Towards improving chemical synthesis planning assistants

Doctoral Dissertation submitted to the
Faculty of Informatics of the Università della Svizzera italiana
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

presented by
Mikhail Andronov

under the supervision of
Prof. Jürgen Schmidhuber and Dr.-Ing. Michael Wand

April 2026

Dissertation Committee

Prof. Cesare Alippi	Università della Svizzera italiana, Switzerland
Prof. Andrea-Emilio Rizzoli	Università della Svizzera italiana, Switzerland
Prof. Rolf Krause	Università della Svizzera italiana, Switzerland
Dr. Djork-Arné Clevert	Pfizer Research and Development, Germany
Dr. Igor Tetko	Institute of Structural Biology, Helmholtz Munich, Germany

Dissertation accepted on 14 April 2026

Research Advisor

Prof. Jürgen Schmidhuber

Co-Advisor

Dr.-Ing. Michael Wand

PhD Program Director

Prof. Walter Binder/ Prof. Stefan Wolf

I certify that except where due acknowledgement has been given, the work presented in this thesis is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; and the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program.

Mikhail Andronov
Lugano, 14 April 2026

Abstract

In this thesis, we explore a set of techniques to improve existing deep learning–based automated chemical synthesis planning systems. First, we develop and analyze methods to reduce the latency of transformer-based synthesis planning systems. We accelerate transformer-based single-step retrosynthesis and reaction prediction models using speculative decoding and a novel speculative beam search algorithm, achieving substantial speedups without loss of accuracy and demonstrating their impact on multi-step synthesis planning. Secondly, we study reagent prediction as an integral but underexplored component of synthesis planning. We formulate it as a sequence-to-sequence learning problem and show that accurate reagent modeling can both enhance reaction prediction and mitigate missing information in reaction datasets. We then propose a self-supervised approach to improve the quality of reagent information in reaction data by grouping reagents by functional role, implemented in an interactive web application for data preparation. Together, our contributions advance the efficiency, completeness, and reliability of AI-driven synthesis planning and provide a foundation for developing more practical and trustworthy next-generation CASP systems.

Acknowledgements

I want to express my deepest appreciation to all the people who enabled my PhD journey and shaped its outcome.

I am very grateful to Prof. Jürgen Schmidhuber for the opportunity to do a PhD in his group, for the research freedom granted to me, and for the conversations that deeply influenced my understanding of AI science. I am also very grateful to my co-supervisor, Michael Wand, with whom I worked most closely, for his utmost supportiveness.

I am greatly thankful to Dr. Igor Tetko for organizing and leading this wonderful AIDD consortium of PhD students, of which I have been a part. All the AIDD schools, seminars, and conferences were immensely useful to my research and will be forever memorable to me. I extend my thanks to all the AIDD fellows who made this project so great: Peter, Emma, Paula, Varvara, Julian, Son, Alan, Ana, Yasmine, Rosa, Vincenzo, Alessio, Arslan, Mathias, and Mariia.

I am most grateful to Dr. Djork-Arné Clevert for his invaluable mentorship, genuine participation, and guidance toward pressing scientific problems. I also thank all colleagues at the MLCS team at Pfizer, who made my time in Berlin so enjoyable and inspiring: Tuan, Kristof, Robin, Marco, Joren, and others. Special thanks to Andreas Poehlmann for all the fun we had together and for the programming tips.

I thank Samuel Genheden for his mentorship during my secondment in Gothenburg and for the opportunity to learn about synthesis planning in production.

I thank Prof. Philippe Schwaller for being a big source of inspiration for my research.

My sincere thanks to my colleagues at USI: Vincent, Aleksandar, Róbert, Dylan, Imanol, Aditya, Anand, Kazuki, Francesco, and Eric, for all the great time we spent in discussions and board games. I would like to thank Sjoerd van Steenkiste in particular for helping me move into my first apartment in Lugano, which made my early PhD months much easier and saved me a lot of time for research.

I thank all my friends with whom we had so many great moments during my time in Switzerland and Germany: Mikhail Batov, Konstantin, Timur, Kristina, Daniil, Artem Shmatko, Artem Lomakin, Gleb, and others. I extend my gratitude to my family for their unwavering support and encouragement.

Finally, I express my deepest gratitude to my wife Natasha, without whom this wonderful scientific and personal PhD journey would not have been possible.

The work presented in this thesis was largely funded by the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie Innovative Training Network European Industrial Doctorate grant agreement No. 956832 "Advanced machine learning for Innovative Drug Discovery."

Contents

Figures

Tables

Chapter 1

Introduction

Artificial Intelligence, enabled by deep learning, is currently leading innovation in computing. Neural networks can handle tasks that are unapproachable by usual algorithms, and their capabilities today seem nearly science-fictional. AI can now generate arbitrary human-like text and high-fidelity images and videos, compose music, control robots, recognize and synthesize speech, discover novel mathematics, simulate immersive three-dimensional video game environments, and do numerous other things. AI promises immense opportunities for work automation, so far in the computer realm and potentially in the physical world as well.

Naturally, AI is proliferating into established fields of science and technology, promising to accelerate and facilitate novel discoveries. Pharmaceutical research and chemistry in general have a long tradition of AI adoption, experimenting with the first artificial neural networks as early as the 1980s [Hoskins and Himmelblau, 1988]. Today, AI technologies have firmly established their place in the drug discovery process, assisting at all stages of the Design-Make-Test-Analyze (DMTA) cycle [Ghiandoni et al., 2024]. AI is used to generate molecules, predict their properties, reconstruct molecules from spectra, etc.

One of the challenging problems in chemical research is chemical synthesis planning, i.e., finding a sequence of chemical transformations that leads from some available compounds to the compound of interest. It is a challenging problem that takes chemists years of formal training and experience to master. Drug discovery and chemical manufacturing would benefit enormously from a faster, cheaper, and more accurate automated synthesis planning. It is also one of the tasks for which AI technologies have significant innovative potential.

