RxnPred: A Generative Framework for Enzymatic Reaction Prediction

Srikrishnan Balamuhunthan[1], Saketha Nath Jagarlapudi[2] and Karthik Raman[1*]

[1] *Department of Data Science and Artificial Intelligence, Indian Institute of Technology Madras, Chennai, India*

[2] *Department of Computer Science and Engineering, Indian Institute of Technology Hyderabad, Hyderabad, India*

[*] *Corresponding author: kraman@dsai.iitm.ac.in*

Enzyme function annotation remains a challenging problem, even with many advancements in machine learning and deep learning. Most computational tools predict Enzyme Commission numbers, which are occasionally reassigned, or retrieve reactions from a candidate pool. We introduce RxnPred, which instead generates reaction SMILES from protein sequence and structure inputs, using an encoder-decoder transformer model trained on protein-reaction pairs. We evaluate RxnPred on three data splits of increasing difficulty in terms of sequence identity, and on three external test sets, comparing against a homology baseline, a contrastive learning reaction screening tool, and four EC prediction tools. On reactions seen during training, RxnPred is competitive with these tools and achieves higher recall under beam search, though at lower precision; the homology baseline nevertheless remains the strongest method. Performance drops when test sequences become more dissimilar from training data, and on unseen reactions, the model does not recover the correct answer, whereas a retrieval framework can rank such reactions because they are supplied as input. Its predictions for these entries are nonetheless closer to the ground truth by fingerprint similarity than those of the homology baseline and an EC prediction tool. The model does generate reaction SMILES absent from its training data, a small fraction of which map to existing reactions, but assigns them to the wrong enzymes. We identify metabolite-level tokenization and protein-reaction alignment, rather than reaction generation, as the main bottleneck, and granular tokenization, chemical validity constraints, and strong alignment as directions for improvement.

Keywords: Enzyme function prediction; reaction SMILES generation; reaction prediction

1. Introduction

Protein function prediction is an active area of research due to the task's challenging nature. As the available corpus of genome sequences grows ever faster, so does the number of proteins, but the functions of many of them are still unknown. Protein functions are commonly defined using Gene Ontology terms with three subcategories: Molecular Function, Cellular Component, and Biological Process. For enzymes, the common way to describe their function is through Enzyme Commission (EC) numbers, which are expert-defined four-digit codes, with each digit having a specific meaning. Apart from experimentally determining what reaction an enzyme catalyses, computational methods have been developed to expedite the process of

© 2026 The Authors. Open Access chapter published by World Scientific Publishing Company and distributed under the terms of the Creative Commons Attribution Non-Commercial (CC BY-NC) 4.0 License.

functional characterisation of new proteins.[1] Traditionally, EC numbers are assigned either by homology search using tools like blastp[2] or by using Hidden Markov Model profiles.[3] Recently, deep learning tools have been developed for function prediction. For example, CLEAN[4] is a contrastive learning tool for predicting EC numbers from protein sequences. CLEAN-Contact[5] utilises protein structure in addition to sequence for EC prediction. HIT-EC[6] is a transformer-based EC tool that predicts EC number hierarchically. While the EC numbers provide a standard nomenclature, they are occasionally reassigned or deprecated, making it difficult to have a consistent representation of the enzyme function.

A more specific way to describe enzyme function is through the exact biochemical reaction(s) it catalyses. Unlike EC numbers, the exact reactions are given by various identifiers across different databases. However, representing the metabolites of the reaction as SMILES (Simplified Molecular Input Line Entry System) strings, and thus the reaction as a reaction SMILES string, will yield a consistent representation. While most existing tools focus on the EC number, very few recent methods have shifted their focus to exact reactions. CLIPZyme[7] is an enzyme screening tool, where enzyme structures and reactions are encoded using Graph Neural Networks. Horizyn-1[8] encodes reactions using reaction fingerprints and encodes proteins using protein language models for reaction or enzyme retrieval. ReactZyme[9] benchmarks different architectures on enzyme-reaction prediction tasks, across diverse challenging data splits. While these reaction-centric tools rank reactions from a given pool, a generative framework generates them from scratch, which none of these tools does. This opens the label space, since the model is not restricted to reactions present in the training, at the risk of generating chemically invalid ones. Improving the accuracy of enzyme function annotation has the potential to accelerate genome-scale metabolic model reconstruction and annotation of metagenomes.

Pretrained and fine-tuned generative models exist for organic chemistry reactions, which are used for forward reaction prediction or retrosynthesis.[10–12] However, biochemical reactions differ from those reactions because they depend on the input enzyme. The amount of data available for organic chemistry reactions is also disproportionately larger than that available for enzymatic reactions. For example, Molecular Transformer was trained on hundreds of thousands of reactions, while the total number of reactions present in Rhea curated database is 18,558.[10,13] These factors likely explain why, to our knowledge, no existing method addresses this problem as a generative task.

