École polytechnique fédérale de Lausanne

Master project

Master in Computational Science and Engineering

An AI toolkit to quantify chemical sustainability

Author:
Adrián Sager La Ganga

EPFL Supervisor:
Martin Jaggi
IBM Research Supervisor:
Amol Thakkar

Contents

1 Acknowledgements 2

2 Introduction 3

3 Background 4
- 3.1 Nomenclature 4
- 3.2 Chemical reaction representation and data availability 4
  - 3.2.1 Data imbalance and classification metrics 4
  - 3.2.2 Embedding-based SMILES fingerprints vs traditional fingerprints 6
- 3.3 Literature on data-driven sustainable chemistry 6
- 3.4 Literature on Computer-Aided Synthesis Planning (CASP) 7
- 3.5 Uncertainty quantification (UQ) and explainable AI 8

4 Models and Methods 10
- 4.1 Pistachio+ECREACT Dataset 10
- 4.2 Multi-Head Attention and BERT 12
- 4.3 Enzymatic reaction classifier 12
  - 4.3.1 BERT classifier 13
  - 4.3.2 Conclusion 16
- 4.4 Solvent classifier for reaction sustainability 17
  - 4.4.1 Extracting all solvents from Pistachio 17
  - 4.4.2 Solvent sustainability distribution in Pistachio 18
  - 4.4.3 Intersection with reaction classes 20
  - 4.4.4 Pistachio+ECREACT solvent extraction 22
  - 4.4.5 Solvent sustainability definition 24
  - 4.4.6 XGBoost baseline 24
  - 4.4.7 BERT hyper-parameter search 25
  - 4.4.8 Conclusion 28
- 4.5 Sustainability metrics 28
  - 4.5.1 AI-based metrics 28
  - 4.5.2 Reaction SMILES augmentation 29
  - 4.5.3 Validating AI metrics 30
  - 4.5.4 Uncertainty Quantification (UQ) for reactions in multiple classes 32
  - 4.5.5 Atom Economy 33
  - 4.5.6 Conclusion 33
- 4.6 Limitations and AI interpretability tools 34
- 4.7 Computer-Aided Synthesis Planning (CASP) Use Case 36
  - 4.7.1 Monte-Carlo tree search for CASP 37
  - 4.7.2 Cost-effective scoring of pathways with the Route Cost scorer 37
  - 4.7.3 Pathway sustainability scoring 38
  - 4.7.4 Assessing metric influence on pathway generation 39
  - 4.7.5 Conclusion 41

5 Results and Discussion 42

6 Conclusion 43

A Figures 51

B Random choice algorithm 52

Abstract

The chemical industry is moving toward automated tools to accelerate sustainable synthesis planning. One of the main challenges to achieving this goal is the definition of helpful reaction sustainability metrics, as they require the prediction of complex reaction conditions. Moreover, since sustainability is a subjective and multi-faceted problem, each chemist will use different metrics for each use case. In this work, we propose a simple yet scalable toolkit as an initial step toward a complete framework for chemical sustainability prediction and integration containing three metrics: *enzyme-sustainability*, *solvent-sustainability*, and a modified *Atom Economy (AE)*. We combine the *Pistachio* and *ECREACT* chemical reaction datasets and achieve an F1 score of 0.986 on enzyme-sustainability prediction and an F1 score of 0.584 on solvent-sustainability prediction using fine-tuned BERT models. With these models, we define *AI-metrics* for sustainability by employing uncertainty quantification (UQ) on the models' predicted likelihood that the reaction is sustainable. By applying these metrics, we enhance a Monte Carlo tree search (MCTS) algorithm from the *AiZynthFinder* retrosynthesis planning tool and demonstrate that users can tune a trade-off between increasing enzyme-sustainability and lowering cost-effectiveness on generated chemical pathways.

1 Acknowledgements

This publication was created as part of the NCCR Catalysis (grant number 180544) Young Talents Fellowship, a National Centre of Competence in Research funded by the Swiss National Science Foundation. My special gratitude goes to NCCR Catalysis for their financial support, which has allowed me to perform my research work in Switzerland, and their wonderful community.

I would also like to particularly thank my supervisor at IBM Research Zürich, Dr. Amol Thakkar, for his invaluable expert guidance during our weekly meetings and extra time spent supporting the project, even when the time was not on our side.

Additionally, this work would not have been possible without the guidance of Prof. Martin Jaggi, alongside his PhD assistant Jean-Baptiste Cordonnier, and the fantastic RXN for Chemistry team from Teodoro Laino at IBM Research Zürich, especially those who directly assisted with the project: Alain Vaucher, Aleksandros Sobczyk, Alessandra Toniato, Alessio Bray, Anubhab Chakraborty, Jannis Born, Miruna Cretu, Nicolas Deutschmann, and Teodoro Laino.

2 Introduction

There is a transformation toward sustainable practices in the chemical field to decrease the negative impact of industries on the environment and evolve from non-renewable fossil fuels. For this goal, automated tools are required to accelerate sustainable development. Related to sustainability, chemical greenness has been a topic of discussion in the literature until recently, where the focus has shifted towards building self-sustaining and environmentally-friendly supply chains, which requires cost-effective planning of chemical reaction routes. However, sustainability is difficult to quantify due to its multi-faceted nature and subjective definitions, which usually results in a lack of consensus on what constitutes a sustainable reaction or compound. In particular, sustainability valuation tools should consider the ecological and socio-economic context the reactions and participating compounds influence (Weber *et al.* (2021); Weber (2022); Sheldon (2018)).

The main contribution of this project is a toolkit with AI-based sustainability metrics, which aims to be a first step toward a unifying framework for sustainability analysis of chemical reaction processes.

This framework exploits the recent surge of data-driven chemistry and performance improvements in natural language processing (NLP) to predict chemical agents and catalysts relevant to our sustainability discussion. Specifically, it includes the following metrics to quantify sustainability in a chemical reaction: potential for biocatalysis, the potential for using renewable and low-hazard solvents, and Atom Economy (AE). We propose to use the confidence of two deep learning classifiers, based on uncertainty quantification (UQ), to score the first two metrics. These models are trained on a dataset of 3.7M reactions which combines synthetic reactions from Pistachio (Nextmove Software (2021b)) and enzymatic reactions from ECREACT (Probst *et al.* (2022a)), from which we extract solvent information to train our solvent predictors. We interpret the predictions in the enzyme classifier and discover an adversarial attack using a wildcard token. We then make the model robust to this attack by randomly substituting some tokens with the wildcard token during training.

With these single-reaction scoring metrics, we can generalize to pathway scoring and integrate our AI-metrics into existing Computer-Aided Synthesis Planning (CASP) software. We will present sustainability improvements in this use-case using the AiZynthFinder (Genheden *et al.* (2020b)) retrosynthesis planning tool where the chemist can specify the weighting of each sustainability aspect.

3 Background

Data-driven chemistry using artificial intelligence (AI) has risen in the field of computer-aided drug design (CADD) due to its improved predictive capability compared to more traditional approaches, which manually encode reactions as templates (Thakkar *et al.* (2020); Szymkuć *et al.* (2016); Segler *et al.* (2018)).

However, data for some chemical properties in reactions is limited, especially on sustainability (Weber *et al.* (2021)).

In what follows, we provide a background on these and the other main topics relevant to this work.

3.1 Nomenclature

Before discussing the literature, we should specify the standard naming conventions used throughout this project.

A chemical reaction describes a chemical transformation from one or more *reactant* molecules to one or more *product* molecules with the assistance of *reagents*. These molecules, such as solvents and catalysts, are not consumed in the reaction process. Reactants and reagents in conjunction form the *precursors* of the reaction. A *feasible* reaction is that which may be successfully performed in practice.

Enzymes are natural catalysts, or *biocatalysts*.

Finally, performing one or more reactions in succession is called *synthesis* or *forward synthesis*, while *retrosynthetic analysis* or *retrosynthesis* is the task of discovering the precursors that can synthesize a target product. *Multi-step retrosynthesis* is the task of generating the tree of reactions that can synthesize a target product from some starting molecules. A tree of reactions that synthesizes some product is also called a *synthesis plan*, a reaction *pathway*, or a reaction *route*.

3.2 Chemical reaction representation and data availability

The type of reaction data representation is a fundamental choice in data-driven chemistry. There are multiple ways of representing reactions (David *et al.* (2020)), such us reaction SMILES (Weininger (1988); Weininger *et al.* (1989)), SMIRKS (Daylight Chemical Information Systems, Inc. (2022)), RInChI (Grethe *et al.* (2018)), the Condensed graph of reaction (CGR, Varnek *et al.* (2005)), Bond electron matrices (BE-matrix, Gasteiger and Jochum (1978)), HORACE (Rose and Gasteiger (1994)), InfoChem CLASSIFY (InfoChem GmbH (2002)), and traditional reaction fingerprints (Schneider *et al.* (2015); Probst *et al.* (2022c)). Due to our data availability, in this work we use the reaction SMILES representation, which is a common format available in the datasets mentioned below.

AI methods achieve state-of-the-art performance for multiple tasks in chemistry owing to the large amount of reactions in available datasets (Thakkar *et al.* (2020)).

Some of the proprietary datasets used in the literature include CASREACT (> 150M reactions, Blake and Dana (1990); American Chemical Society (2023)), Reaxys (57M reactions, Elsevier Limited (2023)), Pistachio (> 13M reactions, Nextmove Software (2021b)), and SPRESIweb (4.6M reactions, InfoChem (2019)).

Two examples of publicly available datasets are USPTO (> 3.3M reactions, Lowe (2017)), and the Open Reaction Database (ORD) (> 2.2M reactions, Kearnes *et al.* (2021)).

In addition, there is a limited amount of datasets to train and benchmark models on multi-step retrosynthesis. Two examples include ASKCOS (Mo *et al.* (2021)), and PaRoutes (Genheden and Bjerrum (2022)).

Data related to sustainability is particularly lacking in the literature (Weber *et al.* (2021)). However, for enzyme reactions there are some publicly available datasets like ECREACT (62'222 reactions, Probst *et al.* (2022a)), and KEGG REACTION (Kanehisa Laboratories (2023)).

The datasets most easily accessible and ready to use for this work were Pistachio and ECREACT, which we present in Section 4.1.

3.2.1 Data imbalance and classification metrics

Due to the large amount of chemical reactions, reaction class datasets may have a large data imbalance, especially when we combine enzymatic with non-enzymatic reactions (Nextmove Software (2021a); Weber *et al.* (2021)).

Previous work has used the Synthetic Minority Oversampling Technique (SMOTE, Chawla *et al.* (2002)) to deal with data imbalance on SMILES (Hung *et al.* (2022), Mahmud *et al.* (2019)). However, when there is extreme imbalance in our dataset and a long tail, this technique would over-sample noisy data, and under-sampling may be preferred in this case as it has been previously shown in Drummond *et al.* (2003). To solve this issue, a weighting technique has been proposed based on the *effective number of samples* per-class (Cui *et al.* (2019)), which has been shown to perform best in average class-wise accuracy in the context of object detection (Phan and Yamamoto (2020)).

Data imbalance also affects the metrics we should use to evaluate our models.

In binary classification tasks, accuracy and F1 score are the most common metrics (Chicco and Jurman (2020)). The F1 score is defined as the harmonic mean of recall and precision,

$$\mathrm{F1} = \frac{2 \cdot \mathrm{TP}}{2 \cdot \mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$$

Where TP are the number of true positives, FP the number of false positives and FN the number of false negatives. Most classification metrics can be computed with TP, FP, FN, and the number of true negatives, TN (Hossin and Sulaiman (2015)).

From its definition, the F1 score is more sensitive to false positives and false negatives than accuracy. This is important in sustainability prediction, as we want our models to have a low amount of false positives, where the reaction is predicted as sustainable when it is not, and a low amount of false negatives, where our model missed that a reaction is sustainable. The F1 score is also more appropiate for imbalanced datasets since it is independent on the number of true negatives. Thus, a model which always predicts that a reaction is non-sustainable will achieve a high accuracy due to the small amount of sustainable reactions in the dataset whereas the F1 score will be zero. For these reasons, when evaluating our models we will present both accuracy and the F1 score but decide which model performs "better" according to the F1 score.

For multi-class classification, we can generalize the classification metrics by taking the *micro* or *macro* average of the score over the classes (Grandini *et al.* (2020)). In *micro* averaging, we first sum TP, TN, FP, and FN individually for all classes and then apply our original formula. Thus, the micro F1 score is defined as,

$$\text{Micro F1} = \frac{2 \cdot \sum_{k \in C} \mathrm{TP}_k}{2 \cdot \sum_{k \in C} \mathrm{TP}_k + \sum_{k \in C} \mathrm{FP}_k + \sum_{k \in C} \mathrm{FN}_k}$$

Where C is the set of classes.

In *macro* averaging, we first calculate the classification score separately for each class and then average over the classes,

$$\text{Macro F1} = \frac{1}{|C|} \sum_{k \in C} \frac{2 \cdot \mathrm{TP}_k}{2 \cdot \mathrm{TP}_k + \mathrm{FP}_k + \mathrm{FN}_k}$$

Therefore, micro averaging represents a global accuracy that does not differentiate between classes while macro averaging represents a class-balanced quantity.

When classifying reactions, Schwaller *et al.* (2021b) use two other metrics: the confusion entropy (CEN,Wei *et al.* (2010)), and the overall Matthews correlation coefficient (MCC, Matthews (1975); Gorodkin (2004)). These metrics are meant for imbalanced **single-label** multi-class datasets. Given the confusion matrix $M_{i,j}$ where $i \in C$ is the true label and $j \in C$ is the predicted label, the CEN is defined as,

$$\begin{aligned} P^{l}_{i,j} &= \frac{M_{i,j}}{\sum_{k \in C} \left(M_{l,k} + M_{k,l}\right)} \\ P_j &= \frac{\sum_{k \in C} \left(M_{j,k} + M_{k,j}\right)}{2 \sum_{k,l \in C} M_{k,l}} \\ \mathrm{CEN} &= -\sum_{j \in C} P_j \sum_{k \in C, k \neq j} \left(P^{j}_{j,k} \log_{2(|C|-1)}\left(P^{j}_{j,k}\right) + P^{j}_{k,j} \log_{2(|C|-1)}\left(P^{j}_{k,j}\right)\right) \end{aligned}$$

And the MCC is defined as,

$$\mathrm{MCC} = \frac{\sum_{i,j,k \in C} \left(M_{i,i} M_{k,j} - M_{j,i} M_{i,k}\right)}{\sqrt{\left[\sum_{i \in C} \left(\sum_{j \in C} M_{j,i}\right) \left(\sum_{k,l \in C, k \neq i} M_{l,k}\right)\right] \cdot \left[\sum_{i \in C} \left(\sum_{j \in C} M_{i,j}\right) \left(\sum_{k,l \in C, k \neq i} M_{k,l}\right)\right]}}$$

A lower CEN and a higher MCC are better.

Since we will classify our reactions using the same dataset as Schwaller *et al.* (2021b) with additional enzyme information and similar methods, we will present these metrics in the reaction classification results alongside the micro and macro F1 scores (Section 4.3.1).

Finally, we will deal with the **multi-label** and multi-class task of classifying participating solvents in a reaction. A recent paper proposes the Multi-Label Confusion Matrix (MLCM, Heydarian *et al.* (2022)), an extension of the traditional single-label confusion matrix which is more representative than the traditional multi-label confusion matrix used to calculate multi-label metrics, implemented for example in Scikit-Learn's `multilabel_confusion_matrix`. This is partially because traditional multi-label and multi-class metrics do not take into account the case where the model predicts that a sample has no labels. Furthermore, traditional precision, recall, and F-score metrics provide inflated results since they isolate their focus on each label individually, ignoring the false negative and false positive relationship between labels, so MLCM provides a new definition for these metrics which overcome these problems. Moreover, the MLCM is equivalent to the traditional confusion matrix when dealing with a single-label dataset and prediction. We will use MLCM-derived metrics to compare models in the classification task of multi-label reaction solvent prediction (Section 4.4).

3.2.2 Embedding-based SMILES fingerprints vs traditional fingerprints

Previous work has used traditional molecule fingerprints for classification tasks (Chandrasekaran *et al.* (2020); Gao *et al.* (2018); Walker *et al.* (2019)). These fingerprints encode molecules as bit-vectors where each feature in the vector roughly indicates if a certain structure is present in the molecule. We focus on the Atom-Pairs (AP) fingerprint (Carhart *et al.* (1985)), which is calculated by taking a numerical representation of the path distance and properties of every pair of heavy atoms, and hashing it to get the index of the fingerprint vector feature to set to 1.

For reactions, we will use the following two fingerprints: the difference atom-pair fingerprint (AP3, Schneider *et al.* (2015)), and the recently published differential reaction fingerprint (DRFP, Probst *et al.* (2022c)). On the one hand, the AP3 reaction fingerprint is the difference of the sum of AP fingerprints between products and reactants, where the AP fingerprints are calculated for atom-pairs with a maximum path length of three. On the other hand, the DRFP fingerprint computes all circular substructures up to radius three in the molecules' graph representation, takes the symmetric difference of the subtructures between the products and precursors, and hashes the resulting substructures into numerical values which, after a modulo operation, indicate the features in the fingerprint vector to set to 1. DRFP fingerprints are sparse bit-vectors, while AP3 are sparse integer vectors.

However, it has been shown that the latent-space of transformer models pre-trained on masked language modelling and then fine-tuned on a reaction classification task has more expressive power than traditional reaction fingerprints (Schwaller *et al.* (2021b)). It has also been shown that attention-based architectures can extract reaction grammar, avoiding the need for expert-crafted features (Schwaller *et al.* (2021a)).

Furthermore, previous work has shown that using SMILES features lead to better performance since they allow for easy data augmentation (Manica *et al.* (2019); Kimber *et al.* (2021)).

In addition, traditional reaction fingerprints have limitations. For example, AP3 fingerprints cluster according to the Dice similarity (Dice (1945); Schneider *et al.* (2015)), however this representation is very sensitive to reaction reagents and it also does not cluster reactions perfectly, which means that some information in the reaction useful for reaction classification may be lost in the process of compressing the SMILES into a lower-entropy representation. On the other hand, feature learning is a well-known property in machine learning, where models learn the proper, or most effective, feature representation for the given task. This is due to the universal approximation theorem (Cybenko (1989); Leshno *et al.* (1993); Pinkus (1999); Zhou (2020); Kratsios and Papon (2022)).

For solvent prediction we will study the performance difference when using reaction fingerprints versus a SMILES-based deep neural network (Section 4.4). Reaction fingerprint vectors can be seen as tabular data, as each feature in the vector corresponds to a different property in the reaction. A recent study has discussed the necessity for deep neural networks on tabular data, arguing that XGBoost (Chen and Guestrin (2016)), a gradient boosting approach, provides better performance with less tuning (Shwartz-Ziv and Armon (2022)). Thus, we will only study the performance of XGBoost models when using reaction fingerprints as the input.

3.3 Literature on data-driven sustainable chemistry

Weber *et al.* (2021) and Weber (2022) provide an extensive review and future avenues for sustainability in data-driven chemistry. One of the main aspects of sustainability assessment is the necessity of quantifiable metrics (Sheldon (2018)). Since sustainability is a multi-faceted problem, many metrics have been proposed and, in fact, the Sustainable Development Goals (SDGs) framework from the United Nations define 231 different sustainability aspects that should be measured (Weber *et al.* (2021)). In practice, to have a complete picture on sustainability we would need to account for monetary, supply chain, environmental impact, and energy efficiency considerations, as well as their synergies and trade-offs. This is a complex system of relationships that no single metric could thoroughly quantify. Thus, in this work we propose a general framework that focuses on sustainability quantification as the potential that a reaction has some desirable sustainable property, which allows the chemist to weight the different properties and subjectively decide which combination makes a reaction sustainable for each use-case. As an example, we focus on two potentially sustainable aspects of a reaction: biocatalysis, and the usage of low-hazard and renewable solvents. We will propose a general AI-based metric approach in Section 4.5.1, which enables our framework to be expanded in the future with models that predict other sustainability aspects in a reaction.

Biocatalysis, which we also refer to as *enzyme-sustainability*, plays an important role in enabling reaction sustainability since enzymatic, or *biocatalyzed*, reactions can result in more cost-efficient reaction synthesis plans (Weber *et al.* (2021); Sheldon (2018)) and because enzymes are nontoxic, produce minimal byproducts, and are renewable (Shoda *et al.* (2016); Kobayashi *et al.* (2001); Puskas *et al.* (2009); Cheng and Gross (2010)). There is a large body of work on applying machine learning in enzyme engineering (Mazurenko *et al.* (2019)). However, the literature is more limited on whether an enzyme can catalyze a reaction or not. There is previous work on this task where the authors use a Gaussian process regression (GPR) model trained on BRENDA (Mellor *et al.* (2016); Schomburg *et al.* (2002)). Another example is Probst *et al.* (2022a), where the authors present the ECREACT dataset, which combines four publicly available datasets including BRENDA, and use the Molecular Transformer (Schwaller *et al.* (2019)) on SMILES to achieve a top-1 accuracy of 49.6% in forward synthesis prediction.

Sustainability on solvents is more difficult to define. Multiple solvent guides from industry and academia have provided guidelines for what constitutes a *hazardous* solvent from a practical point of view, each of them with different *hazardous solvent* definitions and different sets of hazard levels (ETH Zurich (2008); Hargreaves *et al.* (2008); Curzons *et al.* (1999)). Byrne *et al.* (2016) provides an extensive review of these solvent guides and combines them to provide a consensus on the hazard levels for 51 solvents. They base their final ranking mainly on the solvents present in the CHEM21 survey (Prat *et al.* (2016)), which ranks solvents according to safety, health and environment criteria. Additionally, they also classify these solvents according to whether they can come from a renewable source. The authors conclude that "*there is no need for more general purpose solvent selection guides of the familiar format because they are no longer providing any significant advancement in this field*".

There are multiple approaches for solvent prediction using *shallow* learning on fingerprints (AP3 for reaction fingerprints), including multilayer perceptrons (MLPs), *k*-nearest neighbors (*k*-NN), and support vector machines (SVMs) (Chandrasekaran *et al.* (2020); Walker *et al.* (2019); Gao *et al.* (2018)).

Additionally, there is a software for solvent sustainability prediction implemented in Java, SUSSOL (Sels *et al.* (2020)). SUSSOL is a proprietary software, but there is an open source GitHub repository with reduced functionality (De Smet (2020)). The solvent sustainability assumptions in SUSSOL follow the CHEM21 selection guide, and solvent sustainability prediction is achieved using the Self-Organizing Map (SOM, Kohonen (1982)) clustering algorithm.

3.4 Literature on Computer-Aided Synthesis Planning (CASP)

Computer-Aided Synthesis Planning (CASP) has the objective of providing tools to analyze reaction conditions in pathways, suggest optimizations and synthesis plans, and help determine whether a reaction is feasible. In one aspect of CASP, the chemist expert utilizes automated retrosynthesis tools to plan and discover pathways that can synthesize target compounds from a set of starting molecules. In Section 4.7, we will integrate our metrics into the CASP software AiZynthFinder (Genheden *et al.* (2020b)) to search for sustainable synthesis plans. This software uses *template-based* models for retrosynthesis, which classify which known chemical transformations may be applied to generate some target product, whereas *template-free* methods in CASP use models which in principle can predict novel reactions (Sun and Sahinidis (2022)). For example, IBM's RXN platform implements template-free retrosynthesis by employing sequence-to-sequence Transformer models (Vaswani *et al.* (2017)) and a strategy to explore the chemical reaction space as a hyper-graph (Schwaller *et al.* (2020)). Another example of a template-based CASP tool is ASKCOS (Coley *et al.* (2019); Connor Coley and Mo (2021)).

Tree-structured long short-term memory models (tree-LSTM, Tai *et al.* (2015)) are used in the literature to predict pathway properties. In Mo *et al.* (2021), the authors train the model to predict whether a pathway is *patent-like* or generated. Whereas in Genheden *et al.* (2022), the authors use the same model to predict the tree edit distance (TED, Bille (2005); Genheden *et al.* (2021)) between two pathways.

To generate pathways, one approach is to use Monte Carlo tree search (Kocsis and Szepesvári (2006); Coulom (2007)). MCTS is a general search optimization technique for games or planning tasks where the objective space can be modelled by a Markov decision process (MDP) (Browne *et al.* (2012); Sutton and Barto (2018)). MCTS in combination with deep reinforcement learning (RL) has been shown to achieve state-of-the-art performance in multiple game-related tasks (Świechowski *et al.* (2022)). MCTS is used in AiZynthFinder for multi-step retrosynthesis and it is based on the work by Segler *et al.* (2018).

Another approach to generate pathways is inspired by the A* search algorithm (Hart *et al.* (1968)), called Retro* (Chen *et al.* (2020)). It has been recently used in combination with a reaction knowledge base to generate feasible and cost-efficient pathways (Jeong *et al.* (2022)).

Among these two approaches, MCTS is the best prediction method in terms of route quality and route diversity according to the PaRoutes benchmark (Genheden and Bjerrum (2022)).

Another task related to CASP is that of finding the most cost-efficient set of pathways that synthesize a collection of target molecules while minimizing the number of starting compounds used. Gao *et al.* (2020a) tackle this problem using MCTS (Segler *et al.* (2018)) and a mixed-integer linear programming (MILP) solver on 127 target molecules extracted from the WHO Essential Medicines List (EML) from 1977 (WHO Expert Committee (1977)). We will not undertake this challenge in this work, however we consider that this task is relevant for sustainable synthesis planning as it tries to minimize wastes and maximize cost-effectiveness, which are two goals in sustainable chemistry (Weber *et al.* (2021); Weber (2022)).