Although computer-aided synthesis planning (CASP) is an old field of research dating back to early computers and expert systems, it is still a developing technology. Chemists did not adopt early-generation automated synthesis planning systems, and

they are still not eager to use modern ones routinely [Ihlenfeldt and Gasteiger, 1996]. Current CASP systems, powered by neural networks and often leveraging advances in language modeling, have numerous limitations. Among them are suboptimal accuracy, high latency, omission of important details in predicted reactions, hallucinations, failures to process complex molecules, and issues with the training data, which will never be as ample as in CV or NLP, and which, due to the task specifics, cannot tolerate noisy records.

Inspired by the vision of helpful synthesis-planning AI assistants, we set out to address some of the problems of modern AI-based CASP systems to contribute to the development of improved next-generation CASP systems that would be truly useful to chemists. This thesis describes our efforts.

1.1 Contributions

This thesis describes several methods we developed to advance AI-based computer-aided synthesis planning.

In Chapter 2, we provide an overview of cheminformatics, the applications of AI in chemistry, and approaches to automated synthesis planning to introduce the necessary background and context for subsequent chapters.

Chapters 3 and 4 describe our efforts at making state-of-the-art AI-based synthesis planning faster.

In Chapter 3, we tackle the problem of synthesis planning latency and demonstrate how a state-of-the-art transformer model for single-step retrosynthesis and reaction product prediction can be accelerated on inference without compromising accuracy. When those tasks are framed as conditional generation of molecular strings and approached with a transformer model, we can capitalize on the fact that the input and output molecular strings often differ only slightly. To do it, we employ speculative decoding with a simple domain-specific heuristic drafting scheme. We then demonstrate speculative beam search, a novel algorithm that we developed to render beam search and speculative decoding compatible. We show that our method accelerates transformer-based reaction prediction and single-step retrosynthesis by up to 3× without sacrificing accuracy.

In Chapter 4, we explore the effect of the accelerated inference of a transformer-based single-step retrosynthesis model in multi-step synthesis planning. First, we propose an improved drafting scheme for speculative decoding in single-step retrosynthesis by training a ”Medusa” variant of the transformer and comparing it with the previously introduced heuristic drafting scheme. Then, we demonstrate the acceleration of multi-step synthesis planning enabled by a faster single-step retrosynthesis model.

Chapters 5 and 6 are dedicated to our work with reagent information. Reagents, e.g., catalysts or solvents, are integral for any reaction, and an AI-powered synthesis planning system cannot reach its maximum accuracy without a reagent prediction component.

In Chapter 5, we tackle the problem of reagent prediction for reactions. Similar to single-step retrosynthesis and product prediction, we formulate reagent prediction as conditional text generation (SMILES-to-SMILES) and study the capability of an encoder-decoder transformer on this task. Additionally, we describe common reagent-related issues in the reaction data and demonstrate how they can be mitigated using a trained reagent-prediction model that reintroduces missing reagents. We demonstrate that the latter can enable more accurate prediction of reaction products.

In Chapter 6, we dive deeper into the problem of reagent curation in reaction data for the ultimate purpose of training more accurate reagent prediction models. We identify additional issues with reagent information in reaction data and present a solution for curating reagents by grouping them into roles in a self-supervised manner. We implement our method as an interactive web application. We then use our application to correct reagent information in a common open reaction dataset.

Chapter 7 concludes this thesis and outlines directions for further research.

Chapter 2

Background

This chapter aims to provide the background necessary to understand the specifics of machine learning from chemical data. We begin with a brief overview of cheminformatics, its history, and its goals. We then describe chemical data modalities, common molecular representations in data, and sources of chemical reaction data. Finally, we provide a detailed overview of machine learning-based reaction modeling and automated synthesis planning. We do not provide an introduction to machine learning, deep neural architectures, or modern AI technologies, as we assume the reader is already familiar with these topics. However, for a profound introduction to machine learning, deep learning, and neural networks, we invite the reader to consult specialized textbooks [Bishop and Bishop, 2023; Bishop and Nasrabadi, 2006]. For a thorough introduction to cheminformatics, we refer to the classic textbook by Gasteiger and Engel, 2003.

2.1 Cheminformatics

Chemistry, as a natural science, studies the composition, structure, properties, and transformations of matter. Matter consists of various substances, and the range of possible substances for study is virtually limitless. Elemental composition, the number of atoms, various spatial arrangements, and other factors produce a combinatorial explosion in the possibility space of molecules and crystal structures that can exist. As scientists discovered, isolated, and characterized more and more compounds, the need emerged to store, categorize, index, and search chemical information. By the time computers emerged, the accumulated body of chemical knowledge had already exceeded the point where manual literature searches were feasible. Computers were immediately adapted to process chemical information, and since then, chemical science has become unimaginable without them.

Working with chemical information on computers required solving specific practical problems: how to represent molecules and reactions in computer memory, how to implement substructure search, how to visualize arbitrary molecules on screen, how to standardize equivalent representations of the same molecule, and so forth [Willett, 2008]. Pharmaceutical industry workers and medical chemists, who eagerly embraced all such innovations, also formulated a question: how can the wealth of experimental data be used to uncover correlations between molecular structures and their biological activity or other properties? In 1998, Brown coined the term "chemoinformatics" (which later became interchangeable with the term "cheminformatics") in the context of the problems drug discovery specialists face [Brown, 1998]. By chemoinformatics, the author meant a set of programs, protocols, and practices for supporting chemists in decision-making and answering two main questions: "what to test next" and "what to make next." Although other authors later provided more general definitions [Gasteiger and Engel, 2003], it is still drug discovery specialists to whom the achievements in cheminformatics are most relevant. It is important to note that cheminformatics, as a discipline, differs from computational chemistry and does not originate from it. Computational chemistry aims to predict the properties of molecules deductively by approximating the solution to Schrödinger's equation. Chemoinformatics, in contrast, aims to predict molecular properties inductively from large amounts of experimental data. Modern readers immediately recognize that this refers to artificial intelligence and machine learning. And indeed, cheminformatics has been closely intertwined with statistical methods and machine learning from the outset.

