daniel Montezano, and Joanna SG Slusky. Machine learning for enzyme engineering, selection and design. *Protein Engineering, Design and Selection*, 34: gzab019, 2021.

[78] Brian D Weitzner, Yakov Kipnis, A Gerard Daniel, Donald Hilvert, and David Baker. A computational method for design of connected catalytic networks in proteins. *Protein Science*, 28(12):2036–2041, 2019.

[79] Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620 (7976):1089–1100, 2023.

[80] John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. *Advances in neural information processing systems*, 32, 2019.

[81] Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378 (6615):49–56, 2022.

[82] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016.

[83] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[84] Guillem Casadevall, Cristina Duran, and Sílvia Osuna. Alphafold2 and deep learning for elucidating enzyme conformational flexibility and its application for design. *JACS Au*, 3(6):1554–1562, 2023.

[85] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022.

[86] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

[87] Sarp Aykent and Tian Xia. Gbpnet: Universal geometric representation learning on protein structures. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4–14, 2022.

[88] Kenneth Atz, Clemens Isert, Markus NA Böcker, José Jiménez-Luna, and Gisbert Schneider. δ-quantum machine-learning for medicinal chemistry. *Physical Chemistry Chemical Physics*, 24(18):10775–10783, 2022.

[89] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala.

PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[90] William Falcon and The PyTorch Lightning team. PyTorch Lightning, March 2019. URL https://github.com/Lightning-AI/lightning.

[91] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[92] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[93] Uniprot: the universal protein knowledgebase in 2023. *Nucleic Acids Research*, 51(D1): D523–D531, 2023.

[94] Zhuofu Ni, Andrew E Stine, Keith EJ Tyo, and Linda J Broadbelt. Curating a comprehensive set of enzymatic reaction rules for efficient novel biosynthetic pathway design. *Metabolic Engineering*, 65:79–87, 2021.

[95] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[96] Roshan Rao, Jason Liu, Robert Verkuil, Joshua Meier, John F. Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. *bioRxiv*, 2021. doi:10.1101/2021.02.12.430858. URL https://www.biorxiv.org/content/10.1101/2021.02.12.430858v1.

[97] Michael Remmert, Andreas Biegert, Andreas Hauser, and Johannes Söding. Hhblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment. *Nature methods*, 9(2):173–175, 2012.

[98] Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.

[99] Milot Mirdita, Konstantin Schütze, Yoshitaka Moriwaki, Lim Heo, Sergey Ovchinnikov, and Martin Steinegger. Colabfold: making protein folding accessible to all. *Nature methods*, 19(6):679–682, 2022.

[100] Baris E Suzek, Yuqi Wang, Hongzhan Huang, Peter B McGarvey, Cathy H Wu, and UniProt Consortium. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015.

[101] Milot Mirdita, Lars Von Den Driesch, Clovis Galiez, Maria J Martin, Johannes Söding, and Martin Steinegger. Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic acids research*, 45(D1):D170–D176, 2017.

[102] Jean-François Truchon and Christopher I Bayly. Evaluating virtual screening methods: good and bad metrics for the "early recognition" problem. *Journal of chemical information and modeling*, 47(2):488–508, 2007.