3.5 Uncertainty quantification (UQ) and explainable AI

We want to minimize the risk of false positives and false negatives during sustainable synthesis planning, since this would introduce non-sustainable reactions which may go unnoticed by the expert or we may miss feasible and useful sustainable reactions. This is especially troublesome in our case since machine learning models are black-boxes and the reasoning behind their outcomes is difficult to interpret, their predictions are noisy, and they are prone to inference errors, so using their output as-is to define AI-metrics is unreliable (Abdar *et al.* (2021)).

These problems are tackled in the field of uncertainty quantification (UQ), where one tries to extract an accurate and reliable estimate for uncertainty in machine learning predictions (Abdar *et al.* (2021); Hüllermeier and Waegeman (2021)). This notion of uncertainty or confidence in a model's prediction can provide insights into the inductive biases in the model.

There are two types of uncertainty in probabilistic modelling: *aleatoric* and *epistemic* uncertainty. Aleatoric, or statistical, uncertainty is due to inherent noise or inconsistencies in the data and represents the minimum uncertainty any model could achieve; while epistemic, or systematic, uncertainty is due to noise or expressive limitations in the model (Hora (1996); Der Kiureghian and Ditlevsen (2009)). There exist very simple methods to estimate both kinds of uncertainty in deep learning, namely Monte Carlo Dropout (MC Dropout, Gal and Ghahramani (2016)) and test-time data augmentation (Ayhan and Berens (2018)). MC Dropout esitmates epistemic uncertainty and only requires the presence of Dropout layers in the network, while test-time data augmentation estimates aleatoric uncertainty and only requires that the input data is able to be randomly augmented, which is true for reaction SMILES. We will present and explain both methods in Section 4.5.1 and 4.5.2. In contrast, there are more complex UQ techniques for epistemic uncertainty in deep learning under the umbrella of Bayesian deep learning (BDL, Wang and Yeung (2020)), where the parameters of the network are represented as random variables. These models can be trained through Bayesian optimization in frameworks such us BoTorch (Balandat *et al.* (2020)). Another specific example are *deep ensembles* (Lakshminarayanan *et al.* (2017)), a simple approach where combining the output of multiple deep neural networks that have similar accuracy can boost the overall performance and reduce epistemic uncertainty.

UQ will be the basis for our AI-metrics, which we define in Section 4.5.1.

However, UQ alone cannot help us infer the causal connection between the inputs and outputs of a model. To achieve that, the field of *explainability* in AI investigates methods to interpret the internal functioning of AI models (Gilpin *et al.* (2018); Linardatos *et al.* (2020)).

There are multiple methods for the interpretability of the input's influence on the output. One approach in attention-based networks (Vaswani *et al.* (2017)) is to look at attention scores, which has been used before to explain causal inference in the prediction of BERT (Devlin *et al.* (2018)) on reaction classification (Schwaller *et al.* (2021b)). However, it is argued that saliency methods are better suited for the explainability of the tokens influence on the classification prediction (Bastings and Filippova (2020)). One saliency method is that of Integrated Gradients (Sundararajan *et al.* (2017)), implemented in the Captum Python package (Kokhlikyan *et al.* (2020)). This method will be used in Section 4.3.1 to discover an adversarial attack on our enzyme reaction classifier and in Section 4.6 to demonstrate that attention can be misleading on BERT models when trying to rationally connect their predictions to the inputs.

4 Models and Methods

This work introduces a framework that allows chemists and researchers to introduce AI-based reaction sustainability scoring on drug discovery workflows.

In this section, we will review the sustainability objectives of this framework. In particular, we look at the approaches taken to predict enzymatic reactions and solvent sustainability, as well as describe a novel AI-based method for scoring samples using uncertainty quantification (UQ) and, finally, a use case where we integrate the package into AiZynthFinder (Genheden *et al.* (2020b)) to predict sustainable reaction pathways.

4.1 Pistachio+ECREACT Dataset

The two sustainability aspects we look at in this work are biocatalysis prediction and solvent renewability prediction. Our first main challenge is to find a dataset that contains this reaction information as there is limited data on sustainability (Weber *et al.* (2021)). We focus our study on datasets that allow natural language processing (NLP) models to be used. This can be achieved thanks to the *simplified molecular-input line-entry system* (SMILES, Weininger (1988); Weininger *et al.* (1989)) specification for molecules and reactions, which is a text representation that includes atom and structural information. SMILES can completely describe a reaction, but they do not encode the 3D structure of the molecules, and they are not unique in general; thus, a reaction can have multiple SMILES representations.

For our sustainability purposes, we used the following datasets:

1. *Pistachio* (Nextmove Software (2021b)): A commercial dataset of synthetic reactions automatically extracted from patent data. It contains 3.7M reaction samples with SMILES and reaction class information. Pistachio relies on LeadMine to text-mine patent data (Lowe and Sayle (2015)). The reaction classes in Pistachio are classified using NameRxn (Nextmove Software (2021a)), a software that classifies roughly 1'000 different named reactions based on known reaction mechanisms and transformation rules. It defines 12 main reaction types called *superclasses* and a class for unrecognized reactions by the software, 0.0. The original reaction types were first described in Carey *et al.* (2006). Pistachio is continuously updated, but for this work, we utilized version 210403 from 2021 since we were provided with the newer version 220406 from 2022 late in the project. The newer version 220406 will be used in Section 4.4 to study Pistachio's solvent distribution.

2. *ECREACT* (Probst *et al.* (2022b)): An open source dataset of enzyme-catalyzed reactions. It contains 62'222 reaction samples in SMILES format and reaction class information according to the enzymes participating in the reaction. Reactions are classified according to the Enzyme Commission (EC) numbers, which was implemented in 1955 by the International Commission of Enzymes—now the International Union of Biochemistry and Molecular Biology, IUBMB (Webb *et al.* (1992)). EC numbers classify the enzyme groups according to their effect on the reaction. Enzyme-catalyzed reactions and the accompanying EC numbers were retrieved from four databases, namely Rhea, BRENDA, PathBank, and MetaNetX (Alcántara *et al.* (2012); Schomburg *et al.* (2002); Wishart *et al.* (2020); Ganter *et al.* (2013)), and merged into a new data set, named ECREACT. This dataset contains 6'289 different EC classes. Furthermore, the contained SMILES sometimes have a special wildcard "*" token, which indicates that the same reaction transformation works with multiple different structures bonded at that location. These wildcard tokens are only present in SMILES from the Rhea and MetaNetX datasets, which correspond to 42% of ECREACT.

Figure 1 contains detailed examples for each dataset. We standardized the SMILES in Pistachio using RXN chemutils (IBM RXN team (2022)), which removes atom mapping information.

We combined both datasets into a single one, *Pistachio+ECREACT*. Since ECREACT contains the reaction class label inside the reaction SMILES, it was processed to remove this information and follow the format of Pistachio.

We should note that reactions in these datasets are hierarchically classified. For example, a reaction of the level three class 1.2.1 is also part of the level two class 1.2.x and, finally, the superclass 1.x.x (see Figure 1). ECREACT has four levels. Thus, a reaction of class 2.8.1.1 is also in class 2.8.1.x, 2.8.x.x, and 2.x.x.x. We will include the prefix "EC." on all ECREACT classes to differentiate them from the Pistachio classes; thus, "2.8.1.1" becomes "EC.2.8.1.1".

46% of Pistachio classes have less than 100 samples at all three levels (like 1.2.1, 1.2.4), while 54% of ECREACT classes have less than 100 samples at level 2 (like EC.1.1.x.x, EC.1.14.x.x). Therefore, we limit our study to class level two for Pistachio (like 1.2.x, 1.3.x) and level one for ECREACT (like EC.1.x.x.x,

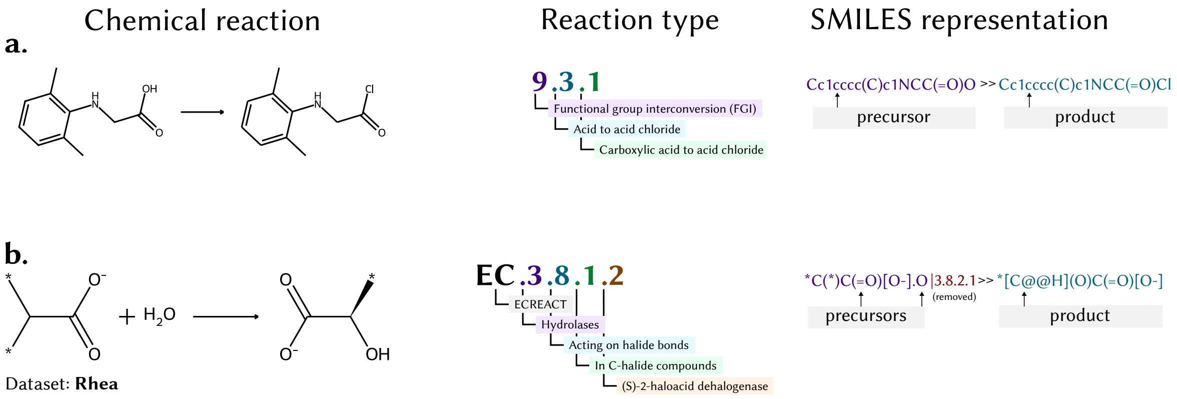


Figure 1: **a**, Pistachio reaction example. **b**, ECREACT reaction example. ECREACT SMILES contains the EC number in the string, so we removed this information when combining Pistachio+ECREACT. Unlike Pistachio, ECREACT reactions may contain a wildcard "*" token

EC.2.x.x.x) to be fine-grained enough to learn subtle differences while having enough samples to do so. The final number of classes is 83.

When checking the class distribution (up to class level 2 in Pistachio and class level 1 in ECREACT), we appreciate a significant imbalance and long tail (see Figure 2).

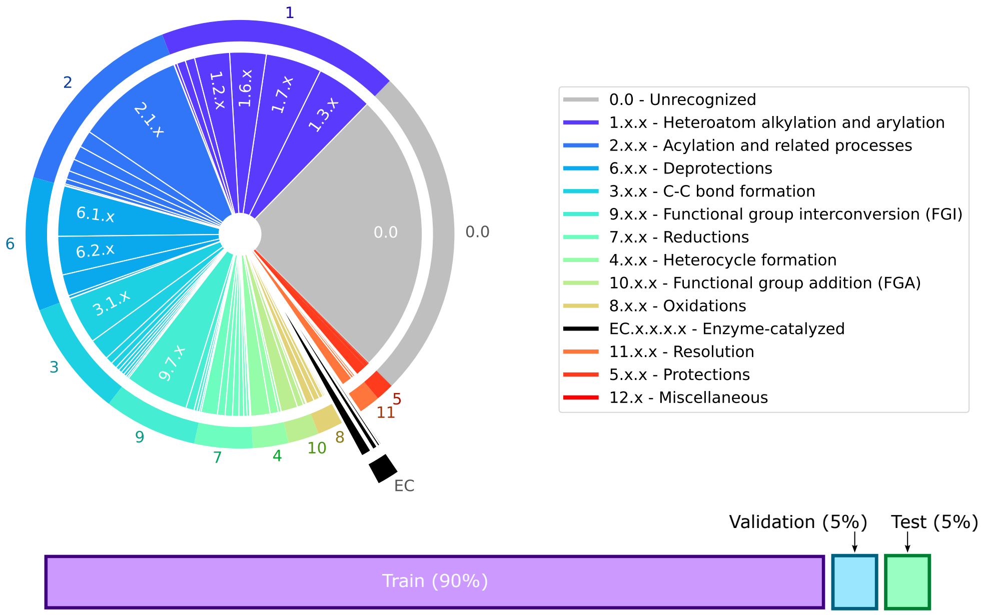


Figure 2: Distribution for the 83 reaction classes in the Pistachio+ECREACT dataset. The "EC" slice corresponds to ECREACT, with EC.2.x.x.x as the most frequent class (33'315 reactions). The description for Pistachio's superclasses and the data split are also indicated

Additionally, two classes contain only one sample: Pistachio 12.1 and 12.2. These outliers were moved to the test set so the training and validation sets contained in total 81 reaction classes each. Table 1 shows these two particular samples.

The Pistachio+ECREACT dataset sample distribution was then randomly split as follows: 90% for training, 5% for validation, and 5% for testing.

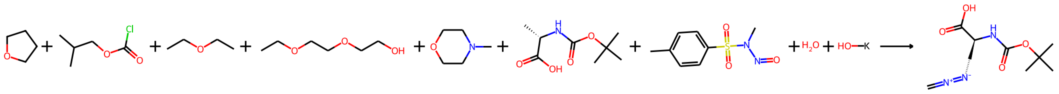
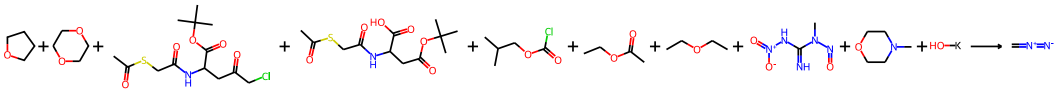
Reaction and SMILES	class
 `C1CCOC1.CC(C)COC(=O)Cl.CCOCC.CCOCCOCCO.CN1CCOCC1.C[C@H](NC(=O)OC(C)(C)C)C(=O)O.Cc1ccc(S(=O)(=O)N(C)N=O)cc1.O.O[K]>>C=[N+]=[N-] C[C@H](NC(=O)OC(C)(C)C)C(=O)O`	12.1
 `C1CCOC1.C1COCCO1.CC(=O)SCC(=O)NC(CC(=O)CCl)C(=O)OC(C)(C)C.CC(=O)SCC(=O)NC(CC(=O)OC(C)(C)C)C(=O)O.CC(C)COC(=O)Cl.CCOC(C)=O.CCOCC.CN(N=O)C(=N)N[N+](=O)[O-].CN1CCOCC1.O[K]>>C=[N+]=[N-]`	12.2

Table 1: The only two reactions part of the *miscellaneous* (12.x) NameRXN superclass

4.2 Multi-Head Attention and BERT

The *Bidirectional Encoder Representations from Transformers* model (BERT, Devlin *et al.* (2018)) has been used in the literature to classify reaction SMILES (Schwaller *et al.* (2021b)). BERT is an encoder model based on the Transformer architecture (Vaswani *et al.* (2017)), which initially introduced the concept of *Scaled Dot-Product Attention*. This kind of attention, given N input tokens, is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{1}{\sqrt{d_k}} QK^T\right) V$$

Where $Q \in \mathbb{R}^{N,d_k}$ are the *queries*, $K \in \mathbb{R}^{N,d_k}$ are the *keys*, $V \in \mathbb{R}^{N,d_v}$ are the *values*, and softmax() is applied independently in each row.

Each row in Q, K, and V corresponds to some token's query, key, and value representations. Furthermore, this function intuitively retrieves similarity between queries Q and keys K via a dot-product and then uses these similarities to do a weighted average of the token values in V.

This definition of attention is permutation-invariant in the tokens. Thus, positional encodings are added to the input tokens in the Transformer to take ordering into account. Moreover, the tokens are one-hot encoded using a *tokenizer*. In our case, we will use the SMILES tokenizer from Schwaller *et al.* (2021b).

Another contribution was the *Multi-Head Attention* layer, which is defined for h heads as:

$$\begin{aligned} \text{MultiHead}(Q', K', V') &= (\text{head}_1 | \cdots | \text{head}_h) W^O \\ \text{head}_i &= \text{Attention}(Q' W_i^Q, K' W_i^K, V' W_i^V) \quad i = 1, \ldots, h \end{aligned}$$

Where $\text{head}_i \in \mathbb{R}^{N \times d_v}$ are the output token values for the ith head, "|" concatenates the vectors, and $W_i^Q, W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ and $W_i^O \in \mathbb{R}^{h d_v \times d_{\text{model}}}$ are the model's paremeters. This time, Q', K', $V' \in \mathbb{R}^{N \times d_{\text{model}}}$ are the query, key, and value representations of all tokens on the multi-head level, while $Q' W_i^Q$, $K' W_i^K$, $V' W_i^V$ are the representations at the single-head level.

See Rush (2018) for practical implementation and further details of the Transformer architecture.

One of BERT novelties in the original paper was that it is a bidirectional language model instead of a left-to-right architecture like the recent auto-regressive models ChatGPT (OpenAI (2022)), BLOOM (Scao *et al.* (2022)), or Galactica (Taylor *et al.* (2022)). This bidirectionality means that it *attends* the relationship each token has with its context both to its left and right.

Like other transformer models, BERT is based on self-attention, and it introduced an unsupervised pre-training task on Masked Language Modelling (MLM), where a token is masked at random from the input text, and the model has to predict it. In detail, BERT predicts logits for all tokens and is trained using the cross-entropy loss function, meaning that MLM is a classification task. Through this task, the model effectively learns the grammar in the dataset (Schwaller *et al.* (2021a)) and encodes the tokens into useful embeddings. Thus, the pre-trained BERT encoder can be fine-tuned on various tasks and achieve high performance (Devlin *et al.* (2018)). The tasks we are interested in are single-label reaction type classification and multi-label reaction solvent classification. We will discuss the former in Section 4.3 and the latter in Section 4.4.

4.3 Enzymatic reaction classifier

In this section we train a BERT classifier on Pistachio+ECREACT to discern if a reaction is enzymatic or not.

We discover an adversarial attack on our model and adapt our training procedure to account for it. We also introduce a loss-balancing parameter to achieve higher performance on low-sample classes.

4.3.1 BERT classifier

We pre-train a BERT model (Devlin *et al.* (2018)) as a Masked Language Model (MLM) on the combined Pistachio+ECREACT dataset using the *Transformers* framework from Hugging Face (Wolf *et al.* (2019)). Following the architecture of Schwaller *et al.* (2021b), we set the hidden size to 256, intermediate size to 512, number of BERT layers to 12, and 4 attention heads. However, unlike this previous work, we increment the maximum sequence length from 512 to 1024 since some ECREACT SMILES contain many tokens. Only one SMILES contains more than 1024 tokens with 1177 tokens and it was present in the training set, but we consider it as unharmful as it is a single outlier reaction and increasing the sequence length requires more memory.

Then, we fine-tune the MLM BERT model on the Pistachio+ECREACT reaction classification task, where we set the learning rate to $3 \cdot 10^{-4}$ since it provides good results.

We noticed that the performance of the model on some classes with low samples was low which may be due to the uniform split of train/validation/test mentioned in the Section 4.1. This split might not represent well the sample distribution over the classes.

Thus, the original Pistachio+ECREACT dataset was split again using stratified sampling, so each class is independently split into 90% train, 5% validation, and 5% test. The same procedure is followed to train BERT, pre-training on a MLM task and fine-tuning on Pistachio+ECREACT classification for 3 epochs.

Afterward, we noticed that the models "cheat" when classifying ECREACT reactions (see Table 2), as 12% of the enzymatic samples contain a wildcard token, "*", which is not present in Pistachio. This was visualized by calculating importance attribution scores on the tokens using the method of *Integrated Gradients* (Sundararajan *et al.* (2017)). This method allows us to understand how each input token affects the prediction, either positively, negatively or without influence. It is defined for each input feature i as:

$$\text{IntegratedGrads}_i(\boldsymbol{x}) = (x_i - x_i') \cdot \int_{\alpha \in [0,\ 1]} \frac{\partial F(\boldsymbol{x}' + \alpha \cdot (\boldsymbol{x} - \boldsymbol{x}'))}{\partial x_i} \mathrm{d}\alpha$$

Where $F(\boldsymbol{x})$ is the neural network, $\boldsymbol{x}$ is the input, and $\boldsymbol{x}'$ is some *baseline*, usually the zero vector.

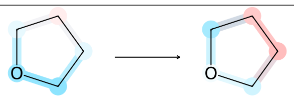
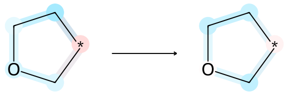
Integrated Gradients quantifies how much the output has been affected with respect to the input baseline. In our case, the input features are tokens and the baseline is the empty SMILES string `""`, which corresponds to 1024 padding tokens, `[PAD]`. We use the Captum Python package (Kokhlikyan *et al.* (2020)) to calculate the Integrated Gradients attribution scores.

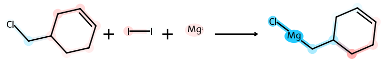
In the reaction of class 9.7.x of Table 2 we can appreciate how the model correctly attributes importance to the magnesium functional group substituted in the reaction when it correctly predicts the reaction class 9.7.x with a likelihood of 100%. Indeed, reaction class 9.7.x includes reactions where a functional group is substituted for another. However, after substituting 2 tokens for "*", the model ends up predicting a reaction class of EC.1.x.x.x with a likelihood of 94% and attributes a negative impact on the newly substituted functional group, meaning it does not consider anymore this rule in its prediction. Thus, we see that the substitution of some tokens for "*" changes the interpretation of the reaction and the structures that influence the prediction.

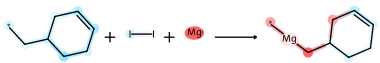
To avoid this cheat, the classification fine-tuning on the stratified dataset was re-done, this time introducing "*" at random into 50% of the Pistachio training samples. The procedure was as follows: we compute from the training set the empirical probability of finding a SMILES in ECREACT with n "*" tokens (see Figure 3); then, for 50% of Pistachio samples we substitute $\min\{n,\ 8\}$ tokens according to the previously computed probabilities. We do not substitute more than 8 tokens since the analysis shows that only 0.4% of ECREACT reactions contain the wildcard token more than 8 times and substituting too many tokens could introduce unwanted noise. The following tokens were not substituted: `[PAD]`, `[CLS]`, `[SEP]`, `(`, `)`, `.`, `>`, `>>`, `=`, `~`, `-`. The resulting model does not "cheat", i.e. it is not influenced by the presence of "*" in its prediction as it can be seen in Table 3. The same reaction of class 9.7.x previously mentioned now is correctly classified with a likelihood of 99% before adding "*" and with a likelihood of 96% after adding "*", and the functional group related to the magnesium atom is correctly attributed as a factor in the prediction.

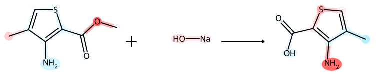
Another challenge in the data was its great imbalance and long tail (see Section 4.1). Weights based on the per-class effective number of samples were added to the cross-entropy loss so as to deal with this imbalance. They are introduced in the work of Cui *et al.* (2019), where the cross-entropy loss is re-written as:

$$\text{CE}_y(\hat{\boldsymbol{z}}) = -\frac{1-\beta}{1-\beta^{n_y}} \log\left(\frac{\exp(\hat{z}_y)}{\sum_{j=1}^{C} \exp(\hat{z}_j)}\right)$$

Reaction with importance attribution	SMILES	True class	Predicted class (prob.)
	`C1CCOC1>>C1CCOC1`	-	0.0 (31%)
	`*1CCOC1>>*1CCOC1`	-	EC.3.x.x.x (50%)

	`ClCC1CC=CCC1.II.[Mg]>>Cl`
	`[Mg]CC1CC=CCC1`
		9.7.x	9.7.x (100%)

	`*CC1CC=CCC1.II.[Mg]>>*`
	`[Mg]CC1CC=CCC1`
		9.7.x	EC.1.x.x.x (94%)

	`COC(=O)c1scc(C)c1N.O[Na]>>`
	`Cc1csc(C(=O)O)c1N`
		6.2.x	6.2.x (100%)

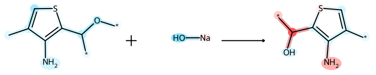
	`*OC(*)c1scc(C)c1N.O[Na]>>`
	`*c1csc(C(*)O)c1N`
		6.2.x	EC.1.x.x.x (85%)

Table 2: Adding "*" in some Pistachio validation examples makes the model predict them as enzymatic, and it makes the model change the structures it considers important for the prediction. The predicted class with its corresponding predicted probability is indicated. Blue indicates higher importance attribution and red lower. White means no attribution. The model used was BERT fine-tuned on the stratified Pistachio+ECREACT dataset. Importance attribution was retrieved through the method of *integrated gradients* (Sundararajan *et al.* (2017))

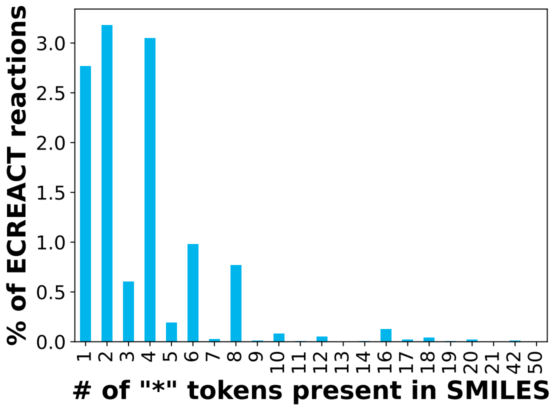
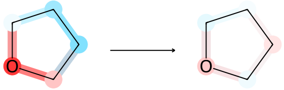
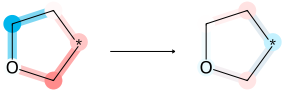


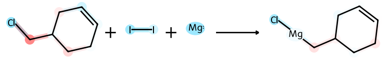
Figure 3: Frequency of the wildcard token in ECREACT reaction SMILES. Reactions usually have 1 or an even number of wildcard tokens. Only 0.40% of reactions contain the token more than 8 times

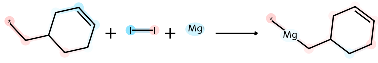
Where $y \in \{1, 2, \cdots, C\}$ is the class index, $\hat{z} \in \mathbb{R}^C$ are the predicted logits, $n_y \in \mathbb{N}^+$ is the number of training samples for class y, and $\beta \in [0, 1)$ is a hyper-parameter.