In 1962, Hansch et al. built a linear regression model on a Clary DE-60 computer to predict the water-octanol partition coefficient for a series of substituted phenoxyacetic acids, motivated by the search for active plant growth factors and the failure of classical physical chemistry models [Hansch et al., 1962]. This work established the cheminformatics subdiscipline of QSAR/QSPR (quantitative structure–activity / structure–property relationships), methods that are now routinely employed in drug discovery across both industry and academia [Cherkasov et al., 2014].

In 1969, Lederberg et al. published a paper with a strikingly modern-sounding title: "Applications of artificial intelligence for chemical inference." In this manuscript, the authors presented a deterministic program written in LISP for generating all possible acyclic isomers of a given organic molecule in a string notation, aiming to facilitate the interpretation of mass spectra [Lederberg et al., 1969]. Today, the interpretation of mass spectra via molecular generation is typically approached using advanced deep learning architectures [Litsa et al., 2023; Shrivastava et al., 2021; Beck et al., 2024], and graph-processing algorithms similar to those described in Lederberg's work form the basis of modern cheminformatics toolkits [*RDKit: Open-source cheminformatics*; Willighagen et al., 2017].

In 1969, Elias Corey (a Nobel laureate in chemistry, renowned for his pioneering contributions to organic synthesis and for formalizing the concept of retrosynthetic analysis [Corey, 1967; Corey and Cheng, 1989]) and his postdoc, Todd Wipke, presented the first computer-aided synthesis planning (CASP) system called OCSS (Organic Chemical Simulation of Synthesis) [Corey and Wipke, 1969]. This sophisticated program ran on a DEC PDP-1 computer equipped with three monitors, a graphical input system, and a plotting device. The user could draw the structure of a target compound on a tablet, after which the program constructed a synthesis plan, displayed it graphically, and optionally produced a printed version. The program recognized the input drawings, validated them, and converted them into special connection tables. The algorithm underlying the tree construction applied transformation heuristics predefined by expert chemists to every molecule in the growing synthesis tree. At each step, the algorithm attempted to simplify the molecular structure. The authors applied their method to the patchouli alcohol molecule and, remarkably, the program successfully proposed elegant, previously undescribed synthetic routes for it. They predicted that in the future, software similar to OCSS would become an indispensable tool for chemists. OCSS and its successor, LHASA, were described by their creators as "artificially intelligent problem solvers" [Corey et al., 1972]. This work established automated synthesis planning as one of the pillars of cheminformatics and associated it with intelligent systems.

Today, cheminformatics has fully embraced modern deep learning and artificial intelligence methods to address its tasks. Molecular property prediction is approached using graph neural networks [Gilmer et al., 2017; Capela et al., 2019; Jiang et al., 2021], transformer models trained on text [Karpov et al., 2019a; Ross et al., 2022], and Large Language Models [Zhuang et al., 2025]. Structure elucidation, i.e., inferring molecular structures from their experimental properties, such as spectra, is done by various encoder-decoder neural architectures [Zhang et al., 2025; Bohde et al., 2025; Tan, 2025]. *De novo* molecular generation, i.e., generation of molecules with desired properties, such as affinity to a target protein, is addressed using recurrent neural networks [Segler, Kogej, Tyrchan and Waller, 2018], LLMs [Bhattacharya et al., 2024; Liu et al., 2024], diffusion models [Hoogeboom et al., 2022; Cremer et al., 2024; Wang et al., 2025], and flow matching models [Cremer et al., 2025]. Large Language Models are now applied to all areas of cheminformatics [Alampara et al., 2025]. New, promising fields have been unlocked by recent advances in machine learning, such as automated chemical data mining from text [Vaucher, Zipoli, Geluykens, Nair, Schwaller and Laino, 2020; Zhang et al., 2024] and the acceleration of computational chemistry methods through various learned optimizations [Smith et al., 2017; Huang and Von Lilienfeld, 2021; del Rio et al., 2023].

The modern pharmaceutical industry is heavily investing in AI to reduce the sub-

stantial time and financial costs associated with drug development [Vijayan et al., 2022]. The traditional Design-Make-Test-Analyze (DMTA) cycle of the preclinical drug discovery process is now difficult to imagine without specialized neural models assisting at all its stages. *De novo* molecular generators are already helping to identify potential drug candidates [Ivanenkov et al., 2023].

2.2 Chemical data representations

Since advances in cheminformatics are, both generally and in our own work, primarily oriented toward drug development, we will focus here on organic molecules and reactions from organic chemistry when discussing chemical data representations.

2.2.1 Molecules

Representing molecules in computers is not trivial, and the best representation depends on the problem at hand (Fig. 2.1). Rephrasing the IUPAC definition [1], a molecule is a stable and electrically neutral entity consisting of n atoms, where $n > 1$. Atoms can be viewed as points in three-dimensional space, held together by chemical bonds. A molecule can participate in chemical reactions, during which bonds between atoms are broken, or new ones are formed. The bonds manifest in 3D as stable distances between atoms. However, the relative positions of bonded atoms fluctuate within small ranges, resulting in varying molecular conformations. In principle, this makes a molecule a four-dimensional object, with the additional dimension reflecting conformational flexibility over time. Molecular representations at this level are necessary, for example, for molecular dynamics simulations. If we are not interested in conformational changes, we can consider a molecule to be either a three-dimensional point cloud of atoms or a 3D surface defined by the electrostatic potential of the atoms. Such a view is useful, for example, for molecular docking. If we are only interested in the connectivity of atoms in the molecule, we can consider it a molecular graph, which makes the molecule a 2D object. Two-dimensional structure diagrams are a common way to represent organic molecules in scientific manuscripts. Furthermore, we can represent a molecule as a sequence of characters and treat it as a 1D object. Empirical formulas and the IUPAC nomenclature of organic chemistry are examples of molecular string notations established before the computer era.