In this work, we introduce RxnPred, a generative framework that predicts reaction SMILES directly from protein sequences and structures (Figure 1A), and evaluate it across three data splits of increasing sequence dissimilarity and three external test sets against a homology baseline, a retrieval based tool, and four EC prediction tools, characterizing where the generative framing helps and where it does not. A generative framework, especially with an open SMILES vocabulary, is not restricted to generating reactions only from the training data. However, we used a full metabolite vocabulary, which shortens the tokenized sequences considerably but is closed: the model cannot predict any metabolite out of the vocabulary. By simulating a test setting in which a model is prompted to predict the functions of enzymes whose associated reactions it has not seen, we assess its ability to generalise in biochemistry. RxnPred

is competitive with a homology baseline on reactions seen during training and outperforms a contrastive retrieval tool, with beam search raising recall at the cost of precision, though the homology baseline remains the strongest method in this setting. It does generate reaction SMILES absent from its training set, but assigns them to the wrong enzymes, so the generative advantage remains unrealized.

2. Methods

2.1. *Training data and preprocessing*

RxnPred was developed using protein and reaction data from SwissProt, a curated subset of UniProt[14] and Rhea.[13] We downloaded the protein sequences, along with their EC numbers and Rhea annotations, from SwissProt (as on 15th Sep 2025). We removed protein sequences shorter than 30 amino acids or longer than 1024 amino acids, retaining only the entries that have at least one Rhea association and not more than 10. We considered only the forward (left-to-right) versions of all the reactions and we filtered out reactions that lacked a valid SMILES string from Rhea. Each metabolite's SMILES string was canonicalised using RDKit.[15] We evaluated our model on three data splits: one random split and two sequence-similarity-based splits. For the sequence similarity-based splits, we used MMseqs2[16] to cluster sequences at two sequence identity thresholds: 30% and 70%. A query coverage of 0.5 was considered in all clustering experiments. Train-validation-test splits were made in such a way that sequences belonging to the same cluster are present within the same split, and there is no leakage across the splits, with an approximate ratio of 0.8/0.1/0.1.

Since our method's main focus is on monofunctional enzymes, validation and test splits contain only enzymes with a single reaction association. However, multifunctional enzymes increase reaction coverage and were therefore included in the training set. Our model being a sequence-to-sequence transformer, each training item is expected to be a pair of an enzyme and a reaction. For these multifunctional enzymes, the sequences were duplicated and differentially masked to create augmented sequences, following CLEAN[4]. We similarly augmented reactions associated with fewer than 10 enzymes. Based on metabolite frequencies in the training split, metabolites in a reaction were ordered from least to most frequent on both the reactant and product sides. We obtained the structures of all the proteins from the AlphaFold structure database v6,[17] using LocalColabFold(`https://github.com/YoshitakaMo/localcolabfold`) for structures absent from AlphaFold database (Supplementary material Section S1.1). The resulting training sets are comparable in size across splits: 241,820 entries (132,565 proteins, 9,643 reactions) for the random split, 238,230 (132,578 proteins, 8,994 reactions) for MMseqs30, and 235,992 (132,580 proteins, 8,827 reactions) for MMseqs70. Data, code, and all supplementary materials are available at `https://github.com/srikrishnan-b/ReactionPred`.

2.2. *Protein and reaction representation*

We used the protein language model ESM2[18] (`esm2_t12_35M_UR50D`) to generate embeddings for the proteins. The embeddings are of shape $L \times d$, where L is the length of the sequence, and d is the embedding dimension, which is 480 for this model. For proteins longer than 512 residues, the first 512 residues were taken. All embeddings were precomputed and stored on

the disk in half-precision (FP16) format. From the protein structures, we used ESM-IF1[19] to generate embeddings of shape $L \times d$, where L is the length of the sequence, and d is the embedding dimension, which is 512 in this case. The final protein representation is obtained by concatenating the structure embedding with the sequence embedding along the embedding dimension, yielding a representation of shape $L \times 992$. Reactions are represented as reaction SMILES, where the metabolites are separated by a "." token and the reactants and products are separated by a ">>" token. ChemBERTa-10M-MLM[20] was used to obtain 384-dimensional embeddings for all the metabolites from their SMILES strings.

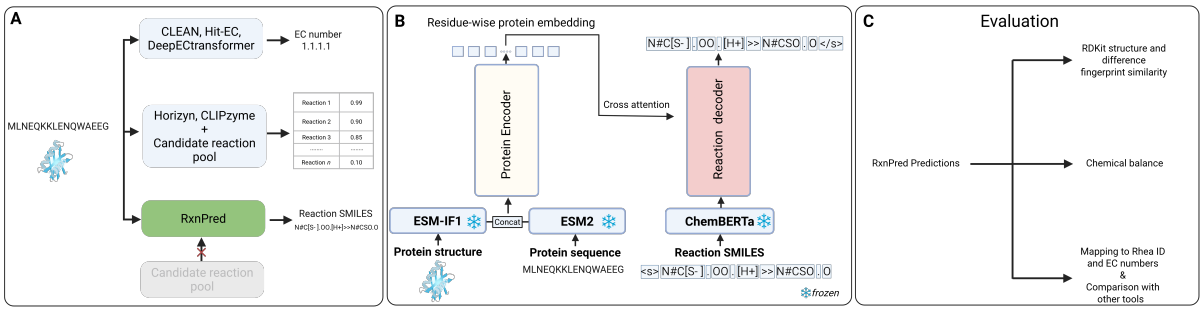


Fig. 1. **RxnPred Overview:** A. Existing tools predict a fixed EC label or rank a supplied candidate pool; RxnPred directly generates reaction SMILES. B. Architecture: fused ESM-2 and ESM-IF1 embeddings feed a protein encoder; the reaction decoder cross-attends and generates metabolite tokens autoregressively. C. Predictions are evaluated by fingerprint similarity, chemical balance, and mapping to Rhea and EC. (Created in BioRender. Raman, K. (2026) https://BioRender.com/rqkj8j8)

2.3. *RxnPred implementation*

RxnPred is a sequence-to-sequence transformer model adopting the BART[21] skeleton, which consists of a bidirectional encoder and an autoregressive decoder. We adopted the underlying encoder and decoder architectures, but the model is initialized randomly and trained from scratch. RxnPred has two transformer modules: a protein encoder and a reaction decoder (Figure 1B). We chose this architecture because the task involves mapping a variable-length input to a variable-length output, which is the standard setting for an encoder-decoder with cross attention. The protein embeddings are processed by the encoder, which consists of self-attention layers, and its last hidden state $H \in \mathbb{R}^{L \times d}$ is used by the decoder. The reaction decoder also has self-attention and cross-attention layers, with the reaction SMILES string as the input during training, and H used for cross-attention. For the decoder, in the input, a `<START>` token acts as the first token and the original tokens are shifted to the right by one position, while the output to be predicted ends with a `<STOP>` token. The reaction SMILES strings are tokenized into individual metabolites and special reaction tokens, making each individual metabolite a token. We chose this tokenization scheme to reduce the complexity arising from the longer input sequences produced by atom-wise tokenizers. For the reaction decoder, we constructed the vocabulary from all the metabolites available in our dataset. The

embedding layer for this was initialized from the precomputed ChemBERTa embeddings and remains frozen throughout the training. The non-metabolite special token (eg., "." and ">>") embeddings are randomly initialized and updated throughout the training.

The model is trained with teacher forcing using cross-entropy loss. The final model uses four encoder layers and six reaction decoder layers, with four attention heads and $d_{\text{model}} = 320$. The feed-forward dimension is 4096 in the encoder and 2048 in the decoder. Models were trained up to 25 epochs with early stopping on validation loss, using AdamW optimizer with a peak learning rate of 4.4×10^{-4}, cosine annealing, and a linear warmup over the first 10% of steps. We used a per-device batch size of 64 with two gradient accumulation steps, giving an effective batch size of 128. Training was implemented with the HuggingFace Transformers `Trainer`[22] (v4.57.1) on PyTorch[23] (v2.13.0) and Python 3.10.0, under Ubuntu 22.04.5 LTS. Experiments were run on NVIDIA RTX PRO 2000 Blackwell and NVIDIA A100 GPUs. Training took approximately 10.4 hours for the longest run on NVIDIA RTX PRO 2000 Blackwell GPU.

We evaluated two modes of inference: greedy decoding and beam search. In beam search, instead of a single output, k (beam width) best partial outputs are maintained at each decoding step, ranked by cumulative log probability.[24] We used a beam width of five for the internal test splits and ten for the external test sets. All k completed beams are retained, ranked by final log probability without length normalisation. This rank is the prediction rank in all top-k evaluations. During inference, the protein sequence and structure embeddings are given as input to the encoder, and the decoder is given a `<START>` token.

2.4. *Evaluation*

For evaluation, we use the internal test set of each split. In addition, we also evaluated the models on three external test sets: Price-149, Halogenase, and New-477. Price is a set of 149 proteins curated by ProteInfer,[25] from a previous work,[26] where they experimentally verified the functions of poorly annotated bacterial proteins, and is used as a benchmark set commonly.[4,5,27] Halogenase is a set of 37 halogenase enzymes, whose functions were assigned experimentally or through expert curation in CLEAN.[4] We constructed a new test set of 477 proteins released after our model's training cutoff date or excluded from our training data, providing a real-world scenario for evaluation. This set contains sequences released on SwissProt from 01/01/2025 to 27/03/2026. We made sure that none of the entries from these internal and external tests was used in the training process. In the external sets, only the entries with a Rhea annotation and that are monofunctional are considered, leaving Halogenase with 30 entries covering 12 Rhea reactions, Price with 110 entries covering 40 reactions, and New with 288 entries covering 194 reactions. We evaluate the model on two tasks: reaction prediction performance primarily, and EC number prediction performance as a proxy for reaction prediction.

The predicted reaction sequence is compared with the true reaction sequence using structure fingerprint Tanimoto similarity (RDKit CreateStructuralFingerprintForReaction) and difference reaction fingerprint Tanimoto similarity (RDKit CreateDifferenceFingerprintForReaction) (Figure 1C). Structure fingerprint encodes substructures present in reactants and products, while the difference fingerprint encodes the changes between them. Tanimoto simi-

larity is defined in Supplementary material Section S1.2. We check what fraction of predictions have similarity equal to or above a given threshold. We also assess whether the predicted reaction is mass- and charge-balanced. Further, all predictions are mapped to any of the existing Rhea IDs and from each Rhea ID to EC numbers. For comparison with other tools, we consider this as multi-label prediction and compute Precision, Recall and F1 scores for both Rhea and EC labels. To compare with a reaction screening tool, we calculate the top-k hit rate at different k values, which computes the fraction of predictions for which at least one of the ground truth reactions is predicted within the top-k ranks. In our model predictions, beams are considered as ranks. Multilabel EC and Rhea metrics are computed per label across the test set and averaged with weighting by label support. Results are reported separately for each internal and external test set.

For comparison with other tools, we used their released versions, except for CLEAN, which we also retrained on our datasets. The full details about the implementation can be found in the Supplementary material Section S1.4. We also performed experiments with a variant in which the structure was removed from the input, and another in which an additional EC decoder was used to predict EC up to 3 digits (Supplementary material Section S1.3).

3. Results

3.1. *Reaction generation performance of RxnPred*

RxnPred was trained and evaluated on three data splits: random, MMseqs30, and MMseqs70. In MMseqs30, no test sequence shares more than 30% identity with any training sequence at 0.5 query coverage, and in MMseqs70, the corresponding threshold is 70%. First, we evaluated the reaction prediction performance of our model on all the test splits. We compared the reaction SMILES generated by our model, obtained via either greedy decoding or beam search, with the ground truth using structure- and difference-fingerprint similarities. A beam width of 5 was used for the test sets, and the maximum similarity score for each entry was taken.

Figure 2 shows the fraction of predictions with similarity equal to or more than a given threshold for six thresholds from 1.0 to 0.5. With beam search, structure fingerprint exact matches fall from 93% on the random split to 83% on MMseqs70 and 56% on MMseqs30; difference fingerprint exact matches were 90% on the random split. Average similarities follow the same ordering (Figure 2).

In general, the structure fingerprint similarity scores are higher than the difference fingerprint scores, indicating that the model predicts similar metabolites even when the captured reaction transformation is incorrect. The reduction in exact matches and average similarity scores in MMseqs splits indicates that the model struggles to maintain accuracy as test set entries become more dissimilar to the training distribution. We assessed how the chemical balance of predicted reactions varies across beams and found that the first-ranked beam was balanced in 99% of predictions in the random test split (Supplementary material Section S2.1). The proportion of balanced predictions decreases as the beam rank increases, pointing towards the need for inference time steering to ensure chemical balance.

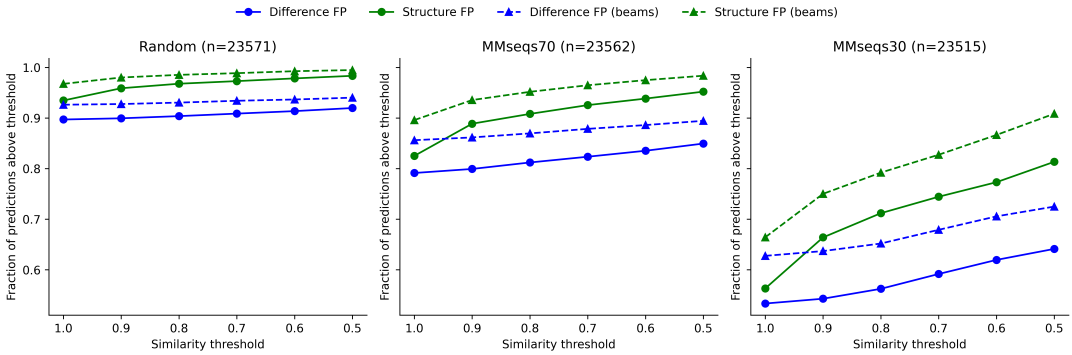


Fig. 2. Cumulative fraction of predictions from greedy decoding (solid lines) and beam search with width 5 (dashed lines) whose difference or structure fingerprint (FP) similarity to the ground truth is at or above the threshold on x-axis, running from 1.0 (exact match) to 0.5; blue – difference FP; green – structure FP. Panels: random (n=23,571), MMseqs70 (n=23,562) and MMseqs30 (n=23,515).

3.2. *Comparison with a homology baseline*

We investigated how our model performs in predicting the reaction compared with DIAMOND[28](v2.1.11) blastp (hereafter blastp), a homology baseline, on the internal test splits and the external test sets. Using blastp, the test set entries were queried against the training data as the target database, and either the top hit or the top-10 hits (top-5 for the random, MMseqs70, and MMseqs30 splits) were considered. The Rhea annotation of the hits is transferred to the test queries. Using our model, we predicted reactions for the test sets using greedy decoding and beam search with a width of 10 (beam width of 5 for the random, MMseqs70, and MMseqs30 splits). Then we computed multilabel classification metrics, precision, recall and F1 scores in the weighted average setting.

Table 1 shows the scores achieved by blastp and RxnPred across all test sets at the top-1 and top-k, where k is 5 for internal test splits and 10 for external test sets. When only the top hit is considered, RxnPred has a higher precision than blastp on Price and leads on all three metrics on the random split, while blastp lies ahead in the other test sets. The recall is also better for blastp across all test sets except the random test at the top-1 level. With beam search, our model's recall increases significantly as we consider more retrieved reactions. With top-k hits, though blastp's recall increases in the external test sets, it hardly increases in the internal test sets. The reason is that the training set contains many sequences similar to the queries and that share the same annotation, so they are retrieved as top hits. For the same reason, precision is not affected by using top-k hits with blastp. Conversely, beam search generates k candidate reactions, most of which are not the ground truth. On the internal splits this raises recall but sharply reduces precision. On the external sets, precision instead increases, because these sets contain far fewer distinct ground-truth reactions and the metric is weighted by label support, so false positives on rare labels contribute little to the average. In general, our model performs poorly on the Halogenase set but better on the Price dataset. Given that generation is a harder task than retrieving similar sequences from training data, matching or exceeding blastp's recall on five of six test sets at top-k is a reasonable outcome, though it comes at lower precision on the internal splits.

Table 1. Reaction prediction as multilabel classification: RxnPred (RxP) compared with blastp (Dmd), using weighted precision, recall, and F1. *Top-5 for internal test sets, Top-10 for external test sets.

	Top-1						Top-5/10*					
	Precision		Recall		F1		Precision		Recall		F1	
Test set	Dmd	RxP	Dmd	RxP	Dmd	RxP	Dmd	RxP	Dmd	RxP	Dmd	RxP
Price	0.52	**0.57**	**0.34**	0.32	**0.38**	**0.38**	0.62	**0.75**	0.46	**0.61**	0.49	**0.64**
Halogenase	**0.42**	0.13	**0.20**	0.13	**0.24**	0.12	**0.48**	0.45	**0.43**	0.33	**0.45**	0.30
New	**0.55**	0.46	**0.51**	0.42	**0.52**	0.43	**0.55**	0.48	0.54	**0.57**	**0.54**	0.50
Random split	0.91	**0.93**	0.90	**0.93**	0.90	**0.93**	**0.90**	0.67	0.90	**0.97**	**0.89**	0.76
MMseqs70	**0.86**	0.84	**0.84**	0.82	**0.85**	0.82	**0.84**	0.62	0.85	**0.89**	**0.84**	0.70
MMseqs30	**0.73**	0.66	**0.65**	0.54	**0.67**	0.55	**0.73**	0.57	**0.65**	**0.65**	**0.67**	0.56

3.3. *Comparison with a retrieval-based framework*

As an alternative to traditional homology-based methods, the problem of reaction assignment has recently been approached as a retrieval task, in which a protein is screened against a set of reactions, and the reactions are ranked, or vice versa. Horizyn-1[8] is a recent work that uses a dual encoder contrastive learning framework to screen enzymes for a given reaction, utilising cosine similarity scores between reaction and enzyme representations. Since this computation is symmetric, we used this framework to evaluate the model's ability to predict reactions for given enzymes. We used both released Horizyn-1 checkpoints, the development checkpoint used in the paper and the inference checkpoint trained on a larger dataset (Supplementary material Section S1.4).

We also report the top-k hit rate obtained using blastp. We used beam search of width 10 to generate results for these datasets from our model. Hence, we considered the Top-k hit rate for k values of 1, 3, 5, 7, and 10. In the combined dataset of size 657 (including multifunctional enzymes, excluding reactions with no SMILES), RxnPred matches or exceeds both Horizyn-1 checkpoints at every k (Figure 3, left).

Interestingly, the Horizyn-1 inference checkpoint was trained on a larger dataset, and we do not know which proteins and reactions were included in it or whether they are present in the test sets. Since the number of unseen reactions for Horizyn-1 and our model differs, for a fair comparison, we evaluated only top-k hit rates on a subset of 426 entries whose reactions were commonly seen by both models during training (Figure 3, right). In this, both blastp and RxnPred outperformed Horizyn-1 across all k levels. Although blastp leads at top-1 and top-3, RxnPred matches or exceeds it from top-5 onward, in both the full set and the subset of commonly seen reactions (Figure 3, left and right). Horizyn-1, however, performed better on a subset of common, unseen reactions for both models, in which it was able to rank unseen reactions in the top 10 at a rate of 0.15, while both blastp and RxnPred failed to predict any unseen reaction. This is possible because of the nature of the screening procedure, in which all possible reactions are presented to the model during inference for ranking, even if they were unseen during training. Blastp does not inherently predict unseen labels, as it just

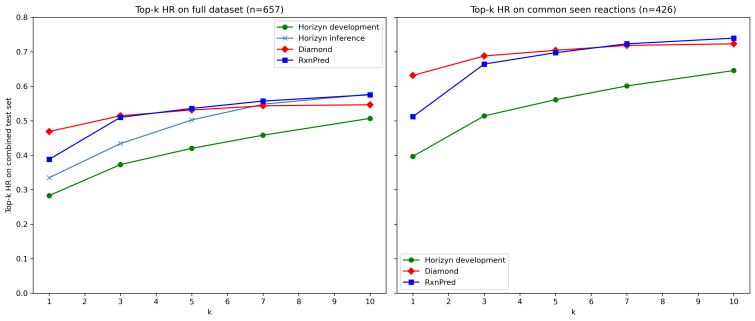


Fig. 3. Top-k hit rates for RxnPred compared with Horizyn-1 and blastp for the combined external test sets ($n = 657$) and the subset of $n = 426$ entries whose reactions were seen during training by both models.

picks candidates from the training data. RxnPred, on the other hand, is prompted to generate reactions for a protein from scratch. Although the generative framework can generate new reactions, our current model could not produce the correct reactions for these unseen entries. The model could, in fact, generate new reactions but for the wrong proteins, as explained in Section 3.5. The generative capabilities can be further improved by experimenting with different tokenization strategies that are more granular, as the notion of unseen metabolites vanishes in those cases, which is part of our future work.

3.4. *Comparison with EC prediction tools*

Next, we evaluated how our model compares to existing EC number prediction tools. For this, we mapped the predicted reaction SMILES to a Rhea identifier, then to an EC number associated with that Rhea ID. We use this comparison as a proxy for reaction prediction and compare it with existing EC tools. The tools used for comparison were DeepECtransformer,[29] CLEAN, CLEAN-Contact, HIT-EC, and blastp. The original released models were used for inference. We also checked the results after retraining CLEAN on our splits. For our model and blastp, results from two versions are reported: greedy and beam search for RxnPred, top-1 and top-10 for blastp. For the released models, we evaluated only on the external test sets, as our test set split might include proteins used in their training. We used our own splits to evaluate blastp and retrained CLEAN. The results for the external sets are presented in Table 2. On Price, as with the Rhea multilabel metrics, RxnPred with beam search showed the best recall and F1 scores. In the Halogenase dataset, CLEAN showed the best precision, recall, and F1 scores.

In the New dataset, recall was highest for RxnPred, while blastp showed higher precision and F1. The pattern of beam search increasing recall continues here as well, suggesting that RxnPred could predict the correct reactions given the chance to generate more candidates, even when its most confident prediction is incorrect. While CLEAN's released model performs well on the Halogenase dataset, its performance declined sharply when retrained on our dataset, suggesting the gap is attributable to training set composition rather than architecture. On the internal splits, blastp and retrained CLEAN lead on precision, while RxnPred with beam search remains competitive on recall (Supplementary material Section S2.3).

Table 2. Multilabel EC label metrics Precision, Recall, and F1 are computed from different models on the three external tests, and compared with RxnPred. Best result is written in bold.

	Price			Halogenase			New		
Method	P	R	F1	P	R	F1	P	R	F1
blastp (top-1)	0.42	0.28	0.31	0.37	0.18	0.21	0.63	0.61	0.62
blastp (top-10)	0.56	0.47	0.47	0.43	0.38	0.40	**0.64**	0.65	**0.64**
DeepECtransformer	0.53	0.31	0.36	0.13	0.15	0.14	0.52	0.47	0.49
CLEAN	0.61	0.49	0.52	**0.76**	**0.71**	**0.72**	0.56	0.54	0.55
CLEAN-Contact	**0.63**	0.53	0.54	0.00	0.00	0.00	0.58	0.56	0.56
HIT-EC	0.41	0.23	0.28	0.41	0.06	0.10	0.52	0.46	0.48
CLEAN (retrained)	0.40	0.33	0.34	0.52	0.41	0.43	0.62	0.59	0.60
RxnPred (greedy)	0.46	0.27	0.31	0.12	0.12	0.11	0.49	0.52	0.49
RxnPred (beam)	0.60	**0.57**	**0.55**	0.40	0.29	0.27	0.49	**0.69**	0.54

3.5. *Generalization to unseen reactions*

We wanted to assess the generalization ability of our model on unseen reactions. We investigated two aspects: 1) Can the model correctly predict an unseen reaction? 2) Can the model predict any unseen reaction at all, whether it is correct or not, being irrelevant. For the second question, the answer is no by design for CLEAN and blastp, as they cannot generate a new reaction that the model has not been trained on or is not present in the training data. To answer the first question, we evaluated RxnPred, blastp, and CLEAN on random test split entries with unseen reactions. We computed the structure and difference fingerprint similarity between the predicted and actual reactions. For CLEAN, we mapped the EC number to all possible reactions and selected the one that showed the maximum similarity. For RxnPred, we used beam search with a width of 5 on our test set and selected the beam with the highest score. For blastp, top-5 hits were considered. For the 226 entries with unseen reactions, RxnPred's predictions were closer to the ground truth than those of blastp and CLEAN on both fingerprints (Figure 4).

Next, we checked whether the model had generated any new reaction that was completely unseen during training, irrespective of whether it was the correct reaction for the given protein. This ability is unique to a generative framework, and we wanted to verify whether the model has learned the general chemistry and can generate new reactions. Using all five beams per test entry, the model generated 17,192, 17,516 and 12,779 distinct reactions for the random, MMseqs70 and MMseqs30 splits, of which 12,200, 12,333 and 9,247 were absent from the training data. We then checked how many of them are actual, valid reactions by mapping them to a Rhea identifier (Table 3). Only 68, 57 and 47 matched an actual Rhea reaction, and under 10% of the novel SMILES were chemically balanced. Therefore, many of the predictions are chemically invalid reactions that the model hallucinated. This also suggests that the model is not merely regurgitating reactions from the training set. In future work, chemistry-guided inference is needed to make sure the model generates only valid reactions. This could also be approached by incorporating validity constraints during training.

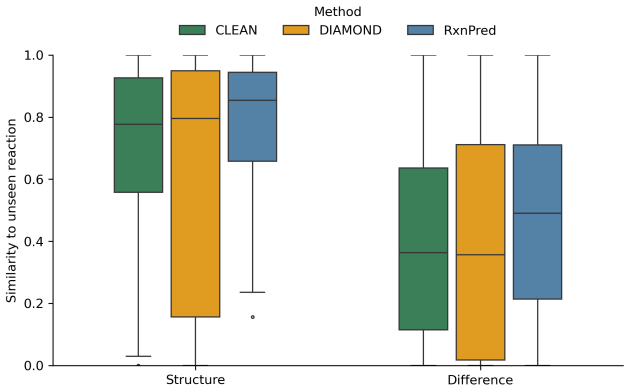


Fig. 4. Similarity distributions of predictions from RxnPred, blastp and CLEAN for 226 entries in the random test split whose reactions were not present in the training set, shown for structure fingerprint (left) and difference fingerprint (right) similarity.

Furthermore, we computed the difference fingerprint similarity between the novel SMILES corresponding to actual Rhea reactions and all SMILES in the training data to assess how different they are from the training distribution. The actual Rhea reactions generated by the model had a mean fingerprint similarity of 0.68 in the random split, 0.74 in the MMseqs70 split and 0.61 in the MMseqs30 split. We also found that one newly generated reaction was correct in each of the MMseqs70 and MMseqs30 splits (Rhea 33023 and 18369); both are similar to reactions present in the training data (Supplementary material Section S2.5). Most of the newly generated reactions were either transport reactions or seen transformations with different metabolites.

Table 3. "Novel SMILES" counts distinct predicted reaction SMILES not present in training, across all beams; the total number of distinct reactions predicted per split is given in the text. "Actual Rhea reactions" counts those matching an actual Rhea entry; "Balanced" gives the count and fraction of novel predictions that are chemically balanced.

Split	Novel SMILES	Actual Rhea reactions	Balanced (n)	Balanced (frac.)
Random	12,200	68	1,157	0.09
MMseqs70	12,333	57	1,199	0.10
MMseqs30	9,247	47	779	0.08

4. Discussion

In our task, there are two alignments to be learned: reaction generation and protein-to-reaction mapping. Our model has learned to generate reactions reasonably well, with a few unseen reactions being generated, but it does not learn the correct mapping to proteins as reliably. The bottleneck, therefore, lies in protein-reaction alignment rather than in reaction generation, and the metabolite-level tokenization does not yield generalization; we regard these as the main findings of this work. In future work, we will focus on improving the protein representation

and on better interaction between the protein and reaction representations, which currently occurs only through cross-attention.

Traditional bioinformatics tools like blastp are highly effective at assigning functions to proteins through homology, as demonstrated in this work as well. However, they operate only at the protein level, not at the reaction level; they are not limited to enzyme functions, can identify similar proteins across any group, and are not specific to biochemistry. Deep learning tools like Horizyn-1 and CLIPZyme, on the other hand, process both protein and reaction information to find similarities. These tools can generalise to new reactions in the sense that, when unseen reactions are screened, they can process them and give relevance scores — but only when the unseen reaction is already known to us. A generative model, in contrast, has the potential to unearth unknown biochemical reaction chemistry. We are not able to achieve this with our current model, which may be due to a lack of diverse data compared with organic chemistry reactions. Despite this, we observe positive signs: new reactions, in the context of the model, are being generated. Although these reactions are not predicted for the right enzyme, this suggests that the identification of new reaction chemistry is not out of reach.

The mapping from proteins to seen reactions is better in our model than in the screening tool Horizyn-1, which in turn has the edge on unseen reactions. Tools like blastp and CLEAN show a superior ability to map proteins to seen reactions compared with our model, but fail on unseen reactions by design. RxnPred is therefore stronger than Horizyn-1 on seen reactions, but has not yet realized the generative advantage on unseen ones. Horizyn-1 can only retrieve reactions already known, whereas our model's limit is the closed metabolite vocabulary, which granular tokenization would remove. With current evidence, blastp and CLEAN remain the better choice for routine annotation, given blastp's higher precision at lower computational cost. RxnPred is preferable where multiple candidate reactions are acceptable, and downstream filtering is available. The advantage gained by including protein structures is marginal, but more experiments are needed. We also found that the addition of an EC number decoder to condition the reaction decoder did not help (Supplementary material Section S2.4).

Finally, we note three limitations of this work. First, for simplicity, all reactions are considered only in the forward direction, which may lead to false metabolic capabilities, especially when the predictions are used to construct metabolic networks. Second, the weighted multilabel metric may inflate the apparent capability of the model by diluting false positives. Although the metrics may appear favourable, a tool that returns the correct function alongside many false positives is less useful in practice than tools like blastp or CLEAN, which achieve better precision. Third, even where the model performs well on one subset of proteins, performance drops on another, which holds for the other tools as well. Developing more specialised models may address this going forward.

Acknowledgments

SB acknowledges funding from the Ministry of Education, Government of India.

References

1. L. Kennedy, M.-E. Harper and M. Cuperlovic-Culf, Computational approaches to enzymatic reaction assignment: a review of methods, validations, and future directions, *Briefings in Bioinformatics* **26**, p. bbaf635 (2025).
2. S. F. Altschul, W. Gish, W. Miller, E. W. Myers and D. J. Lipman, Basic local alignment search tool, *Journal of molecular biology* **215**, 403 (1990).
3. S. R. Eddy, Profile hidden markov models., *Bioinformatics (Oxford, England)* **14**, 755 (1998).
4. T. Yu, H. Cui, J. C. Li, Y. Luo, G. Jiang and H. Zhao, Enzyme function prediction using contrastive learning, *Science* **379**, 1358 (2023).
5. Y. Yang, A. Jerger, S. Feng, Z. Wang, C. Brasfield, M. S. Cheung, J. Zucker and Q. Guan, Improved enzyme functional annotation prediction using contrastive learning with structural inference, *Communications Biology* **7**, p. 1690 (2024).
6. L. Dumontet, S.-R. Han, J. H. Lee, T.-J. Oh and M. Kang, Trustworthy prediction of enzyme commission numbers using a hierarchical interpretable transformer, *Nature Communications* **17**, p. 1146 (2026).
7. P. G. Mikhael, I. Chinn and R. Barzilay, Clipzyme: Reaction-conditioned virtual screening of enzymes, *arXiv preprint arXiv:2402.06748* (2024).
8. J. W. Rocks, D. P. Truong, D. Rappoport, S. Maddrell-Mander, D. A. Martin-Alarcon, T. M. Lee, S. Crossan and J. E. Goldford, Dual-encoder contrastive learning accelerates enzyme discovery, *Proceedings of the National Academy of Sciences* **123**, p. e2520070123 (2026).
9. C. Hua, B. Zhong, S. Luan, L. Hong, G. Wolf, D. Precup and S. Zheng, Reactzyme: A benchmark for enzyme-reaction prediction, *Advances in Neural Information Processing Systems* **37**, 26415 (2024).
10. P. Schwaller, T. Laino, T. Gaudin, P. Bolgar, C. A. Hunter, C. Bekas and A. A. Lee, Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction, *ACS central science* **5**, p. 1572 (2019).
11. R. Irwin, S. Dimitriadis, J. He and E. J. Bjerrum, Chemformer: a pre-trained transformer for computational chemistry, *Machine Learning: Science and Technology* **3**, p. 015022 (2022).
12. T. Sagawa and R. Kojima, Reactiont5: a pre-trained transformer model for accurate chemical reaction prediction with limited data (2025).
13. P. Bansal, A. Morgat, K. B. Axelsen, V. Muthukrishnan, E. Coudert, L. Aimo, N. Hyka-Nouspikel, E. Gasteiger, A. Kerhornou, T. B. Neto *et al.*, Rhea, the reaction knowledgebase in 2022, *Nucleic acids research* **50**, D693 (2022).
14. Uniprot: the universal protein knowledgebase in 2023, *Nucleic acids research* **51**, D523 (2023).
15. G. Landrum *et al.*, RDKit: Open-source cheminformatics `https://www.rdkit.org`, (2026), Version 2026.03.3, `https://doi.org/10.5281/zenodo.20446949`.
16. M. Steinegger and J. Söding, Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets, *Nature biotechnology* **35**, 1026 (2017).
17. D. Bertoni, M. Tsenkov, P. Magana, S. Nair, I. Pidruchna, M. Querino Lima Afonso, A. Midlik, U. Paramval, D. Lawal, A. Tanweer *et al.*, Alphafold protein structure database 2025: a redesigned interface and updated structural coverage, *Nucleic Acids Research* **54**, D358 (2026).
18. Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli *et al.*, Evolutionary-scale prediction of atomic-level protein structure with a language model, *Science* **379**, 1123 (2023).
19. C. Hsu, R. Verkuil, J. Liu, Z. Lin, B. Hie, T. Sercu, A. Lerer and A. Rives, Learning inverse folding from millions of predicted structures, in *International conference on machine learning*, 2022.
20. W. Ahmad, E. Simon, S. Chithrananda, G. Grand and B. Ramsundar, Chemberta-2: Towards

chemical foundation models, *arXiv preprint arXiv:2209.01712* (2022).
21. M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov and L. Zettlemoyer, Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020.
22. T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, Transformers: State-of-the-art natural language processing, in *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 2020.
23. J. Ansel, E. Yang, H. He, N. Gimelshein, A. Jain, M. Voznesensky, B. Bao, P. Bell, D. Berard, E. Burovski *et al.*, Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation, in *Proceedings of the 29th ACM international conference on architectural support for programming languages and operating systems, volume 2*, 2024.
24. M. Freitag and Y. Al-Onaizan, Beam search strategies for neural machine translation, in *Proceedings of the First Workshop on Neural Machine Translation*, 2017.
25. T. Sanderson, M. L. Bileschi, D. Belanger and L. J. Colwell, Proteinfer, deep neural networks for protein functional inference, *elife* **12**, p. e80942 (2023).
26. M. N. Price, K. M. Wetmore, R. J. Waters, M. Callaghan, J. Ray, H. Liu, J. V. Kuehl, R. A. Melnyk, J. S. Lamson, Y. Suh *et al.*, Mutant phenotypes for thousands of bacterial genes of unknown function, *Nature* **557**, 503 (2018).
27. Z. Yang, B. Su, J. Chen and J.-R. Wen, Interpretable enzyme function prediction via residue-level detection, *arXiv preprint arXiv:2501.05644* (2025).
28. B. Buchfink, K. Reuter and H.-G. Drost, Sensitive protein alignments at tree-of-life scale using diamond, *Nature methods* **18**, 366 (2021).
29. G. B. Kim, J. Y. Kim, J. A. Lee, C. J. Norsigian, B. O. Palsson and S. Y. Lee, Functional annotation of enzyme-encoding genes using deep learning with transformer layers, *Nature Communications* **14**, p. 7370 (2023).