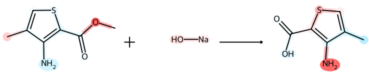
In the original article, β depends on *class imbalance*, which is defined as $\max_y(n_y)/\min_y(n_y)$. They show that as β gets closer to 1 the approximation $\frac{1-\beta}{1-\beta^{n_y}} \simeq \frac{1}{n_y}$ becomes a better approximation for larger n_y. This inverse frequency weighting would be a naïve approach of balancing the classes; however, in this case $\frac{1-\beta}{1-\beta^{n_y}} \to 1 - \beta = \text{const.}$ for $n_y \to \infty$ which, unlike in the naïve weighting, prevents very large classes from being underrepresented.

Our dataset has an imbalance of ~24'400—the division of the largest class, 0.0 in Pistachio, and the lowest, 4.4.x also in Pistachio—whereas the maximum imbalance factor in the original work is of 500. In that case,

Reaction with importance attribution	SMILES	True class	Predicted class (prob.)
	`C1CCOC1>>C1CCOC1`	-	EC.1.x.x.x (53%)
	`*1CCOC1>>*1CCOC1`	-	11.6.x (40%)

	`ClCC1CC=CCC1.II.[Mg]>>Cl`
	`[Mg]CC1CC=CCC1`
		9.7.x	9.7.x (99%)

	`*CC1CC=CCC1.II.[Mg]>>*`
	`[Mg]CC1CC=CCC1`
		9.7.x	9.7.x (96%)

	`COC(=O)c1scc(C)c1N.O[Na]>>`
	`Cc1csc(C(=O)O)c1N`
		6.2.x	6.2.x (100%)

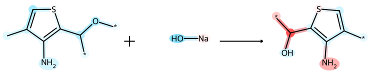
	`*OC(*)c1scc(C)c1N.O[Na]>>`
	`*c1csc(C(*)O)c1N`
		6.2.x	6.3.x (100%)

Table 3: After accounting for “*” during training the model does not “cheat”

the authors achieve best results with $\beta = 0.999$. We noticed that $\beta \leq 0.999$ resulted in a similar constant weighting of $1 - \beta \geq 0.001$ on most of the classes due their large amount of samples. Therefore, we decided to evaluate performance for $\beta \in \{0.9999, 0.99999, 0.999999\}$, which provided a more diverse set of weights across classes. For each β, we fine-tuned the stratified MLM model and performed grid search for different learning rates based on the macro average recall on the validation set. The models were trained for 3 epochs substituting wildcard tokens at random as described. We noticed that values of β closer to 1 required smaller learning rates for convergence, although in all cases the learning rate of 10^{-4} gave the highest macro average recall in validation.

β	Learning-rate search space	Best learning rate	Best macro average recall
0.9999	$\{10^{-5},\ 5\cdot 10^{-5},\ 10^{-4},\ 3\cdot 10^{-4}\}$	10^{-4}	96.90%
0.99999	$\{10^{-5},\ 5\cdot 10^{-5},\ 10^{-4},\ 3\cdot 10^{-4}\}$	10^{-4}	**97.37%**
0.999999	$\{10^{-5},\ 5\cdot 10^{-5},\ 10^{-4},\ 3\cdot 10^{-4}\}$	10^{-4}	96.98%

Table 4: Hyper-parameter search for weighted loss. Models were fine-tuned for 3 epochs

Table 4 shows that $\beta = 0.99999$ provides the best results in terms of the macro average recall (97.37%), and in Figure 4 we can see that the class-balanced model disallows low recall on any class, no matter its amount of samples. This is important since, as discussed in Section 4.1, the enzymatic classes have much fewer samples than the Pistachio classes.

Tables 5 and 6 contain a summary of the results. The evaluation metrics CEN and MCC are the same ones used in Schwaller *et al.* (2021b). Table 6 also includes the classification accuracy and F1 score when identifying that a reaction can be enzymatic.

Both in pre-training and fine-tuning for all approaches we used an AdamW optimizer (Loshchilov and Hutter (2017)) on 2 NVIDIA A100 GPUs. In MLM pre-training we used a batch size of 48 per device and learning-rate of $3 \cdot 10^{-4}$ with linear decay for 10 epochs, while in fine-tuning we used a batch size of 32 and learning-rate of $3 \cdot 10^{-4}$ with linear decay for 3 epochs, except for the class-balanced approach in which we used a learning-rate of 10^{-4}.

We note that the wildcard token substitution slightly reduced overall accuracy by 0.06% and micro F1 score by 0.001 on all classes.

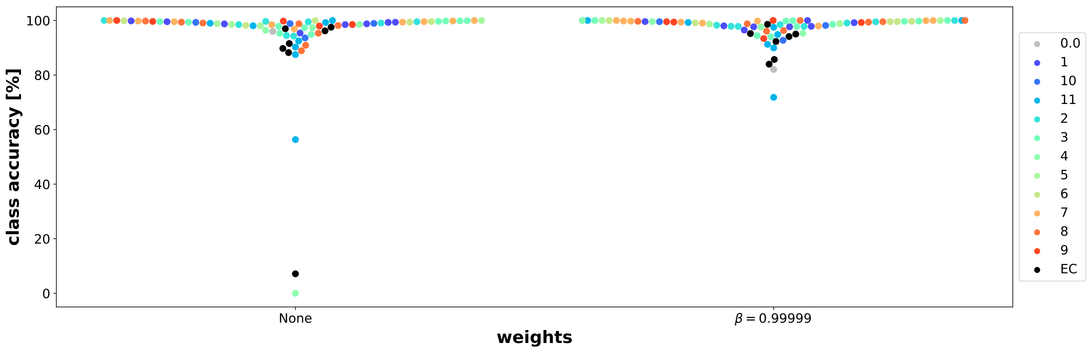


Figure 4: Validation recall for all 81 classes present in validation. Classes are colored according to their superclass. Left: unweighted loss. Right: weighted loss. The macro average recall for the non-balanced model is 94.81% and for the balanced model is 97.37%

MLM approach	Final Validation Accuracy
Non-stratified	99.04%
Stratified	99.06%

Table 5: Summary of MLM approaches. Models were pre-trained for 10 epochs. Pre-training for each model took ~ 44h on 2 NVIDIA A100 GPUs

	Enzymatic prediction		EC.7.x.x.x	Pistachio+ECREACT all classes				
Approach	Acc.	F1	F1	Acc.	Micro F1	Macro F1	MCC	CEN
Non-stratified	99.96%	0.987	0.111	98.04%	0.980	0.938	0.979	0.024
Stratified	99.96%	0.989	0.000	98.26%	0.983	0.943	0.981	0.022
Stratified + ‘*’ token	99.96%	0.987	0.133	98.20%	0.982	0.943	0.980	0.022
Stratified + ‘*’ token + β = 0.99999	99.93%	0.978	**0.338**	93.95%	0.940	0.901	0.935	0.058

Table 6: Summary of classification approaches. Models were fine-tuned for 3 epochs. Note that the non-stratified apporach has a different set of training samples compared to the stratified dataset used in the other approaches

Regarding the loss-balanced model, it improved the F1 score by 0.205 on the translocase-based catalysis prediction (EC.7.x.x.x class) which is the lowest sample class in ECREACT. Thus, by reducing the accuracy by 0.03 percentage points and the F1 score by 0.009 on whether the reaction can be enzymatic, we gain the ability to better identify rare biocatalyzed reactions.

4.3.2 Conclusion

Our greatest challenge in this section was training BERT on our imbalanced Pistachio+ECREACT dataset, due to the required hyper-parameter search for the optimal loss-balancing parameter and learning-rate.

We also demonstrated how the method of Integrated Gradients (Sundararajan *et al.* (2017)) can be used for model-driven discovery of adversarial attacks. A more thorough study of this adversarial attack could be performed by randomly replacing tokens with “*” on some reactions and evaluating the distribution of the resulting likelihood that the reactions are enzymatic.

Previous work has looked at reactions in Pistachio, achieving an accuracy of 98.2%, MCC of 0.988, and CEN of 0.010 (Schwaller *et al.* (2021b)); however, it did not include enzymatic reactions. When accounting for biocatalysts, our adversarially-aligned BERT model without loss-balancing achieves a similar accuracy of 98.2%, MCC of 0.980, and CEN of 0.022 on all Pistachio+ECREACT classes. However, the focus of our work is sustainability prediction, and for this task our loss-balanced model achieves an accuracy of 99.93% and an F1 score of 0.978 in validation.

4.4 Solvent classifier for reaction sustainability

In this section, we first verify that there is enough low-hazard and renewable solvent data in our dataset. Then, we study whether solvent sustainability and enzymatic reactions are clearly disjoint aspects in our dataset to justify the need for a solvent predictor. Afterward, we describe the solvent extraction procedure from Pistachio+ECREACT and provide the particular list of solvents we will consider sustainable in this work. Finally, we train XGBoost and BERT models for sovent prediction and evaluate their performances.

4.4.1 Extracting all solvents from Pistachio

To carry out a solvent sustainability study on Pistachio (Nextmove Software (2021b)) we first must study which solvents are available in the dataset.

We extracted solvent information from version 220406 of Pistachio. This version from 2022 contains 13M reactions, and it is more recent than the one used in Section 4.3 (version 210403). In particular, the dataset contains solvent information for 7M out of the 13M Pistachio reactions (53%) either by solvent name and its SMILES representation or only by solvent name (as mined from the patents' text).

The following challenges arise due to the imperfect nature of the patent-mining procedure used in Pistachio and had to be overcome:

- Not all reactions contain solvent information (47% do not contain solvent information), and some reactions contain multiple solvents.
- A solvent name can correspond to multiple SMILES.
- A SMILES can correspond to multiple solvent names.
- A solvent name can have no corresponding SMILES.

For the latter case, the number of reactions with solvent names without their corresponding SMILES is 26'099 (0.37% of solvent reactions). Since they are few reactions, these solvent names were removed for the rest of the analysis.

The solvent extraction procedure is as follows:

1. For each solvent name, we select the SMILES it is most commonly paired with in the reactions.
2. We canonicalize all SMILES using RXN chemutils (IBM RXN team (2022)) and combine solvents that have the same canonicalized SMILES. SMILES with an incorrect format are removed.
3. With the list of solvent SMILES retrieved in the previous step, a list of *synonyms* for each solvent is compiled by looking at all solvent names paired with each SMILES from the list. The most frequent *synonym* is kept as the "common name" for the solvent.

After curating the data, we found 80'182 different solvent names in Pistachio and 11'177 unique canonicalized solvent SMILES. Some solvents are composed of multiple compounds, like dichlormethane methanol, or THF methanol.

Figure 5 displays the 100 most frequent solvents. Analogous to the reaction class distribution, there is a high data imbalance and long tail.

We conclude that Pistachio contains enough solvent information to perform a solvent sustainability study, which follows in the next section.

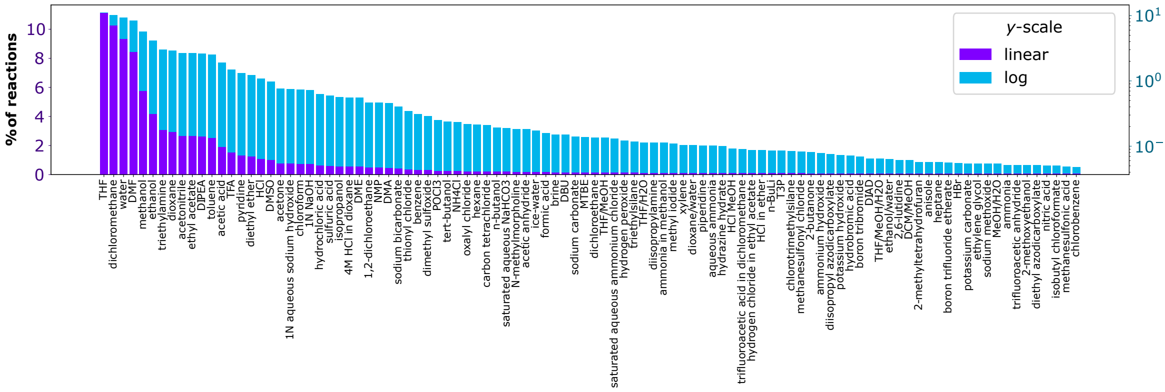


Figure 5: 100 most frequent solvents in Pistachio 2022. Solvent names displayed are the most common names found in Pistachio for each solvent. The same distribution is included in log-scale in blue

4.4.2 Solvent sustainability distribution in Pistachio

Solvent sustainability is a complex and multi-faceted problem, which is often defined following solvent selection guides from academia and industry (Prat *et al.* (2016); ETH Zurich (2008)). In Byrne *et al.* (2016), the authors provide a complete review of solvent selection guides and provide a final classification of solvents according to their hazard level and source renewability by combining multiple selection guides. In particular, we follow the solvent sustainability classification matrix of Byrne *et al.* (2016).

As in the previous section, we perform our analysis on the newer Pistachio version 220406 with 13M reactions.

First, the SMILES representation for each of the 51 solvents was retrieved using the list of solvent synonyms from the previous section. Then, using these SMILES we can perform a first analysis on the frequency of each solvent in the Pistachio dataset using the results from last section (Figure 5): the 3 least common solvents are methyl acetate, methylcyclohexane, and cyclohexanone which appeared in 550, 337, and 305 reactions respectively; while the 3 most common are THF, DCM, and water, with 1.4M, 1.3M, and 1.2M reactions respectively. It is important to note that the reactions counted for this paragraph do not necessarily contain SMILES information.

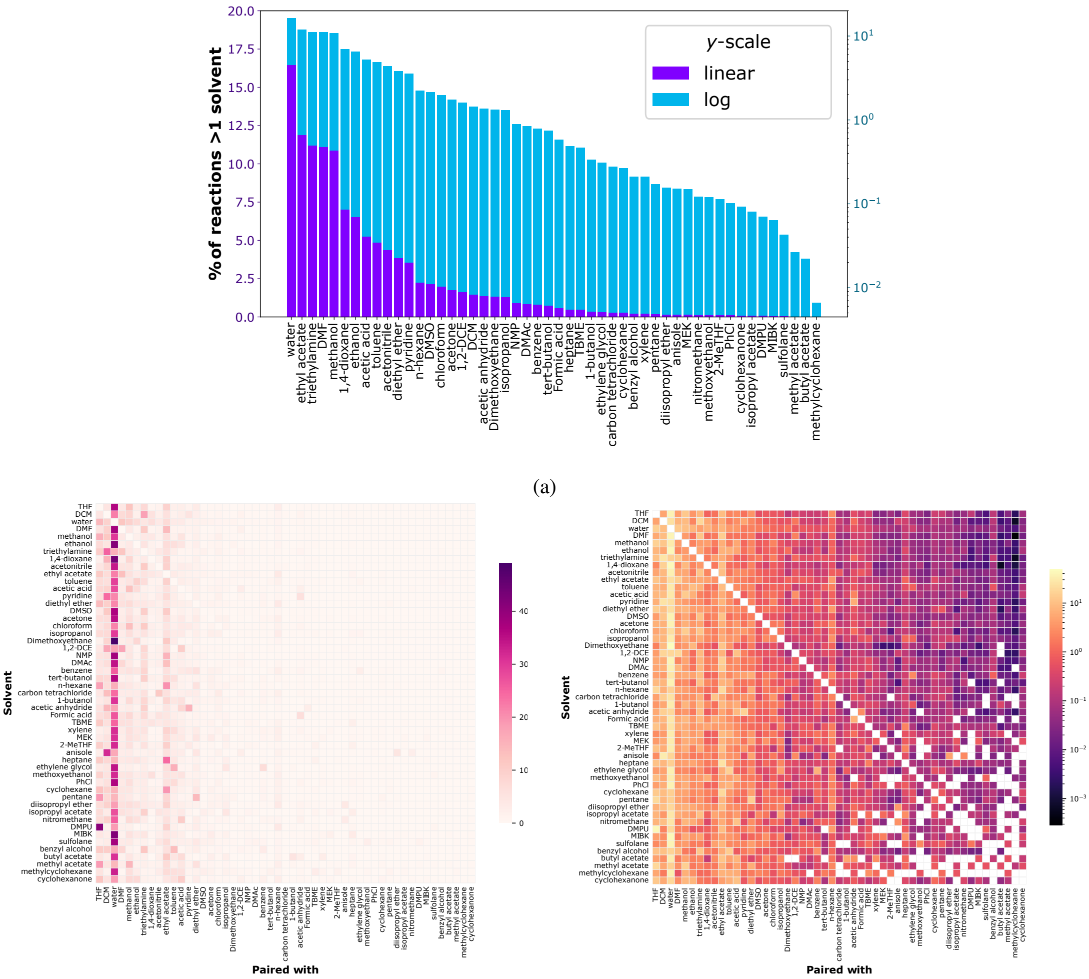
To further analyze the solvents present in the Pistachio reactions and avoid possible noisy or incomplete data in Pistachio's solvent information, we look directly for solvents in the reaction SMILES. We extract all reactions that contain SMILES information and standardize them using RXN chemutils (IBM RXN team (2022)). The final amount of unique standardized reaction SMILES is 4M. With this list we can discover the participating solvent molecules by searching for their compound SMILES. We simply use RDKit (Landrum *et al.* (2022)) to extract all participating compounds in a reaction SMILES and match them to our list of solvent SMILES. We should note that through this method *solvent molecules* are not necessarily acting as *solvents* in the reaction, but we can still justify our sustainability analysis since the solvent sustainability information from Byrne *et al.* (2016) depends on the origin and external impact of the solvent compounds, not whether they act as solvents in the reactions.

First, we note that reactions can have multiple participating solvent molecules. In fact, 38% of the 4M Pistachio reaction SMILES contain multiple solvents. Figure 6a shows how frequently each solvent appears in multi-solvent reactions. Not surprisingly, water is the most common companion solvent. However, some exceptions include triethylamine, most commonly paired with DCM, pyridine with DCM, n-hexane with ethyl acetate, anisole with DCM, heptane with ethyl acetate, and DMPU with THF (see Figure 6b).

Then, we classify reactions according to the two previously mentioned qualities of sustainable solvents, hazard level and source renewability:

- **Hazard level**: We look at the distribution of hazardous Pistachio reactions according to the solvents present in the 4M reaction SMILES. Since each reaction can have multiple solvent molecules, we say that the hazard level of a reaction is the hazard level of its most hazardous solvent. Figure 8a shows the reaction hazard level distribution. As it can be seen, reactions tend to be hazardous in Pistachio.

  We also look at the most common solvents present in "*recommended*" reactions, which is the lowest hazard level (see Figure 8b). Both water and ethanol are the most common, each taking part in around half of the recommended reactions.



(a)

(b)

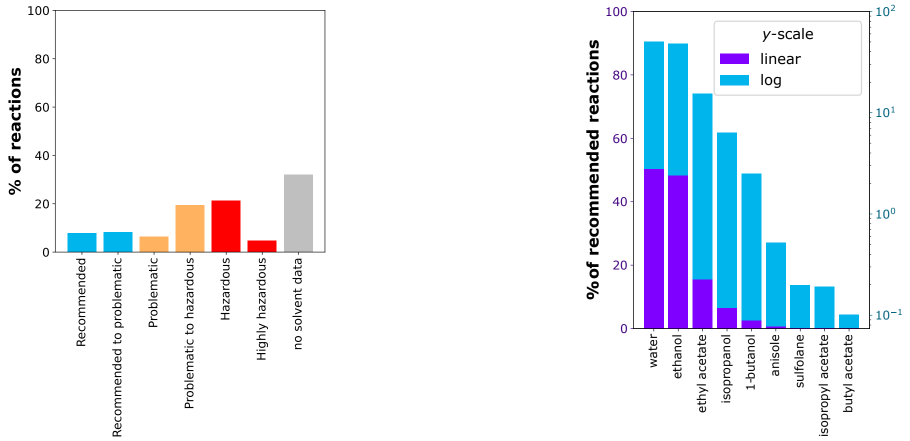
Figure 6: **a**, Frequency of solvents when participating in reactions with multiple solvent molecules in Pistachio 2022. *y*-axis is in the interval 0% to 20%. The same plot is included in log-scale in blue. **b**, Distribution of paired solvents. Left: Distribution of solvents (columns) in reactions where another solvent appears (rows). Solvents are sorted from most to least common (top to bottom, left to right). Right: same plot in log-scale. Blank squares are solvents which do not pair

- **Source renewability**: Similar to the hazard level, we classify reactions according to the least renewable solvent. Figure 8a shows the reaction renewability distribution. We consider that the only reactions we can confidently label *non-sustainable* are those which are "*not bio-based*", that is, they contain a solvent which is definitely not renewable.

  Figure 8b shows how often each "*bio-based*" solvent appears in bio-based reactions. As in hazard level, water and ethanol are the most common appearing in roughly 55% of reactions.

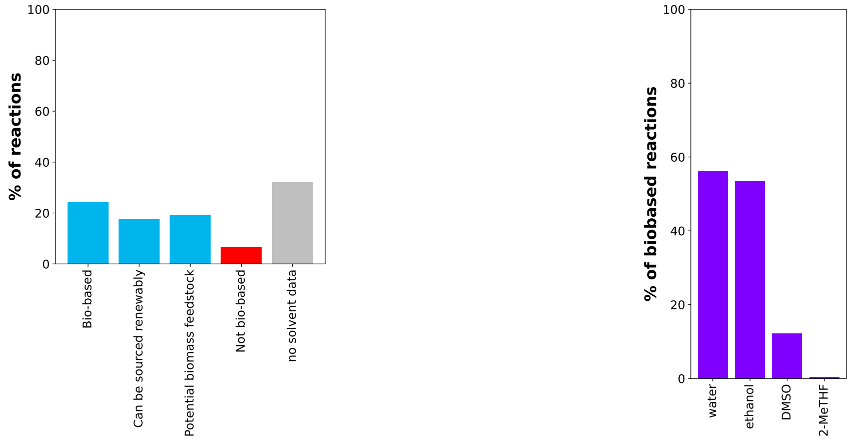
The solvent distributions of Figures 7b and 8b are not surprising since they follow the general solvent frequency distribution in all Pistachio reactions.

Since 68% of the 4M Pistachio reaction SMILES contain solvent molecules from Byrne *et al.* (2016) and there are enough low-hazard and renewable reactions, as seen in Figures 7a and 8a, we conclude that this solvent sustainability classification approach covers enough data to allow AI models to learn solvent sustainability.



(a) (b)

Figure 7: **a**, Distribution of reactions in Pistachio 2022 according to their hazard level. 32% of the 4M Pistachio reactions do not contain solvent molecules from Byrne *et al.* (2016). **b**, Distribution of recommended solvents in Pistachio 2022, also included in log-scale in blue



(a) (b)

Figure 8: **a**, Distribution of reactions in Pistachio 2022 according to their renewability. 32% of the 4M Pistachio reactions do not contain solvent molecules from Byrne *et al.* (2016). **b**, Distribution of bio-based solvents in Pistachio 2022

4.4.3 Intersection with reaction classes

To compare how the enzyme and solvent sustainability factors overlap, we take a look at the intersection between the discussed solvents from Byrne *et al.* (2016) and our Pistachio+ECREACT dataset introduced in Section 4.3.

For each reaction, we see if a solvent molecule is present in the reaction SMILES, either as a precursor or product. In this way we can count for each reaction class the number of reactions the solvent appears in.

Figure 9 shows the distribution of reaction classes per solvent, i.e. the types of NameRXN superclasses (Nextmove Software (2021a)) and including enzyme-catalyzed reactions, where the solvent participates in. We ignore all reactions that do not have solvent information, which excludes 85% of enzyme-catalyzed reactions (see Figure 10a). Enzyme-catalyzed reactions are very rare, amounting to 0.2% of reactions, while the "*miscellaneous*" class of reactions, 12.1 and 12.2, only include one reaction each (as discussed in Section 4.3).

As it can be seen in Figure 9, we lack the class information for almost 25% of reactions, which are labelled as 0.0. In fact, it is usually the case that given a solvent the reaction class is unknown in the dataset. Regarding the intersection with the enzyme-catalyzed reactions, we see that solvents like water, MEK, benzyl alcohol, cyclohexane, methyl acetate and methylcyclohexane participate relatively more in these kinds of reactions compared to the rest of the solvents. However, the intersection is so small in all cases that sustainability considerations for enzyme-catalyzed reactions and those based on sustainable solvents can be considered in-

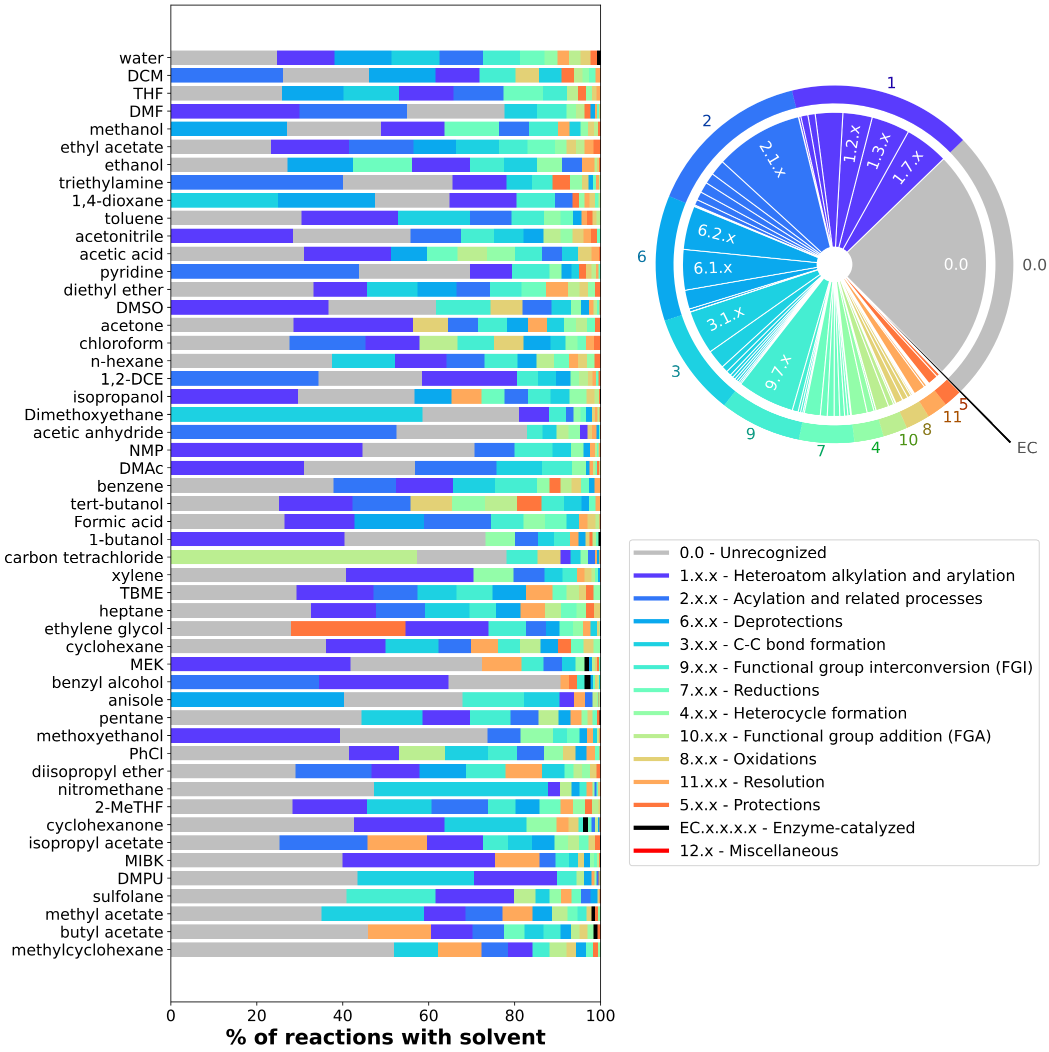
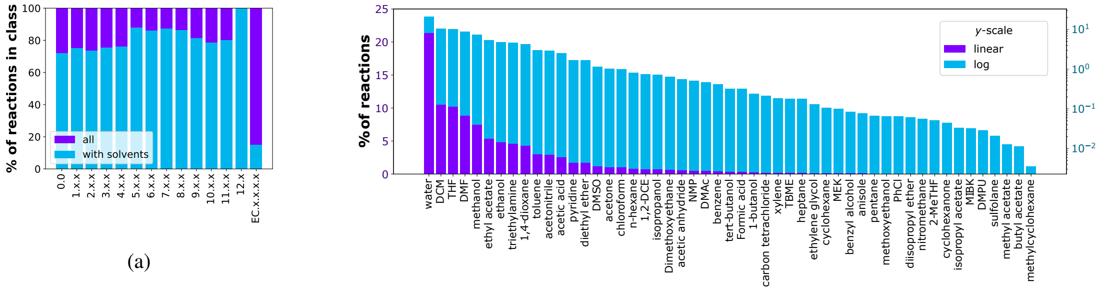


Figure 9: NameRXN superclasses (Nextmove Software (2021a)) in which each solvent molecule participates in, including enzyme-catalyzed reactions. Solvent molecules are sorted from most to least common (top to bottom). Reaction superclasses for each solvent molecule are sorted from most to least common (left to right). The pie chart indicates the reaction superclass distribution for reactions with solvent molecules. Frequent sub-classes are indicated

dependent discussions. This motivates the need for a separate model for predicting the solvents participating in a reaction in addition to the enzyme classifier model presented in Section 4.3, as it can provide a different perspective on reaction sustainability.

You can find the same plot without the unrecognized 0.0 class in Appendix A (Figure A.1).



(a)

(b)

Figure 10: **a**, Percentage of reactions that contain solvent information per NameRXN superclass (Nextmove Software (2021a)), including enzyme-catalyzed reactions. **b**, Percentage of reactions information where each solvent participates in. The percentages are over the reactions that contain solvent information. The same distribution is included in log-scale in blue

4.4.4 Pistachio+ECREACT solvent extraction

In order to train models to predict the solvents present in a reaction we need to first create a dataset with reactions without solvents and their extracted solvents as labels. Thus, we will extract these solvent labels from the Pistachio+ECREACT dataset presented in Section 4.1.

An important first consideration is that reactions in Pistachio+ECREACT contain fragment bonds ~, which indicate the interaction between two molecules. Thus, solvent molecules interacting through fragment bonds have to be taken into account and extracted. Figure 11 shows an example reaction with fragment bonds and solvent molecules.

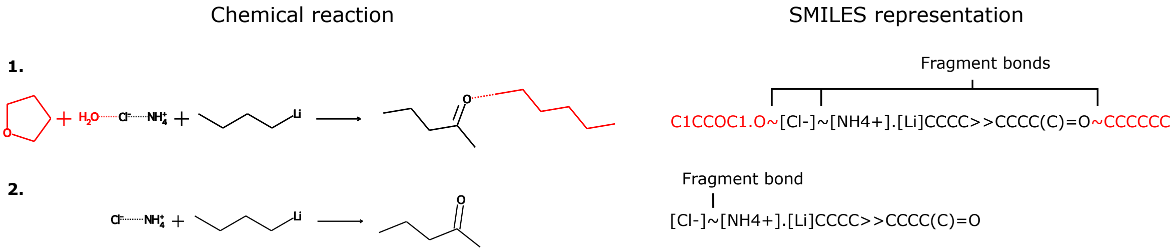

Chemical reaction
SMILES representation
1.
Fragment bonds
C1CCOC1.O~[Cl-]~[NH4+].[Li]CCCC>>CCCC(C)=O~CCCCCC
2.
Fragment bond
[Cl-]~[NH4+].[Li]CCCC>>CCCC(C)=O

Figure 11: Example reaction with fragment bonds "~". **1.**, Original reaction from Pistachio+ECREACT. The identified solvent molecules from Byrne *et al.* (2016) are highlighted in red. **2.**, The same reaction without the identified solvent molecules

One of the main difficulties in solvent extraction is that reactions can have multiple solvents, and each reaction can have multiple *variants* with a different set of solvents each. Figure 12 shows three example reactions which appear multiple times in the dataset, each time with a different set of solvents participating in the reaction. Furthermore, molecules that are usually solvents sometimes act as reactants in the reaction. For example, methoxyethanol in reaction **3.** of Figure 12 participates as a reactant in the reaction, not as a solvent. This is not a concern for our purposes, since the sustainability information we use for these molecules is related to their origin and environmental impact (Byrne *et al.* (2016)), not whether they act as solvents or reactants in the reaction.

Reactions with multiple variants correspond to 4.3% of all reactions in the dataset (117'582 reactions). Enzyme-catalyzed reactions with multiple variants are only 0.46% (32 reactions) of all enzyme-catalyzed reactions. Since these percentages are small, for each reaction we will combine solvent sets from all variants into one set in the solvent extraction process, as assuming a single solvent set per reaction simplifies the model architecture. Moreover, we can see that variant A in reaction **3.** of Figure 12 contains incomplete information as methoxyethanol, being a reactant, should be present in the reaction. So, even if our model could account for multiple variants it may be very difficult to generalize since some variants in the trainset contain incomplete information.

Thus, the solvent extraction procedure is as follows:

1. For each reaction variant, extract all solvent molecules that coincide with those from Byrne *et al.* (2016) by matching their SMILES representation, taking into account that they may be interacting through fragment bonds ~.

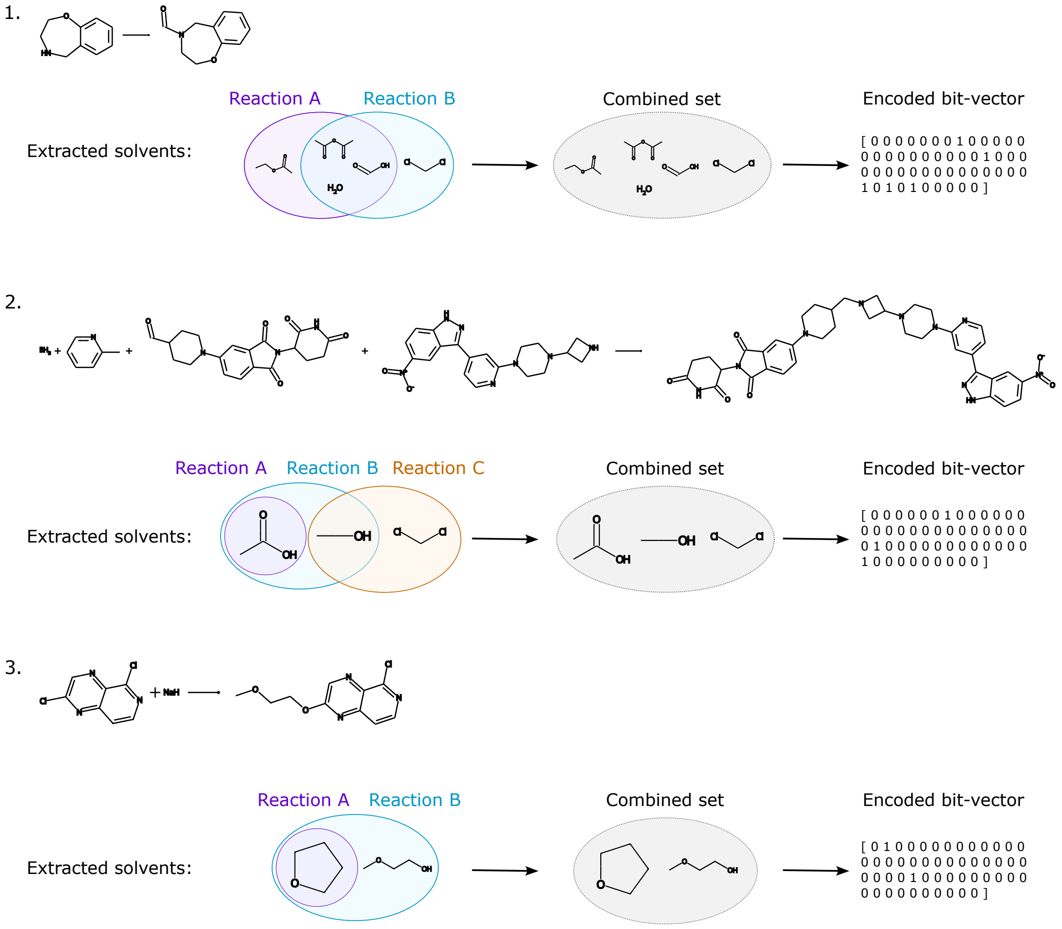


Figure 12: Example reactions that are multi-solvent and multi-label and the solvent extraction procedure. The reactions presented have all solvents removed. In example **1.**, there are 2 variants in the dataset (A and B) which coincide in 3 solvent molecules. In example **2.**, there are 3 variants (A, B and C) and variant A's solvents are contained in variant B's solvents, while variant A and B do not coincide in any solvents. In example **3.**, there are 2 variants (A and B) and again variant A's solvents are contained in variant B's solvents

2. For each reaction, combine all solvent molecules in all reaction variants into a single set of solvent molecules.

3. For each reaction, encode the combined solvent set $S = \{\text{solvent}_{j_1}, \text{solvent}_{j_2}, \ldots\}$, for some $j_k \in \{1, \ldots, \#\text{ of solvents}\}$, into a bit-vector $l \in \{0, 1\}^{\#\text{ of solvents}}$ where $l_i = 1$ if $\text{solvent}_i \in S$ and $l_i = 0$ otherwise.

Due to the multi-faceted nature of solvent sustainability and noise in the data we do not implicitly use sustainability information in the solvent extraction process; for example, by extracting only the most sustainable variant. It is important that models predict all possible solvents that could partake in a reaction so that the chemist can directly validate whether the predicted solvents could indeed take place in the reaction and are sustainable in their context.

4.4.5 Solvent sustainability definition

Sustainability is a multi-factorial problem, thus we should allow the user to heuristically choose what aspect of sustainability to give more weight to. Therefore, all our models should be trained to predict all solvents that could participate in the reaction and only after the solvent sustainability metric is computed according to some rule chosen by the chemist.

For solvents, the chemist should decide which hazard level and type of renewable source are acceptable. For this work, we base the solvents' greenness and renewability on Table 5 from Byrne *et al.* (2016). In particular, we arbitrarily decide that sustainable solvents are those either "*recommended*" or "*inbetween recommended and problematic*" and also either "*bio-based*", "*can be sourced renewably*", or a "*potential biomass feedstock*" (see Table 7). We considered all solvent sources except those not bio-based since renewability is one of the current core objectives in sustainable chemistry (Weber *et al.* (2021); Weber (2022)) and the listed solvents in all of these source types are industrially-viable as discussed in Byrne *et al.* (2016). "*Inbetween recommended and problematic*" solvents come from a non-consensus on whether the solvent is recommended or problematic according to safety, health and environment considerations (Prat *et al.* (2016)). This hazard level has been included to cover a wide space of reactions while not having reactions being considered problematic (~ 45% of Pistachio reactions as seen in Figure 7).

	Source		
Hazard level	**Bio-based**	**Can be sourced renewably**	**Potential biomass feedstock**
Recommended	Ethanol Water	1-Butanol Ethyl acetate	1-Butyl acetate Isopropanol Isopropyl acetate
Inbetween recommended and problematic		Acetic acid Acetone Ethylene glycol Methanol	Acetic anhydride *t*-Butanol Methyl acetate MIBK

Table 7: Solvents defined as "*sustainable*" in this work

As discussed in the previous Section, reactions are multi-label and multi-class in terms of the solvent molecules participating in them. To classify a reaction's sustainability according to its participating solvent molecules we should take the worst case solvent molecules. Thus we have the following definition:

Definition 4.4.1. *A reaction is* solvent-sustainable *if there are no* non-sustainable *solvent molecules participating in the reaction as reactants, agents, or products.*

For example, the reaction presented in Figure 11 is not *solvent-sustainable* since it contains THF (with SMILES `C1CCOC1`), which is not one of the sustainable solvents we have defined in Table 7.

We will refer to Table 7 in the rest of this work when scoring solvent sustainability, although our implementation is invariant to how the set of sustainable solvents is defined.

4.4.6 XGBoost baseline

We train an XGBoost model and use the atom-pair difference (AP3) fingerprints from Schneider *et al.* (2015) as implemented in RDKit (Landrum *et al.* (2022)), and the differential reaction fingerprint (DRFP, Probst *et al.* (2022c)).

Micro and macro F1 used are those defined by the MLCM Python package (Heydarian *et al.* (2022)).

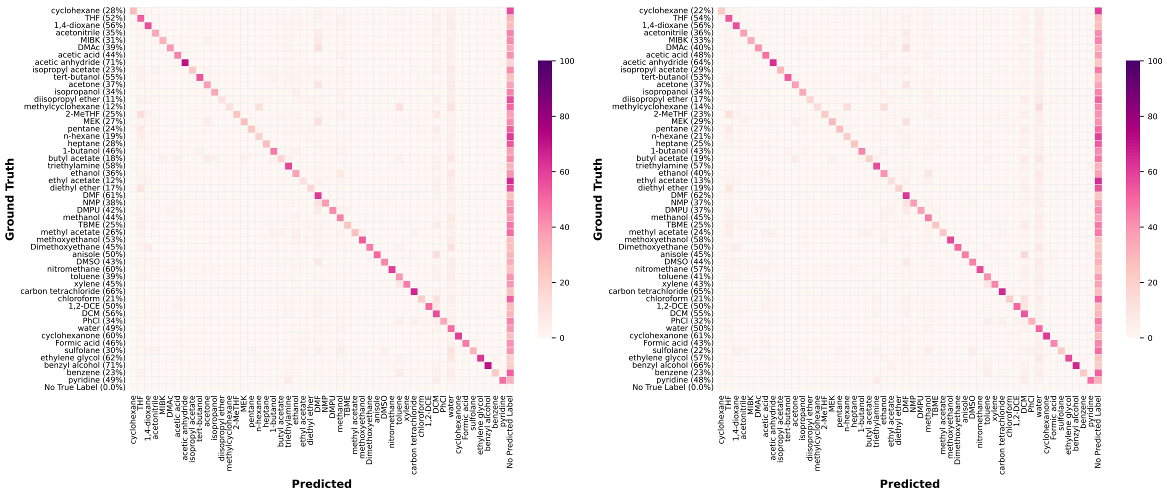
We also compare both approaches according to their accuracy and F1 score when classifying reaction sustainability according to the participating solvent molecules.

Both fingerprint methods perform very similarly. However, DRFP provides slightly better performance in solvent sustainability prediction with an F1 score of 0.615 compared to the F1 score of 0.604 from the AP3 fingerprint, with the disadvantage of having a x20 slower conversion rate from SMILES to the reaction fingerprint vector compared to AP3.

	Solvent sustainability prediction		All solvents		Conversion rate
Fingerprint	**Accuracy**	**F1**	**Micro F1**	**Macro F1**	**[SMILES/s]**
AP3 2048-bit	72.37%	0.604	**0.403**	**0.502**	**1'040**
DRFP 2048-bit	**73.60%**	**0.615**	0.401	0.500	51

Table 8: Summary of XGBoost performance for different fingerprint approaches. Micro and macro F1 scores are calculated following the definition of MLCM (Heydarian *et al.* (2022)). The rate of conversion from SMILES to the reaction fingerprint vector is also included

Figure 13 shows the MLCM confusion matrices for both approaches on the validation dataset. We can see that the models usually predict that reactions do not contain any solvent molecules. The AP3 and DRFP models predict that 24.83% and 23.59% of validation reactions do not contain solvent molecules, respectively. However, all reactions in the validation set contain solvent molecules. This indicates that the greatest limitation in these models is their low predicted likelihood for all solvents.



(a) AP3. No-solvent predictions: 24.83%

(b) DRFP. No-solvent predictions: 23.59%

Figure 13: MLCM confusion matrices and percentage of "no-solvent" predictions. **a**, AP3 2048 bit fingerprint confusion matrix. **b**, DRFP 2048 bit fingerprint confusion matrix. Recall for each solvent class is also included in parenthesis. "No True Label" is 0.0% since reactions without solvents are excluded in this study

Analogously to the performance improvements of DRFP over AP3 in Probst *et al.* (2022a), DRFP outperforms AP3 in solvent sustainability prediction.

4.4.7 BERT hyper-parameter search

Analogously to the approach from Section 4.3.1, we pre-train BERT on a Masked-Language Modelling (MLM) task on the stratified Pistachio+ECREACT dataset. Compared to the enzyme classification model, we increase the hidden size and intermediate size to 512 and 1024, respectively. The MLM model is pre-trained for 3 epochs and achieves a validation accuracy of 99%.

Due to the data imbalance, we evaluate both a non-weighted sigmoid cross-entropy (CE) loss, and the same loss weighted by the effective number of samples (Cui *et al.* (2019)). This is analogous to the balancing loss from Section 4.3.1. In our case, we have a data imbalance of ~4480, which is 5.40 times smaller than the Pistachio reaction class imbalance seen in that previous section.

To choose the optimal loss-balancing parameter, we do a hyper-parameter search by also including the learning-rate and weight decay in the search. Since the search space is too big, we restrict our training set during the search to 10% of the samples.

Unlike the fine-tuning approach of the enzyme classifier, we cannot use a linear learning-rate decay since we are not dealing with the complete training set. Instead, we use the inverse square-root learning rate scheduler introduced in the original Transformer architecture (Vaswani *et al.* (2017)), since it has proven good results when scaling visual transformers against other scheduling approaches (Zhai *et al.* (2022)) and it does not depend on the number of training epochs. This independence on the number of epochs allows us to cut training during the hyper-parameter search and still be able to meaningfully scale training to multiple epochs after we have found the best hyper-parameters. The learning rate update rule is:

$$\begin{cases} \gamma_t = \gamma_0 \cdot \frac{t}{\text{warmup}} & \text{if } t < \text{warmup} \\ \gamma_t = \gamma_0 \cdot \left(\sqrt{\frac{t + (\text{timescale} - \text{warmup})}{\text{timescale}}} \right)^{-1} & \text{if } t \geq \text{warmup} \end{cases}$$

Where γ_0 is the original learning rate, γ_t is the scaled learning rate at step t, *timescale* is a scaling parameter, and *warmup* is the number of warmup steps. Following the implementation of Zhai *et al.* (2022), *timescale* is set to equal *warmup*.

For the hyper-parameter search, we use AdamW as the optimizer (Loshchilov and Hutter (2017)), a batch size of 32 per device, and two NVIDIA A100 GPUs. We utilize Optuna (Dusenberry *et al.* (2020)) through the Transformers Python package (Wolf *et al.* (2019)) for our hyper-parameter search. Due to time constraints, we employ Hugging Face's Optimum (Hugging Face (2022)) with ONNX Runtime (Microsoft (2022)) to accelerate training by ~2.7, and limit the number of trials to 20. Finally, the number of warmup steps chosen is 1'800, which is ~ 5% of an epoch, and we only consider β values above 0.99 since lower values result in equal weighting across solvent classes.

Table 9 shows the search results. We divide $(1 - \beta)$ to the validation loss in the loss balancing approaches when presenting the results to compare them on an equal footing, as this is a constant factor that simply re-scales the loss.

Loss balancing	Best trial	Best validation loss	Best learning-rate	Best weight decay
None	12	**0.0812**	$2.1 \cdot 10^{-4}$	0.463
$\beta = 0.99$	11	$(1 - \beta) \cdot$ **0.0812**	$2.3 \cdot 10^{-4}$	0.401
$\beta = 0.999$	10	$(1 - \beta) \cdot 0.0982$	$3.4 \cdot 10^{-4}$	0.385
$\beta = 0.9999$	18	$(1 - \beta) \cdot 0.2996$	$5.7 \cdot 10^{-4}$	0.091
$\beta = 0.99999$	18	$(1 - \beta) \cdot 2.4405$	$5.7 \cdot 10^{-4}$	0.091

Table 9: Hyper-parameter search results. A Tree-structured Parzen Estimator (TPE) sampler was used with the same initial seed in all cases. Each trial took ~36min on 2 NVIDIA A100 GPUs, including training and evaluation

After retrieving the best hyper-parameters, we train each model for 3 epochs using the best learning-rate and weight decay. The results are summarized in Figure 14. As it can be seen, the model trained with loss balancing $\beta = 0.99$ achieves the best results in solvent sustainability prediction with an F1 score of 0.648 and accuracy 68.64%. All models except $\beta = 0.99999$ outperform in the F1 score our XGBoost baseline trained on DRFP fingerprints which achieved an F1 score of 0.615 (see Table 8).

Since the model with loss balancing $\beta = 0.99$ achieves the best performance in sustainability prediction, we continue fine-tuning it for 3 more epochs. The F1 score in sustainability prediction increases 0.012 points from 0.648 to 0.660. As this increase is small, we do not continue fine-tuning for further epochs. Figure 15 shows the MLCM confusion matrix for this BERT model. Analogously to the XGBoost baseline, the BERT model often predicts that reactions do not have any solvents, classifying 22.62% of the validaton set as having no solvents. This is similar to the XGBoost DRFP model, which classifies 23.59% of reactions as having no solvents.