With the emergence of computers and large chemical databases to index and query, researchers designed specialized line notations to facilitate working with chemical objects in software [Gasteiger and Engel, 2003]. One particularly popular notation is

[1] IUPAC compendium of chemical terminology. URL: https://doi.org/10.1351/goldbook.M04002

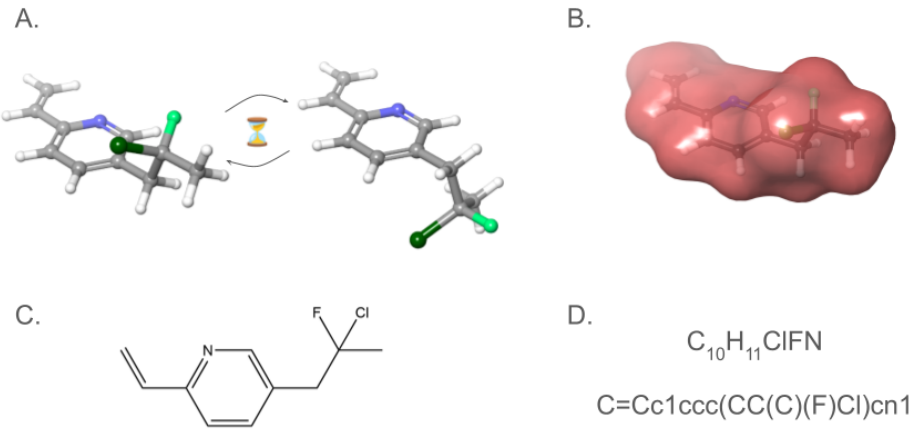

Figure 2.1. Various representations of the same molecule. **A**: The molecule can be represented with an animation in 3D, since chemical bonds in molecules may rotate and slightly oscillate in length. **B**: The molecule can be represented as a point cloud in 3D and optionally a surface in 3D. **C**: The molecular topology can be represented as a 2D diagram or as a graph. **D**: The molecule can be encoded as a string.

SMILES (Simplified Molecular Input Line Entry System) [Weininger, 1988]. SMILES provides a concise, intuitive way to represent organic molecules by linearizing a spanning tree of a molecular graph (Fig. 2.2). Since the spanning tree can be rooted at any atom, a molecule can have multiple different equivalent SMILES representations; however, all of them can be converted to a canonical SMILES string for disambiguation. Although SMILES was initially designed solely for the exchange of chemical information, this notation turned out to be remarkably compatible with deep learning models used in the domain of natural language processing (NLP): researchers have relied on SMILES as training data for molecular representation learning [Winter et al., 2019; Karpov et al., 2019a], property prediction and QSAR modeling [Ross et al., 2022], de novo molecular generation [Gómez-Bombarelli et al., 2018], structure elucidation from spectra [Shrivastava et al., 2021], reaction and synthesis prediction [Schwaller et al., 2019], etc. Moreover, SMILES prompted the development of string notations specifically tailored for molecular generation with neural models, such as DeepSMILES [O'Boyle and Dalke, 2018] and SELFIES [Krenn et al., 2022]. Research on SMILES-based deep learning models is actively developing.

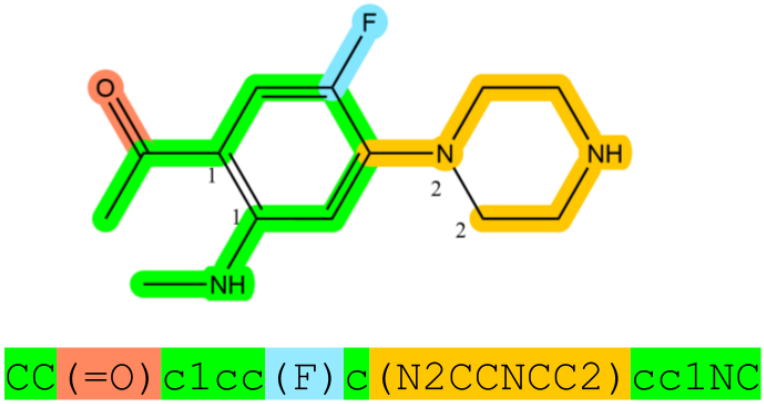

Figure 2.2. Organic molecules can be represented as strings in the SMILES (simplified molecular input line entry system) notation. A SMILES string is a linearization of a spanning tree in a molecular graph.

2.2.2 Reactions

A chemical reaction [2] is a process that results in the interconversion of chemical species. A reaction in organic chemistry is usually represented as a transformation between molecular graphs. In principle, the total mass and charge of the reacting species are conserved in any chemical reaction. However, organic chemists often omit all products except the one they are most interested in when recording reactions. Figure 2.4A shows an example of a reaction scheme, a graphical depiction of an overall reaction, often drawn without explicit mass balance, that organic chemists commonly use in manuscripts. Reactions can be either elementary or stepwise. The end products of a stepwise reaction result from a sequence of elementary reactions and unstable intermediates. The detailed description of such a sequence for a reaction is called a reaction mechanism. A generic mechanism defines a reaction type that various molecules with specific structural motifs may undergo. Figure 2.4F displays the mechanism of the reductive amination reaction. There are many hundreds of known reaction types [Li, 2006] with mechanisms of varying complexity. Determining the mechanism of a novel reaction type is a hard problem that always requires considerable effort.

Figure 2.3 shows a reaction scheme corresponding to the instance of the reaction

[2] IUPAC compendium of chemical terminology. URL: https://doi.org/10.1351/goldbook.C01033

type called Suzuki coupling [Miyaura and Suzuki, 1995] - a staple of modern organic synthesis. A reaction scheme typically involves reactants, products, and reagents. The reactants are placed to the left of the arrow, the products to the right, and the reagents may be placed above the arrow, below it, or to the left, together with the reactants. The definition of products is obvious, but the distinction between reactants and reagents is somewhat blurry. IUPAC defines a reactant as a substance consumed during a chemical reaction [3], and suggests that reagents and reactants are synonyms. In cheminformatics, however, "reagents" often refers specifically to molecules that enable a reaction but do not contribute atoms to the final products. For example, catalysts and solvents are reagents. Reagents may be necessary to enable a reaction mechanism by contributing atoms to unstable intermediates, but not to the final products. In practice, even substances with other roles, such as reducing and oxidizing agents, may also be considered reagents because chemists often write them above the arrow in chemical literature, even though they may contribute atoms to the products. The distinction between reactants and reagents is important for synthesis planning; however, there is still no standard definition, and various researchers define this distinction based on the tasks they face. Together, reactants and reagents form the set of precursors.

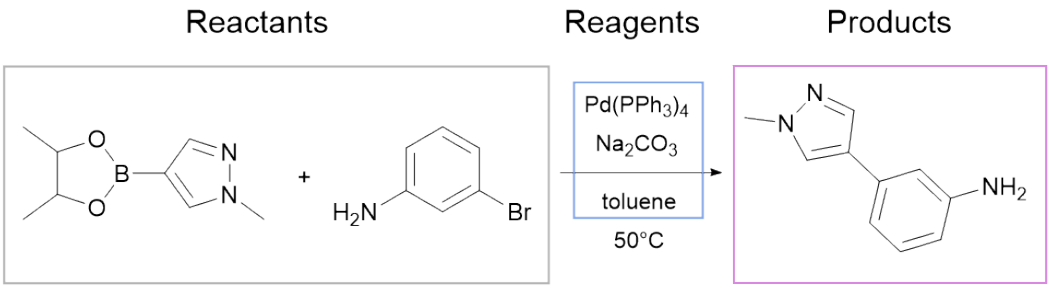

Figure 2.3. An instance of the Suzuki coupling reaction. Reagents are normally written above or below the arrow. Various reaction types are often enabled by specific reagents.

Since individual molecules can be represented in text as SMILES, reactions have a text representation by extension. To make a reaction SMILES, one writes the SMILES of all reactants separated by dots, then the SMILES of all reagents separated by dots, and the SMILES of all products, and separates the three strings with ">" characters (Fig. 2.4B). Since only some of the bonds are broken and formed in a chemical reaction, it is possible to extract a generic transformation representation called either a SMIRKS [Daylight Chemical Information Systems, Inc.] or a SMARTS template [Coley,

[3] IUPAC compendium of chemical terminology. URL: https://doi.org/10.1351/goldbook.R05163

Green and Jensen, 2019] (Fig. 2.4D). A reaction SMARTS template can be applied to SMILES in chemical data manipulation software to deterministically turn one SMILES string into another. The template can be applied in both directions: to convert products into reactants and reactants into products. Reaction SMILES can include atom-to-atom mapping (AAM), which assigns numerical labels to corresponding atoms in reactants and products (Fig. 2.4C). This atom mapping is necessary to extract reaction templates in SMIRKS format. Atom mapping placement for arbitrary reactions is a challenging task that requires complex algorithms or machine learning [Schwaller et al., 2021; Nugmanov et al., 2022]. Although it is never solved with absolute accuracy [Lin, Dyubankova, Madzhidov, Nugmanov, Verhoeven, Gimadiev, Afonina, Ibragimova, Rakhimbekova, Sidorov et al., 2022], the accuracy of the latest methods is remarkable [Chen, An, Babazade and Jung, 2024].

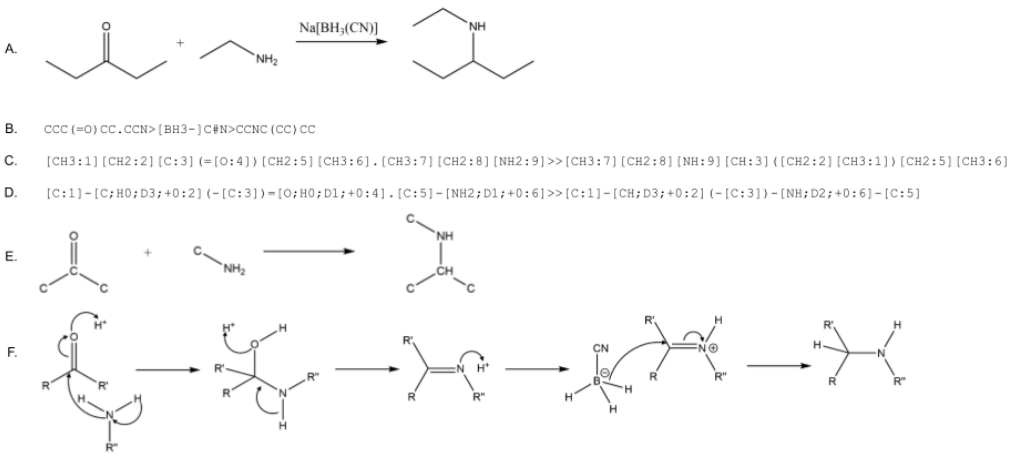

Figure 2.4. Various representations of an instance of the reductive amination reaction. **A**: Reaction scheme, a 2D diagram that expresses the overall transformation. **B**: Reaction SMILES. **C**: Reaction SMILES with atom-atom mapping (AAM). **D**: SMARTS/SMIRKS template of the reaction, a string expression that encodes the change of atom connectivity in the reaction. It can be applied to the reactants to generate the products and vice versa. **E**: Reaction template displayed as a diagram. **F**: a diagram of the actual general mechanism of the reaction. Curved arrows depict the movement of an electron pair. Reaction mechanisms typically consist of multiple elementary steps and unstable intermediates.

2.3 Sources of chemical data

Applying machine learning to make meaningful predictions at the molecular and reaction levels requires access to large amounts of high-quality data. Unfortunately, cheminformatics has never been as data-rich as fields like computer vision (CV) or natural language processing (NLP). Neural networks in CV and NLP are trained on datasets that can be scraped from the internet in virtually unlimited quantities and curated with relative ease. In contrast, chemical data is scarce, highly sensitive to the slightest corruptions (which can render it invalid), and requires expert curation by trained specialists. Although pharmaceutical companies amassed ample chemical knowledge in their repositories, most of them are guarded as company secrets and unavailable to the broader research community. Nevertheless, in recent decades, several open datasets have emerged that provide molecular structures, physicochemical properties, biological activities, chemical reactions, and other relevant information. These datasets form the backbone of modern cheminformatics and serve as the primary training material for machine learning models.

2.3.1 Molecules

For drug discovery purposes, one highly useful source of data on molecules is the ChEMBL database [Zdrazil et al., 2024] - a manually curated collection of bioactive, drug-like molecules and their measured bioactivities from medicinal chemistry literature. ChEMBL contains about 2.4 million unique structures, each annotated with standardized activity data, and is widely used for training QSAR models. A larger alternative to ChEMBL is the PubChem database [Kim et al., 2025], which contains approximately 119 million unique compounds. Most ChEMBL compounds are also included in PubChem; however, PubChem also collects data from sources other than journal articles, and its bioactivity data tends to be less uniformly curated and of variable quality. An even larger molecular dataset is ZINC [Tingle et al., 2023], a catalog of compounds that can be synthesized on demand, containing 37 billion molecules. ZINC primarily does not offer bioactivity information, but provides 3D representations of molecules ready for molecular docking, and is useful for training de novo molecular generators. Finally, GDB-17 [Ruddigkeit et al., 2012] is the largest molecular dataset with 166 billion molecules. However, the molecules in it are generated by combinatorial enumeration of reasonable structures, most of which have never been synthesized, and no information beyond SMILES strings is provided in the database. Overall, the more specific the information required to train a machine learning model on molecules, the smaller the datasets that provide it, and they are relatively hard to assemble.

2.3.2 Reactions

Reaction data is scarcer than molecular data and harder to obtain and standardize. When describing reactions in manuscripts and laboratory notebooks, chemists often omit details that seem obvious to them but are hard for database maintainers to reconstruct. Moreover, reactions carry valuable metadata, such as experimental yield and the exact experimental procedure, which are often lost, insufficient, or unreliable in reaction databases. Large reaction databases assembled by manual extraction of reaction data from manuscripts spanning many decades, such as Elsevier's Reaxys, which features around 55 million reaction records of varying quality, are commercial and subject to restrictive licensing terms. Therefore, they are usually not an option for most researchers.

The situation improved in 2012, when Daniel Lowe, a PhD student from Cambridge University, published the first open dataset of chemical reactions derived from text mining of US patents, known as the USPTO dataset [Lowe, 2012]. USPTO enabled rapid progress in machine learning for reaction prediction and synthesis planning, spawned a family of specialized data subsets, and prompted further development of open reaction datasets such as ORD [Kearnes et al., 2021], and ORDerly [Wigh et al., 2024]. Unfortunately, the quality of the data in USPTO is far from ideal. It contains approximately 1.5 million unique reaction SMILES, which may or may not be accompanied by reaction yields. However, it lacks information on experimental procedures and comprehensive reaction conditions. In the available reaction SMILES, atom-atom mapping is often incomplete or incorrect, reactions from different experiments are occasionally combined into a single entry, and reagents are sometimes missing. Also, the distribution of reaction types is far from balanced or complete [Thakkar et al., 2020; Zheng et al., 2019]. Such issues are not minor and severely limit the capabilities of machine learning models trained on USPTO. Researchers often invent ways to clean and improve the USPTO data [Chen and Li, 2024; Toniato et al., 2021] or turn to a better-curated version of USPTO called Pistachio, which, however, is proprietary and not openly available. Unlike molecules in GDB-17, reaction data cannot be generated by brute-force enumeration (although some researchers attempt to do it [Deng et al., 2025]), because an arbitrary reaction scheme is not guaranteed to describe an actually occurring process. In principle, one could envision an expert system that encodes chemical rules and exceptions, and then generates plausible reactions that include reagents and conditions. Indeed, Kayala and colleagues [Kayala and Baldi, 2011] used a closed expert system to assemble a dataset for training product prediction models. However, such systems remain limited to simple reaction types, and the generated examples are still not universally justified.

The availability and quality of data are the primary limiting factors in machine

learning for chemical reactions today, and with improved data, models of impressive capability will be developed in the future.

2.4 Computer-aided synthesis planning

Planning of chemical syntheses is an inherently algorithmic task that can be carried out on a computer. In this section, we give an overview of the historical and modern approaches to synthesis planning.

2.4.1 Historical approaches

In the early days of computing, it was already clear that computers could be highly useful for planning chemical syntheses. George Vléduts, a pioneer of chemical information systems, wrote in 1963: "It is comparatively easy to imagine how a logico-information machine will be able, starting from the given final compound, to construct the various possible 'chains' of synthesis and 'trees' describing the formation of this final product from all possible 'precursor compounds'" [Vleduts, 1963]. This vision did not take long to be fulfilled. Around the same time, Elias Corey developed and described a systematic methodology of organic synthesis planning [Corey, 1967] which is now known as "retrosynthetic analysis" [Corey and Cheng, 1989]. Today, any university course in organic chemistry is unimaginable without it. Still, back then, it presented a major change in mindset: chemists do not usually reason in terms of retrosynthetic analysis, but rather in analogous syntheses and reactions they are used to. Although retrosynthetic analysis is, in principle, the most powerful approach to synthesis planning, it would require more time and cognitive effort from a chemist than simple analogy-based thinking; however, it could be formalized on a computer. Corey immediately implemented retrosynthetic analysis on the available computer, and in 1969 presented OCSS - the first software for systematic planning of organic synthesis [Corey and Wipke, 1969], which started the discipline of computer-aided synthesis planning. In 1990, Corey was awarded the Nobel Prize in Chemistry for the development of retrosynthetic analysis.

OCSS was an expert system: it relied on a set of reaction rules defined by the expert chemist and incorporated into the program's source code. The rules would define the transformations and the corresponding structural filters for substrates. When the user would input a desired synthesis product, the program would apply several fitting transforms in the order of their priorities, trying to simplify the input molecule in a certain sense. The program would propose only single steps of the plan, returning control to the chemist to choose the next molecule to transform. Many follow-up works aimed at improving various aspects of the described design. LHASA [Corey et al., 1972] was an

updated version of OCSS with incremental improvements. SYNCHEM [Gelernter et al., 1977] was the first to build synthesis plans autonomously using heuristic best-first tree search. SECS [Wipke et al., 1978] supported an external database for reaction rules, allowing the addition of new transforms to the system's library without changing the source code. SYNGEN [Hendrickson and Toczko, 1989] supported convergent synthesis planning - an efficient general strategy that is rarely considered even in modern CASP systems [Torren-Peraire et al., 2025]. EROS [Gasteiger and Jochum, 1978] and other systems developed by Johann Gasteiger, a pioneer of neural networks in cheminformatics, encoded transforms using special bond-electron matrices [Dugundji and Ugi, 1973] instead of scripts in a programming language, and focused heavily on verifying retrosynthetic steps with a reaction feasibility subroutine. Overall, by the 1990s, there were around 10 expert systems for organic synthesis planning. However, none of them were particularly successful. As Gasteiger and Ihlenfeldt wrote in their 1996 review: "Computer-assisted synthesis planning has been met with utter skepticism, even hostility, from the majority of chemists; ... not a single system has become widely accepted as a routine tool, and the generally very critical attitude towards this kind of system has been corroborated". The available software was rarely activated and demonstrated no user retention [Ihlenfeldt and Gasteiger, 1996]. The problems with expert systems for CASP were, among others, the constant need to update the transforms library to cover various use cases, the overwhelming amount of generated synthesis trees that chemists were unwilling to review, unfeasible reactions in synthesis plans, and the general remoteness of CASP from the intuition of a human chemist. Gasteiger and Ihlenfeldt, however, believed that the field of CASP is "far from dead" and that all its problems will be overcome in the future. Today, the interest in computer-aided synthesis planning has indeed been reignited by the success of deep learning and its promise to distill raw data into concentrated chemical knowledge that will be applied appropriately and autonomously.

Some research groups still adhere to the expert system approach to synthesis planning. For example, the group led by Prof. B. Grzybowski in Poland is developing a proprietary Synthia/Chematica system [Grzybowski et al., 2018]. It relies on expert-defined chemical rules for single-step expansions of the synthesis tree and enjoys commercial success. However, the overall research interest has largely shifted towards deep learning approaches to CASP.

2.4.2 Machine learning-based CASP

The possibility to efficiently train deep neural networks on graphics processing units (GPUs), demonstrated decisively in early 2010s image classification competitions [Cireşan et al., 2011; Cireşan et al., 2012; Krizhevsky et al., 2012], revived widespread interest

in machine learning, establishing it as the most promising path toward artificial intelligence. Cheminformaticians drew inspiration from advances in computer vision and natural language processing and began exploring the potential of machine learning for synthesis planning.

A comprehensive synthesis planning system relies on solutions to a range of individual reaction modeling problems, such as reaction product prediction, single-step retrosynthesis prediction, conditions prediction, yield prediction, and so on. We will give an overview of approaches to some of them. [4]

2.4.3 Reaction prediction

The problem of predicting reaction products from precursors, also known as product prediction, forward prediction, or reaction prediction, is one of the foundational problems in automated synthesis planning and an important part of a comprehensive CASP system. In principle, the exact solution of the reaction prediction problem for arbitrary sets of reacting molecules would be tremendously challenging and would require detailed thermodynamic modeling of every step of the reaction mechanism. Alternatively, one could train a neural network on a dataset of known reactions to predict the products.

Machine learning approaches to reaction prediction broadly divide into two schools: template-based methods and template-free methods. Template-based methods rely on the availability of reaction templates and use them as target classes for supervised classification. The simplest template-based method is to select a template using a simple classifier trained on vector representations of reactant sets. The selected template is then applied to the SMILES of the reactants to generate the product SMILES. The team of Prof. Alán Aspuru-Guzik [Wei et al., 2016] demonstrated this approach by training a simple MLP on a small dataset manually extracted from a chemistry textbook using concatenated molecular fingerprints of reactant molecules as input. Segler and Waller [Segler and Waller, 2017] scaled the same approach to 3.5 million reactions from Reaxys. Coley et al. [Coley, Barzilay, Jaakkola, Green and Jensen, 2017] proposed a variant of this method in which several matching reaction templates enumerate a range of possible products, which are then embedded in a vector space and ranked by a dense neural network. Template-based methods are straightforward to implement and mirror the paradigm of expert systems, but they share the same limitation: they cannot predict reactivity beyond predefined templates.

In contrast to template-based methods, template-free methods bypass the classification into templates when predicting reaction outcomes. Template-free methods

[4] See Tables E.1 and E.2 in the Appendix for a summary of ML approaches to reaction prediction and single-step retrosynthesis.

are split into several branches. One branch of template-free methods is based on the general idea of moving from one molecule to another via graph editing operations. In graph-edit-based methods, a neural network works over embeddings of atoms and bonds in a starting molecular graph and models a multi-step process that gradually leads to the reaction product. Earlier works described a gated graph neural network trained on a subset of USPTO to repeatedly classify pairs of atom nodes to add or remove an edge in the reactant graph until it turns into the desired graph [Bradshaw et al., 2018; Do et al., 2019]. Sacha et al. [Sacha et al., 2021] used an encoder-decoder graph attention network and a more extensive set of graph editing actions, training the model for both forward prediction and single-step retrosynthesis. Coley and his colleagues [Jin et al., 2017; Coley, Jin, Rogers, Jamison, Jaakkola, Green, Barzilay and Jensen, 2019] transferred their earlier product-ranking method [Coley, Barzilay, Jaakkola, Green and Jensen, 2017] into the template-free domain using a graph convolutional network called WLDN. The network learns to output likelihood scores for each bond change between each atom pair in the reactant graph. The candidate products are then ranked to obtain the most likely product. As a follow-up, they extended this approach by augmenting learned atom embeddings with descriptors obtained from quantum-chemical calculations and provided in a separate dataset [Stuyver and Coley, 2022].

The competing direction in template-free methods draws inspiration from neural language modeling, a field of neural network research with a long history [Forcada and Ñeco, 1997; Mikolov and Zweig, 2012; Graves, 2012; Sutskever et al., 2014], and aims to generate the SMILES of the products conditioned on the reactant SMILES. Earlier methods relied on Seq2Seq [Bahdanau, 2014] models based on LSTM [Hochreiter and Schmidhuber, 1997] to ”translate” the reactant SMILES into product SMILES [Nam and Kim, 2016; Schwaller et al., 2018]. When the Transformer architecture [Vaswani et al., 2017; Schmidhuber, 1992b; Schlag et al., 2021] crystallized as the most scalable architecture for training on contemporary hardware and started replacing Seq2Seq LSTM in conditional sequence generation, it was quickly adapted to reaction prediction in the seminal Molecular Transformer paper [Schwaller et al., 2019]. The Molecular Transformer is an encoder-decoder transformer with 12 million parameters and a shared vocabulary between the encoder and decoder, trained on text pairs of input SMILES and output SMILES from the USPTO. This straightforward approach achieved state-of-the-art accuracy in reaction product prediction and became the brand name for template-free methods in reaction modeling, inspiring numerous follow-up works from synthesis planning researchers.

There are also other original branches of research in reaction prediction that branch away from the main line of template-based and template-free methods. The laboratory of Prof. Pierre Baldi at the University of California, Irvine, has been developing the

”mechanistic” approach to reaction prediction since the early 2010s [Kayala and Baldi, 2011; Kayala et al., 2011; Kayala and Baldi, 2012; Fooshee et al., 2018]. The idea is to predict reactions at the level of the mechanistic steps of the true reaction mechanisms. The primary aim of the method is to predict reaction products in an explainable way, and the defining challenge of the mechanistic approach lies in data preparation. Since USPTO or large proprietary databases do not include information on reaction mechanisms (although efforts to augment USPTO with mechanistic information are emerging [Chen, Babazade, Kim, Han and Jung, 2024]), the data for the mechanistic approach must be assembled manually or through a closed-source expert system, which hurts both scalability and often the reproducibility of the approach. Sadly, the mechanistic approach receives little attention compared to other methods, even though it could be extremely useful, for example, to verify the steps of an automated synthesis plan and to filter out infeasible retrosynthetic transformations in CASP systems. Another outstanding reaction prediction method is non-autoregressive electron redistribution modeling [Bi et al., 2021; Meng et al., 2023a], in which a neural network is trained to rearrange all bonds in a molecular graph in one forward pass by predicting the difference between the adjacency matrix of the reactants and the products, each represented as a single graph. This method is notable for its inference speed and the ability to produce predictions that are balanced in mass with the reactants. However, it does not extend to single-step retrosynthesis.

2.4.4 Single-step retrosynthesis

Single-step retrosynthesis is the reverse task of reaction prediction. Given a target product molecule, the goal is to suggest plausible sets of reactants that could yield that product in a single reaction. This task is fundamentally more challenging than reaction prediction because of its inherent one-to-many ambiguity: multiple distinct sets of reactants may react to form a given product. In reaction prediction, the same reactants may sometimes result in several possible products, but the ambiguity of single-step retrosynthesis is much greater.

Coley et al. [Coley, Rogers, Green and Jensen, 2017] demonstrated RetroSim—a simple approach to single-step retrosynthesis built around deterministic molecular similarity. For a query product, it first retrieves a set of reactions from a database based on molecular similarity s_{prod} between their products and the query product; then, it extracts the SMIRKS template from every reaction and applies it to the query product; then, for every template, it calculates the similarity s_{reac} between the reactants generated for the query and the precedent reactants that gave the template; finally, it orders the generated products by the corresponding $s_{\text{prod}} \cdot s_{\text{reac}}$. This method demonstrated surprisingly high accuracy for its simplicity and served as a baseline for subsequent

machine learning-based approaches to single-step retrosynthesis.

Neural single-step retrosynthesis methods can be broadly classified into three categories. