Digital Discovery

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue

Check for updates

Cite this: *Digital Discovery*, 2025, **4**, 3588

Open Access Article. Published on 30 October 2025. Downloaded on 2/8/2026 10:48:01 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

Leveraging large language models for enzymatic reaction prediction and characterization

Lorenzo Di Fruscia[a] and Jana M. Weber*[b]

Received 8th May 2025
Accepted 21st September 2025

DOI: 10.1039/d5dd00187k

rsc.li/digitaldiscovery

Predicting enzymatic reactions is crucial for applications in biocatalysis, metabolic engineering, and drug discovery, yet it remains a complex and resource-intensive task. Large Language Models (LLMs) have recently demonstrated remarkable success in various scientific domains, *e.g.*, through their ability to generalize knowledge, reason over complex structures, and leverage in-context learning strategies. In this study, we systematically evaluate the capability of LLMs, particularly the Llama-3.1 family (8B and 70B), across three core biochemical tasks: enzyme commission number prediction, forward synthesis, and retrosynthesis. We compare single-task and multitask learning strategies, employing parameter-efficient fine-tuning *via* LoRA adapters. Additionally, we assess performance across different data regimes to explore their adaptability in low-data settings. Our results demonstrate that fine-tuned LLMs capture biochemical knowledge, with multitask learning enhancing forward- and retrosynthesis predictions by leveraging shared enzymatic information. We also identify key limitations, for example challenges in hierarchical EC classification schemes, highlighting areas for further improvement in LLM-driven biochemical modeling.

1 Introduction

Biochemistry plays a fundamental role in nearly every aspect of daily life, from medicine development to food production, from the creation of fuels to personal care items, significantly contributing to improved quality of life. Developing novel biocatalysts and discovering and optimizing biochemical reactions hold immense promise for addressing global challenges. However, these discoveries are inherently complex, requiring a deep understanding of enzyme-substrate relationships, and they remain experimentally expensive and time-intensive.[1–3]

To mitigate these experimental costs, computational approaches date back many decades: early Computer-Aided Synthesis Planning (CASP) began with hand-crafted expert systems such as LHASA[4] and SECS,[5] that encoded chemistry libraries of reaction transformation rules to propose synthetic routes. These rules were applied to produce templates, specific atom-mapping patterns describing the molecualr transformations. These approaches evolved into more recent and extensive methods such as SYNTHIA,[6] ICSYNTH[7] and RetroSim[8] while still being primarily template-based. However, templates require manual rule curation and heuristic atom-mapping, which limits their scalability as reaction databases grew. To overcome these limitations, the scientific community started developing template-free methods. Data-driven at their core, these approaches learn reaction patterns directly from molecular strings or graphs with limited to no reliance on crafted rules. Early work applied motif- and profile-based statistical learners such as PRIAM[9] for enzyme detection and classification, and EFICAz[10] for enzyme function inference. These statistical methods have since been surpassed by deep architectures such as the CNN-based DEEPre[11] and DeepEC[12] for enzyme function prediction, and RNN-based ones such as the sequence-to-sequence model from ref. 13 for retrosynthetic reaction prediction.

More recently transformers,[14] architectures suited for applications like language translation, sentiment analysis and text completion, have proven to be effective for tasks such as chemical reaction product prediction with the molecular transformer[15] and for molecule optimization overall.[16,17] In biochemistry, several ML models have been tailored for prediction tasks, including approaches where enzymes are represented using natural language (*e.g.* the enzymatic transformer[18]), numerical classification schemes,[19,20] or amino acid sequences.[21] Recent work has also shown that transformer models trained on protein sequences only (also known as Protein Language Models or PLMs) can be used on downstream tasks such as predicting EC numbers from the amino-acid sequence,[22,23] usually obtained by adapting large protein encoder-only models, *e.g.* ProteinBERT[24] and ESM-2.[25] While these specialized models deliver impressive results, they are typically constrained to specific tasks and require extensive domain-specific data and expertise for their development and for the incorporation of biochemical knowledge.

[a] *Department of Intelligent Systems Delft University of Technology, Delft 2629 HZ, The Netherlands. E-mail: l.difruscia@tudelft.nl*

[b] *Department of Intelligent Systems Delft University of Technology, Delft 2629 HZ, The Netherlands. E-mail: j.m.weber@tudelft.nl*

© 2025 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 30 October 2025. Downloaded on 2/8/2026 10:48:01 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

We are now witnessing the emergence of foundation models like Large Language Models (LLMs),[26,27] that have found their application in chemistry as well.[28] These transformer-based architectures consist of up to hundreds of billions of parameters and are trained on text corpora comprising trillions of tokens. Despite being trained for next token prediction, these models have shown emergent abilities that were not foreseeable for smaller sized models:[29] they are capable of more than just completing phrases in natural language, as to some extent they are able to answer questions, understand examples and reason over problems. Foundation models can be capable of solving multiple tasks at once. Building on top of existing LLMs is straightforward to implement and they require relatively little expertise to use, circumventing the need to train a multitude of specialized models. LLMs excel in low-data regimes, adapting on the fly from context and examples such as in real-world lab scenarios. They can ground their outputs *via* Retrieval Augmented Generation (RAG),[30] or knowledge-graph databases access,[31] support agentic behaviour through external tool integration,[32] and can expose their step-by-step reasoning to guide experimental workflows.[33,34]

Efforts to integrate LLMs into chemistry generally fall into two distinct categories. The first focuses on building chemistry agents that leverage the LLMs planning abilities to work with task-specific tools and improve reasoning.[35] For instance, in Bran *et al.*,[36] researchers augmented LLMs by providing access to expert-designed tools for drug discovery, materials design and organic synthesis. The second category involves using LLMs directly for downstream tasks such as property prediction, reagent selection and molecule captioning.[37–40] In Guo *et al.*,[37] they benchmarked LLMs in zero- and few-shot settings, demonstrating their capabilities in explaining, understanding and reasoning over chemistry. In Jablonka *et al.*,[38] they show how by fine-tuning GPT family models from OpenAI,[41] they easily adapt them to solve various tasks involving classification, regression, inverse design of chemicals, and many more. Their model proved to be useful especially in the low-data regime, where the LLM performed at least as good as the conventional ML models. Additionally, comprehensive instruction datasets for the chemical and biochemical domains have been introduced.[42,43] These datasets, encompassing millions of examples across applications like molecule generation, name conversion and reaction prediction, enable small fine-tuned LLMs to surpass prompted SOTA LLMs, demonstrating the role of high quality datasets in enhancing performance in molecular domains.

While previous studies mainly focused on the investigation of LLMs for chemical and materials tasks, we are interested in understanding LLMs potential for biochemical reaction characterization, discovery, and optimization. Specifically, we are interested in whether a single general-purpose LLM can be adapted on multiple tasks with one interface, with little engineering and limited labeled data. With such a model, scientists would be able to query multiple aspects possibly under changing conditions in natural language. This study investigates to which extent one LLM is sufficient, and in which aspects the connection to specialized models is still required.

The scientific community is building upon recent discoveries that scaling up LLMs in size and training data leads to promising zero- and few-shot capabilities for in-context learning.[44] One key problem of learning from context is the high variance in the outputs returned by the model: slight changes in prompts can greatly affect the model performance, ranging from barely above chance, to near state-of-the-art (SOTA) level.[45] Additionally, LLMs may produce made-up or irrelevant content, a phenomenon known as hallucinations. To address these instabilities, research has explored advanced prompting strategies such as Chain-of-Thought (CoT), a technique that guides the model to break down answers as a series of connected thoughts. By explicitly decomposing complex problems into step-by-step reasoning, CoT reduces output variability and enhances accuracy, particularly for tasks requiring logical progression or multi-step calculations. By acting in a way that mimics human reasoning, CoT showed to improve the reliability of responses and to therewith make LLMs more robust.[46]

Another key task adaptation strategy is fine-tuning, which modifies the weights of the pretrained model. It offers the advantage of not being constrained by a limited context window for input data, but it typically results in a model specialized for a single task. However, prior research[47] showed that fine-tuning outperforms in-context learning strategies in both in-domain and out-of-distribution tasks for models of comparable size, with performance gains increasing as more training data becomes available. Fine-tuning limitations in principle include the need for significant training expertise and computational resources, with a reduced reusability compared to in-context learning strategies. These shortcomings are partially mitigated by Parameter-Efficient Fine-Tuning (PEFT).[48,49] PEFT techniques selectively adjust only a small portion of parameters, leaving the rest unchanged. This approach preserves the base model general-purpose capabilities while adding task-specific expertise in a modular way, enabling greater adaptability to new tasks.

We focus on enzymatic reactions represented using SMILES (Simplified Molecular Input Line Entry System) notation[50] for chemicals and EC numbers for enzyme classification. Specifically, we design tasks that test the model's ability to predict EC numbers, reaction products (forward synthesis), and substrates (retrosynthesis). By introducing a multitask learning setup, we investigate whether training on multiple tasks simultaneously makes use of shared biochemical knowledge compared to single-task fine-tuning. Finally, we perform ablation studies to examine the impact of several data regimes and fine-tuning setups on different models' performance.

2 Methods

2.1 Tasks and dataset description

2.1.1 Task selection. We assemble a representative set of biochemical prediction tasks. The selected tasks are designed to evaluate the capabilities of Large Language Models (LLMs) in understanding and predicting enzymatic reactions, when the chemicals are presented in string format and the enzyme in the EC numerical classification scheme. Specifically:

- EC number prediction: we assess whether LLMs can accurately assign EC numbers given the substrates and the products of each reaction.

Open Access Article. Published on 30 October 2025. Downloaded on 2/8/2026 10:48:01 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

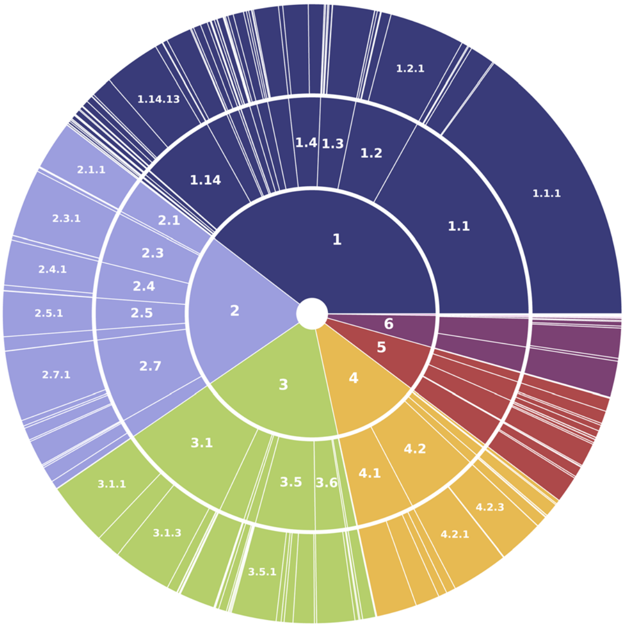


Fig. 1 Distributions of samples across EC levels for the BRENDA dataset. The innermost layer represents the main class (EC1 digit), and the middle and outer layers represent levels EC2 and EC3 respectively. The label for enzyme class 7 (translocases) is not visible due to the limited data available (<20 samples).

• Product prediction: here we explore the model's ability to predict reaction products given substrates and the EC number associated to the reaction (forward synthesis).

• Substrate prediction: we test the model's capabilities of predicting substrates based on reaction products and the EC number (retrosynthesis).

Given the inherent similarities among the three tasks, we investigate whether the model can improve its performance when trained on all tasks simultaneously, by leveraging shared information in a synergistic manner. To test this, we introduce a multitask (MT) setup, in which a single model is trained concurrently on all three tasks, inspired by what has been done in Yu *et al.*[43] This setup allows us to evaluate whether a multitask-trained model can outperform individually fine-tuned models for each task (single-task, ST) producing a more general model eventually capable of handling diverse biochemistry tasks involving enzymes. The following sections explain data selection and the data split suitable for both ST and MT experiments. To ensure that the selected tasks are supported by high-quality data, we preprocess the data to minimize biases and data leakage.

2.1.2 Dataset preparation. We make use of the ECREACT dataset curated by Probst *et al.*[19] This dataset results from the combination of data coming from four different databases: MetaNetX, Rhea, PathBank and BRENDA.[51–54] The authors screened the enzymatic reactions, and determined the corresponding Enzyme Commission (EC) number for each of them. Further processing simplified and generalized the dataset. They removed products also occurring as reactants in the same reaction, co-enzymes, common by-products, and reactions without reactants or multiple or missing products. In each reaction, substrates and products are represented in SMILES, whereas EC numbers are tags for the reaction in the form of a 4-digit tag '*X.X.X.X*'. The digits follow a hierarchy, with the first digit (EC1) representing

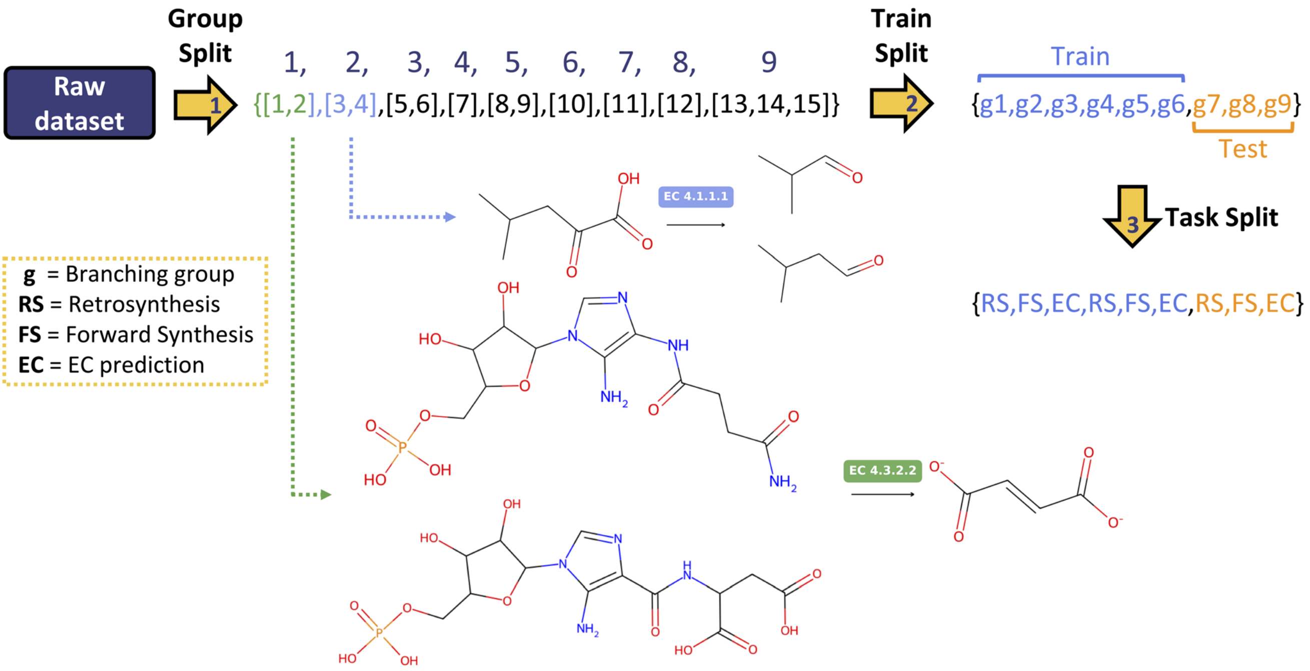


Fig. 2 Data preprocessing pipeline. Individual reactions sharing the same {product, EC} or {substrate, EC} pair are grouped together (here groups are numbered from 1 to 9, first row). The dataset is split into training and test set, while keeping each group intact. Within training and test, each group is assigned to one of the three tasks on a rotating basis to balance the splits. Groups are randomly shuffled at the beginning of the procedure, here we keep indices in order for visual clarity.

Open Access Article. Published on 30 October 2025. Downloaded on 2/8/2026 10:48:01 AM. This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

the main class of the enzymatic reaction. From these databases, we only focus on BRENDA, for a total of $n = 8496$ enzyme-catalyzed reactions covering all seven different EC classes. This is mainly due to computational constraints, as fine-tuning large-scale LLMs on the full dataset without parallelized infrastructure would require several weeks. The distribution of reactions according to their respective EC numbers is shown in Fig. 1. We include all four EC digits (thus up to EC4) in the dataset, but our subsequent analyses will focus on up to sublevel EC3, as many subcategories for EC4 consist of only a single enzyme-substrate example. Class 7 will not be included as well due to the limited sample size for the class (<20 samples).

2.1.3 Data splitting. We implement several preprocessing steps: canonicalization of SMILES representations using the RDKit library parsing functions to remove redundant entries, grouping reactions that share the same {product, EC} or {substrate, EC} pair, but differ in the remaining molecule, and avoidance of task-specific leakage, ensuring that if *e.g.* a reaction appears in forward synthesis, it must not appear in retrosynthesis as well. More details about these steps are reported in Appendix Section A.1.

These points imply that to maintain dataset integrity, each above-mentioned reaction group is assigned exclusively to one task and one dataset split (either training or test). By addressing these issues preemptively, we also ensure a consistent random dataset split for both single-task and multitask setups, enabling fair comparisons between the two methodologies. Fig. 2 better illustrates this approach.

We perform a 70–30 train–test split, ensuring that the fraction of groups assigned to each of the two sets maintains a balanced ratio. Of the train set, 10% is used for validation. The test sets remain constant across all training regimes and varying

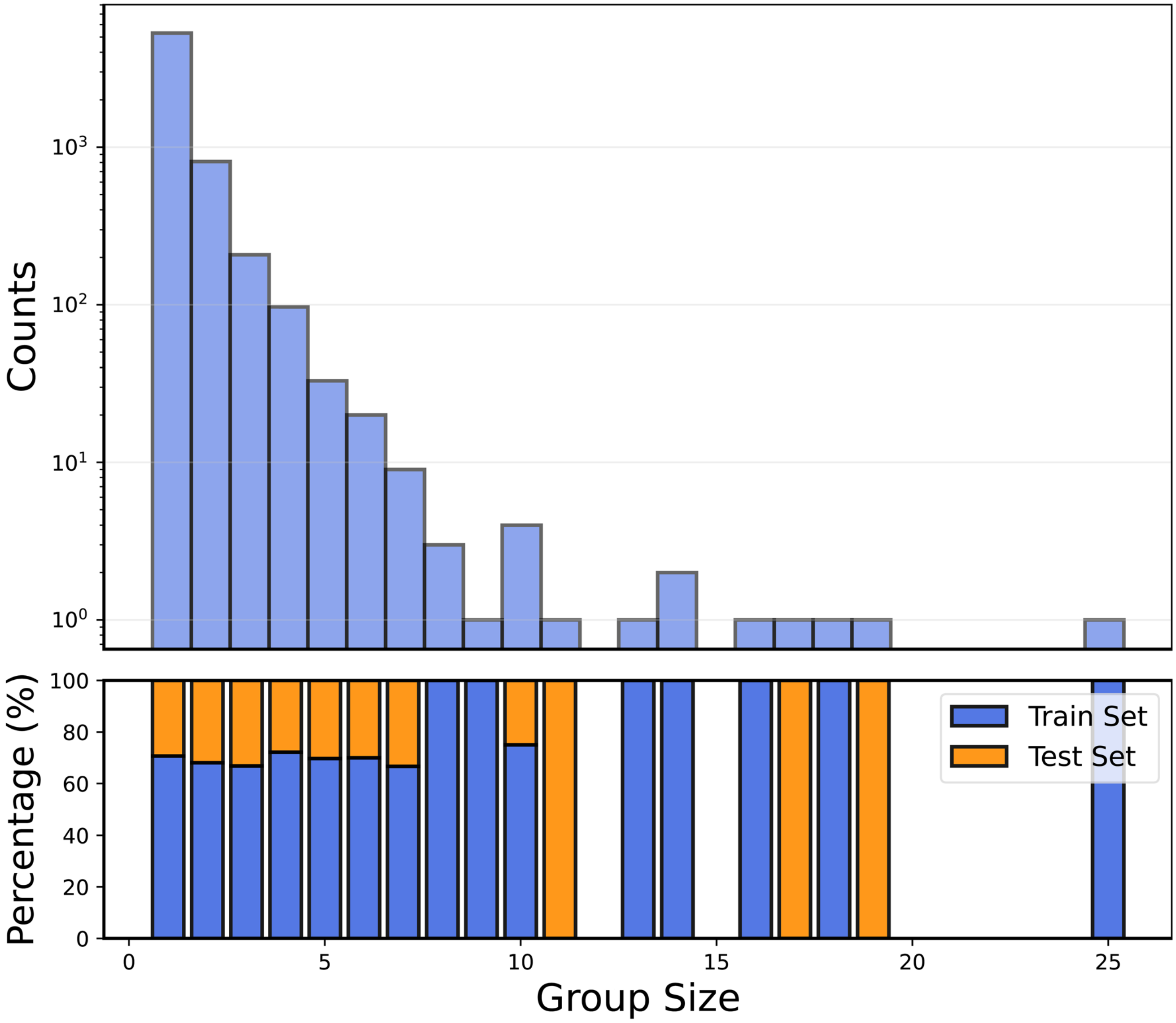


Fig. 3 Distribution of reaction groups with repeating substrates and/or products. Unique reactions are included as elements with group size equal to 1. Group sizes with a number of counts >10 closely follow the required 70–30 split ratio between train and test set.

Open Access Article. Published on 30 October 2025. Downloaded on 2/8/2026 10:48:01 AM. This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

training set sizes. EC prediction, forward synthesis and retrosynthesis tasks have fixed test sets of size 855, 857, and 837 examples respectively. The above-mentioned preprocessing steps prevent basic information leakage that could artificially inflate performance metrics, yet our similarity analysis shows that substantial analogue overlap persists, that we report in Appendix Fig. 12. It is also worth noting that by being public it is likely that the LLM may have had access to this data during its extensive pretraining. Fig. 3 illustrates the final distribution of groups across the train and test sets.

2.1.4 Instruction creation. We manually craft question–answer textual pairs by converting reaction SMILES strings and EC labels into clear natural language queries. Each reaction data point is converted into a single question-answer pair, but we generate multiple templates per task to avoid overfitting on a single prompt style. For EC number prediction, example templates include:

- `Can you tell me the potential EC number of an enzymatic reaction that has <SMILES> substrate»product </SMILES> as the substrate and product?`
- `Using <SMILES> substrate » product </SMILES> as the substrate and product, tell me the probable EC number.`
- `Predict a valid Enzyme Commission (EC) number from the listed reactant and product:<SMILES> substrate » product </SMILES>.`

The model is instructed to return only the answer (the EC number in this case), without additional text. We vary synonyms *e.g.* “EC number” *vs.* “Enzyme commission number”, and qualifiers such as “feasible” *vs.* “probable” across up to 14 unique templates per task.

2.2 LLM interaction and adaptation

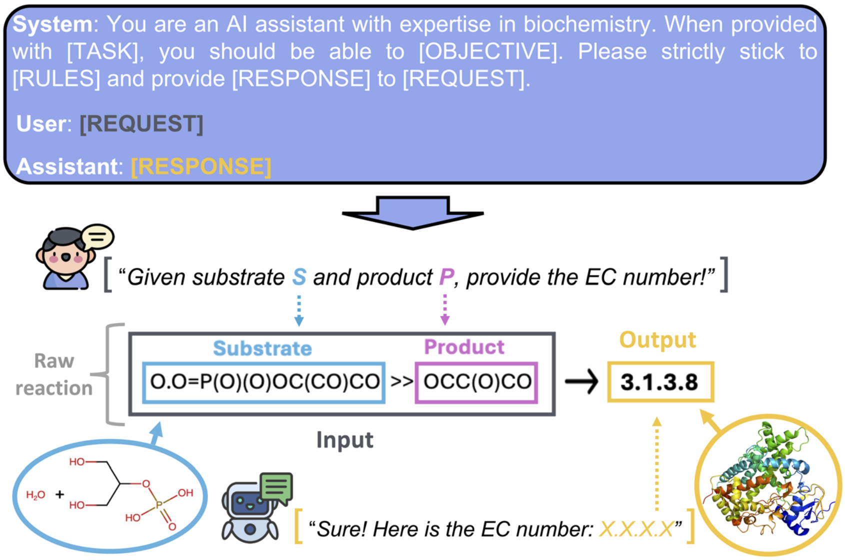
2.2.1 In-context learning. In In-Context Learning (ICL), we interact with a LLM solely through prompting. Prompting means giving a set of instructions to the model in natural language in order to make it perform a task: answering, reasoning, story-writing, conversation, tool-access and so on. With ICL, the model is not retrained, so no parameters are changed. Instead, the model uses its existing knowledge to generalize “on the fly” within that single interaction, which we refer to as zero-shot. LLMs are powerful zero-shot learners and can easily adapt to examples to improve their understanding, which is called few-shot prompting.[44] This approach is flexible and immediate, but its performance can vary significantly with prompt phrasing, the context provided and even the order of words.

When interacting with the LLM, each data point is formatted as a conversation between a user and an assistant (the model itself), as follows:

- A general system prompt assigns the model the role of a biochemically knowledgeable assistant.
- The user prompt specifies the task, phrasing it in a flexible way to ensure a certain degree of variability. As mentioned above, diverse templates are used in order to prevent overfitting on specific question structures.
- The assistant provides a direct answer, formatted with tagging elements such as ⟨EC⟩ … ⟨/EC⟩ (for the EC number prediction task), to enhance consistency and ease of parsing.

A visual example of this is shown in Fig. 4.

2.2.2 Fine-tuning. Fine-tuning refers to the process of adapting a pretrained model to perform specific tasks by updating its parameters on a new dataset. This approach allows the model to specialize in a narrower domain while retaining its general pretrained knowledge. As models sizes grow into tens or hundreds of billions of parameters, retraining every weight for each new task becomes prohibitively expensive in memory and compute. To fine-tune our models efficiently, we use Parameter-Efficient Fine-Tuning (PEFT) techniques that leave most of the base model weights unchanged in the process. Specifically, we



System: You are an AI assistant with expertise in biochemistry. When provided with [TASK], you should be able to [OBJECTIVE]. Please strictly stick to [RULES] and provide [RESPONSE] to [REQUEST].

User: [REQUEST]

Assistant: [RESPONSE]

[“Given substrate S and product P, provide the EC number!”]
Raw reaction
Substrate
Product
O.O=P(O)(O)OC(CO)CO >> OCC(O)CO
Input
Output
3.1.3.8
H₂O
HO
HO
OH
HO
O
P
[“Sure! Here is the EC number: X.X.X.X”]

Fig. 4 Example of a zero-shot prompt for the EC number prediction task. The model first receives the system prompt: a general prompt with instructions that inform it about the task to perform. The [TASK] here is EC number prediction, and the [OBJECTIVE] is to assign the 4 digits of the EC number given only reactants and product in SMILES notation. After that, the model receives the reaction SMILES from the user as a [REQUEST], and the model associates an EC number to it as the [RESPONSE].

Open Access Article. Published on 30 October 2025. Downloaded on 2/8/2026 10:48:01 AM. This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

use Low-Rank Adaptation (LoRA).[55] LoRA allows fine-tuning by updating only a small subset of the model's parameters, significantly reducing computational demands. Instead of directly updating a weight matrix of the pretrained model $W \in \mathbb{R}^{n \times m}$, LoRA models the update as the product $\Delta W = AB$, where $W \in \mathbb{R}^{n \times m}$ and $B \in \mathbb{R}^{r \times m}$ are matrices with a rank $r \ll \min(n, m)$. The rank determines the size of the two matrices, and during forward passes the effective weight matrix becomes

$$W' = W + \Delta W = W + AB \tag{1}$$

The small rank is what ensures that A and B contain far fewer parameters than W and this drastically reduces memory footprint and fine-tuning time. An illustration of the algorithm is shown in Fig. 5. While the model can be loaded in a quantized format for efficient memory usage, fine-tuning occurs on a limited percentage of parameters that are stored in full/half precision. This approach has shown to yield performance levels close to those of full model fine-tuning, while maintaining the model's general reasoning abilities and core capabilities.[55,56] Importantly, during fine-tuning, the model is provided with interrogation pairs in exactly the same way as in In-Context Learning, using the same prompt template and tagging conventions. Only now each question-answer pair is used to adapt its internal parameters.

The use of LoRA adapters is particularly advantageous for LLMs like ours. These adapters can be "plugged in" for domain-specific tasks and subsequently removed to revert to the base model, which remains unaffected by fine-tuning, thereby keeping computational costs under control.

2.2.2.1 Model selection. In selecting a model for our biochemical prediction tasks, our primary selection criteria are:

- Power: the model's ability to handle complex tasks and achieve high accuracy;
- Flexibility: its ability to tackle diverse tasks both in in-context learning and fine-tuning settings;
- Efficiency: the model's computational cost-effectiveness, particularly in resource-constrained environments.

We aimed to use a LLM that balances computational power with flexibility, ensuring it can be customized for specialized biochemical applications. We prioritize general-purpose LLMs to evaluate their adaptability and scalability across multiple biochemical tasks. Equally important was choosing an open-source model to facilitate accessibility and enable further development by other researchers. Given these requirements, we selected models from Meta AI's Llama 3.1 family,[57] specifically the 8B and 70B parameter versions. The smaller 8B model offers a trade-off between efficiency and flexibility for exploratory or lower-resource settings, while the 70B model provides greater power. Further, we employed the instruct versions of these models, both for in-context learning and fine-tuning. These variants are fine-tuned on instruction-response pairs, helping them generate responses that align with the given instructions. Lastly, we utilize both base models in the 4-bit quantized format to reduce computational costs and inference time.

2.3 Evaluation metrics

For the EC prediction task, a prediction is correct if the digits match exactly those of the ground truth. If only the first digit is correct, the model correctly predicted the EC class. If the first two digits are correct, the prediction is correct up to digit EC2, and so on. For the accuracy, we always compute the macro-average to show performance across classes, treating each class as equally important. Additionally, for the main class we report F1 score, precision and recall, as they help provide a more complete picture especially with imbalanced datasets such as ours. To evaluate product and substrate predictions, we categorize predicted SMILES strings into five distinct groups:

- Invalid (I): if the RDKit parsing fails, the prediction is not a valid SMILES string, either because chemically implausible or incorrectly formatted. If the parsing succeeds, the pipeline proceeds to the next steps;

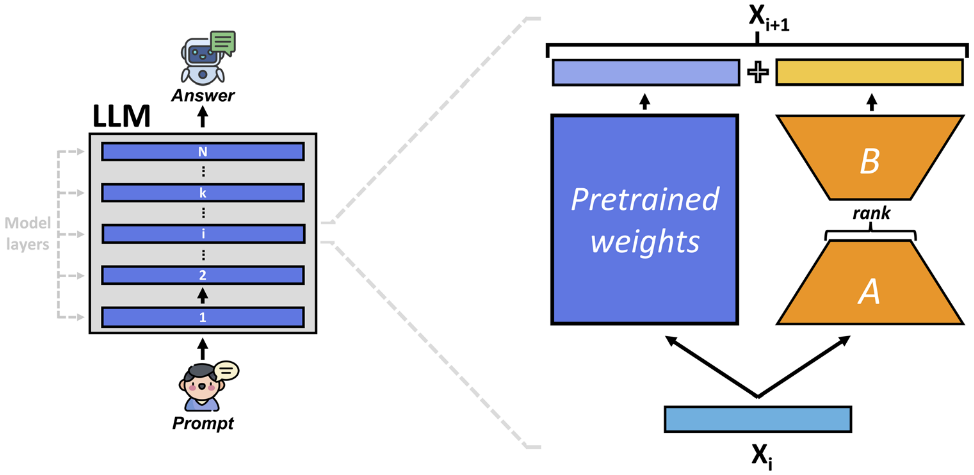


Fig. 5 Illustration of LoRA framework. The input vector x_i at layer i is passed through both the frozen i-th weight matrix of the pretrained model, and the i-th LoRA head. After both blocks process the input, the two representations are summed together to obtain a new representation x_{i+1}. This procedure is repeated for all layers.

Open Access Article. Published on 30 October 2025. Downloaded on 2/8/2026 10:48:01 AM. This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

- Non-canonical valid (NCV): the predicted SMILES does not correspond to the ground truth molecule, but it represents a plausible molecule in a non-canonical SMILES representation;
- Canonical valid (CV): the predicted SMILES does not correspond to the ground truth molecule, but it represents a plausible molecule in the canonical SMILES representation;
- Non-canonical match (NCM): the predicted SMILES does represent the ground truth molecule, but in a non-canonical SMILES format;
- Canonical match (CM): the predicted SMILES string does represent the ground truth molecule in the canonical format: the predicted string exactly matches the ground truth one.

For valid chemicals, we additionally examine molecular similarity to determine the potential relevance of the generated SMILES, using the Tanimoto similarity coefficient after computing daylight fingerprints[58] for each molecule. A Tanimoto similarity >0.85 is often considered indicative of structurally similar molecules, suggesting that even incorrect predictions may still be chemically meaningful. High similarity scores could for example suggest that the LLM-generated molecule might serve as an alternative substrate in retrosynthetic applications, potentially offering novel biochemical insights. It is important to note that the SOTA results that we mention are taken from existing studies and are based on models trained on the entire ECREACT dataset, which comprises unique $n = 62\,222$ enzymatic reactions aggregated from four different databases. In contrast, our experiments are conducted using only the reactions from the BRENDA database. While this difference in training data size limits direct comparisons with SOTA models, our setup allows for a holistic experimental design within reasonable computational limits. While this limits the comparison to certain extent, it allows us to focus on a single well-curated database, we can systematically evaluate different model sizes, fine-tuning strategies, and data regimes, while still capturing a diverse range of enzymatic reactions. All results are averaged over $N = 3$ experiments to provide robust performance metrics, with standard deviations reported where applicable.

2.4 Fine-tuning setup

All models are trained with a learning rate $\mathrm{lr} = 0.002$ using a linear decay scheduler, and $\{\alpha = 32, r = 16\}$ for the LoRA adapter. Minimal hyperparameter tuning was performed ($\mathrm{lr} \in [0.0005, 0.005]$, $\alpha \in \{32, 64\}$, $r \in \{16, 32\}$). We explore two new LoRA setups in addition to the default one, to evaluate the trade-off between fine-tuning parameter count and model performance. We adopted the same setups choice by ref. 43, to ensure consistency with prior LoRA fine-tuning literature in the chemical domain. Here we list them, including in parenthesis the number of trainable parameters and their percentage with respect to the pretrained, base model:

- LoRA light (6.8 M, 0.09% for the 8B, 32.8 M, 0.05% for the 70B): we only fine-tune the query and key matrices within the attention modules [q_{proj}, k_{proj}].
- LoRA attention (13.6 M, 0.17% for the 8B, 65.5 M, 0.09% for the 70B): we extend fine-tuning to all matrices within the attention mechanism [q_{proj}, k_{proj}, v_{proj}, o_{proj}].
- LoRA (41.9 M, 0.52% for the 8B, 207 M, 0.29% for the 70B): this is the basic setting and the one used throughout the paper. The adapter tunes all the attention modules and the feed-forward networks (FFN).

For training we use a single NVIDIA Tesla A100 80 GB GPU. To isolate the effect of model size, we report single-task training durations: the 8B model takes ∼30 min per epoch, while the 70B model takes ∼4 h per epoch with a batch size of 8. Inference latency is ∼2.5 s per sample for the 8B model and ∼5.7 s per sample for the 70B model on a Intel XEON E5-6448Y 32C 2.1 GHz CPU.

3 Results and discussion

In this section, we present the results of fine-tuning the selected Llama models. The analysis encompasses ST and MT setups, along with experiments designed to evaluate performance in low-data regimes and across different fine-tuning schemes. For each task, the performance is compared against baselines.

3.1 Single-task fine-tuning

Llama-3.1 models accurately predict the highest level of EC number classification, yet show a decline when tasked with the second and third digit. In a single-task settup, the Llama-3.1 model family exhibits some difficulties with exact product and substrate prediction tasks. Interestingly, we find that reasonably large percentages of uncorrect predictions show a high Tanimoto similarity with the correct predictions, which can potentially still be useful in biochemical workflows.

3.1.1 EC prediction task. The 70B model accuracy for EC class prediction is consistent across most classes, with an average accuracy of 91.7%. This indicates that it is fairly simple for the fine-tuned model to correctly assign the highest EC number given any reactant, product pair as request. However, class 4 exhibits a noticeable performance dip, despite not being the least-represented class in the dataset. To explore the model's misclassification patterns, we present the confusion matrix for EC class prediction in Fig. 6. The matrix reveals that classes 4 and 5 are sometimes wrongly assigned to each other. In classes 1, 2 and 3 rare instances of misclassifications either happen between 1 and 2 or assign the reactions to class 4.

For EC2 predictions, we see that the model frequently misclassifies subclasses within the same main class. This relates to the EC2 category distribution per main class. For instance, class 1.*X*.*X*.*X* has numerous subclasses, whereas classes 5.*X*.*X*.*X* and 6.*X*.*X*.*X* only have a few. Rare subclasses, such as 2.2.*X*.*X* or 4.99.*X*.*X*, show clear exceptions with the model misclassifying outside the main class, likely due to their underrepresentation. Additionally, structural similarities within main classes may further contribute to confusion, independent of dataset imbalance. The confusion matrix for the EC2 level, alongside the test set distribution for that depth, is shown in Fig. 7.

The accuracy of the model declines at deeper EC levels, reflecting the increasing challenge of capturing hierarchical enzyme relationships. These difficulties also stem from

© 2025 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 30 October 2025. Downloaded on 2/8/2026 10:48:01 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

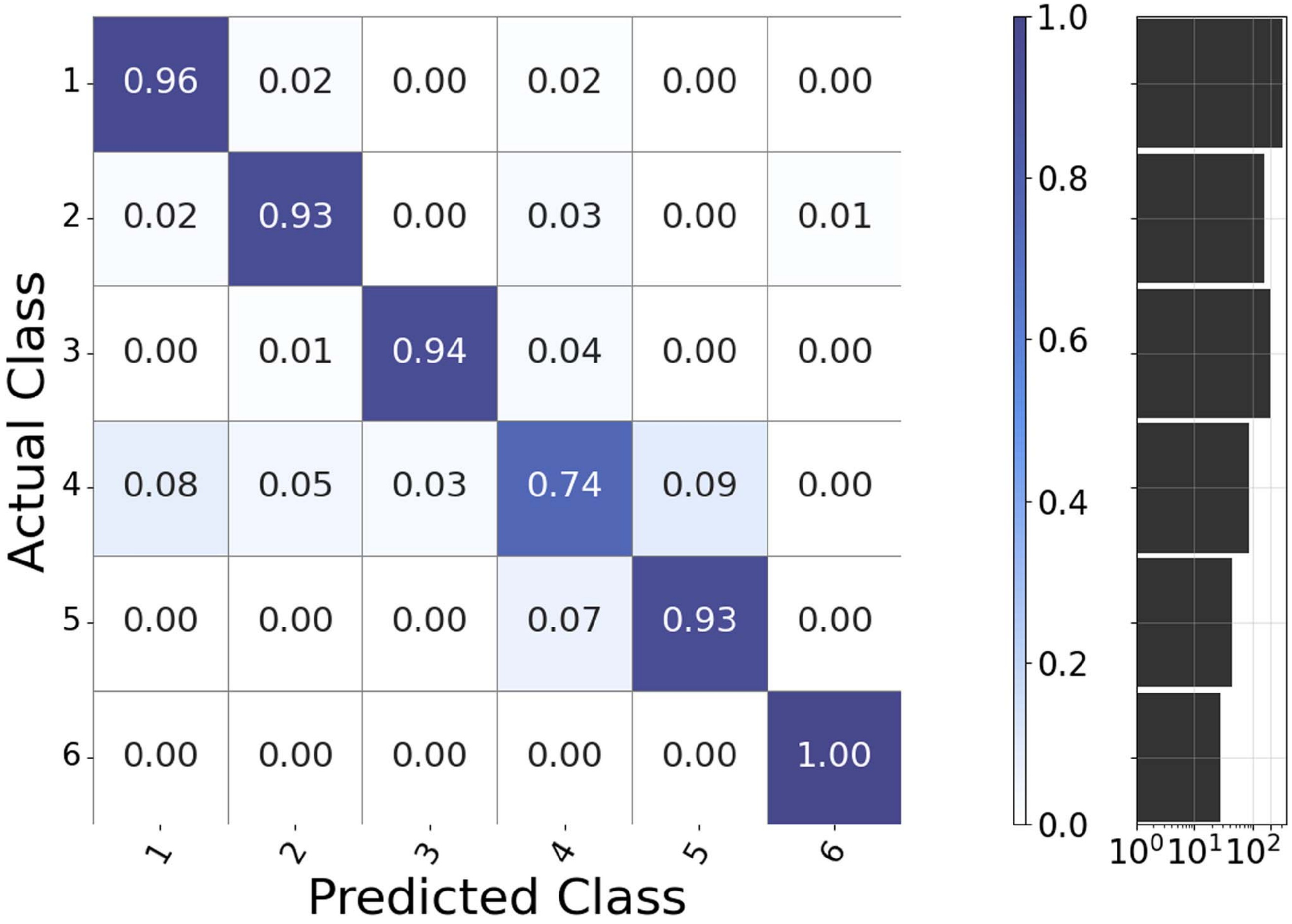


Fig. 6 Confusion matrix representing Llama-3.1 70B accuracy in predicting the enzyme class given reactants and substrates, for one experiment. The out-of-diagonal elements show how examples are misclassified. The histogram on the right shows the test set distribution stratified by main class, roughly following how training data is distributed.

increased combinatorial complexity of sublevels and class imbalance. In fact, at level EC2 the model performs best for class 6 and worst for class 1, a result that aligns with the dataset distribution shown in Fig. 1: EC class 1 has a highly branched EC2 structure, with 1.1.*X*.*X* accounting for almost half of the samples, introducing class imbalance. Conversely, class 6 has a limited number of balanced subcategories (6.2.*X*.*X* and 6.3.*X*.*X*), simplifying subclass predictions. Fig. 8 illustrates the model's performance in predicting EC numbers up to level EC3, stratified by main class.

The fine-tuned 70B model comes on top of the fine-tuned 8B model predicting EC digits at any depth. However, the compared SOTA retains a significant edge across all levels (EC1 accuracy: 96.2%, EC2 accuracy: 93.4.6%, EC3 accuracy: 91.6%).[20] Please note that the authors have performed a micro-average, while we perform a macro-average that takes class imbalance into account. Macro-averaged accuracy at any EC level depth is computed by first calculating the accuracy within each EC number class, and then taking the simple mean of those per-class accuracies. In this way, each class contributes equally regardless of its size. In contrast, micro-averaged accuracy is computed by averaging all test examples across classes, so larger classes carry proportionally more weight, a limitation noted by the SOTA authors.[20] Extended metrics (F1 score, precision, recall) for the EC class prediction task are reported in Appendix Fig. 14. Additionally, we compare our fine-tuned models with a zero-shot baseline with Llama-3.1 70B, in Table 1. We see that the zero-shot prompting approach lacks far behind the fine-tuned models of this size and general capabilities. This indicates that at present it seems inevitable to fine-tune the general purpose model for a complex and domain-specific task such as EC classification in biochemistry.

3.1.2 Product and substrate prediction tasks. The 70B model generates a high proportion of chemically valid molecules in canonical format, with canonical matches (the output string matches the ground truth string as it is) accounting for 24.9% and 13.0% for products and substrates respectively. While FS shows a higher percentage of canonical matches, RS has a greater proportion of chemically valid but incorrect predictions, indicating that retrosynthesis may involve more complex structural reasoning. Chemically invalid predictions are minimal (<5% of the total test set for both tasks), and wrong generations due to *e.g.* formatting errors are rare (<2%). This demonstrates that the LLMs can easily adhere to complex

Open Access Article. Published on 30 October 2025. Downloaded on 2/8/2026 10:48:01 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

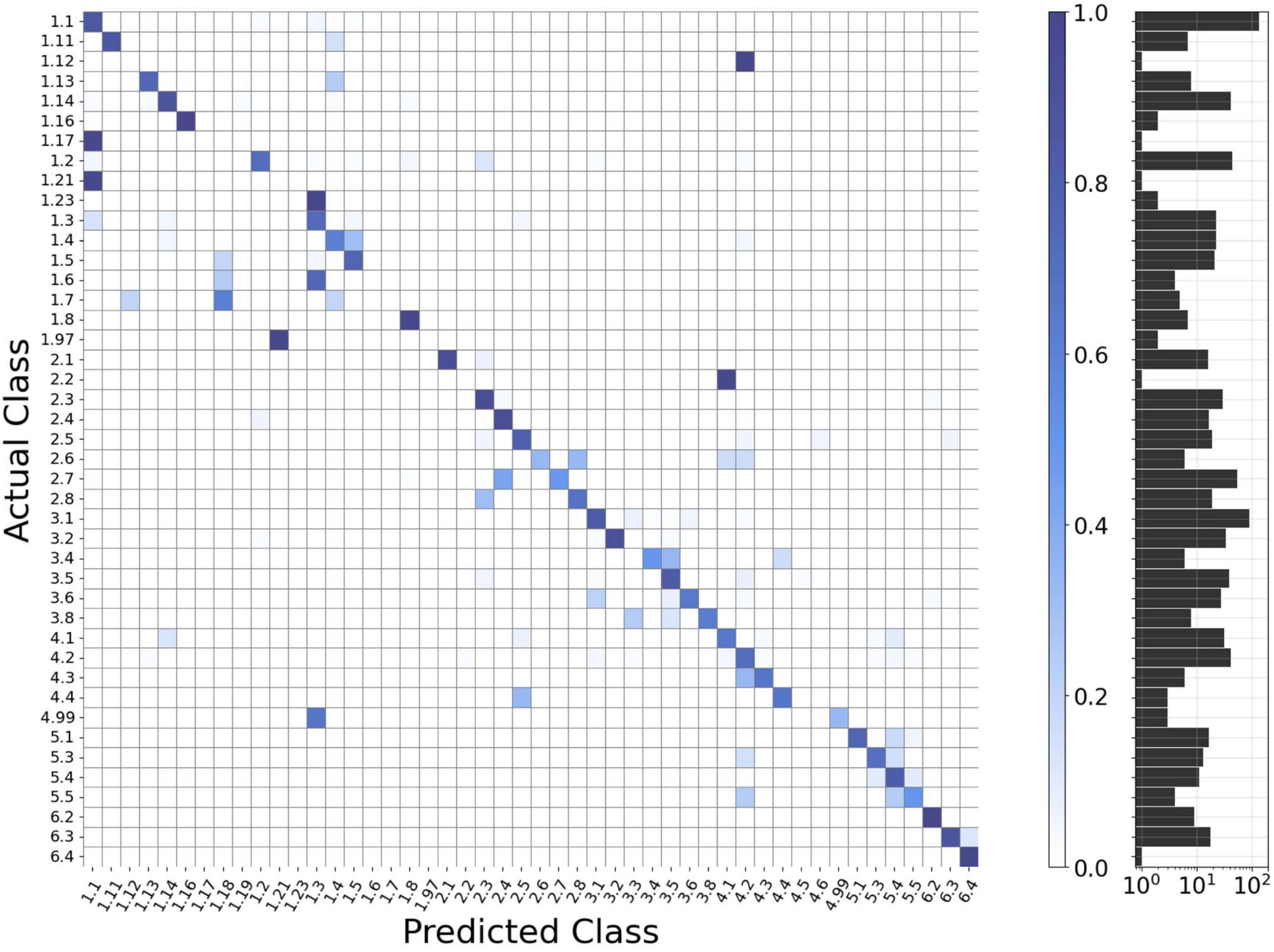


Fig. 7 Confusion matrix representing Llama-3.1 70B accuracy in predicting the EC number up to the second digit (EC2), given reactants and substrates, for one experiment. The out-of-diagonal elements show how examples are misclassified. The misalignment in the diagonal elements is due to the set of predicted classes having elements that are not present in the test set, like subclasses 3.3.*X*.*X* and 4.5.*X*.*X*, that the model predicts in a few cases. The histogram on the right shows the test set distribution stratified by EC2 subclass.

domain specific grammar like SMILES and to requested output formats, which is a useful property for the analysis of model results. However, these results are not yet competitive with the SOTA model[19] (49.6% and 60.0% accuracy for exact matches for FS and RS respectively). Note that the dataset used for our models is not exactly the same as the one from SOTA, making the results not directly comparable. Pie charts in Fig. 9 display the distribution of predictions across the five categories for FS and RS tasks, respectively, for Llama-3.1 70B.

When the model fails to predict the exact molecule, it generates relevant alternatives that may hold biochemical utility in 12% and 35% of the cases, for products and substrates respectively. We classify such an output with biochemical utility if the generated molecule shows a high Tanimoto similarity to the correct output. Focusing on the set of valid chemicals, Tanimoto similarity scores are computed and shown in Fig. 10. In the dataset, SMILES for products are shorter than substrates on average, and we also observe that for branching reactions, the set of products that are possible from certain substrates in a forward synthesis task, is generally smaller than the set of possible substrates reachable from a product in a retrosynthesis task, as observed in the Appendix Fig. 11. Thus, for products, the model either predicts a molecule very close to matching the ground truth, or it gets the wrong chemical. For substrates on the other hand, having longer strings and more options in the RS task leads to generating many substrates that are not correct, but show a relatively high Tanimoto score. Analyzing the highest Tanimoto values, we see that 7.1% of chemically valid products, and 6.7% of chemically valid substrates, report a score equal to 1. Examples of these chemicals are reported in Appendix Fig. 17 and 18. We summarize the results for both Llama-3.1 8B and Llama-3.1 70B on FS and RS tasks, including a baseline 0-shot performance and comparison to SOTA in the Appendix Table 6.

3.1.3 Generalization over unseen tasks.

When fine-tuned on a single biochemical task, the model not only retains its general capabilities on unseen, related tasks within the same sub-domain but also improves its performance compared to its zero-shot baseline. To evaluate this generalization effect, we test each of the three single-task (ST) fine-tuned models on the two tasks they were not trained on, comparing their performance to the respective zero-shot baseline. Results show that fine-tuning on either the FS or RS task significantly improves EC class

Open Access Article. Published on 30 October 2025. Downloaded on 2/8/2026 10:48:01 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

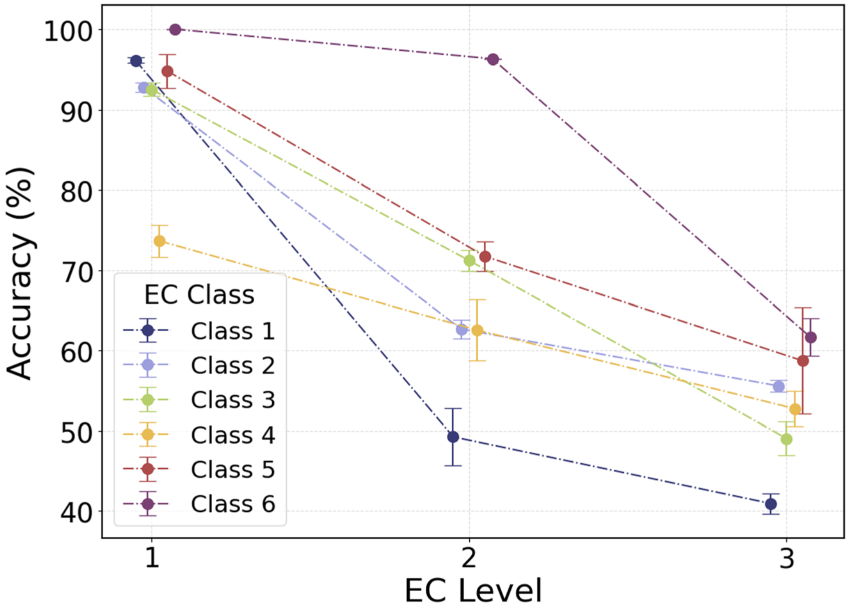


Fig. 8 Llama-3.1 70B accuracy in predicting the EC number up to level EC3, organized by main class. Accuracy measures if the model correctly matches the ground truth EC number up to the EC level specified on the *x*-axis. Accuracies are computed considering each (sub)class as equally weighted. These distribution patterns influence model performance, irrespective of reaction complexity or SMILES grammar.

Table 1 Performance comparison between Llama-3.1 70B and Llama-3.1 8B models fine-tuned for the EC prediction task, from predicting level EC1 only, to all digits up to EC3 included. A baseline 0-shot prompting approach with the 70B model is reported as well. We also show our models performance in micro-average next to the SOTA model[20] in micro-average. Note that the dataset is not exactly the same (see Subsection 2.1) and thus results are still not entirely comparable

Metric	Llama 8B	Llama 70B	Llama 70B 0-shot	Llama 70B micro-avg	SOTA
EC1 accuracy (%)	86.4 ± 0.6	**91.7 ±0.5**	29.6 ± 0.7	92.4 ± 0.2	96.2
EC2 accuracy (%)	56.5 ± 1.5	**61.7 ± 1.1**	8.7 ± 0.5	75.6 ± 0.1	93.4
EC3 accuracy (%)	40.5 ± 0.6	**49.2 ±0.7**	5.7 ± 0.4	68.1 ± 0.1	91.6
Validity (%)	>99.9	**100.0**	89.4 ± 0.3	100.0	—

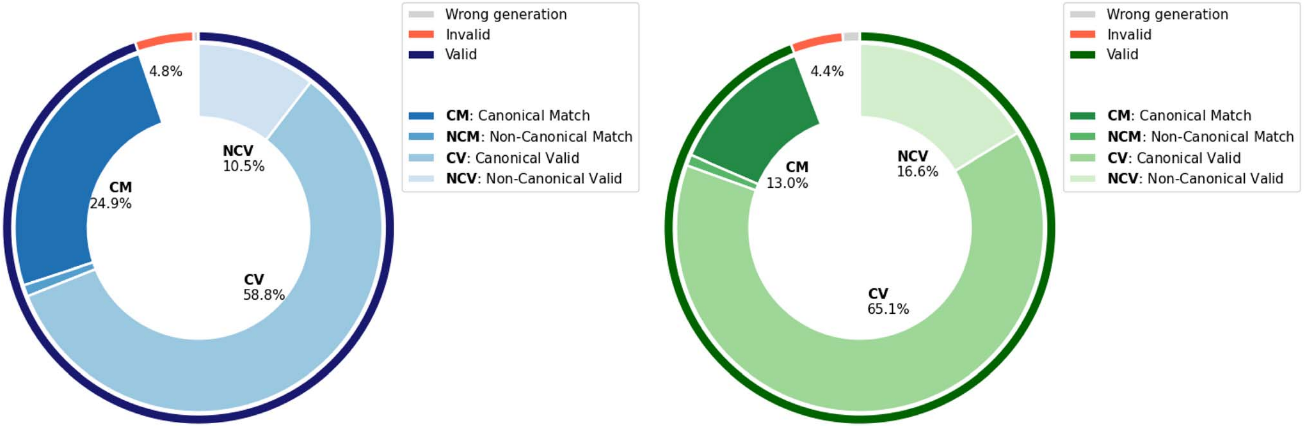


Fig. 9 Pie charts showing the average distribution of predictions for forward synthesis (FS, left) and retrosynthesis (RS, right) for Llama-3.1 70B. The outer layer indicates the proportion of correctly generated (blue/green), invalid chemicals (red), and wrongly generated predictions (grey), while the inner layer differentiates correct outputs from structurally valid but incorrect outputs. Invalid and wrongly formatted predictions remain <5% and <2% for both tasks, respectively. Results for each category are obtained averaging over $N = 3$ experiments, with standard deviations below 5% of each category value. Percentages are shown for >2% slices only.

Open Access Article. Published on 30 October 2025. Downloaded on 2/8/2026 10:48:01 AM. This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

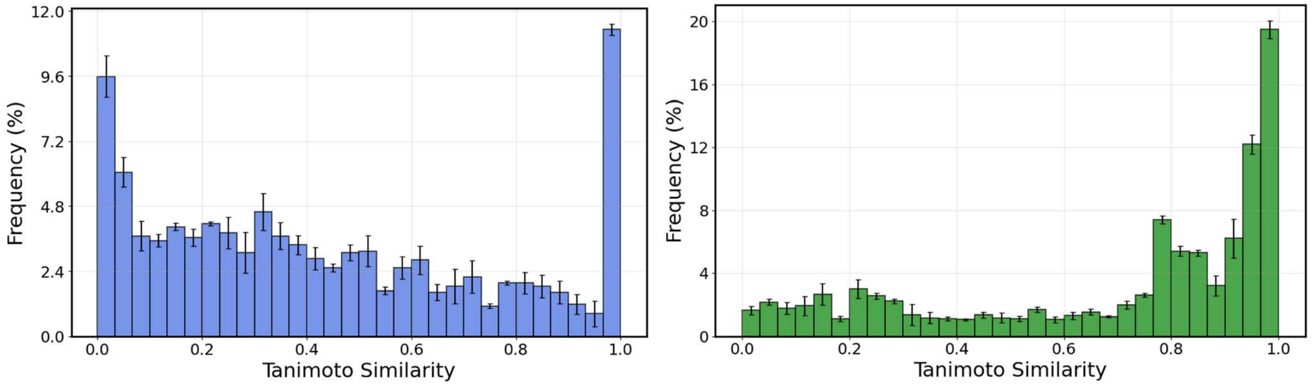


Fig. 10 Histograms of Tanimoto similarities of ground truths against products (left) and substrates (right), that the model predicts as chemically possible but not corresponding to the ground truth.

Table 2 Generalization of ST fine-tuned Llama-3.1 70B models when tested on the unseen related biochemical tasks. The zero-shot baseline is reported for comparison. Performance on the original fine-tuned task is omitted to emphasize cross-task generalization. The reported match values are here considered regardless of canonicity. The invalid category includes both incorrect SMILES notation as well as wrongly formatted output from the LLM

	EC		FS		RS	
Fine-tuned on	EC1 ↑ (%)	Invalid ↓ (%)	Match ↑ (%)	Invalid ↓ (%)	Match ↑ (%)	Invalid ↓ (%)
EC	—	—	**12.9**	31.1	0.3	55.3
FS	**54.3**	**0.3**	—	—	**1.4**	**3.3**
RS	42.1	6.3	0.6	**5.5**	—	—
ICL 0-shot	29.6	10.6	<0.1	53.5	<0.1	82.4

prediction accuracy, nearly doubling the zero-shot baseline performance. Likewise, a model fine-tuned exclusively on EC number prediction improves FS match accuracy from nearly 0% to 12.9% while also reducing invalid predictions by half. Table 2 presents the generalization results, where each fine-tuned model is tested on the two unseen tasks.

3.2 Multitask fine-tuning

Using a multitask setup we show that we can improve performance through the use of synergistic information from the related task, in particular for FS and RS tasks. For these the model performance for matches (regardless of canonicity) increases by 7.9% and 5.3% respectively. The three ST datasets are merged together to provide the dataset used for the MT setup. The Llama 70B and 8B models are both fine-tuned, using the best-performing configuration identified in the single-task experiments. Performance is compared against single-task setups to assess multitask learning benefits, with the main results reported in Table 3.

3.3 Exploring low-data regimes

Fine-tuned LLMs show promise in low data regimes: for Llama-3.1 70B, we report almost double EC class accuracy when comparing zero-shot prompting (29.6%) with the fine-tuned version with only $N = 200$ training samples (55.3%). We replicate low-data scenarios to evaluate how the models perform with significantly reduced training samples. Specifically, we analyze performance degradation when the training set size is limited to 600 and 200 compared to our default training (~1800 samples per task). This analysis is conducted for both models, to provide insights into their scalability when data availability becomes the bottleneck. Both models show a steady performance increase when training data is increased. The larger architecture holds an edge over the smaller one regardless of data size across almost all tasks, confirming again its greater capabilities.

For a fairer comparison, we include a simple XGBoost baseline. XGBoost[59] is a gradient boosting model that performs well with structured data and does not rely on large-scale pretraining, making it a suitable reference for evaluating whether LLM fine-tuning truly adds value in data-limited biochemical prediction tasks. We find that across all tasks and for each data scenario, our models outperform the XGBoost model. We report our findings in Table 4. More details on how XGBoost is trained are reported in the Appendix in Subsection A.6.

3.4 Impact of different LoRA setups

We observe that adding more trainable parameters can lead to performance improvement for most tasks. This indicates the importance of parameter-efficient learning strategies in domains where fine-tuning is essential. We see the trend that LoRA default performs better than LoRA attention and LoRA light in almost all settings. In most tested cases for the 8B

Open Access Article. Published on 30 October 2025. Downloaded on 2/8/2026 10:48:01 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

Table 3 Performance comparison between single-task and multitask setups for Llama-3.1 8B and Llama-3.1 70B. Blue cells represent performance improvement, orange cells represent performance reduction. The reported match values are here considered regardless of canonicity. The categories "Match + (TS = 1)" and "Match + (TS > 0.95)" add to the previous one the share of valid chemicals with a Tanimoto score equal to 1 and greater than 0.95 respectively. Numbers are presented in bold if the best performance improvement does not fall within one standard deviation from the second-best

		Llama-3.1 70B		
Task	Metric (%)	ST	MT	Δ
EC	Accuracy EC1 ↑	**91.7**	86.4	-5.3
	Accuracy EC2 ↑	61.7	**65.1**	+3.7
	Accuracy EC3 ↑	49.2	48.9	−0.3
FS	Match ↑	25.9	**33.8**	+7.9
	Match + (TS = 1) ↑	33.0	**44.4**	+11.4
	Match + (TS > 0.95) ↑	34.2	**45.4**	+11.2
	Invalid ↓	4.8	4.9	+0.1
RS	Match ↑	13.9	**19.2**	+5.3
	Match + (TS = 1) ↑	20.6	**30.1**	+9.5
	Match + (TS > 0.95) ↑	36.1	**45.4**	+9.3
	Invalid ↓	4.4	**3.0**	-1.4

model, LoRA attention performs slightly better than LoRA light, while for the 70B model, LoRA light performs slightly better than LoRA attention in most tested cases. Performance across all tasks with different LoRA setups are reported in Table 5.

3.5 Context: where do LLMs fit?

Computational tools for reaction prediction have progressed from rule libraries to deep task-specific models, following waves that traded interpretability for coverage. Below, we contrast the historical approaches, explain how LLMs complement specialized tools, and how such generalist models currently fit and will evolve in the current tool landscape.

- Template-based models (LHASA,[4] SYNTHIA[6]) are deterministic and transparent. The outputs are easy to inspect, but every new reaction type demands manual rule curation, leading to rule explosion and limited ability to generalize.
- Data-driven statistical learners (EFICAz[10]) can exploit larger databases, yet lose interpretability and can inherit the human biases embedded in handcrafted descriptors.
- Deep learning models (such as the molecular[15] and enzymatic[18] transformers) reach state-of-the-art accuracy by discovering hidden patterns. They are, however, computationally intensive, data hungry and largely less interpretable.
- LLMs require large-scale pretraining, yet could cover a broad task spectrum.[38,39] They offer tentative interpretability *via* natural language interrogation, although hallucinations remain a risk.

Our benchmarks confirm that pretrained LLMs still require task-specific tuning before tackling complex biochemical pipelines. Overall applicability depends on the goal: even after fine-tuning them, for high-precision tasks, specialized models win. However for breadth, adaptability and human-centric interaction, LLMs are compelling. Thanks to their fast repurposing and unified conversational interface, they can work as control layers for the existing toolbox. Compact code logic, plug-in nature and access to legacy tools raise the bar for automation. Researchers already query LLMs for literature, accessing collected knowledge more effectively. It is important to note that the LLM evaluation is not as reliable as human evaluation in chemical reasoning. Practitioners should choose LLMs for exploratory or hypothesis-generation stages, rapidly changing tasks, or settings with sparse data, and stick to specialized models for best accuracy.

Standard LLMs are not yet robust for complex zero- or few-shot biochemical tasks, so for now they serve as sparring partners rather than oracles. But progress is rapid: during this study we saw multimodal LLMs emerge, along with reasoning

Table 4 Performance of Llama-3.1 8B and Llama-3.1 70B across all tasks and for different training set sizes. The reported match values are here considered regardless of canonicity. Each task is trained on a slightly different amount of samples (±20) because of how data has been split, thus we report a reference number of 1800 samples in the corresponding rows. Numbers are presented in bold if the best performance does not fall within one standard deviation from the second-best. A baseline XGBoost model is reported for comparison

		EC			FS		RS	
Model	Train set size	EC1 ↑ (%)	EC2 ↑ (%)	EC3 ↑ (%)	Match ↑ (%)	Invalid ↓ (%)	Match ↑ (%)	Invalid ↓ (%)
LLama-3.1 8B	200	43.5	15.5	8.5	2.6	4.6	0.2	11.2
	600	65.6	30.1	17.4	8.3	6.4	2.8	10.2
	∼1800	86.4	56.5	40.5	18.4	9.4	**15.1**	**4.3**
LLama-3.1 70B	200	55.3	28.5	17.7	7.7	7.7	2.9	7.3
	600	73.5	45.8	33.1	11.0	4.4	4.1	7.2
	∼1800	**91.7**	**61.7**	**49.2**	**25.9**	4.8	13.9	**4.4**
XGBoost	200	32.7	4.9	<0.1	<0.1	—	<0.1	—
	600	40.9	6.0	1.7	1.9	—	2.5	—
	∼1800	54.0	23.7	15.9	5.1	—	3.6	—

Open Access Article. Published on 30 October 2025. Downloaded on 2/8/2026 10:48:01 AM. This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

Table 5 Performance of Llama-3.1 8B and Llama-3.1 70B across all tasks and for different fine-tuning setups. Perfomance for all tasks increases with the number of fine-tuned parameters, the only exception being the attention fine-tuning for Llama-3.1 70B, where an increase in FS performance comes with a degradation in RS and EC prediction tasks. The reported Match values are here considered regardless of canonicity. Numbers are presented in bold if the best performance does not fall within one standard deviation from the second-best

		EC			FS		RS	
Model	LoRA type	EC1 ↑ (%)	EC2 ↑ (%)	EC3 ↑ (%)	Match ↑ (%)	Invalid ↓ (%)	Match ↑ (%)	Invalid ↓ (%)
LLama-3.1 8B	light	72.0	44.1	30.3	10.2	9.1	4.7	12.8
	attention	82.0	48.4	31.9	11.3	7.9	6.9	10.1
	default	86.4	56.5	40.5	18.4	9.4	**15.1**	4.3
LLama-3.1 70B	light	85.8	58.5	45.2	21.4	6.0	13.7	3.9
	attention	78.8	48.0	34.9	**25.6**	5.5	9.8	3.3
	default	**91.7**	**61.7**	**49.2**	**25.9**	4.8	13.9	4.4

variants that spend extra inference time or call external tools to verify answers and surface their chain-of-thought. We therefore expect future workflows to resemble small teams of models: one LLM engages the user, then hands off to specialist agents for planning or calculation. As these guard-railed systems mature, a single general-purpose LLM may absorb many routine tasks. Until then, pairing a LLM with specialist tools remains the safest and most productive path.

3.6 Limitations

While our study demonstrates the potential for researchers to work with LLMs when studying biochemical reactions, several limitations must be acknowledged. Addressing these will be key to improving both model accuracy and applicability in real-world biochemical workflows.

• Potential data leakage: although we fine-tune the LLM to evaluate performance in low-data regimes, it is possible that the model has already been exposed to similar biochemical reaction data during pretraining, as such datasets are available online. For a fairer comparison, future evaluations should ensure that test sets are composed of truly held-out reactions that cannot be scraped or indirectly inferred from pretraining text on the internet. This would provide a clearer measure of the model's generalization ability beyond memorization. Moreover, our similarity analysis (Appendix A.1) shows that, even after grouping branching reactions, substantial analogue overlap remains between train and test substrates and products, indicating that stricter split protocols are required.

• Data constraints: our study is based on the BRENDA subset of the ECREACT dataset, which, while extensive, does not fully cover the diversity of enzymatic reactions and does not allow a direct comparison to current SOTA model. The limited representation of certain EC subclasses affects generalization. Expanding training to the full ECREACT dataset or integrating additional reaction databases could mitigate this issue and enhance model robustness, yet also here, ECREACT has been preprocessed and simplifies complex biochemical reaction mechanisms to a certain degree.

• Computational constraints: fine-tuning LLMs is computationally expensive, even with PEFT strategies like LoRA, limiting accessibility for resource-constrained environments.

• Interpretability analysis: we focus on predictive metrics only and do not analyze how the model assigns EC numbers or predicts reaction outcomes, nor whether its intermediate reasoning aligns with biochemical knowledge. SMILES strings are not inherently human-readable, but can be converted into molecular graphs for deeper analysis of the model's prediction. Inspecting LLM-generated rationales step-by-step *via* chain-of-thought prompts is a promising direction for future work, both for user's interpretability and to feedback them back into the model for more robust responses.

4 Conclusions

In this study, we systematically evaluated the potential of Large Language Models (LLMs) for biochemical reaction prediction, focusing on enzyme commission classification, forward synthesis, and retrosynthesis. By fine-tuning Llama-3.1 models, we demonstrated that LLMs can answer biochemical questions, although they are not yet fully competitive with specialized models. Fine-tuning significantly improves performance over in-context learning, with Llama-3.1 70B achieving 91.7% accuracy in EC class classification. Fine-tuning on a single task does not degrade the 70B model capabilities on unseen related tasks, as we observe performance improvement compared to zero-shot baselines that use the base, pretrained model. Multitask learning enhances forward synthesis and retrosynthesis predictions, with a match accuracy of 33.8% and 19.2% respectively, indicating that leveraging shared biochemical knowledge improves generalization. Additionally, LLMs have potential in low-data regimes, making them valuable for applications where labeled data is scarce. The choice of fine-tuning strategy impacts the performance, with LoRA offering an efficient and scalable adaptation method. Despite these strengths, several challenges remain: LLMs struggle with handling rare EC subclasses and ensuring reliable predictions. As LLM architectures continue to evolve, their integration into biochemical workflows has the potential to accelerate discoveries in enzyme-substrate prediction and biocatalysis design.

© 2025 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 30 October 2025. Downloaded on 2/8/2026 10:48:01 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

Author contributions

J. M. Weber and L. Di Fruscia conceptualized the study. L. Di Fruscia led data curation, formal analysis, investigation, methodology, software development, validation, visualization, and drafting of the manuscript. J. M. Weber supervised the project, provided resources, and reviewed and edited the manuscript.

Conflicts of interest

There are no conflicts to declare.

Data availability

Data and code for this article are available at https://github.com/Intelligent-molecular-systems/LLM_finetuning_for_biochemistry. This study was carried out using publicly available data at https://github.com/rxn4chemistry/biocatalysis-model. DOI: https://doi.org/10.5281/zenodo.17224080.

A Appendix

A.1 Data preprocessing and analysis

The original dataset comprises 8496 reaction examples, distributed across the seven EC classes as follows: Class 1 (3361), Class 2 (1700), Class 3 (1596), Class 4 (964), Class 5 (504), Class 6 (352), and Class 7 (19). We implemented a series of preprocessing steps to ensure a fair split across training and test set and across tasks:

- Canonicalization of SMILES representations: reactions with substrates or products in different SMILES representations are unified by converting all SMILES strings to their canonical forms using the RDKit library. This ensures that duplicate {substrate, product} pairs, differing only in molecular representation, are identified and removed. In this step, 362 reactions (4.2% of the total) are reformatted, and no reactions are discarded.
- Grouping of related reactions: reactions that represent the same underlying biochemical process but differ slightly due to variations in substrate or product representations are grouped: whenever a {substrate, EC} or {product, EC} pair maps to multiple valid counterparts, we treat all those reactions as a group that must stay together in any train/test split. A group of size N contains N distinct reactions sharing the same {substrate, EC} or {product, EC}. With this definition, a group size of $N = 1$ indicates a unique reaction with no branching alternatives. We refer to this as substratebranching and productbranching respectively. All reactions within a group are allocated to the same dataset split (training or test) to avoid leakage.
- Avoidance of task-specific leakage: in forward synthesis (FS) and retrosynthesis (RS), if a reaction appears in FS, then any of its counterparts with the same product and EC number but different substrates, must not appear in RS. This prevents the model from gaining undue advantage by being exposed to related information in the training phase.

Branching groups distribution varies a lot on whether we look at the products or the substrates, as most of the branching substrates only lead to 2 or 3 possible products, but the reverse task has a wider spread. We report this in Fig. 11. We further analyze the substrates and products to assess whether overlapping reactions across groups are present. This is needed because *e.g.* a substrate, while branching into multiple products, may also be part of a set of substrates reachable from a specific product. We follow this by merging those overlapping groups together and removing redundant entries.

A.1.1 Similarity check across splits. To determine whether our cold substrate and cold product splits still leak information through highly similar molecules, we compute the pairwise Tanimoto similarity between the training and test sets, separately for substrates and products. For each training molecule, we compute the mean of its ten most similar test set neighbours. The resulting distributions reveal that 38% of training products and 67% of training substrates have a Tanimoto score >0.85. Such extensive overlap persists despite our grouping of duplicate {substrate, EC} and {product, EC}

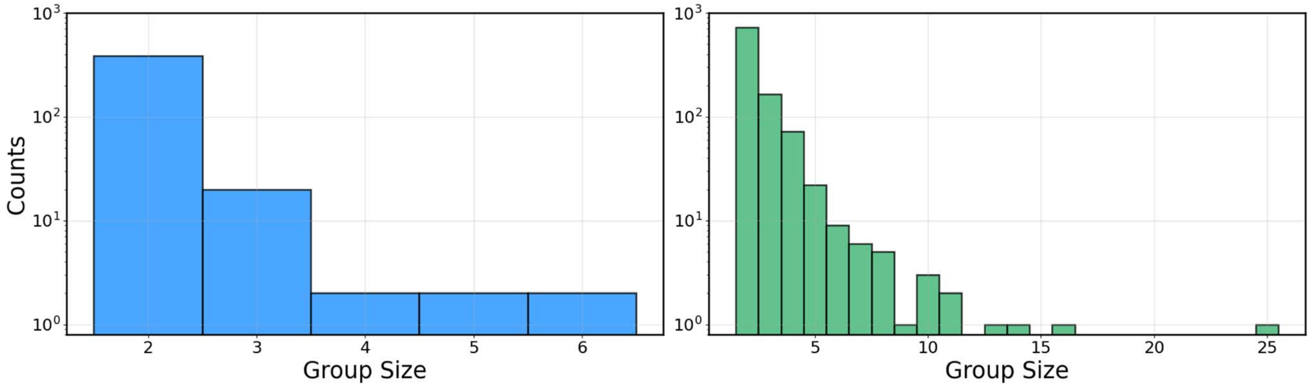


Fig. 11 Histograms of group size for duplicate {substrate, EC} (left) and duplicate {product, EC} (right). We can observe that while most duplicate reactions branch into two possible products, substrates tend to branch into larger groups.

Open Access Article. Published on 30 October 2025. Downloaded on 2/8/2026 10:48:01 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

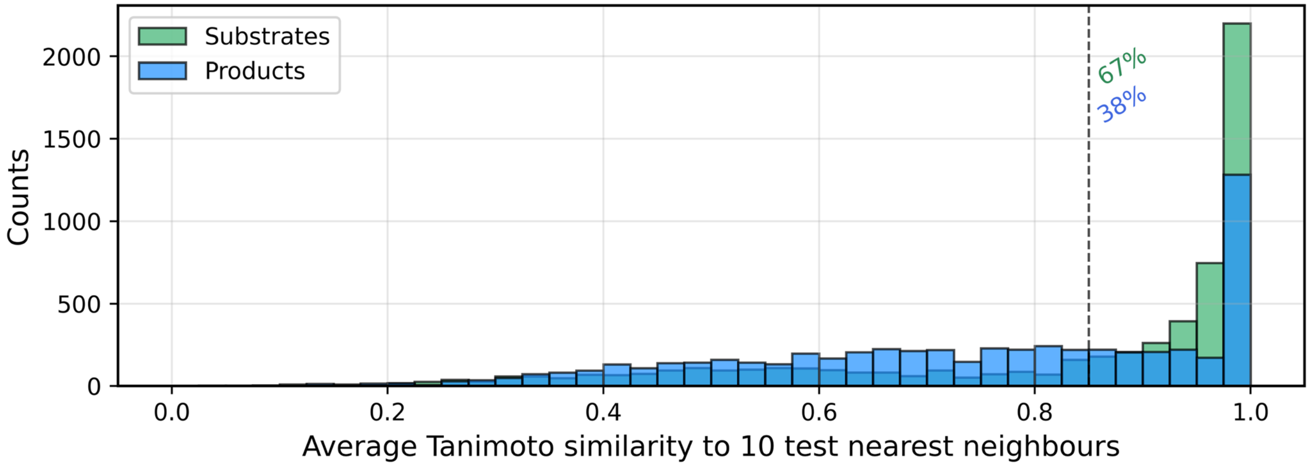


Fig. 12 Histogram of mean top-10 Tanimoto similarities between training and test molecules. Substrates are shown in green and products in blue. Each bar aggregates training molecules whose average similarity to their ten closest test set analogues falls in that interval. The dashed line at 0.85 marks the high-similarity regime, that includes 67% of substrates and 38% of products.

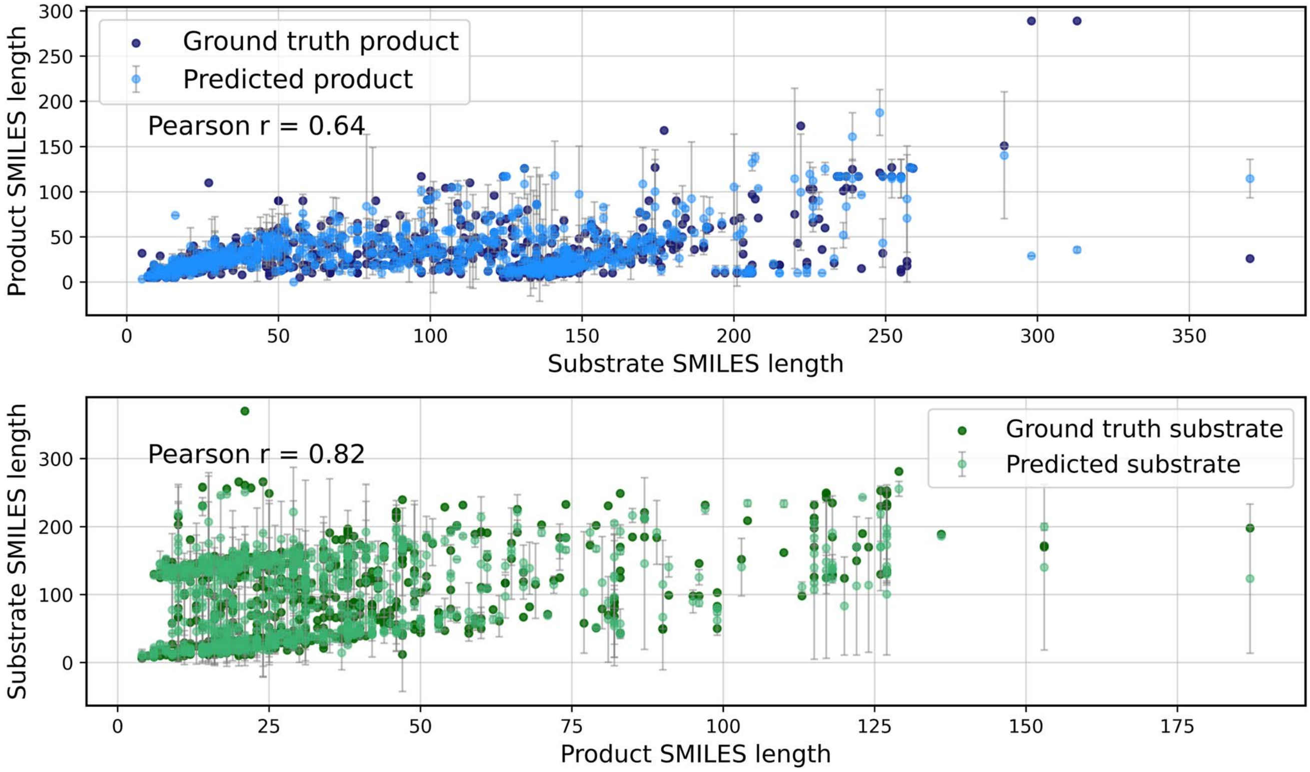


Fig. 13 Scatterplots of SMILES-length correlations. Top: forward synthesis (Pearson $r = 0.39$). Bottom: retrosynthesis (Pearson $r = 0.38$). This shows that true input and output lengths are weakly correlated. On the plots is further reported the correlation coefficient between predictions and ground truths length, showing strong correlation for both tasks.

reactions, highlighting that exact-match splitting alone is insufficient to eliminate analogue leakage in enzyme reaction datasets.

A.1.2 **Input-output SMILES length correlation.** To quantify the intrinsic relationship between substrate and product SMILES lengths in our test set, we first compute the Pearson correlation coefficient r between true substrate and true product lengths, obtaining $r \sim 0.39$, which indicates only a weak linear association. We then evaluate how well our model reproduces that trend by correlating predicted with true lengths: for

Open Access Article. Published on 30 October 2025. Downloaded on 2/8/2026 10:48:01 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

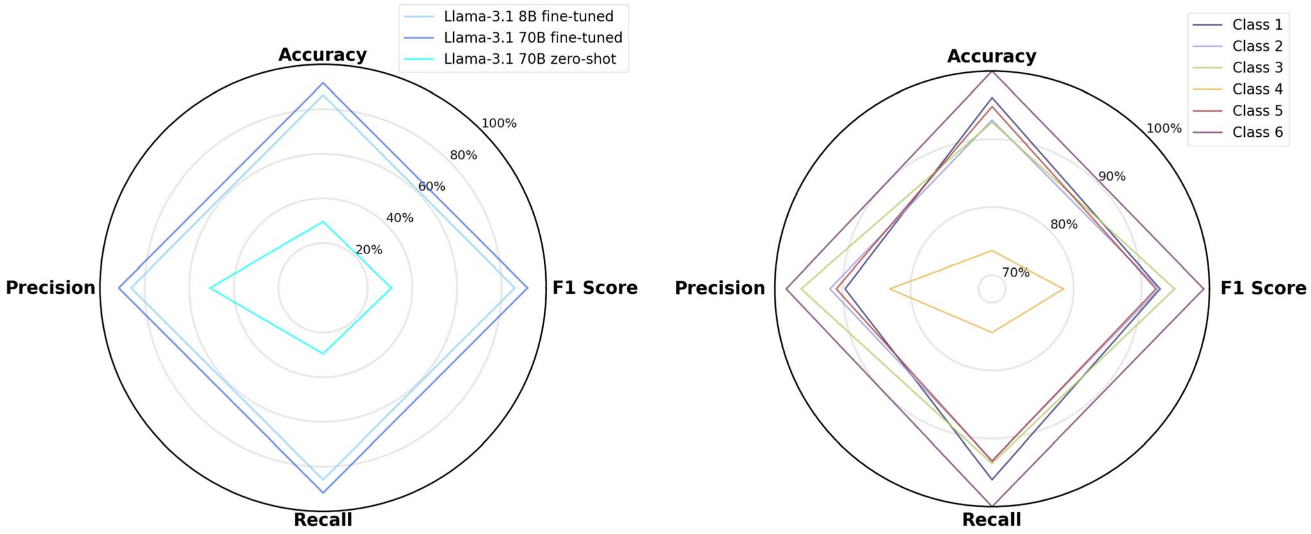


Fig. 14 Left: radar plot showing accuracy, precision, recall and F1 score for Llama-3.1 70B and Llama-3.1 8B averaged over $N = 3$ experiments. The plot demonstrates consistent outperformance of the larger model over the 8B variant across all metrics. For comparison, we also show the 70B model perforsmance when it is not fine-tuned, in a zero-shot format. Right: EC class accuracy for the fine-tuned Llama-3.1 70B stratified by the class.

Table 6 Performance comparison between Llama-3.1 8B and Llama-3.1 70B models for forward- and retrosynthesis. All values for our fine-tuned models are obtained averaging over $N = 3$ experiments, with standard deviations below 5% of each category value. A zero-shot baseline on the pretrained 70B model is reported for comparison. We also report the SOTA model[19] performance at the end. Note that the dataset is not exactly the same (see Subsection 2.1) and thus results are still not entirely comparable. Numbers are presented in bold if the best performance does not fall within one standard deviation from the second-best. NCM, CV, and NCV categories taken alone do not reflect model improvement, thus we do not bold them. CM:canonical matching, NCM:non-canonical matching, CV:canonical calid, NCV:non-canonical valid

Model	Task	CM ↑ (%)	NCM (%)	CV (%)	NCV (%)	Invalid ↓ (%)
Llama-3.1 8B	FS	17.6	0.8	53.8	14.0	9.4
	RS	14.0	1.1	67.8	11.7	4.3
Llama-3.1 70B	FS	**24.9**	1.0	58.8	10.5	**4.8**
	RS	13.0	0.9	65.1	16.6	4.4
Llama-3.1 70B 0-shot	FS	<0.1	0	40.5	5.9	53.5
	RS	0	<0.1	12.7	4.9	82.4
SOTA	FS	49.6	—	—	—	—
	RS	60.0	—	—	—	—

forward synthesis we observe $r = 0.64$, and for retrosynthesis $r = 0.82$. We report our results in Fig. 13.

A.2 EC class prediction radar plots

Computing precision, recall and F1 score alongside accuracy, we observe that these four metrics are all consistent with each other for both of our fine-tuned model sizes, with Llama-3.1 70B beating Llama-3.1 8B in every metric. We compare them to a 0-shot prompting setup with the pretrained Llama-3.1 70B as a baseline, observing the clear performance gap between in-context learning with the larger model, against the fine-tuned 8B version. Focusing on the fine-tuned 70B model, a stratification by main class shows us again that the values for the four metrics are consistent with each other, per class, with class 4 being the most unbalanced. These findings are reported in Fig. 14.

A.3 Forward- and retrosyntesis comparison with fine-tuned llama 8B

The 70B model performs better than the 8B one for forward synthesis, and are both comparable when it comes to retrosynthesis. We report the main results in Table 6, alongside the SOTA model.

A.4 Average Tanimoto scores in ground truth branching

We observe that for the equally valid ground truths that the database stores for a given reaction, many examples show a relatively low similarity score. Focusing on the product prediction only, some of the reasons this happen can be due to having a co-factor recorded in place of the main product, or some entries may report products that correspond to different

Open Access Article. Published on 30 October 2025. Downloaded on 2/8/2026 10:48:01 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

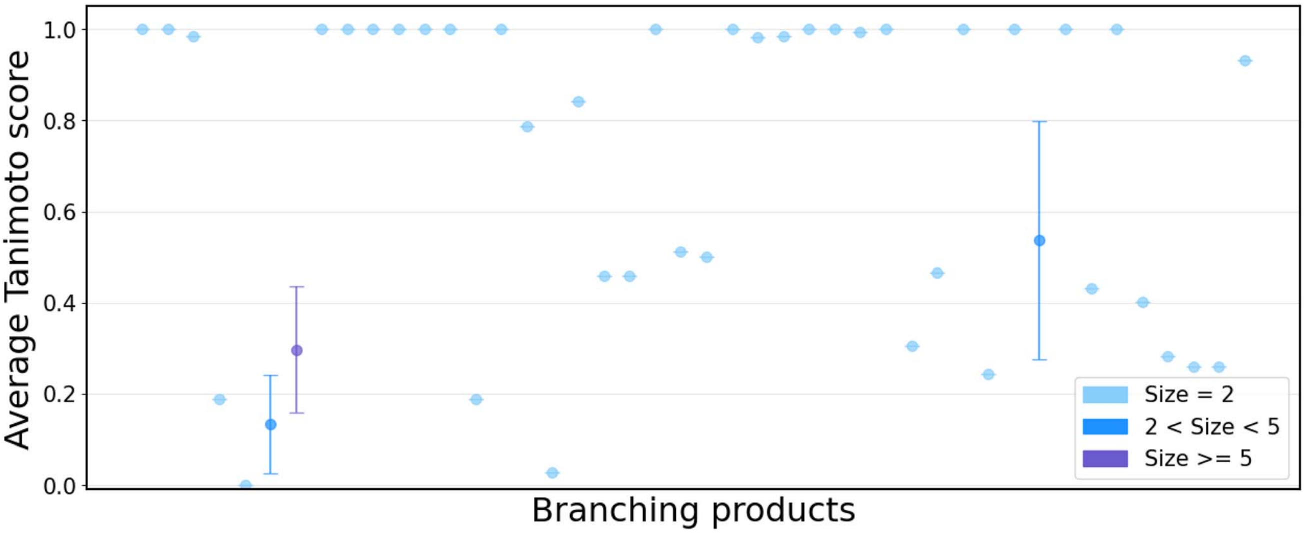


Fig. 15 Average Tanimoto score computed across a ground truth product and each of its ground truth branching counterparts, for all groups and stratified by group size. For branching groups of size 2, no standard deviation is shown as we only have one Tanimoto score computed between the reference ground truth and its alternative option.

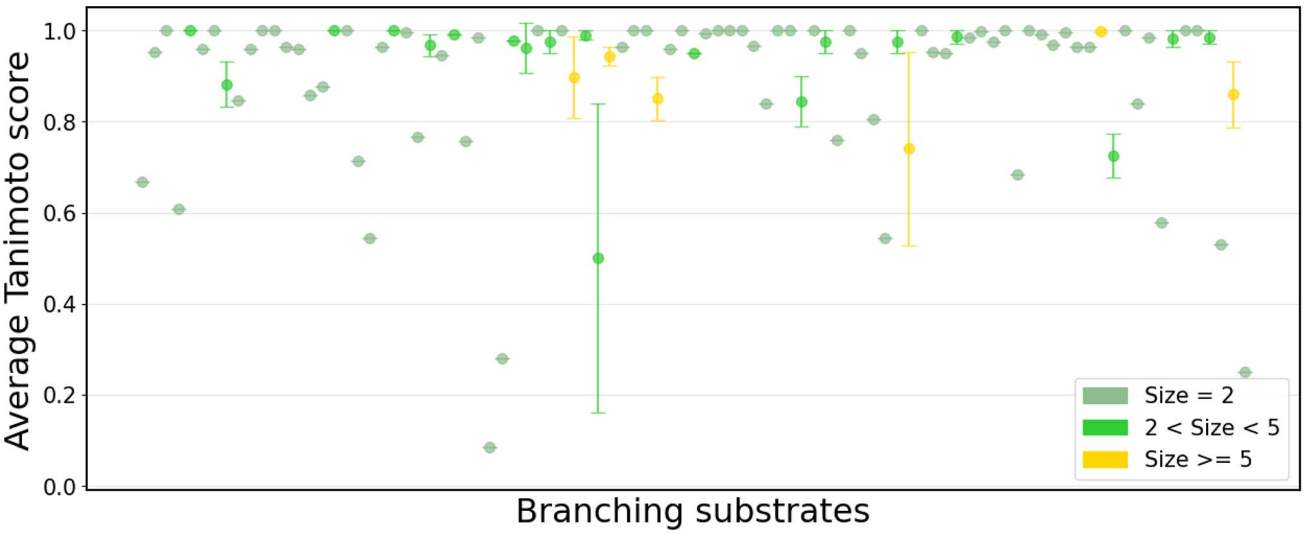


Fig. 16 Average Tanimoto score computed across a ground truth substrate and each of its ground truth branching counterparts, for all groups and stratified by group size. For branching groups of size 2, no standard deviation is shown as we only have one Tanimoto score computed between the reference ground truth and its alternative option.

reaction intermediates, a problem that strictly relates to the presence of branching reactions in the dataset. We compute the average Tanimoto score across ground truth chemicals that belong to the same set of branching product/substrates, to get insights over the chemical diversity of alternatives products/substrates that are reported in the dataset.

Given a group of size N, we compute the Tanimoto scores between one element of the set and the remaining $N - 1$ chemicals. Then, we compute the average Tanimoto score and its standard deviation for that group. If the chemicals are all similar to each other, we observe a high average with a relatively small standard deviation. On the other end, if the chemicals present more variability, we expect to see a lower average with a wider spread in the standard deviation. We report the findings in Fig. 15 and 16.

Open Access Article. Published on 30 October 2025. Downloaded on 2/8/2026 10:48:01 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

A.5 Predictions with Tanimoto score equal to 1 for products and substrates

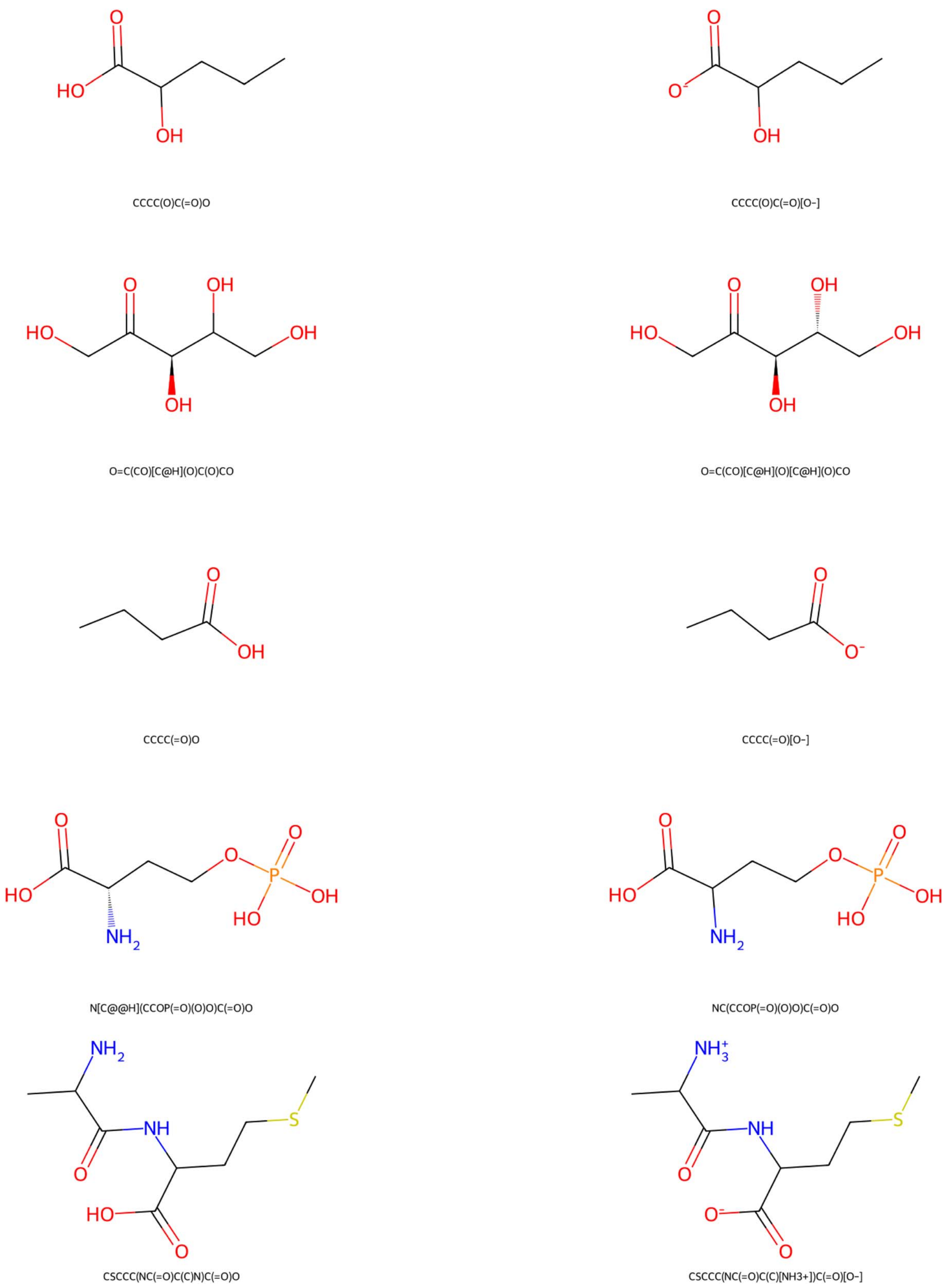


Fig. 17 Examples of predicted (left) *vs.* ground truth (right) products, when the prediction is not correct but produces a Tanimoto score equal to 1. We see that some predictions have an additional hydrogen (resulting in an OH group) while the ground truth recorded an oxygen ion (O−) (rows 1, 3), while some others have a mismatch in chirality (rows 2 and 4).

Open Access Article. Published on 30 October 2025. Downloaded on 2/8/2026 10:48:01 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

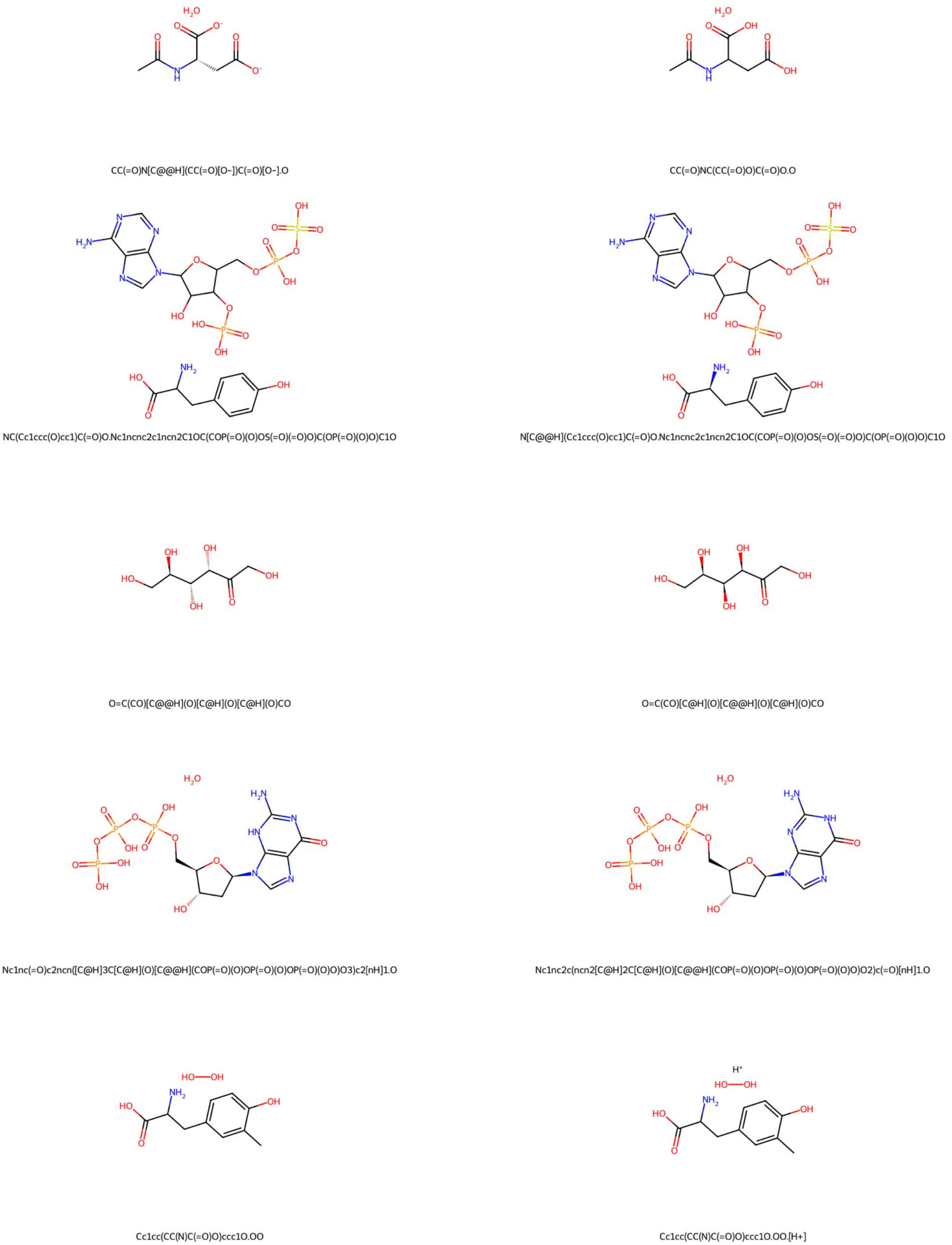

CC(=O)N[C@@H](CC(=O)[O-])C(=O)[O-].O
CC(=O)NC(CC(=O)O)C(=O)O.O
NC(Cc1ccc(O)cc1)C(=O)O.Nc1ncnc2c1ncn2C1OC(COP(=O)(O)OS(=O)(=O)O)C(OP(=O)(O)O)C1O
N[C@@H](Cc1ccc(O)cc1)C(=O)O.Nc1ncnc2c1ncn2C1OC(COP(=O)(O)OS(=O)(=O)O)C(OP(=O)(O)O)C1O
O=C(CO)[C@@H](O)[C@H](O)[C@H](O)CO
O=C(CO)[C@H](O)[C@@H](O)[C@H](O)CO
Nc1nc(=O)c2ncn([C@H]3C[C@H](O)[C@@H](COP(=O)(O)OP(=O)(O)OP(=O)(O)O)O3)c2[nH]1.O
Nc1nc2c(ncn2[C@H]2C[C@H](O)[C@@H](COP(=O)(O)OP(=O)(O)OP(=O)(O)O)O2)c(=O)[nH]1.O
Cc1cc(CC(N)C(=O)O)ccc1O.OO
Cc1cc(CC(N)C(=O)O)ccc1O.OO.[H+]

Fig. 18 Examples of predicted (left) *vs.* ground truth (right) substrates, when the prediction is not correct but produces a Tanimoto score equal to 1. We see that some predictions have a missing hydrogen (resulting in an oxygen ion O−) while the ground truth recorded an OH group (row 1), while some others have a mismatch in chirality (*e.g.* rows 2 and 3).

Open Access Article. Published on 30 October 2025. Downloaded on 2/8/2026 10:48:01 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

A.6 XGBoost data preprocessing and training

For each task, we encode the biochemical inputs into a structured format that XGBoost can process efficiently. Given its reliance on tabular data, molecular and enzymatic information is transformed into numerical feature vectors before being fed into the model:

- Molecular representation: for the product and substrate prediction tasks, we represent molecules using Morgan fingerprints to encode molecular structures into a fixed-length binary vector. Each molecule is transformed in a 256-bit binary vector, where each bit represents the presence or absence of a specific chemical substructure.
- Reaction representation: for the EC number prediction task, the entire biochemical reaction (substrates + products) is encoded as a 1024-bit reaction fingerprint. This representation captures reaction-specific features, such as changes in molecular structures and functional groups.
- EC number representation: we encode them in a way that preserves their hierarchical relationships. Instead of treating whole EC numbers as simple categorical labels, which would ignore relationships between enzymes within the same category, we encode them as four separate numerical features, one for each EC digit. Each of these four digits is first label-encoded, then converted into a continuous representation *via* standardization, approaching it as a regression task where similar EC numbers remain closer in feature space.

For all tasks, EC number label encoding is done on the full set of EC numbers, while standardization is performed using only the training set statistics, preventing information leakage from the test set.

A.6.1 Training and evaluation. XGBoost models are trained separately for each task using the same training and test splits as the LLM experiments. We run the model for 100 boosting rounds and include early stopping to avoid overfitting. For the EC number prediction task, the problem is framed as a regression task with a squared loss, whereas for the other two tasks we use a logistic regression for the output bit-vector.

- EC prediction task: the 1024-bit reaction fingerprint and the standardized, 4D vector of the encoded EC number, represent input and output respectively. Evaluation is done by reverting the standardization process for the prediction and checking whether the categorical encoding of the predicted EC digits matches the true labels exactly.
- Product and substrate prediction: the input is represented by a concatenation of the 256-bit Morgan fingerprint with the 4D encoding of the EC number, and the output is a 256-bit Morgan fingerprint. Since the fingerprints are binary, the output is considered correct if the generated fingerprint exactly matches the ground truth fingerprint, as an upper bound proxy of our "molecule matching" prediction task.

Since the EC number contributes with only four features to an input vector of hundreds of dimensions, we conducted additional experiments to explore its impact. Specifically we inflated the relative importance of the EC number by multiplying its four components by factors ranging from 5 to 100. We also completely removed the EC number from the input to test its effect on performance. Our tests show that the best performance is achieved by including the EC number with the default scaling factor of 1, confirming that enzymatic information contributes meaningfully to reaction prediction, even when it constitutes a small fraction of the feature space.

Acknowledgements

This work is supported by the AI4b.io program, a collaboration between TU Delft and dsm-firmenich, and is fully funded by dsm-firmenich and the RVO (Rijksdienst voor Ondernemend Nederland).

References

1 B. Wiltschi, T. Cernava, A. Dennig, M. G. Casas, M. Geier, S. Gruber, *et al.*, Haberbauer, Marianne. Enzymes revolutionize the bioproduction of value-added compounds: From enzyme discovery to special applications, *Biotechnol. Adv.*, 2020, **40**, 107520.

2 A. R. Alcántara, P. Domínguez de María, J. A. Littlechild, M. Schürmann, R. A. Sheldon and R. Wohlgemuth, Biocatalysis as key to sustainable industrial chemistry, *ChemSusChem*, 2022, **15**(9), e202102709.

3 R. A. Sheldon, Green chemistry and biocatalysis: Engineering a sustainable future, *Catal. Today*, 2024, **431**, 114571.

4 W. T. Wipke and E. J. Corey, Computer-assisted design of complex organic syntheses, *Science*, 1969, **166**(3905), 178–192.

5 W. T. Wipke, *et al.*, Secs—simulation and evaluation of chemical synthesis: Strategy and planning, *Computer Representation and Manipulation of Chemical Information*, 1977.

6 S. Szymkuć, *et al.*, Computer-assisted synthetic planning: The end of the beginning, *Angew Chem. Int. Ed. Engl.*, 2016, **55**(20), 5904–5937.

7 A. Bøgevig, *et al.*, Route design in the 21st century: The icsynth software tool as an idea generator for synthesis prediction, *Org. Process Res. Dev.*, 2015, **19**(2), 357–368.

8 C. W. Coley, L. Rogers, W. H. Green and K. F. Jensen, Computer-assisted retrosynthesis based on molecular similarity, *ACS Cent. Sci.*, 2017, **3**(12), 1237–1245.

9 C. Claudel-Renard, C. Chevalet, T. Faraut and D. Kahn, Enzyme-specific profiles for genome annotation: Priam, *Nucleic Acids Res.*, 2003, **31**(22), 6633–6639.

10 W. Tian, A. K. Arakaki and J. S. Eficaz, a comprehensive approach for accurate genome-scale enzyme function inference, *Nucleic Acids Res.*, 2004, **32**(21), 6226–6239.

11 Y. Li, *et al.*, Deepre: sequence-based enzyme ec number prediction by deep learning, *Bioinformatics*, 2018, **34**(5), 760–769.

12 J. Y. Ryu, H. U. Kim and S. Y. Lee, Deep learning enables high-quality and high-throughput prediction of enzyme commission numbers, *Proc. Natl. Acad. Sci. U. S. A.*, 2019, **116**(28), 13996–14001.

© 2025 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 30 October 2025. Downloaded on 2/8/2026 10:48:01 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

13 B. Liu, *et al.*, Retrosynthetic reaction prediction using neural sequence-to-sequence models, *ACS Cent. Sci.*, 2017, **3**(10), 1103–1113.

14 A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, Attention is all you need, *arXiv*, 2017, preprint, arXiv:1706.03762, DOI: 10.48550/arXiv.1706.03762.

15 P. Schwaller, T. Laino, T. Gaudin, P. Bolgar, C. Bekas and A. A. Lee, Molecular transformer - a model for uncertainty-calibrated chemical reaction prediction, *ACS Cent. Sci.*, 2019, **5**(9), 1572–1583.

16 G. Pesciullesi, P. Schwaller, T. Laino and J.-L. Reymond, Transfer learning enables the molecular transformer to predict regio- and stereoselective reactions on carbohydrates, *Nat. Commun.*, 2020, **11**(1), 4874.

17 R. Irwin, S. Dimitriadis, J. He and E. J. Bjerrum, Chemformer: a pre-trained transformer for computational chemistry, *Mach. Learn.: Sci. Technol.*, 2022, **3**(1), 015022.

18 D. Kreutter, P. Schwaller and J.-L. Reymond, Predicting enzymatic reactions with a molecular transformer, *Chem. Sci.*, 2021, **12**(25), 8648–8659.

19 D. Probst, M. Manica, Y. G. N. Teukam, A. Castrogiovanni, F. Paratore and T. Laino, Biocatalysed synthesis planning using data-driven learning, *Nat. Commun.*, 2022, **13**(1), 964.

20 W. Qian, X. Wang, Y. Kang, P. Pan, T. Hou and C.-Y. Hsieh, A general model for predicting enzyme functions based on enzymatic reactions, *J. Cheminf.*, 2024, **16**(1), 38.

21 Y. G. Nana Teukam, L. Kwate Dassi, M. Manica, D. Probst, P. Schwaller and T. Laino, Language models can identify enzymatic binding sites in protein sequences, *Comput. Struct. Biotechnol. J.*, 2024, **23**, 1929–1937.

22 G. B. Kim, *et al.*, Functional annotation of enzyme-encoding genes using deep learning with transformer layers, *Nat. Commun.*, 2023, **14**(1), 7370.

23 J. Capela, *et al.*, Comparative assessment of protein large language models for enzyme commission number prediction, *BMC Bioinf.*, 2025, **26**, 68.

24 N. Brandes, D. Ofer, Y. Peleg, N. Rappoport and M. L. Proteinbert, a universal deep-learning model of protein sequence and function, *Bioinformatics*, 2022, **38**(8), 2102–2110.

25 A. Rives, *et al.*, Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences, *Proc. Natl. Acad. Sci. U. S. A.*, 2021, **118**(15), e2016239118.

26 A. Radford and K. Narasimhan. *Improving language understanding by generative pre-training*. Semantic Scholar, 2018.

27 A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, *Improving language understanding by generative pre-training*, Semantic Scholar, 2019.

28 A. D. White, *et al.*, Assessment of chemistry knowledge in large language models that generate code, *Digital Discovery*, 2023, **2**(2), 368–376.

29 J. Wei *et al.*, Emergent abilities of large language models, *arXiv*, 2022, preprint, arXiv:2206.07682, DOI: 10.48550/arXiv.2206.07682.

30 P. Lewis *et al.*, Retrieval-augmented generation for knowledge-intensive nlp tasks, *arXiv*, 2021, preprint, arXiv:2005.11401, DOI: 10.48550/arXiv.2005.11401.

31 S. Pan, *et al.*, Unifying large language models and knowledge graphs: A roadmap, *IEEE Trans. Knowl. Data Eng.*, 2024, **36**(7), 3580–3599.

32 L. Wang, *et al.*, A survey on large language model based autonomous agents, *Front. Comput. Sci.*, 2024, **18**(6), 186345.

33 M. C. Ramos, C. J. Collison, and A. D. White. A review of large language models and autonomous agents in chemistry, *arXiv*, 2024, preprint, arXiv:2407.01603, DOI: 10.48550/arXiv.2407.01603.

34 D. A. Boiko, R. MacKnight, B. Kline and G. Gomes, Autonomous chemical research with large language models, *Nature*, 2023, **624**(7992), 570–578.

35 OpenAI *et al.*, Gpt-4 technical report, *arXiv*, 2024, preprint, arXiv:2303.08774, DOI: 10.48550/arXiv.2303.08774.

36 A. M. Bran, S. Cox, O. Schilter, C. Baldassari, A. D. White, and P. S. Chemcrow, Augmenting large-language models with chemistry tools, *arXiv*, 2023, preprint, arXiv:2304.05376, DOI: 10.48550/arXiv.2304.05376.

37 T. Guo *et al.*, What can large language models do in chemistry? a comprehensive benchmark on eight tasks, *arXiv*, 2023, preprint, arXiv:2305.18365, DOI: 10.48550/arXiv.2305.18365,.

38 K. M. Jablonka, P. Schwaller, A. Ortega-Guerrero and B. Smit, Leveraging large language models for predictive chemistry, *Nat. Mach. Intell.*, 2024, 1–9.

39 K. M. Jablonka, *et al.*, 14 examples of how llms can transform materials science and chemistry: a reflection on a large language model hackathon, *Digital Discovery*, 2023, **2**(5), 1233–1250.

40 Z. Zheng, *et al.*, Shaping the water-harvesting behavior of metal–organic frameworks aided by fine-tuned gpt models, *J. Am. Chem. Soc.*, 2023, **145**(51), 28284–28295.

41 https://openai.com/.

42 Y. Fang *et al.*, Mol-instructions: A large-scale biomolecular instruction dataset for large language models, *arXiv*, 2024, preprint, arXiv:2306.08018, DOI: 10.48550/arXiv.2306.08018.

43 B. Yu, F. N. Baker, Z. Chen, X. Ning, and H. Sun, Llasmol: Advancing large language models for chemistry with a large-scale, comprehensive, high-quality instruction tuning dataset, *arXiv*, 2024, preprint, arXiv:2402.09391, DOI: 10.48550/arXiv.2402.09391.

44 T. B. Brown *et al.*, Language models are few-shot learners, *arXiv*, 2020, preprint, arXiv:2005.14165, DOI: 10.48550/arXiv.2005.14165.

45 T. Z. Zhao, E. Wallace, S. Feng, D. Klein and S. Singh, Calibrate before use: Improving few-shot performance of language models, *Proceedings of the 38th International Conference on Machine Learning*, 2021, PMLR 139:12697-12706.

46 J. Wei *et al.*, Chain-of-thought prompting elicits reasoning in large language models, *arXiv*, 2023, preprint, arXiv:2201.11903, DOI: 10.48550/arXiv.2201.11903.

47 M. Mosbach, T. Pimentel, S. Ravfogel, D. Klakow, and Y. Elazar, Few-shot fine-tuning vs. in-context learning: A

© 2025 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 30 October 2025. Downloaded on 2/8/2026 10:48:01 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

fair comparison and evaluation, *arXiv*, 2023, preprint, arXiv:2305.16938, DOI: **10.48550/arXiv.2305.16938**.

48 Z. Han, C. Gao, J. Liu, J. Zhang, and S. Q. Zhang, Parameter-efficient fine-tuning for large models: A comprehensive survey, *arXiv*, 2024, preprint, arXiv:2403.14608, DOI: **10.48550/arXiv.2403.14608**.

49 L. Xu, H. Xie, S.-Z. J. Qin, X. Tao, and F. L. Wang, Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment, *arXiv*, 2023, preprint, arXiv:2312.12148, doi: DOI: **10.48550/arXiv.2312.12148**.

50 D. Weininger, Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules, *J. Chem. Inf. Comput. Sci.*, 1988, **28**(1), 31–36.

51 M. Ganter, T. Bernard, S. Moretti, J. Stelling and M. Pagni, Metanetx.org: a website and repository for accessing, analysing and manipulating metabolic networks, *Bioinformatics*, 2013, **29**, 815–816.

52 R. Alcántara, *et al.*, Rhea – a manually curated resource of biochemical reactions, *Nucleic Acids Res.*, 2012, **40**, D754–D760.

53 D. S. Wishart, *et al.*, Pathbank: a comprehensive pathway database for model organisms, *Nucleic Acids Res.*, 2020, **48**, D470–D478.

54 S. Ida, C. Antje and S. Dietmar, Brenda, enzyme data and metabolic information—pubmed, *Nucleic Acids Res.*, 2002, **30**, 47–49.

55 E. J. Hu *et al.*, Low-rank adaptation of large language models, *arXiv*, 2021, preprint, arXiv:2106.09685, DOI: **10.48550/arXiv.2106.09685**.

56 T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, Qlora: Efficient finetuning of quantized llms, *arXiv*, 2023, preprint, arXiv:2305.14314, DOI: **10.48550/arXiv.2305.14314**.

57 A. Grattafiori *et al.*, The llama 3 herd of models, *arXiv*, 2024, preprint, arXiv:2407.21783, DOI: **10.48550/arXiv.2407.21783**.

58 Inc. Daylight Chemical Information Systems, Daylight theory manual: Fingerprints.

59 T. Chen and C. Guestrin, Xgboost: A scalable tree boosting system, *Proceedings of the 22nd ACM SIGKDD*, 2016.