In conclusion, we will use the fine-tuned BERT model with loss balancing parameter $\beta = 0.99$ and trained for 6 epochs as our solvent sustainability predictor. However, we should note that performance results from

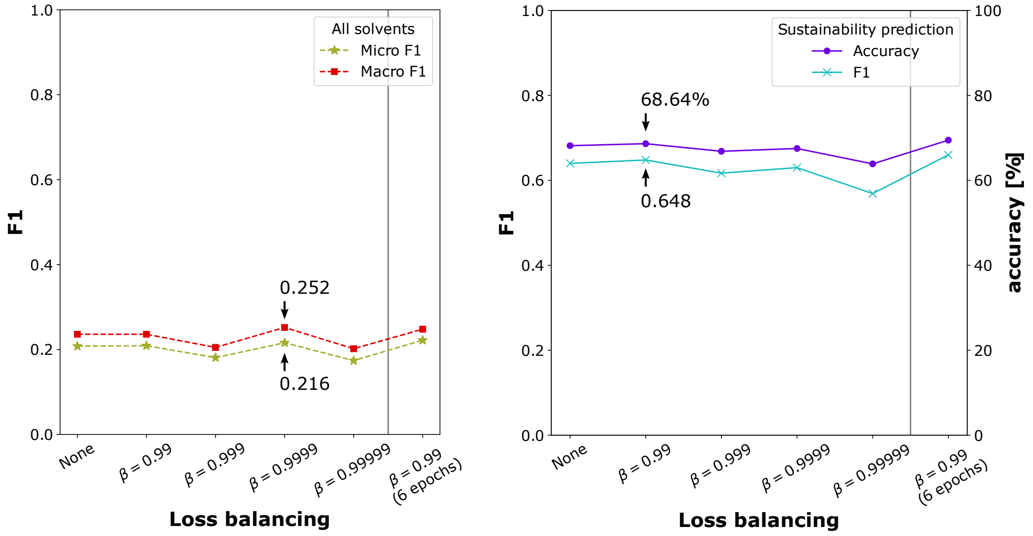


Figure 14: Summary of BERT loss balancing approaches for solvent prediction. For each approach, the best hyper-parameters from Table 9 were used. Models were trained for 3 epochs except for one model with $\beta = 0.99$, which was trained for 6 epochs. Micro and macro F1 scores are calculated following the definition of MLCM (Heydarian *et al.* (2022)). All models were fine-tuned on 2 NVIDIA A100 GPUs

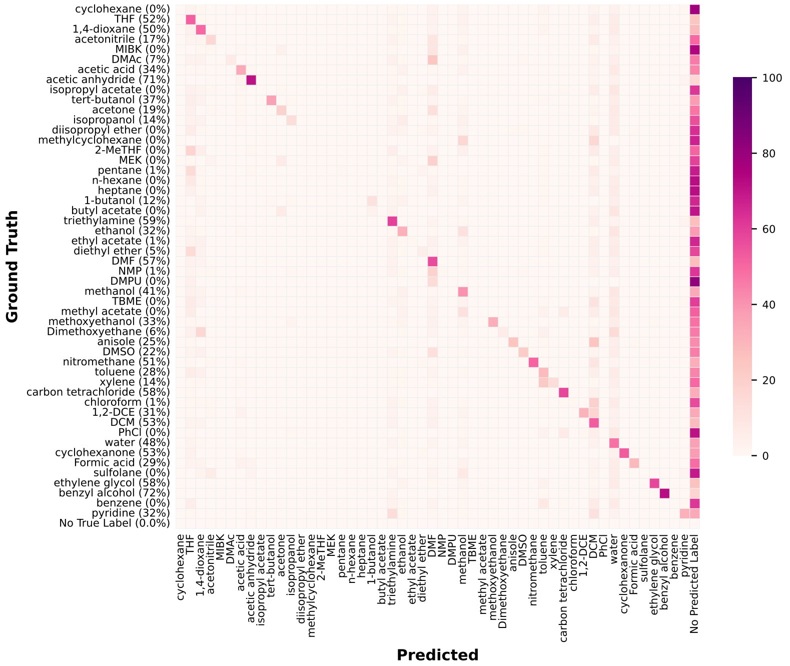


Figure 15: MLCM confusion matrix for BERT model with $\beta = 0.99$ and fine-tuned for 6 epochs. Recall for each solvent is indicated in parenthesis. Amount of no-solvent predictions: 22.62%

Figure 14 may not be final since models could be improved with better hyper-parameters. This is because our hyper-parameter search was limited due to the small fraction of the training set it used and the low amount of trials required to finish the search in a reasonable time. The *timescale* parameter in the learning-rate scheduler may also play an important role in the speed of convergence since it dictates how quickly the learning-rate decays, so including it in the hyper-parameter search may lead to better performance.

4.4.8 Conclusion

Our XGBoost baseline using AP3 fingerprints achieves a micro F1 score of 0.403 and macro F1 of 0.502, while the most performing fine-tuned BERT model with a loss-balancing parameter of $\beta = 0.9999$ achieves a micro F1 of 0.216 and macro F1 of 0.252. Despite this great score difference in multi-label classification, our loss-balanced BERT model with balancing parameter $\beta = 0.99$ achieves a better F1 score of 0.648 in solvent sustainability prediction compared to XGBoost, which achieves an F1 of 0.615. This discrepancy may be because BERT models achieve high accuracy only when predicting common non-sustainable solvents, which define whether or not the reaction is solvent-sustainable (see Definition 4.4.1).

	Solvent sustainability prediction		All solvents	
Model	**Accuracy**	**F1**	**Micro F1**	**Macro F1**
XGBoost (AP3)	72.37%	0.604	**0.403**	**0.502**
XGBoost (DRFP)	**73.60%**	0.615	0.401	0.500
BERT (β = 0.99, 6 epochs)	69.47%	**0.660**	0.222	0.248

Table 10: Summary of solvent prediction models

Due to its better performance in F1 solvent sustainability prediction, we use the fine-tuned BERT with $\beta = 0.99$ in the rest of this work for solvent sustainability scoring of reactions.

4.5 Sustainability metrics

In this section we introduce our definition for an AI-metric, which requires a test-time SMILES augmentation algorithm which will be explained, and validate them. Then, using the techniques introduced in the AI-metric definition, we analyze the capability of our BERT model to correctly classify outlier reactions which have multiple class labels. In particular, we will look at reactions which are both sustainable and non-sustainable. Finally, we introduce a deterministic metric, the Atom Economy (AE).

4.5.1 AI-based metrics

If our AI models were perfect, we could classify reactions as enzymatic or solvent-sustainable without any uncertainty. Unfortunately, AI models rely on statistical approximation where data is noisy and limited, optimization gets stuck in local minima, and the function space from a model's architecture may not contain the solution that best generalizes for the task. Thus, any metrics based on AI models should quantify *how likely* the input is of some class.

In particular, we want our AI-metrics to have the following properties:

- **Normalization**: All metrics are in the range [0, 1], where 0 means the reaction is not sustainable and 1 means the reaction is sustainable.
- **Reliability**: Metrics should quantify how much we can rely on a model's prediction. If the model is perfectly *confident* that a reaction is sustainable and there is no noise in the input data, the metric should be 1. On the other hand, if the model classifies a reaction as sustainable but it is not confident about its prediction or there is a very large amount of noise in the input, the metric should be close to 0.

We will approximate the reliability property, as there is no perfect approach. In fact, the field of uncertainty quantification (UQ) tries to address this problem and there are multiple competing techniques (Abdar *et al.* (2021); Hüllermeier and Waegeman (2021)).

Thus, we base our AI metrics on the probability and confidence in the classification predictions. In particular, given our model with parameters $\boldsymbol{\theta}$, we base our metrics on the likelihood that the input sample $\boldsymbol{x}$ can be classified as some class y,

$$\text{likelihood}_y = p(y \mid \boldsymbol{x},\, \boldsymbol{\theta})$$

We then define the AI-metrics as the maximum expected likelihood, conditioned on our training set $\mathcal{D}$, scaled by the confidence in the prediction over the set of classes C,

$$\text{AI-metric} = \max_{y \in C} \ \mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}\left[\text{likelihood}_y\right] \cdot \text{confidence}_y \quad \in \ [0,\ 1] \tag{1}$$

In our case, C is the set of sustainable classes which, for enzyme sustainability scoring, would be all reaction classes EC.x.x.x.x and, for solvent sustainability scoring, would be all solvents the user defines as sustainable.

The confidence quantifies uncertainty in the prediction and it is defined following the work of Markert *et al.* (2020). Thus, given the minimum and maximum possible standard deviations in the likelihood $\sigma_{\min}$ and $\sigma_{\max}$, the confidence is,

$$\text{confidence}_y = 1 - \frac{\text{stddev}_{p(\boldsymbol{\theta}|\mathcal{D})}\left(\text{likelihood}_y\right) - \sigma_{\min}}{\sigma_{\max} - \sigma_{\min}} \quad \in \ [0,\ 1]$$

Where stddev() denotes the standard deviation. Since the likelihood is a probability, it is a random variable in the range [0, 1], so $\sigma_{\min} = 0$ and $\sigma_{\max} = \frac{1}{2}$.

The expected likelihood and standard deviation are estimated through uncertainty quantification. Following Markert *et al.* (2020), we use Monte-Carlo Dropout (MC Dropout) for epistemic uncertainty estimation, and test-time data augmentation for aleatoric uncertainty estimation.

MC Dropout (Gal and Ghahramani (2016)) is a simple yet effective technique where the dropout layers in a deep neural network, commonly used as regularization during training, are also active during inference. Thus, since a different random subset of the network's nodes is active for each forward inference, it is as if each forward pass uses a different model with different parameters.

In test-time data augmentation (Ayhan and Berens (2018)) we augment the input samples by applying transformations which do not alter the true classification of the sample. In our case, we augment reaction SMILES, which can easily be achieved with RDKit (Landrum *et al.* (2022)) and random permutations. We detail the SMILES augmentation algorithm in the next section.

Combining both techniques, for each SMILES augmentation we can sample with MC Dropout multiple likelihoods. With these samples, we can approximate the expected likelihood as the samples' mean and the standard deviation of the likelihood as the samples' standard deviation. In specific, we augment the SMILES 10 times and do 10 forward MC Dropout inferences for each augmentation.

With Equation 1 we have defined a normalized and reliable metric based on AI models. However, with our estimation of the confidence, the reliability is put into question. By estimating the expected likelihood and the confidence with MC Dropout and test-time data augmentation, the resulting metric quantity is stochastic in nature, thus when presenting it in the following sections we will provide confidence intervals. In what follows we first describe how we augment the SMILES for the confidence estimation and then perform an analysis on the empirical validity of these metrics.

4.5.2 Reaction SMILES augmentation

As stated in the previous section, to compute our AI metrics we need to augment the input reaction SMILES. For that, we devise a SMILES augmentation algorithm which outputs duplicates with low probability.

We divide our augmentation algorithm in four conceptual steps, outlined in Figure 16.

First, we augment each compound in the reaction separately by drawing at random from RDKit's `Chem.MolToSmiles()` function, which can return duplicates. The number of augmentations for each compound, M, has to be large enough so that we can later retrieve our desired k reaction SMILES by permuting the augmentations, but small enough so that the execution time of the algorithm is reasonable. We arbitrarily set $M = \lfloor 20 \cdot k^{1/\text{compounds}} \rfloor$, where *compounds* is the number of compounds in the reaction and $\lfloor \cdot \rfloor$ is the floor function. The intuition is that in this way we get $M^{\text{compounds}} = 20^{\text{compounds}} \cdot k$ total permutations, which is a constant multiple of our desired k permutations. Each compound permutation corresponds to a different reaction SMILES (see step 2. in Figure 16).

In practice, we draw $A \cdot \min\{A^2,\ M\}$ compound augmentations from `Chem.MolToSmiles()` removing duplicates, where A is the number of atoms in the compound, and then keep M of those augmentations. If there are less than M augmentations we duplicate the augmentations until there are M of them. We observed that this approach reduced the number of duplicates, although the number of draws is arbitrary.

Then, we notice that we can uniquely index each compound permutation. The largest index is $M^{\text{compounds}} - 1$.

With this observation, we draw k indices from 0 to $M^{\text{compounds}} - 1$ without replacement, each corresponding to a different reaction SMILES augmentation.

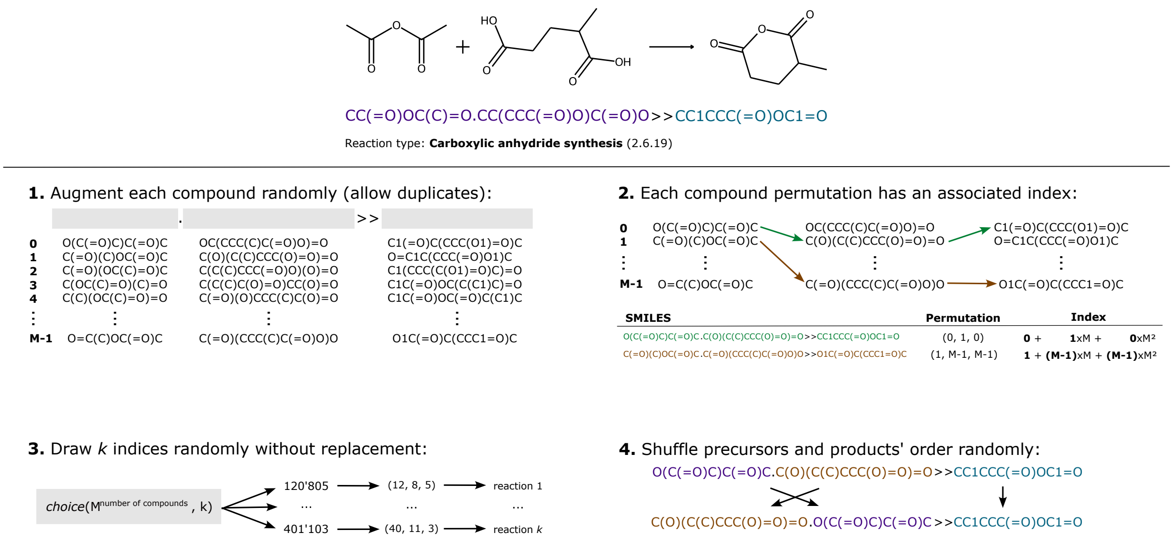

HO
O
OH
+
CC(=O)OC(C)=O.CC(CCC(=O)O)C(=O)O>>CC1CCC(=O)OC1=O
Reaction type: Carboxylic anhydride synthesis (2.6.19)
1. Augment each compound randomly (allow duplicates):
. >>
0 O(C(=O)C)C(=O)C OC(CCC(C)C(=O)O)=O C1(=O)C(CCC(O1)=O)C
1 C(=O)(C)OC(=O)C C(O)(C(C)CCC(O)=O)=O O=C1C(CCC(=O)O1)C
2 C(=O)(OC(C)=O)C C(C(C)CCC(=O)O)(O)=O C1(CCC(C(O1)=O)C)=O
3 C(OC(C)=O)(C)=O C(C(C)C(O)=O)CC(O)=O C1C(=O)OC(C(C1)C)=O
4 C(C)(OC(C)=O)=O C(=O)(O)CCC(C)C(O)=O C1C(=O)OC(=O)C(C1)C
M-1 O=C(C)OC(=O)C C(=O)(CCC(C)C(=O)O)O O1C(=O)C(CCC1=O)C
2. Each compound permutation has an associated index:
0 O(C(=O)C)C(=O)C OC(CCC(C)C(=O)O)=O C1(=O)C(CCC(O1)=O)C
1 C(=O)(C)OC(=O)C C(O)(C(C)CCC(O)=O)=O O=C1C(CCC(=O)O1)C
M-1 O=C(C)OC(=O)C C(=O)(CCC(C)C(=O)O)O O1C(=O)C(CCC1=O)C
SMILES Permutation Index
O(C(=O)C)C(=O)C.C(O)(C(C)CCC(O)=O)=O>>CC1CCC(=O)OC1=O (0, 1, 0) 0 + 1xM + 0xM²
C(=O)(C)OC(=O)C.C(=O)(CCC(C)C(=O)O)O>>O1C(=O)C(CCC1=O)C (1, M-1, M-1) 1 + (M-1)xM + (M-1)xM²
3. Draw k indices randomly without replacement:
choice(M^number of compounds , k)
120'805 (12, 8, 5) reaction 1
... ...
401'103 (40, 11, 3) reaction k
4. Shuffle precursors and products' order randomly:
O(C(=O)C)C(=O)C.C(O)(C(C)CCC(O)=O)=O>>CC1CCC(=O)OC1=O
C(O)(C(C)CCC(O)=O)=O.O(C(=O)C)C(=O)C>>CC1CCC(=O)OC1=O

Figure 16: Reaction SMILES augmentation process

Finally, for all drawn reaction SMILES we randomly shuffle the precursors' order and the products' order.

Of course, since `Chem.MolToSmiles()` outputs duplicates, some reaction SMILES may be duplicated, but it is a rare instance and we have mainly observed it in short input SMILES like: `c1cc[nH]c1>>c1ccc2ccccc2c1`.

We will now go over one important challenge we encountered when drawing the indices (step 3. in Figure 16).

To draw our k indices without replacement we first tried NumPy's `numpy.random.choice()` function. Unfortunately, to draw *without replacement* `numpy.random.choice()` creates an array of size $N = M^{\text{compounds}}$ which requires an infeasible amount of memory for reactions with many compounds. However, when sampling *with replacement* the same function only requires $O(k)$ memory. Thus, we devise a memory efficient algorithm to sample *without replacement*:

Algorithm 4.5.1 An algorithm to efficiently draw without replacement. $k > 0$ items are taken without replacement from a universe of size N. $choice(N, k)$ samples k items with replacement with a space complexity of $O(k \log N)$:

```
Require: $N \geq (2 + \sqrt{2}) \cdot k$
$X \leftarrow \{choice(N, 1)\}$
while $|X| \neq k$ do
    $s \leftarrow choice(N, 1)$
    if $s \notin X$ then
        $X \leftarrow X \cup \{s\}$
    end if
end while
return $X$
```

This algorithm uses $O(k)$ memory and, for our purposes, it draws $1 + 1.003k$ indices with probability less than $20^{-\text{compounds}}$, which is 0.25% for two compounds. A general proof of this last statement and a complete analysis of the algorithm can be found in Appendix B.

Overall, this augmentation algorithm could be improved by modifying `Chem.MolToSmiles()` so that we can draw compound SMILES without replacement and by analyzing what values of M allow us to retrieve a statistically diverse set of augmented reaction SMILES.

4.5.3 Validating AI metrics

We have seen that our definition for AI-metrics is approximated through Monte-Carlo (MC) Dropout and test-time reaction SMILES augmentation. Thus, our AI-metrics are stochastic and an approximation of the real definition, Equation 1.

To validate these AI-metrics we look at their capability to discern sustainable versus non-sustainable reactions. For this purpose, we must quantify non-sustainability, which is easily achieved by inverting the set of classes C in Equation 1. In this way, we have the non-sustainability counterpart for each sustainable AI-metric. We plot both values for each reaction for each of our AI-metrics: enzyme sustainability, and solvent sustainability.

Before discussing the results, we note that AI-metrics require an AI model. For enzyme sustainability, we choose our loss-balanced BERT model from Section 4.3.1, which achieves an F1 score of 0.978 in enzyme prediction (see Table 6). For solvent sustainability, we choose our BERT model fine-tuned for 6 epochs with loss-balance parameter $\beta = 0.99$, which achieves an F1 score of 0.660 in solvent sustainability prediction (see Table 10).

Figure 17 shows how the enzymatic AI-metric successfully separates sustainable and non-sustainable classes. In fact, any value of the AI-metric above 0.1 accurately isolates enzymatic reactions from the rest. The unrecognized class 0.0 in this case is neither sustainable nor non-sustainable, thus it was excluded from the set of classes C in Equation 1 for the sustainable AI-metric calculation and its non-sustainable counterpart.

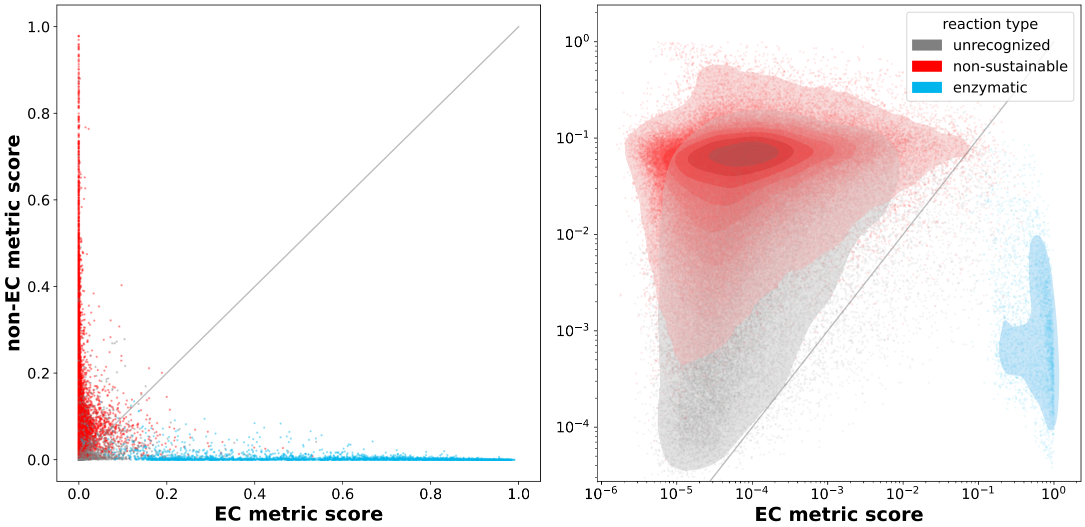


Figure 17: Enzymatic sustainability metric on validation set. Unrecognized reactions of class 0.0 are included in grey. Dense regions are marked on the right log-log plot through kernel density estimation (KDE)

Figure 18 shows the results for the solvent sustainability AI-metric. Following Definition 4.4.1, sustainable reactions are those which do not contain a non-sustainable solvent and, for this work, sustainable solvents are those defined in Table 7. The plot shows how this AI-metric does not clearly separate the solvent-sustainable reactions, which may be due to the low performance of the BERT solvent predictor. However, for AI-metric values above 0.75 the model is able to better isolate solvent-sustainable reactions from the rest. In Appendix A you can find the same plot when defining a reaction solvent-sustainable if it contains *any* sustainable solvent (Figure A.2).

To conclude, our enzyme metric can accurately discern enzyme-sustainable reactions even for low metric scores, while the solvent-sustainability metric can only distinguish solvent-sustainable reactions for very high scores. In particular, with our current models the solvent-sustainability metric is unreliable for metric scores up to 0.75.

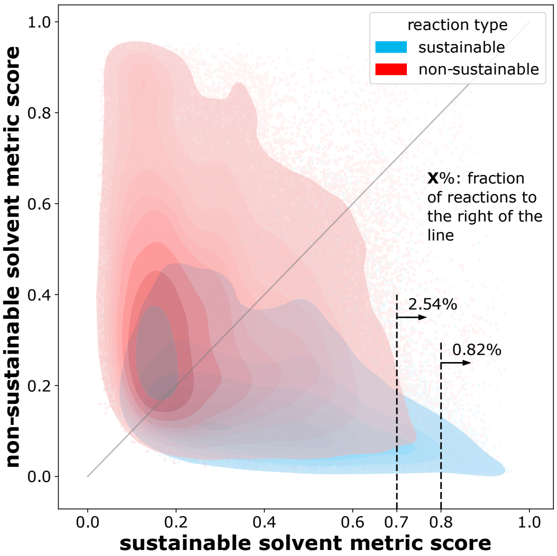


Figure 18: Solvent-sustainability metric on validation set. The fraction of reactions with a sustainable score > 0.7 and > 0.8 are indicated. Dense regions are marked through kernel density estimation (KDE)

4.5.4 Uncertainty Quantification (UQ) for reactions in multiple classes

When combining the Pistachio and ECREACT datasets, 32 reactions SMILES are present in both datasets. These are Pistachio synthetic reactions where we alternatively have the option to catalyze them with enzymes. We will study whether we can infer the secondary class label in these reactions with our enzyme model, which is trained on a single label task. To achieve this, we use Uncertainty Quantification (UQ). In particular, we use the Monte-Carlo (MC) Dropout and test-time data augmentation techniques introduced to approximate the AI-metrics in Section 4.5.1. These approaches allow us to estimate the likelihood distribution for all of the reaction classes and so we can see the most likely predictions and their uncertainty.

We place 16 of the 32 multi-label reactions in the training set and the other 16 in the test set. Analogously to the AI-metric computation, we do 10 MC Dropout forward passes for each SMILES augmentation, where we augment the SMILES 10 times, for a total of 100 likelihood samples. Our model is BERT with loss-balance parameter $\beta = 0.99999$ and F1 score of 0.978 on enzyme prediction (see Table 6).

Figure 19 shows the UQ of the likelihood for four example reactions in the training set and four example reactions in the test set.

We make the following note-worthy observations:

- For 16 out of 16 training set cases the class with highest expected likelihood is enzymatic, while 12 out of 16 test set cases are most expected as enzymatic. The four cases not expected as enzymatic are expected as unrecognized (class 0.0).
- The top-1 class is the correct enzymatic class in 16 out of 16 training set cases, and 12 out of 16 test set cases. In only one test case the true enzymatic class does not appear in the top-5.
- The true Pistachio class appears in the top-5 in 10 out of 16 training set cases, and 10 out of 16 test set cases.

In conclusion, when there is ambiguity about the reaction class, the model gives priority in its prediction to enzyme-catalyzed classes while keeping a high accuracy. This means that our model may prioritize sustainability, however a proper analysis with many more multi-label reactions would be required to confirm this claim.

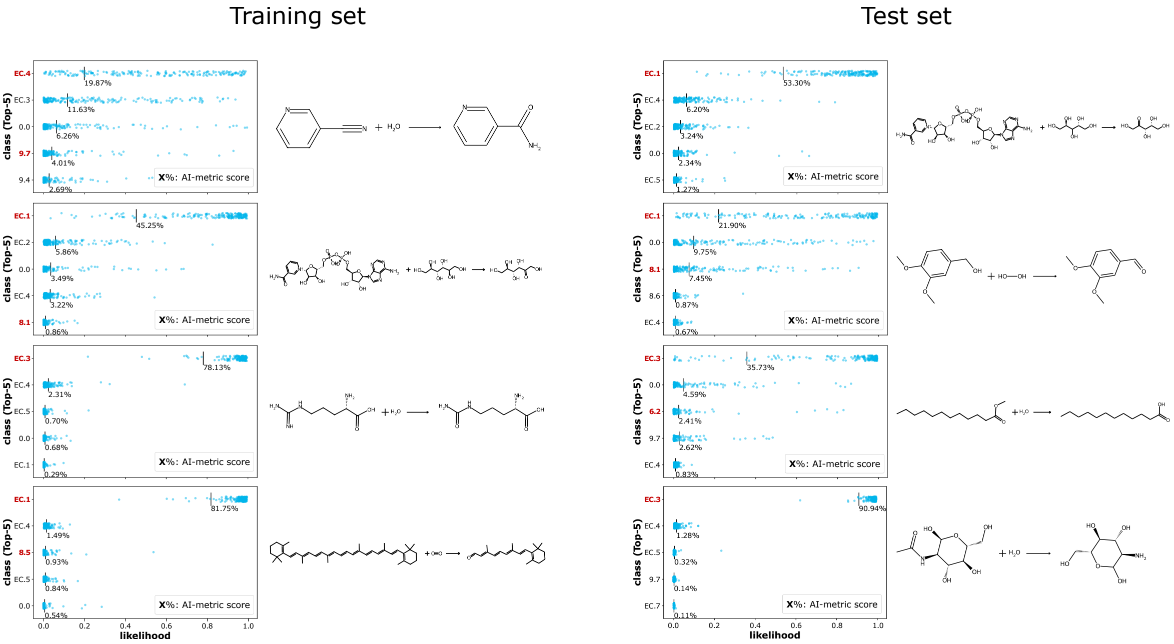


Figure 19: Eight example multi-label reactions in the training set and test set and their corresponding likelihoods for the top-5 predicted classes. True classes are indicated in red, if present

4.5.5 Atom Economy

Atom Economy (AE) is a common greenness metric used when estimating sustainability (Sheldon (2018); Weber *et al.* (2021)). It is usually defined as the ratio between the products' total molecular weight and the reactants' total molecular weight. It should be a metric between 0% and 100%; however, the following problems arise due to the imperfect nature of our Pistachio+ECREACT dataset:

- Reactions have incomplete reactant and product information. Thus, a reaction may have a bigger product molecular weight than the reactants', resulting in an AE score above 100%.
- Reactions do not contain stoichiometry information (how many repetitions of each compound is necessary for the reaction to happen). This can again result in an AE score above 100%.

To resolve these issues, we modify the definition of atom economy:

$$AE = 1 - \frac{|\text{mass in products} - \text{mass in reactants}|}{\max\{\text{mass in products},\ \text{mass in reactants}\}}$$

The intuition is that AE should quantify how much atomic mass is "wasted" in the reaction process. This definition also ensures that the AE score is between 0% and 100%. Similarly to the original definition, an AE score of 0% implies that no product is made and 100% implies that all atoms in the reactants appear in the products. Furthermore, if the products' mass is less than or equal the reactants' mass this AE score is equivalent to the original definition. Finally, unlike the original definition, 0% can also imply the pathological case that the product is generated from no reactants.

We use this definition of Atom Economy as an additional metric alongside the enzyme and solvent AI-metrics.

4.5.6 Conclusion

We have outlined the three different reaction metrics: enzymatic sustainability, solvent-sustainability, and Atom Economy (AE).

These metrics can already help integrate some form of sustainability automation into synthesis planning workflows. However, we will see in the next section that these metric quantities can be misleading and so they serve more as a guide for sustainability that a human expert must interpret.

4.6 Limitations and AI interpretability tools

In this section we show some dangerous pitfalls of blindly relying on AI predictions.

Our BERT reaction classifier from Section 4.3.1 achieves a macro F1 score of 0.901 and accuracy of 93.95%, however it missclassifies the reaction presented in the top-row of Table 11 as being catalyzed by an isomerase enzyme (class EC.5.x.x.x), when in reality it is of a non-sustainable class "*carboxylic acid to acid chloride*" (class 9.3.1). Furthermore, it misclassifies it with a metric score of 0.52 ± 0.07, where we computed the metric ten times.

This is not an isolated example of this type. We can look at similar reactions in terms of the Dice coefficient (Dice (1945)) on the AP3 reaction fingerprints. The Dice coefficient has been used in previous work to cluster AP3 fingerprints (Schneider *et al.* (2015)), and it is equivalent to:

$$\text{Dice}(\boldsymbol{a}, \boldsymbol{b}) = \frac{1}{N} \cdot \sum_{i=1}^{N} \mathbf{1}_{a_i = b_i}$$

Where $\boldsymbol{a}$, $\boldsymbol{b}$ are fingerprint vectors, N is the length of the fingerprints (in our case 2048), and $\mathbf{1}_{(\cdot)}$ is the indicator function.

Table 11 shows three other reactions with high Dice similarity. All three reactions, like the original, have a true class of 9.3.1 and the model predicts they are of class EC.5.x.x.x. Their sustainability metrics are 0.42 ± 0.02, 0.29 ± 0.01, and 0.48 ± 0.00.

Therefore, even if our metrics are close or above 0.50 it does not mean that the reaction is enzyme-sustainable.

For this reason we provide easy-to-use AI interpretability tools in our package to assess the deep neural networks' predictions. We continue by applying these tools to study our missclassified prediction.

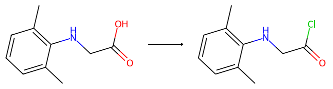
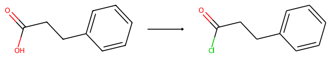
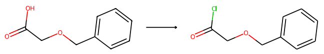
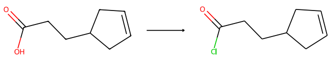
Reaction	Dice similarity	True class	Predicted class (prob.)	Metric score
	(original)	9.3.1	EC.5.x.x.x (97%)	0.52 ± 0.07
	0.996	9.3.1	EC.5.x.x.x (94%)	0.42 ± 0.02
	0.992	9.3.1	EC.5.x.x.x (82%)	0.29 ± 0.01
	0.992	9.3.1	EC.5.x.x.x (97%)	0.48 ± 0.00

Table 11: Misclassified reaction (top-row) and similar reactions in terms of Dice coefficient on the AP3 fingerprint. The likelihood for the predicted class is in parenthesis. The metric score is calculated ten times for each reaction and averaged. The error is the standard deviation

The first interpretability tool uses uncertainty quantification (UQ) to estimate likelihood uncertainty. This tool was used in Section 4.5.4 to assess if our model could identify secondary reaction classes in reactions with multiple class labels. For our particular example, Figure 20 shows the top-5 reaction classes with highest likelihood. As it can be seen, all five classes are enzymatic and class 9.3.1 is nowhere to be seen, thus the model predicts that the non-sustainable class is unlikely.

The second interpretability tool is attention visualization. We can plot attention matrices for each head in each layer. Figure 21 shows the attention matrices for the first and last layers of our BERT model. The figure also displays the average attention scores per atom token per head in the first layer, and the attention received by the `[CLS]` token in the last layer. This kind of attention visualization on the `[CLS]` token has been used before to demonstrate that BERT can learn the atoms that take part in a reaction and use this information for

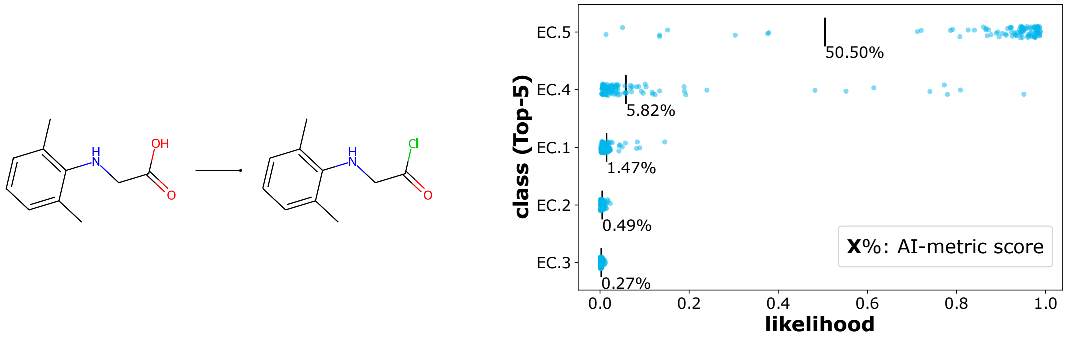
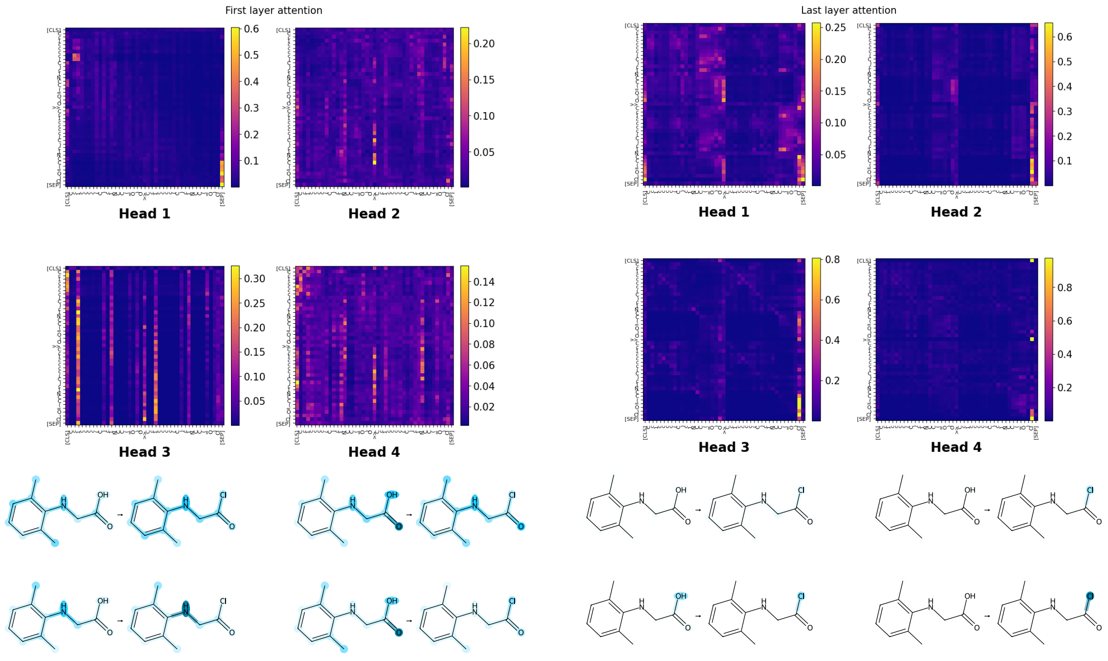


Figure 20: Top-5 reaction classes for missclassified example

its prediction (Schwaller *et al.* (2021b)). However, we can see in Figure 21b how, despite the fact that the reaction is confidently missclassified, the model correctly places the [CLS] token attention on the exchanged atom groups OH and Cl, which is precisely the rule that describes reactions in class 9.3.1.



(a) Average attention per-token in the first layer

(b) Last layer's attention for token [CLS]

Figure 21: BERT attention for the first and last layers. Reactions are shown for each head: head one (top) to head four (bottom). Reaction atoms are colored according to their attention score

In fact, attention, although useful to visualize the inner workings of the model, is misleading and not very meaningful to understand the input's influence on a BERT model prediction (Bastings and Filippova (2020)). This is because attention in the first layer can only explain the activation for the next layer, while the token embeddings in the last layer before classification or any layer in-between do not encode the individual token, but rather a global property in the text. Thus, token attention in Figure 21b does not explain the influence of each input token in the final prediction. For that, we can look into our next interpretability tool, the method of Integrated Gradients (Sundararajan *et al.* (2017)).

We used Integrated Gradients in Section 4.3.1 to discover an adversarial attack on our model. When applying it to all four missclassified reactions we observe that the exchanged atom groups are not the only atoms that influence the prediction. In fact, the exchanged atom group OH now has a negative attribution, and instead the model is influenced by other structures in the reactant. We can conclude that reactions with structures such us those with positive attribution in Figure 22 will be confidently missclassified as class EC.5.x.x.x.

In this section we presented different techniques to debug our AI models and applied them to a missclassi-

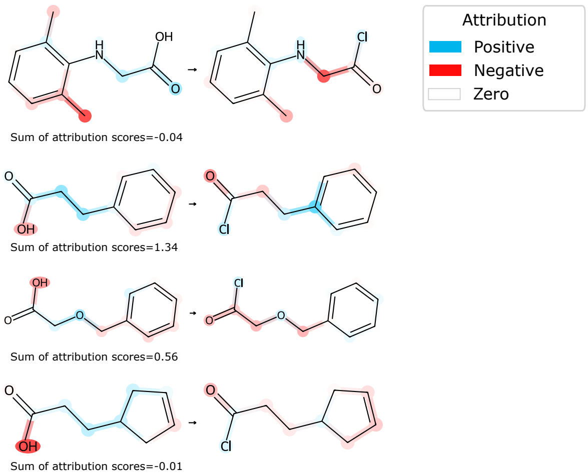

Attribution
🟦 Positive
🟥 Negative
□ Zero
Sum of attribution scores=-0.04
Sum of attribution scores=1.34
Sum of attribution scores=0.56
Sum of attribution scores=-0.01

Figure 22: Importance attribution on missclassified reactions

fied example. Due to this problem on AI reliability, we remark that sustainability assessment using AI models cannot be completely automated, even though these models could help accelerate sustainable development, and there should be a human to verify the results.

4.7 Computer-Aided Synthesis Planning (CASP) Use Case

In this section we bring our metrics into practice and discuss how they can be integrated for reaction *pathway* generation.

Chemical reaction *pathways* or *routes* are paths of compounds and reactions which describe the reactions that have to take place to synthesize a target compound from starting compounds. Starting compounds are also called the *leaves* of the routes. We say that a compound is *in stock* if it is part of our valid starting compound dataset.

The generation of pathways, also called retrosynthesis analysis, is a particular area in Computer-Aided Synthesis Planning (CASP) in which we predict the reaction route backward from the target compound to the starting compounds.

One approach to generate pathways uses Monte-Carlo tree search (MCTS), which we explain in the following section. Then, we describe the scoring function we introduce into this algorithm by combining an approximation of the routes' cost and our metrics. Finally, we use the MCTS algorithm implemented in AiZynthFinder (Genheden *et al.* (2020b)) to analyze the impact of our scorer on 127 target compounds extracted from the WHO Model List of Essential Medicines (EML) from 1977 (WHO Expert Committee (1977)).

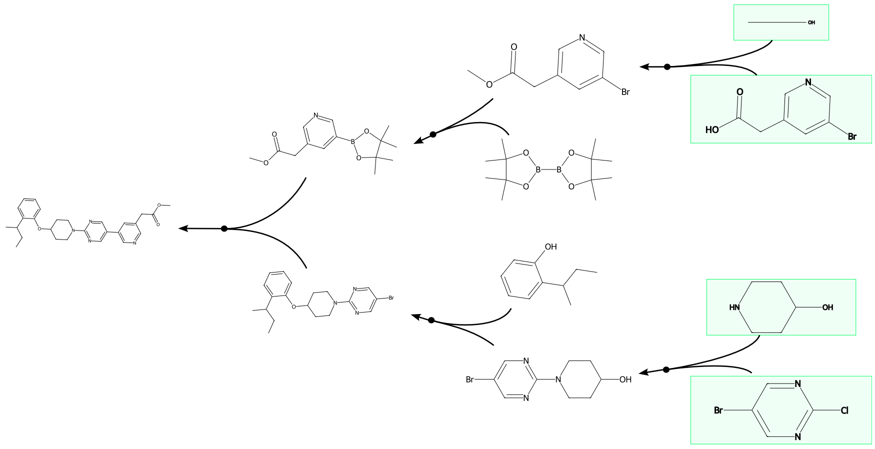


Figure 23: Example *pathway* or *route*. This pathway is part of the PaRoutes dataset (Genheden and Bjerrum (2022)) and was extracted from the US20100004245A1 patent. Solid circles represent reactions. Starting compounds are indicated in green

4.7.1 Monte-Carlo tree search for CASP

The approach we will focus on for computer-aided retrosynthesis is the Monte Carlo tree search (MCTS) algorithm implemented in AiZynthFinder (Genheden *et al.* (2020b)).

In Segler *et al.* (2018), which was later implemented in AiZynthFinder, the authors combine MCTS with deep reinforcement learning (RL) and symbolic rules to achieve state-of-the-art performance. Their MCTS algorithm finds new pathways in each iteration until the limit number of iterations or a time limit are reached. Each iteration has the following steps:

1. **Selection**: Starting from the target compound for which to generate the pathway, called the *root node*, the currently generated *node tree* is traversed following a greedy policy. A *node* or *node state* s is a collection of starting compounds in a reaction route. The *node tree* is the currently generated tree of nodes in the MCTS iteration.

   The node selection policy is repeated until a leaf node is reached by traversing different state-action pairs in the node tree (s_t, a_t), where actions a_t are reactions which have as product some compound in s_t. This policy greedily chooses the reaction a_t which maximizes for all a the linear combination of a value function $Q(s_t, a)$ and a prior probability $P(s_t, a)$. This prior probability can be calculated with a deep neural network.

2. **Expansion**: A neural network, called the **expansion policy**, predicts the single-step retrosynthesis for each molecule in the selected node. By repeating the retrosynthesis multiple possible sets of reactants are generated for each molecule, which are then filtered using a second neural network, the **in-scope filter**.

   These reactants correspond to new nodes in the MCTS tree. The most promising node is selected for the next step.

3. **Rollout**: Multi-step retrosynthesis is performed using a third neural network, the **rollout policy**, from the chosen node until the maximum depth is reached or all molecules are solved. In AiZynthFinder, the rollout policy is the same as the expansion policy by default.

4. **Update**: The value for each MCTS node state and selected action pair is updated. In particular, the authors modify the Q-learning algorithm (Watkins (1989)) with the following state-action value function update:

   $$Q(s, a) \leftarrow \frac{1}{N(s, a)} \sum_{i=1}^{n} \mathbf{1}_i(s, a) \cdot z_i \cdot W(b_i)$$

   Where $N(s, a)$ is the total visit count for the state-action pair (s, a), n is the current iteration, $\mathbf{1}_i(s, a)$ is the indicator function which equals one if (s, a) was visited in the ith iteration, z_i is the reward received in iteration i, and $W(b_i)$ is a custom objective function which takes as input the branch of node states b_i selected at iteration i.

Since this algorithm can use three different networks (the expansion policy, rollout policy and in-scope filter) the authors name it *3N-MCTS*.

After the iteration loop finishes, MCTS has generated a tree with thousands of possible pathways. In AiZynthFinder, these pathways are sorted according to some post-processing scorer and the top-5 pathways are extracted.

4.7.2 Cost-effective scoring of pathways with the Route Cost scorer

One of the pillars for sustainable retrosynthesis planning is finding competitive routes that are cost-effective so they can realistically be implemented in industry (Weber *et al.* (2021); Sheldon (2018)).

Although our AI-metrics do not provide cost estimation, the synthesis planning tool AiZynthFinder (Genheden *et al.* (2020b)) does include a scorer based on the cost of stock molecules and fixed cost of reactions. This scorer has been used in Badowski *et al.* (2019) to generate cost-effective and chemically diverse reaction pathways.

This *Route Cost* scorer defines a compound's cost recursively from the starting compounds. The cost for the starting compounds is retrieved through a compound stock database, which by default is ZINC (Irwin and Shoichet (2005)). Then, the Route Cost scorer defines all non starting compounds c cost as:

$$\text{cost}(c) = \min_{r \in \text{pred}(c)} \text{cost}(r)$$

Where r are reactions and $\text{pred}(c)$ is the set of reactions that generate compound c. A reaction's cost is defined as:

$$\text{cost}(r) = \omega_{\text{cost}}(r) + \sum_{c \in \text{pred}(r)} \frac{\text{cost}(c)}{\text{yield}(r, c)}$$

Where $\omega_{\text{cost}}(r)$ is the operational cost of carrying out reaction r, $\text{pred}(r)$ are the reactants of r, and $\text{yield}(r, c)$ is the product yield of reaction r with respect to the reactant c. The less the yield, the more amount of reactant is necessary to synthesize the product, so the higher is the cost. Note how the Route Cost scorer inherently penalizes pathways with high depth due to the fixed costs and inverse yield scaling.

Since AiZynthFinder does not predict fixed costs of reactions nor product yields, the Route Cost scorer sets a constant value for them for all reactions. In particular, by default it sets $\omega_{\text{cost}}(r) = 1$ and $\text{yield}(r, c) = 0.8$ for all r and c. We keep these defaults in our analysis.

4.7.3 Pathway sustainability scoring

We have defined our metrics in the context of individual reactions (see Section 4.5).

To generalize to pathway sustainability scoring we simply define the score for some pathway P in terms of some metric $m(r)$, which takes as input a reaction r, as the mean over the reactions:

$$S_{P,m} = \frac{1}{|P|} \sum_{r \in P} m(r) \quad \in \ [0,\ 1]$$

Where $|P|$ is the number of reactions in the pathway.

As explained in Section 4.7.1, Monte-Carlo tree search (MCTS) generates thousands of nodes, each corresponding to a valid pathway, and then AiZynthFinder selects the top-5 pathways according to some objective score during post-processing (see Figure 24).

We want to introduce both cost-effectiveness and our metrics into this objective score, thus we use the Route Cost score defined in the previous section in combination with our metrics.

When selecting our top-5 pathways from MCTS nodes, the objective score should take into account that sustainable reactions can offset their cost individually. Thus, in our objective score we use the sum of our metrics over the reactions instead of the mean, which is equivalent to: $\sum_{r \in P} m(r) = |P| \cdot S_{P,m}$. Then, the objective score is the weighted sum of our scores $S_{P,m}$ and the Route Cost score (defined in the previous section):

$$\text{objective\_score}(P) = w_{\text{enzyme}} \cdot |P| \cdot S_{P,\text{enzyme}} + w_{\text{solvent}} \cdot |P| \cdot S_{P,\text{solvent}} + w_{\text{AE}} \cdot |P| \cdot S_{P,\text{AE}} - w_{\text{RC}} \cdot \text{RC}(P) \quad (2)$$

Where $S_{P,\text{enzyme}}$ is the score based on our enzymatic AI-metric, $S_{P,\text{solvent}}$ is the score for our solvent-sustainable AI-metric, $S_{P,\text{AE}}$ is the score for our Atom Economy (AE) metric, and $\text{RC}(P)$ is the Route Cost score for pathway P. w_m is the weight attributed to the score with metric m and w_{RC} is the weight for the Route Cost scorer.

We subtract the Route Cost score since AiZynthFinder tries to maximize the objective objective_score(P) and we aim to generate more cost-effective pathways.

Since we combine all metrics into a single score for the post-processing algorithm, it is useful to interpret the meaning of the different weights w_m. First, we note that the reaction cost scorer sets by default a cost of "1 unit" to carry out any reaction and also "1 unit" as the default cost for any source compound. Then, we note that $|P| \cdot S_{P,m}$ approximately counts the amount of sustainable reactions in the pathway P according to the metric m. Thus, we can interpret our metrics in terms of the Route Cost's "unit cost". In particular, if we set the Route Cost weight $w_{\text{RC}} = 1$ and a weight of $w_m = 1$ for our metric m, then we imply that achieving a reaction score of 1 with that metric is equivalent to offsetting "1 unit" of costs. For example, the fixed cost of a reaction is balanced out if we are confident that it can be made enzyme-catalyzed. Thus, if $w_{\text{RC}} = 1$ our metric weights w_m are in units of cost as defined by the Route Cost scorer.

We hypothesize that increasing our metric weights w_m will result in more sustainable pathways generated by the MCTS algorithm, but their Route Cost will also increase.

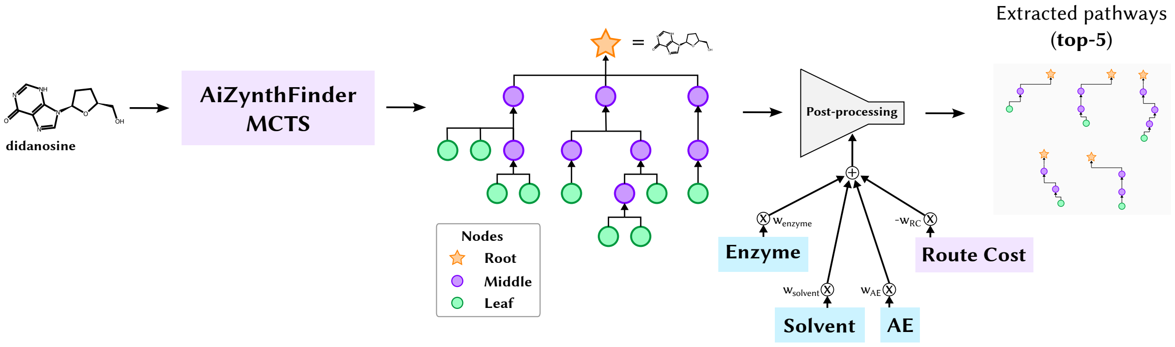


Figure 24: Diagram of the pathway generation process for a target compound from the WHO Essential Medicines List (EML) (WHO Expert Committee (1977)). The MCTS algorithm generates a tree with thousands of nodes, from which the top-5 pathways are extracted during post-processing using some objective score. Our objective score is a weighted sum of the different sustainability metrics

4.7.4 Assessing metric influence on pathway generation

We are now ready to use our metrics to influence pathway generation.

To quantify the performance of our metrics, we generate pathways using AiZynthFinder (Genheden *et al.* (2020b)) on target compounds extracted from the WHO Model List of Essential Medicines (EML) from 1977. 127 targets from this list were extracted and used in previous work, where the authors devised an algorithm to minimize the set of starting materials required to synthesize all 127 compounds (Gao *et al.* (2020a)). We will use the same 127 targets for our study.

We generate pathways using the default expansion policy model from AiZynthFinder (explained in Section 4.7.1). In particular, the model is a template-based model trained on the USPTO reaction dataset (Genheden (2022)), i.e. it predicts the reactants from a target product by combining a deep learning model and known chemical reaction transformations. The rollout policy model is the same as the expansion policy model, and we do not specify any in-scope filter to prevent a possibly biased generation of non-sustainable reactions.

For each target molecule, we run Monte-Carlo tree search (MCTS) with a limit of either 200 iterations or 400 seconds and we set the maximum depth per pathway to 10 reactions. This maximum pathway depth was also used in Gao *et al.* (2020a). We increased these parameters from AiZynthFinder's default 100 iterations, 100 seconds and maximum depth of six to allow the MCTS algorithm to retrieve higher quality pathways. This is because with more iterations and depth the MCTS algorithm covers a larger subset of the chemical reaction space and there are higher chances of finding the optimal pathway for each target.

The default stock database in AiZynthFinder is ZINC (Irwin and Shoichet (2005); Genheden *et al.* (2020a)), where it searches for starting compound's availability on the market and their price. However, when using this limited database we noticed that it is very difficult for MCTS to find pathways where most starting compounds are in stock, even if we increase the number of iterations to 1'000 and the time limit to 20 minutes. Therefore, we instead follow Gao *et al.* (2018) and define as our starting compounds all those with less than 10 carbon atoms, three nitrogen atoms, and five oxygen atoms.

Regarding the cost of the starting compounds, the Route Cost scorer (explained in Section 4.7.2) will set the cost for compounds not found in the stock database to 10 units instead of the default one unit. Since MCTS commonly generates pathways where some starting compounds are not in stock, this results in some routes with very large Route Cost. This large Route Cost almost completely overshadows our metrics. We change this default "not-found" cost to two units, so a starting compound not in stock has twice the cost than the default. This lowers the routes' costs and allows our metrics to have a bigger impact in the search.

However, pathways with starting compounds not in stock are undesirable, since it may not be possible or economically viable to utilize these compounds as starting materials in a real-world setting. We take this into account and for each target compound we choose the pathway with the lowest number of starting compounds not in stock out of the top-5 pathways selected in post-processing.

To assess the impact of each metric in the MCTS pathway generation we perform an ablation study where we isolate the effect of each of our metrics by setting the weights of the other metrics to zero. In particular, we look at how the metric score changes for each pathway of the 127 WHO EML targets by calculating the difference between the metric scores achieved using some weights versus the metric score achieved for some baseline weights. In our case, the baseline weights are those where we only optimize for cost: zero-weight for all of our metrics, and a weight of one for the Route Cost scorer. Table 14 shows the baseline average pathway metric scores for all 127 WHO EML target compounds. The baseline enzyme metric score is 11% on average, while the average solvent-sustainability score is 21%, 84% for Atom Economy (AE), and 6 cost units for the Route Cost. Note that our definition of cost as presented in Section 4.7.2 is currency-agnostic, thus we do not specify any particular units.

Input metric weights				Baseline scores			
Enzyme	**Solvent**	**AE**	**Route Cost**	**Enzyme [%]**	**Solvent [%]**	**AE [%]**	**Route cost**
0	0	0	1	11 ± 12	21 ± 6	84 ± 14	6 ± 5

Table 12: Baseline metric scores. Targets are the WHO Essential Medicines List (EML) of 127 target compounds. Errors are the standard deviations

Figure 25 shows the impact of the different metric weights in the pathway scores. Due to our definition of the post-processing objective (Equation 2), positive weights mean that we want more sustainable pathways while negative weights mean that we want less sustainable pathways. As it can be seen, the scores for almost all pathways change, thus it is clear that our metrics have an impact in the retrosynthesis algorithm. However, in most cases the sustainability scores do not significantly increase when setting positive weights for our metrics. The enzymatic metric is the exception, as its average score increases by 7.9% if we set $w_{\text{enzyme}} = 1$ and all other weights to zero, including the Route Cost weight. The trade-off comes with the Route Cost score in this case, which increases on average by 57 cost units.

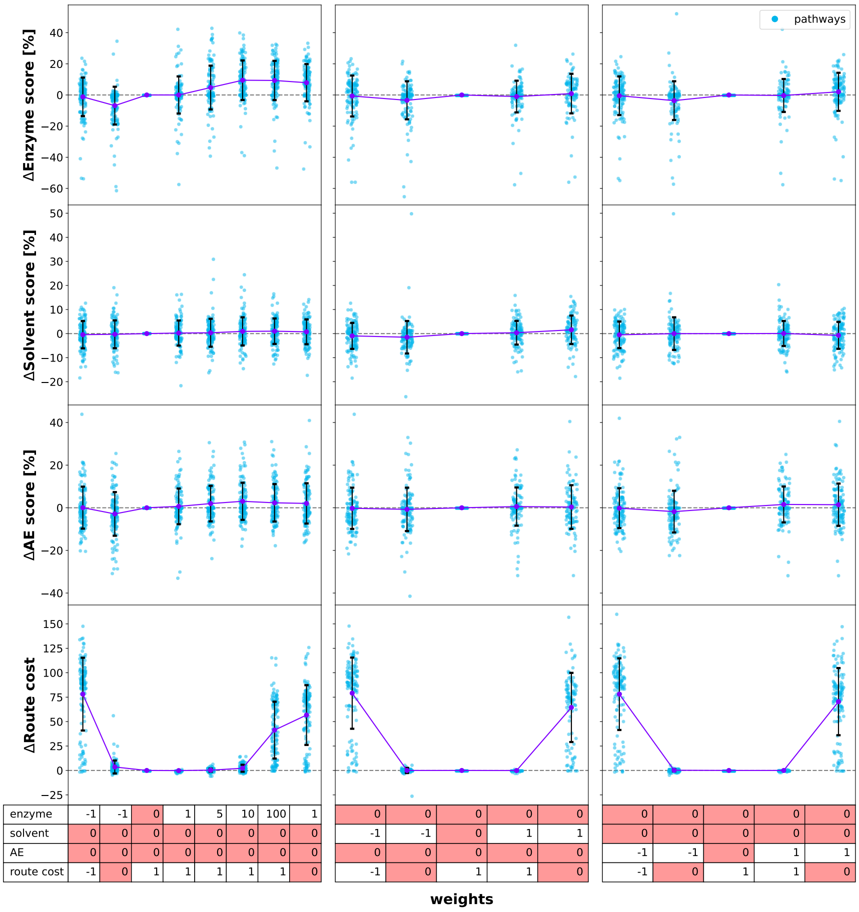


Figure 25: Change in pathway metric scores from the baseline for different metric weights. A negative weight on one of our metrics means we are minimizing that score, while a negative Route Cost weight means we are maximizing it

A surprising result is that if we minimize any of our sustainability metrics while setting the Route Cost weight to zero the Route Cost for the pathways becomes very close to the baseline (see Figure 25). In other words, minimizing sustainability has the same effect as minimizing costs. However, we should note that in most cases minimizing our metrics does not significantly decrease their score. The exception is again the enzymatic metric, which decreases by 6.8% on average.

We should remark that the generated pathways are not necessarily practical due to inaccuracies in the

AiZynthFinder and our models, the limited scope MCTS has of the reaction space, and especially since the viability of a pathway is not verified until brought into practice. In particular, the expansion policy model is a multi-class predictor for reaction templates, and these templates are pre-defined transformations which may be biased towards non-sustainable reactions.

The assessment approach presented in this section is independent of the retrosynthesis algorithm and it could be applied to any algorithm that optimizes some quantifiable objective score.

4.7.5 Conclusion

We have demonstrated Computer-Aided Synthesis Planning (CASP) as a use-case for our metrics, and defined an assessment approach to quantify the impact of additional metrics in the retrosynthesis algorithm.

Moreover, with the results of the previous section we have seen a trade-off between enzyme-sustainability and Route Cost.

We have also discovered that minimizing sustainability in AiZynthFinder's post-processing objective also minimizes Route Cost, although it did not negatively affect the resulting average sustainability scores significantly.

5 Results and Discussion

Table 13 shows the sustainability prediction results on the test set for our best models presented in this work.

For the enzymatic prediction, the BERT model trained without loss balancing outperforms the BERT model with loss-balancing parameter $\beta = 0.99999$. However, the balanced model achives an F1 score of 0.464 in isomerase prediction (enzyme class EC.7.x.x.x), while the non-balanced model has an F1 score of 0.000 in isomerase prediction. Thus, our balanced model is more accurate in rare biocatalysis prediction with only a small decrease in overall sustainability prediction: −0.008 in F1 score and −0.02% in accuracy compared to the non-balanced model (see Table 13).

For the solvent sustainability prediction there is a discrepancy between the results in the test set and the validation set. In the test set the XGBoost baseline with DRFP fingerprints achieves the highest F1 score with 0.618, while our balanced BERT model achieves 0.584. However, in the validation set we had seen XGBoost with DRFP achieve an F1 of 0.615 and BERT an F1 of 0.660 (see Table 10). This may be in part because the hyper-parameter search we performed on BERT overfit to some extent the validation set, thus the model is not as robust as XGBoost.

		Sustainability prediction	
Task	**Model**	**Accuracy**	**F1**
Enzymatic prediction	BERT	**99.95%**	**0.986**
Enzymatic prediction	BERT (balanced)	99.93%	0.978
Solvent prediction	XGBoost (AP3)	72.34%	0.605
Solvent prediction	XGBoost (DRFP)	**73.79%**	**0.618**
Solvent prediction	BERT (balanced)	70.97%	0.584

Table 13: Summary of sustainability classification approaches and their performance on the test set. *(balanced)* indicates that the train loss was balanced according to the effective number of samples (Phan and Yamamoto (2020))

As shown in Section 4.7, applying our metrics for Computer-Aided Synthesis Planning (CASP) allows us to generate pathways which are more enzyme-sustainable. We achieved this through the Monte Carlo tree search (MCTS) algorithm implemented in AiZynthFinder (Genheden *et al.* (2020b)). However, a higher enzyme-sustainability score results in pathways which have a higher cost, so there is a trade-off between increasing enzyme-sustainability and retrieving cost-effective pathways. Table 14 summarizes this trade-off and shows the average scores over the pathways for the relevant input metric weights. For the complete ablation study on the input weights, see Section 4.7.4.

Input metric weights				Pathway scores			
Enzyme	**Solvent**	**AE**	**Route cost**	**Enzyme [%]**	**Solvent [%]**	**AE [%]**	**Route cost**
0	0	0	1	11 ± 12	21 ± 6	84 ± 14	**6** ± 5
1	0	0	0	19 ± 9	22 ± 5	**87** ± 12	63 ± 31
10	0	0	1	**21** ± 14	22 ± 6	**87** ± 14	8 ± 7
0	1	0	0	12 ± 7	**23** ± 5	85 ± 13	75 ± 36
0	0	1	0	13 ± 7	20 ± 5	86 ± 12	77 ± 36

Table 14: CASP results summary. Targets are the WHO Essential Medicines List (EML) of 127 target compounds used in the work of Gao *et al.* (2020a). Errors are the standard deviations over the predicted pathways

Optimizing for the solvent metric or Atom Economy (AE) individually did not significantly change their average score over the pathways compared to row one of Table 14. However, the route cost still increased. For the solvent metric (∼ 22% on average), this may be due to the low accuracy of BERT on solvent prediction, which missclassifies the solvents participating in the reactions, and because the MCTS algorithm is biased towards the generation of reactions with non-sustainable solvents. For AE, we can again point to the MCTS algorithm, since even when setting this metric's weight to zero we achieve a high pathway AE (∼ 85% on average) which may be due to the retrosynthesis model used in MCTS, which generates reactants for a target product by classifying the most likely transformation. This reaction classification approach makes the change in atoms deterministic according to the template, which may disallow many atoms from disappearing in the reaction, thus resulting in a high AE in most cases which is hard to improve.

6 Conclusion

Automated tools are required to accelerate sustainable development to move toward more sustainable practices in the chemical field (Weber *et al.* (2021); Weber (2022)).

In this work, we have defined novel metrics based on artificial intelligence (AI) to estimate sustainability in reactions using uncertainty quantification (UQ) techniques and introduced them into a new package for chemical sustainability quantification. Afterward, we demonstrated how these metrics could be integrated into a Computer-Aided Synthesis Planning (CASP) software, AiZynthFinder, to generate more enzyme-sustainable pathways, with a trade-off on the cost-effectiveness of the routes.

Our metrics quantify sustainability through the AI models' confidence in their sustainability prediction in the form of a percentage, which is closer to 100% the higher the confidence. We pre-trained BERT on a Masked-Language Modelling (MLM) task on reaction SMILES until they achieved an accuracy of 99%, and fine-tuned them either on reaction class classification for enzyme-sustainability prediction or multi-label solvent classification for solvent-sustainability prediction. We achieve an accuracy of 99.95% and an F1 score of 0.986 in enzyme prediction; and an accuracy of 70.97% and an F1 score of 0.584 in solvent-sustainability prediction. Furthermore, we have shown that XGBoost with DRFP reaction fingerprints achieves a solvent-sustainability accuracy of 73.79% and an F1 score of 0.618.

Another focus of our package is to provide AI interpretability tools to help the user debug the AI models used in the metrics. We presented three main visualizations: UQ for expected likelihood estimation, token attention, and importance attribution through Integrated Gradients. Through a misclassified example where the model is confident in the wrong prediction, we showed that attention is misleading and should not be used for attributing the impact of individual tokens on the predicted class. Instead, we showed that Integrated Gradients provides a better picture, and we used it to discover an adversarial attack on our enzyme prediction model where substituting some SMILES tokens with wildcard tokens "*" pushed the model to mispredict the reaction as enzymatic. We then counteracted this adversarial attack by randomly substituting SMILES tokens with "*" during training.

We also generalized our metrics to pathways by scoring them according to the average metric score over the pathway's reactions. Then, we defined a post-processing objective pathway score to maximize in AiZynthFinder as the weighted sum of our metric scores and the negative route cost estimated with AiZynthFinder's Route Cost scorer. These weights allow the user to specify how much impact each sustainability aspect should have on the prediction, and we demonstrated how to tweak the sustainability-cost trade-off during pathway generation through these weights. With this method, we increased the average enzyme sustainability from 11% to 21% while slightly increasing the average Route Cost from 6 to 8 cost units on the 127 target molecules from the WHO Essential Medicines List (EML). These results could be improved as current retrosynthesis planning tools are biased toward generating non-sustainable pathways due to their models and the datasets they are trained on. A recent advance has proposed a hybrid approach combining the previously developed template-based model from ASKCOS, which is aware of traditional non-sustainable chemical transformations, and a novel model trained on enzymatic templates (Levin *et al.* (2022)). Our sustainability metric toolkit could be introduced in this approach as a post-processing step to generate pathways that are more enzyme-sustainable for the same Route Cost we achieved with AiZynthFinder.

We should also remark that in our CASP use-case study, we did not manually inspect the pathways to evaluate their feasibility, including whether they use a chemically diverse set of transformations, which is an important challenge that multi-step retrosynthesis approaches face commonly (Badowski *et al.* (2019)). Future work should evaluate the impact of the AI-metrics and the viability of the generated pathways.

However, we hope this work serves as an initial step toward a more complete chemical sustainability integration framework.

More AI-based metrics could be added to expand on the toolkit, such as E-factor prediction, catalyst prediction, or starting compound toxicity prediction. Deep learning models such as ToxSmi could be used for toxicity prediction (Markert *et al.* (2020)). Additionally, efforts should be made to change the architecture of the models used in the metrics to make them easily interpretable through formal guarantees while keeping high accuracy; however, this would require significant advancements in the machine learning and AI explainability communities. Finally, we consider that other objectives could be helpful for sustainable synthesis planning, including the direct translation of chemical reaction pathways into more sustainable ones and the optimization of the minimum and most sustainable set of starting and waste compounds that can synthesize a target molecule library, which at the moment is only partially solved with slow mixed-integer optimization techniques in recent works that do not focus on sustainability (Gao *et al.* (2020a,b)).

References

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R. *et al.* (2021) A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion* **76**, 243–297.

Alcántara, R., Axelsen, K. B., Morgat, A., Belda, E., Coudert, E., Bridge, A., Cao, H., De Matos, P., Ennis, M., Turner, S. *et al.* (2012) Rhea—a manually curated resource of biochemical reactions. *Nucleic acids research* **40**(D1), D754–D760.

American Chemical Society (2023) Cas reactions. https://www.cas.org/cas-data/cas-reactions. Accessed: 2023-01-18.

Ayhan, M. S. and Berens, P. (2018) Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks .

Badowski, T., Molga, K. and Grzybowski, B. A. (2019) Selection of cost-effective yet chemically diverse pathways from the networks of computer-generated retrosynthetic plans. *Chemical science* **10**(17), 4640–4651.

Balandat, M., Karrer, B., Jiang, D., Daulton, S., Letham, B., Wilson, A. G. and Bakshy, E. (2020) Botorch: a framework for efficient monte-carlo bayesian optimization. *Advances in neural information processing systems* **33**, 21524–21538.

Bastings, J. and Filippova, K. (2020) The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? *arXiv preprint arXiv:2010.05607* .

Bille, P. (2005) A survey on tree edit distance and related problems. *Theoretical computer science* **337**(1-3), 217–239.

Blake, J. E. and Dana, R. C. (1990) Casreact: more than a million reactions. *Journal of chemical information and computer sciences* **30**(4), 394–399.

Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S. and Colton, S. (2012) A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games* **4**(1), 1–43.

Byrne, F. P., Jin, S., Paggiola, G., Petchey, T. H., Clark, J. H., Farmer, T. J., Hunt, A. J., Robert McElroy, C. and Sherwood, J. (2016) Tools and techniques for solvent selection: green solvent selection guides. *Sustainable Chemical Processes* **4**(1), 1–24.

Carey, J. S., Laffan, D., Thomson, C. and Williams, M. T. (2006) Analysis of the reactions used for the preparation of drug candidate molecules. *Organic & biomolecular chemistry* **4**(12), 2337–2347.

Carhart, R. E., Smith, D. H. and Venkataraghavan, R. (1985) Atom pairs as molecular features in structure-activity studies: definition and applications. *Journal of Chemical Information and Computer Sciences* **25**(2), 64–73.

Chandrasekaran, A., Kim, C., Venkatram, S. and Ramprasad, R. (2020) A deep learning solvent-selection paradigm powered by a massive solvent/nonsolvent database for polymers. *Macromolecules* **53**(12), 4764–4769.

Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. (2002) Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research* **16**, 321–357.

Chen, B., Li, C., Dai, H. and Song, L. (2020) Retro*: learning retrosynthetic planning with neural guided a* search. In *International Conference on Machine Learning*, pp. 1608–1616.

Chen, T. and Guestrin, C. (2016) Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794.

Cheng, H. and Gross, R. A. (2010) Green polymer chemistry: biocatalysis and biomaterials. In *Green Polymer Chemistry: Biocatalysis and Biomaterials*, pp. 1–14. ACS Publications.

Chicco, D. and Jurman, G. (2020) The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics* **21**(1), 1–13.

Coley, C. W., Thomas III, D. A., Lummiss, J. A., Jaworski, J. N., Breen, C. P., Schultz, V., Hart, T., Fishman, J. S., Rogers, L., Gao, H. *et al.* (2019) A robotic platform for flow synthesis of organic compounds informed by ai planning. *Science* **365**(6453), eaax1566.

Connor Coley, Mike Fortunato, H. G. P. P. M. C. M. L. Y. W. T. S. J. L. and Mo, Y. (2021) ASKCOS. `https://github.com/ASKCOS/ASKCOS`. Accessed: 2023-01-19.

Coulom, R. (2007) Efficient selectivity and backup operators in monte-carlo tree search. In *International conference on computers and games*, pp. 72–83.

Cui, Y., Jia, M., Lin, T.-Y., Song, Y. and Belongie, S. (2019) Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9268–9277.

Curzons, A., Constable, D. and Cunningham, V. (1999) Solvent selection guide: a guide to the integration of environmental, health and safety criteria into the selection of solvents. *Clean Products and Processes* **1**(2), 82–90.

Cybenko, G. (1989) Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems* **2**(4), 303–314.

David, L., Thakkar, A., Mercado, R. and Engkvist, O. (2020) Molecular representations in ai-driven drug discovery: a review and practical guide. *Journal of Cheminformatics* **12**(1), 1–22.

Daylight Chemical Information Systems, Inc. (2022) Smirks - a reaction transform language. `https://www.daylight.com/dayhtml/doc/theory/theory.smirks.html`. Accessed: 2023-01-16.

De Smet, H. (2020) SUSSOL software. `https://github.com/SUSSOLKDG/Sussol`. Accessed: 2022-12-16.

Der Kiureghian, A. and Ditlevsen, O. (2009) Aleatory or epistemic? does it matter? *Structural safety* **31**(2), 105–112.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* .

Dice, L. R. (1945) Measures of the amount of ecologic association between species. *Ecology* **26**(3), 297–302.

Drummond, C., Holte, R. C. *et al.* (2003) C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on learning from imbalanced datasets II*, volume 11, pp. 1–8.

Dusenberry, M., Jerfel, G., Wen, Y., Ma, Y., Snoek, J., Heller, K., Lakshminarayanan, B. and Tran, D. (2020) Efficient and scalable bayesian neural nets with rank-1 factors. In *International conference on machine learning*, pp. 2782–2792.

Elsevier Limited (2023) Reaxys. `https://new.reaxys.com/`. Accessed: 2023-01-16.

ETH Zurich (2008) Ehs assessment tool. `https://emeritus.setg.ethz.ch/research/downloads/software---tools/ehs-tool.html`. Accessed: 2023-01-06.

Gal, Y. and Ghahramani, Z. (2016) Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059.

Ganter, M., Bernard, T., Moretti, S., Stelling, J. and Pagni, M. (2013) Metanetx. org: a website and repository for accessing, analysing and manipulating metabolic networks. *Bioinformatics* **29**(6), 815–816.

Gao, H., Coley, C. W., Struble, T. J., Li, L., Qian, Y., Green, W. H. and Jensen, K. F. (2020a) Combining retrosynthesis and mixed-integer optimization for minimizing the chemical inventory needed to realize a who essential medicines list. *Reaction Chemistry & Engineering* **5**(2), 367–376.

Gao, H., Pauphilet, J., Struble, T. J., Coley, C. W. and Jensen, K. F. (2020b) Direct optimization across computer-generated reaction networks balances materials use and feasibility of synthesis plans for molecule libraries. *Journal of Chemical Information and Modeling* **61**(1), 493–504.

Gao, H., Struble, T. J., Coley, C. W., Wang, Y., Green, W. H. and Jensen, K. F. (2018) Using machine learning to predict suitable conditions for organic reactions. *ACS central science* **4**(11), 1465–1476.

Gasteiger, J. and Jochum, C. (1978) Eros a computer program for generating sequences of reactions. In *Organic Compunds*, pp. 93–126. Springer.

Genheden, S. (2022) Paroutes 2.0 and uspto-based models. `https://doi.org/10.5281/zenodo.7341155`.

Genheden, S. and Bjerrum, E. (2022) Paroutes: a framework for benchmarking retrosynthesis route predictions .

Genheden, S., Chadimova, V. and Engkvist, O. (2020a) AiZynthFinder: a fast, robust and flexible open-source software for retrosynthetic planning .

Genheden, S., Engkvist, O. and Bjerrum, E. (2021) Clustering of synthetic routes using tree edit distance. *Journal of Chemical Information and Modeling* **61**(8), 3899–3907.

Genheden, S., Engkvist, O. and Bjerrum, E. (2022) Fast prediction of distances between synthetic routes with deep learning. *Machine Learning: Science and Technology* **3**(1), 015018.

Genheden, S., Thakkar, A., Chadimová, V., Reymond, J.-L., Engkvist, O. and Bjerrum, E. (2020b) Aizynthfinder: a fast, robust and flexible open-source software for retrosynthetic planning. *Journal of cheminformatics* **12**(1), 1–9.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M. and Kagal, L. (2018) Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pp. 80–89.

Gorodkin, J. (2004) Comparing two k-category assignments by a k-category correlation coefficient. *Computational biology and chemistry* **28**(5-6), 367–374.

Grandini, M., Bagli, E. and Visani, G. (2020) Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756* .

Grethe, G., Blanke, G., Kraut, H. and Goodman, J. M. (2018) International chemical identifier for reactions (rinchi). *Journal of Cheminformatics* **10**(1), 1–9.

Hargreaves, C. R., Manley, J. and ACS GCI, P. R. (2008) Collaboration to deliver a solvent selection guide for the pharmaceutical industry.

Hart, P. E., Nilsson, N. J. and Raphael, B. (1968) A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics* **4**(2), 100–107.

Heydarian, M., Doyle, T. E. and Samavi, R. (2022) Mlcm: multi-label confusion matrix. *IEEE Access* **10**, 19083–19095.

Hora, S. C. (1996) Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management. *Reliability Engineering & System Safety* **54**(2-3), 217–223.

Hossin, M. and Sulaiman, M. N. (2015) A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process* **5**(2), 1.

Hugging Face (2022) Optimum. `https://huggingface.co/docs/optimum/index`. Accessed: 2023-01-19.

Hüllermeier, E. and Waegeman, W. (2021) Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning* **110**(3), 457–506.

Hung, T. N. K., Le, N. Q. K., Le, N. H., Van Tuan, L., Nguyen, T. P., Thi, C. and Kang, J.-H. (2022) An ai-based prediction model for drug-drug interactions in osteoporosis and paget's diseases from smiles. *Molecular informatics* **41**(6), 2100264.

IBM RXN team (2022) Rxn chemutils. `https://rxn4chemistry.github.io/rxn-chemutils/`. Accessed: 2023-01-06.

InfoChem (2019) Spresiweb. `https://www.spresi.com/`. Accessed: 2023-01-16.

InfoChem GmbH (2002) CLASSIFY. https://www.int-conf-chem-structures.org/eng/downloads/Classify.pdf. Accessed: 2023-01-16.

Irwin, J. J. and Shoichet, B. K. (2005) Zinc- a free database of commercially available compounds for virtual screening. *Journal of chemical information and modeling* **45**(1), 177–182.

Jeong, J., Lee, N., Shin, Y. and Shin, D. (2022) Intelligent generation of optimal synthetic pathways based on knowledge graph inference and retrosynthetic predictions using reaction big data. *Journal of the Taiwan Institute of Chemical Engineers* **130**, 103982.

Kanehisa Laboratories (2023) Kregg reaction. https://www.genome.jp/kegg/reaction/. Accessed: 2023-01-16.

Kearnes, S. M., Maser, M. R., Wleklinski, M., Kast, A., Doyle, A. G., Dreher, S. D., Hawkins, J. M., Jensen, K. F. and Coley, C. W. (2021) The open reaction database. *Journal of the American Chemical Society* **143**(45), 18820–18826.

Kimber, T. B., Gagnebin, M. and Volkamer, A. (2021) Maxsmi: maximizing molecular property prediction performance with confidence estimation using smiles augmentation and deep learning. *Artificial Intelligence in the Life Sciences* **1**, 100014.

Kobayashi, S., Uyama, H. and Ohmae, M. (2001) Enzymatic polymerization for precision polymer synthesis. *Bulletin of the Chemical Society of Japan* **74**(4), 613–635.

Kocsis, L. and Szepesvári, C. (2006) Bandit based monte-carlo planning. In *European conference on machine learning*, pp. 282–293.

Kohonen, T. (1982) Self-organized formation of topologically correct feature maps. *Biological cybernetics* **43**(1), 59–69.

Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S. and Reblitz-Richardson, O. (2020) Captum: A unified and generic model interpretability library for pytorch.

Kratsios, A. and Papon, L. (2022) Universal approximation theorems for differentiable geometric deep learning. *Journal of Machine Learning Research* **23**(196), 1–73.

Lakshminarayanan, B., Pritzel, A. and Blundell, C. (2017) Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems* **30**.

Landrum, G., Tosco, P., Kelley, B., Ric, sriniker, gedeck, Vianello, R., Cosgrove, D., NadineSchneider, Kawashima, E., N, D., Dalke, A., Jones, G., Cole, B., Swain, M., Turk, S., AlexanderSavelyev, Vaucher, A., Wójcikowski, M., Take, I., Probst, D., Ujihara, K., Scalfani, V. F., guillaume godin, Pahl, A., Berenger, F., JLVarjo, strets123, JP and DoliathGavid (2022) rdkit/rdkit: 2022_09_3 (q3 2022) release. https://doi.org/10.5281/zenodo.7415128.

Leshno, M., Lin, V. Y., Pinkus, A. and Schocken, S. (1993) Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks* **6**(6), 861–867.

Levin, I., Liu, M., Voigt, C. A. and Coley, C. W. (2022) Merging enzymatic and synthetic chemistry with computational synthesis planning. *Nature Communications* **13**(1), 1–14.

Linardatos, P., Papastefanopoulos, V. and Kotsiantis, S. (2020) Explainable ai: A review of machine learning interpretability methods. *Entropy* **23**(1), 18.

Loshchilov, I. and Hutter, F. (2017) Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* .

Lowe, D. (2017) Chemical reactions from US patents (1976-Sep2016) .

Lowe, D. M. and Sayle, R. A. (2015) Leadmine: a grammar and dictionary driven approach to entity recognition. *Journal of cheminformatics* **7**(1), 1–9.

Mahmud, S. H., Chen, W., Jahan, H., Liu, Y., Sujan, N. I. and Ahmed, S. (2019) idti-cssmoteb: identification of drug–target interaction based on drug chemical structure and protein sequence using xgboost with over-sampling technique smote. *IEEE Access* **7**, 48699–48714.

Manica, M., Oskooei, A., Born, J., Subramanian, V., Sáez-Rodríguez, J. and Rodríguez Martínez, M. (2019) Toward explainable anticancer compound sensitivity prediction via multimodal attention-based convolutional encoders. *Molecular Pharmaceutics* **16**(12), 4797–4806.

Markert, G., Born, J., Manica, M., Schneider, G. and Rodriguez Martinez, M. (2020) Chemical representation learning for toxicity prediction. In *PharML Workshop at ECML-PKDD (European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases)*.

Matthews, B. W. (1975) Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure* **405**(2), 442–451.

Mazurenko, S., Prokop, Z. and Damborsky, J. (2019) Machine learning in enzyme engineering. *ACS Catalysis* **10**(2), 1210–1223.

Mellor, J., Grigoras, I., Carbonell, P. and Faulon, J.-L. (2016) Semisupervised gaussian process for automated enzyme search. *ACS synthetic biology* **5**(6), 518–528.

Microsoft (2022) ONNX Runtime. `https://onnxruntime.ai/`. Accessed: 2023-01-19.

Mo, Y., Guan, Y., Verma, P., Guo, J., Fortunato, M. E., Lu, Z., Coley, C. W. and Jensen, K. F. (2021) Evaluating and clustering retrosynthesis pathways with learned strategy. *Chemical science* **12**(4), 1469–1478.

Nextmove Software (2021a) NameRXN. `http://www.nextmovesoftware.com/namerxn.html`. Accessed: 2022-12-06.

Nextmove Software (2021b) Pistachio. `https://www.nextmovesoftware.com/pistachio.html`. Accessed: 2022-12-06.

OpenAI (2022) Chatgpt: Optimizing language models for dialogue. `https://openai.com/blog/chatgpt/`. Accessed: 2023-01-19.

Phan, T. H. and Yamamoto, K. (2020) Resolving class imbalance in object detection with weighted cross entropy losses. *arXiv preprint arXiv:2006.01413* .

Pinkus, A. (1999) Approximation theory of the mlp model in neural networks. *Acta numerica* **8**, 143–195.

Prat, D., Wells, A., Hayler, J., Sneddon, H., McElroy, C. R., Abou-Shehada, S. and Dunn, P. J. (2016) Chem21 selection guide of classical-and less classical-solvents. *Green Chemistry* **18**(1), 288–296.

Probst, D., Manica, M., Nana Teukam, Y. G., Castrogiovanni, A., Paratore, F. and Laino, T. (2022a) Biocatalysed synthesis planning using data-driven learning. *Nature communications* **13**(1), 1–11.

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

Probst, D., Schwaller, P. and Reymond, J.-L. (2022c) Reaction classification and yield prediction using the differential reaction fingerprint drfp. *Digital discovery* **1**(2), 91–97.

Puskas, J. E., Sen, M. Y. and Seo, K. S. (2009) Green polymer chemistry using nature's catalysts, enzymes. *Journal of Polymer Science Part A: Polymer Chemistry* **47**(12), 2959–2976.

Rose, J. R. and Gasteiger, J. (1994) Horace: an automatic system for the hierarchical classification of chemical reactions. *Journal of chemical information and computer sciences* **34**(1), 74–90.

Rush, A. M. (2018) The annotated transformer. In *Proceedings of workshop for NLP open source software (NLP-OSS)*, pp. 52–60.

Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S., Yvon, F., Gallé, M. *et al.* (2022) Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100* .

Schneider, N., Lowe, D. M., Sayle, R. A. and Landrum, G. A. (2015) Development of a novel fingerprint for chemical reactions and its application to large-scale reaction classification and similarity. *Journal of chemical information and modeling* **55**(1), 39–53.

Schomburg, I., Chang, A. and Schomburg, D. (2002) Brenda, enzyme data and metabolic information. *Nucleic acids research* **30**(1), 47–49.

Schwaller, P., Hoover, B., Reymond, J.-L., Strobelt, H. and Laino, T. (2021a) Extraction of organic chemistry grammar from unsupervised learning of chemical reactions. *Science Advances* **7**(15), eabe4166.

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

Schwaller, P., Petraglia, R., Zullo, V., Nair, V. H., Haeuselmann, R. A., Pisoni, R., Bekas, C., Iuliano, A. and Laino, T. (2020) Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy. *Chemical science* **11**(12), 3316–3325.

Schwaller, P., Probst, D., Vaucher, A. C., Nair, V. H., Kreutter, D., Laino, T. and Reymond, J.-L. (2021b) Mapping the space of chemical reactions using attention-based neural networks. *Nature Machine Intelligence* **3**(2), 144–152.

Segler, M. H., Preuss, M. and Waller, M. P. (2018) Planning chemical syntheses with deep neural networks and symbolic ai. *Nature* **555**(7698), 604–610.

Sels, H., De Smet, H. and Geuens, J. (2020) SUSSOL—using artificial intelligence for greener solvent selection and substitution. *Molecules* **25**(13), 3037.

Sheldon, R. A. (2018) Metrics of green chemistry and sustainability: past, present, and future. *ACS Sustainable Chemistry & Engineering* **6**(1), 32–48.

Shoda, S.-i., Uyama, H., Kadokawa, J.-i., Kimura, S. and Kobayashi, S. (2016) Enzymes as green catalysts for precision macromolecular synthesis. *Chemical reviews* **116**(4), 2307–2413.

Shwartz-Ziv, R. and Armon, A. (2022) Tabular data: Deep learning is not all you need. *Information Fusion* **81**, 84–90.

Sun, Y. and Sahinidis, N. V. (2022) Computer-aided retrosynthetic design: fundamentals, tools, and outlook. *Current Opinion in Chemical Engineering* **35**, 100721.

Sundararajan, M., Taly, A. and Yan, Q. (2017) Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328.

Sutton, R. S. and Barto, A. G. (2018) *Reinforcement learning: An introduction*. MIT press.

Świechowski, M., Godlewski, K., Sawicki, B. and Mańdziuk, J. (2022) Monte carlo tree search: A review of recent modifications and applications. *Artificial Intelligence Review* pp. 1–66.

Szymkuć, S., Gajewska, E. P., Klucznik, T., Molga, K., Dittwald, P., Startek, M., Bajczyk, M. and Grzybowski, B. A. (2016) Computer-assisted synthetic planning: the end of the beginning. *Angewandte Chemie International Edition* **55**(20), 5904–5937.

Tai, K. S., Socher, R. and Manning, C. D. (2015) Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075* .

Taylor, R., Kardas, M., Cucurull, G., Scialom, T., Hartshorn, A., Saravia, E., Poulton, A., Kerkez, V. and Stojnic, R. (2022) Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085* .

Thakkar, A., Kogej, T., Reymond, J.-L., Engkvist, O. and Bjerrum, E. J. (2020) Datasets and their influence on the development of computer assisted synthesis planning tools in the pharmaceutical domain. *Chemical science* **11**(1), 154–168.

Varnek, A., Fourches, D., Hoonakker, F. and Solov'ev, V. P. (2005) Substructural fragments: an universal language to encode reactions, molecular and supramolecular structures. *Journal of computer-aided molecular design* **19**(9), 693–703.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I. (2017) Attention is all you need. *Advances in neural information processing systems* **30**.

Walker, E., Kammeraad, J., Goetz, J., Robo, M. T., Tewari, A. and Zimmerman, P. M. (2019) Learning to predict reaction conditions: relationships between solvent, molecular structure, and catalyst. *Journal of chemical information and modeling* **59**(9), 3645–3654.

Wang, H. and Yeung, D.-Y. (2020) A survey on bayesian deep learning. *ACM Computing Surveys (CSUR)* **53**(5), 1–37.

Watkins, C. J. C. H. (1989) Learning from delayed rewards .

Webb, E. C. *et al.* (1992) *Enzyme nomenclature 1992. Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzymes.* Number Ed. 6. Academic Press.

Weber, J. (2022) *Circular chemistry through network science and optimisation on big data*. Ph.D. thesis, University of Cambridge.

Weber, J. M., Guo, Z., Zhang, C., Schweidtmann, A. M. and Lapkin, A. A. (2021) Chemical data intelligence for sustainable chemistry. *Chemical Society Reviews* .

Wei, J.-M., Yuan, X.-J., Hu, Q.-H. and Wang, S.-Q. (2010) A novel measure for evaluating classifiers. *Expert Systems with Applications* **37**(5), 3799–3809.

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

Weininger, D., Weininger, A. and Weininger, J. L. (1989) Smiles. 2. algorithm for generation of unique smiles notation. *Journal of chemical information and computer sciences* **29**(2), 97–101.

WHO Expert Committee (1977) The selection of essential drugs : report of a who expert committee [meeting held in geneva from 17 to 21 october 1977].

Wishart, D. S., Li, C., Marcu, A., Badran, H., Pon, A., Budinski, Z., Patron, J., Lipton, D., Cao, X., Oler, E. *et al.* (2020) Pathbank: a comprehensive pathway database for model organisms. *Nucleic acids research* **48**(D1), D470–D478.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M. *et al.* (2019) Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771* .

Zhai, X., Kolesnikov, A., Houlsby, N. and Beyer, L. (2022) Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12104–12113.

Zhou, D.-X. (2020) Universality of deep convolutional neural networks. *Applied and computational harmonic analysis* **48**(2), 787–794.

A Figures

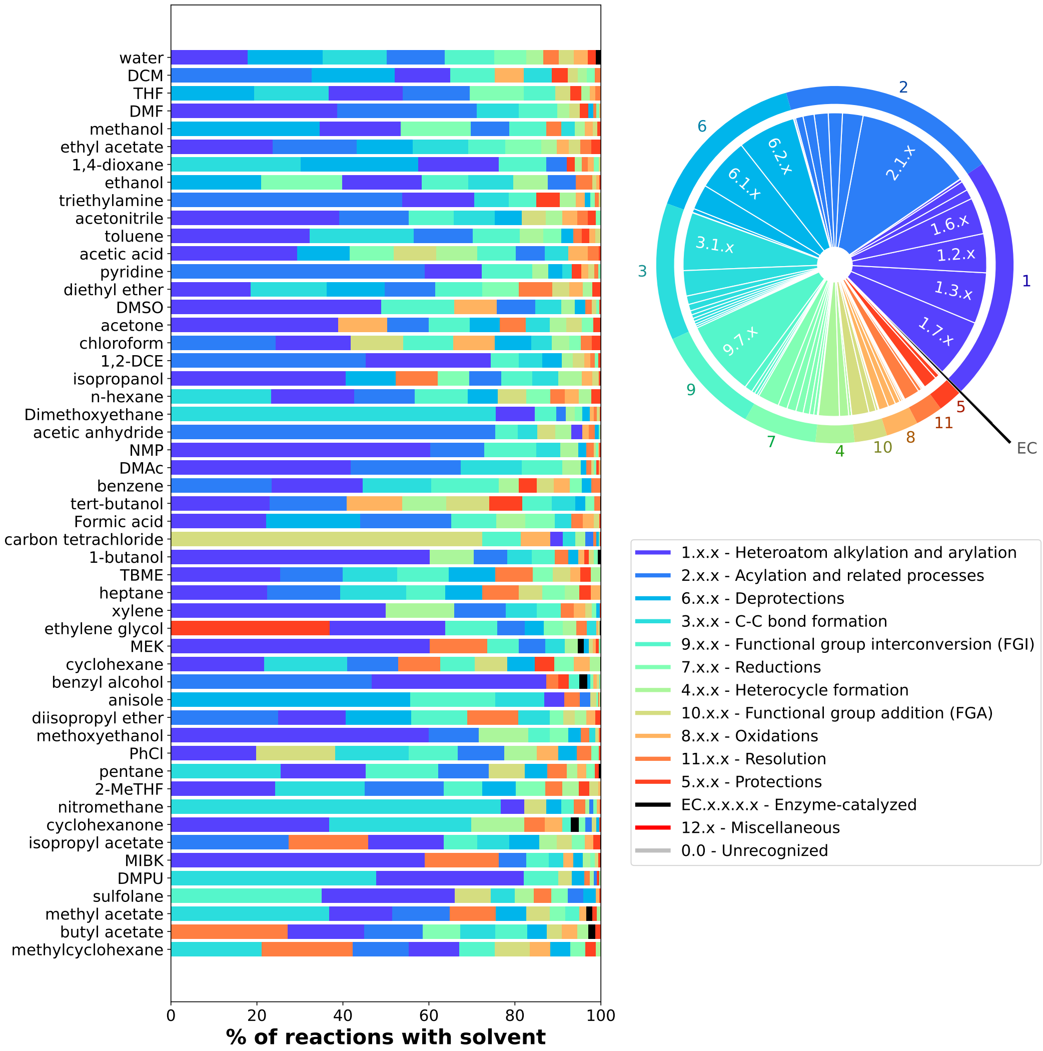


Figure A.1: Solvent distribution in Pistachio+ECREACT dataset. All reactions with class 0.0 have been removed. NameRXN superclasses (Nextmove Software (2021a)) in which each solvent participates in, including enzyme-catalyzed reactions. Solvents are sorted from most to least common (top to bottom). Reaction superclasses for each solvent are sorted from most to least common (left to right). The pie chart indicates the reaction superclass distribution. Frequent sub-classes are indicated

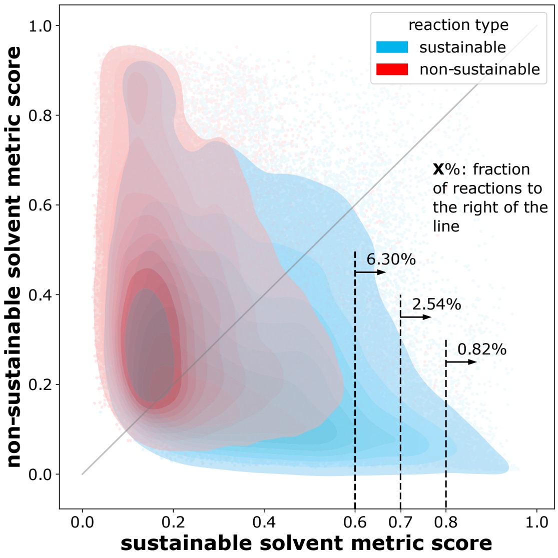


Figure A.2: Solvent-sustainability metric on validation set. Here, a reaction is sustainable if one of the solvents it contains is sustainable. The fraction of reactions with a sustainable score > 0.6, > 0.7 and > 0.8 are indicated. Dense regions are marked through kernel density estimation (KDE)

B Random choice algorithm

Given that we have a method *choice*(N, k) to draw a set of k items from a universe of size N *with replacement*, we can devise an algorithm to draw *without replacement*:

Algorithm B.1 An algorithm to efficiently draw without replacement. $k > 0$ items are taken without replacement from a universe of size N. *choice*(N, k) samples k items with replacement with a space complexity of $O(k \log N)$:

```
Require: $N \geq (2 + \sqrt{2}) \cdot k$
1: $X \leftarrow \{choice(N, 1)\}$
2: while $|X| \neq k$ do
3:     $s \leftarrow choice(N, 1)$
4:     if $s \notin X$ then
5:         $X \leftarrow X \cup \{s\}$
6:     end if
7: end while
8: return $X$
```

Note that lines 4 and 5 are $O(1)$ due to the hashing nature of sets, and given that the sets' hash table is big enough to handle $O(1)$ hash collisions per query.

We demonstrate that this algorithm has a space complexity of $O(k \log N)$ and a time complexity of $\Theta(k)$. Furthermore, we prove that the algorithm is $O(2k)$ in the number of drawn items w.p. (with probability) less than $(k - 1)/N$, and that if $N \geq 100 \cdot k$ then the algorithm draws $1 + 1.021 \cdot k$ items w.p. less than 1%.

Proposition B.1. Algorithm B.1 *has a space complexity of* $O(k \log N)$.

Proof. Both X and *choice*(N, k) occupy $O(k \log N)$ space, since $|X|$ is at most k and our largest number, N, requires $O(\log N)$ bits to be stored. □

Proposition B.2. Algorithm B.1 *has an expected time complexity of* $O(k)$*, i.e., it has a time complexity of* $\Theta(k)$.

Proof. In order to advance in the while loop, we need to add new elements to X. Thus:

$$\begin{aligned} P\left[\text{sampling unique item after } i+1 \text{ draws}\right] &= \left(\frac{|X|}{N}\right)^i \cdot \left(1 - \frac{|X|}{N}\right) \\ &\leq \left(\frac{k-1}{N}\right)^i \cdot 1 \end{aligned}$$

Using this probability bound, the expected number of items drawn is:

$$\begin{aligned}\mathbb{E}[\text{\# of items drawn}] &\leq 1 + \mathbb{E}[\text{\# of draws to increment } |X| = 1 \text{ by } 1]\\ &\quad + \cdots \\ &\quad + \mathbb{E}[\text{\# of draws to increment } |X| = (k-1) \text{ by } 1]\\ &\leq 1 + (k-1)\sum_{i=1}^{\infty} i\left(\frac{k-1}{N}\right)^i\end{aligned}$$

Since we have to draw 1 item in line 1 of the algorithm and then we have to draw $(k-1)$ new unique items in the while loop.

We now note that the algorithm requires that $N \geq (2+\sqrt{2})\cdot k$, so:

$$\begin{aligned}N \geq (2+\sqrt{2})\cdot k &\geq 2\cdot k\\ \frac{k-1}{N} &\leq 2^{-1}\\ &\downarrow\\ \mathbb{E}[\text{\# of items drawn}] &\leq 1 + (k-1)\sum_{i=1}^{\infty} i\cdot 2^{-i}\\ &= 1 + (k-1)\cdot 2\\ &= O(k)\end{aligned}$$

□

However, this is not all. We have used the fact that $N \geq 2k$ in our proof, but now let's generalize to $N \geq ak$ with $a > 1$. Note that $a = 1$ would include the possibility that $N = k$, in which we just sample all items in the universe of size N, and this is not interesting for our purposes. Also, $a < 1$ includes cases where $N < k$, which is impossible and our algorithm would not halt.

Lemma B.1. *If $N \geq ak$ for $a > 1$, the expected number of items drawn is less than* $1 + (k-1)\cdot a/(a-1)^2$.

Proof. $N \geq ak$ implies that,

$$\begin{aligned}\frac{k-1}{N} &\leq a^{-1}\\ &\downarrow\\ \mathbb{E}[\text{\# of items drawn}] &\leq 1 + (k-1)\sum_{i=1}^{\infty} i\cdot a^{-i}\\ &= 1 + (k-1)\frac{a}{(a-1)^2}\end{aligned}$$

Since,

$$\sum_{i=1}^{\infty} i\cdot a^{-i} = \frac{a}{(a-1)^2} \qquad \forall a, |a| > 1$$

□

With this lemma we can prove the following proposition.

Proposition B.3. Algorithm B.1 *draws O(2k) items w.p. less than* $(k-1)/N$.

Proof. We define $Y := (\text{\# of items drawn} - 1)$, which is the amount of items drawn in the while loop. We recall Markov's inequality:

$$P[Y \geq \delta\cdot\mathbb{E}[Y]] \leq \frac{1}{\delta} \qquad \delta > 0$$

So that, using Lemma B.1,

$$P\left[Y \geq \delta\cdot(k-1)\cdot\frac{a}{(a-1)^2}\right] \leq P[Y \geq \delta\cdot\mathbb{E}[Y]] \leq \frac{1}{\delta}$$

Now, we set $\delta = N/(k-1)$, and consider the maximum value of a, $a^* = N/k$,

$$P\left[Y \geq N \cdot \frac{a^*}{(a^*-1)^2}\right] \leq \frac{k-1}{N}$$

Note that this is true since Lemma B.1 still applies when $a = a^*$. We now observe that,

$$N\frac{a^*}{(a^*-1)^2} = k\frac{N^2}{(N-k)^2} = k\left(1-\frac{k}{N}\right)^{-2}$$

Thus finally,

$$P\left[Y \geq k \cdot \left(1-\frac{k}{N}\right)^{-2}\right] \leq \frac{k-1}{N}$$

We can now apply our algorithm requirement $N \geq (2+\sqrt{2}) \cdot k$ so we get,

$$N \geq \left(2+\sqrt{2}\right) \cdot k \quad \rightarrow \quad \left(1-\frac{k}{N}\right)^{-2} \leq \left(1-\frac{1}{2+\sqrt{2}}\right)^{-2} = 2$$

Thus, in the worst case we always draw $O(1+2k) = O(2k)$ items w.p. less than $(k-1)/N$. □

This is very useful. It means that if N is massive and k is minuscule in comparison, we won't draw more than $1+2k$ items with high probability. In fact, if N is 100 times k, then we will draw $1+1.021 \cdot k$ items w.p. less than 1%.

To end this exploration, note that in general if we had chosen $\delta = N/(k-1)k^{\gamma}$ for some $\gamma > 0$, then we would have that the algorithm is $O\left(1+k^{1+\gamma}\left(1-\frac{k}{N}\right)^{-2}\right)$ w.p. less than $k^{1-\gamma}/N$.

For example, the algorithm is $O(k^2)$ w.p. less than $1/N$. This means that drawing 10 items without replacement from a universe of 1'000 items using this algorithm will require drawing 103 items w.p. less than 0.1%.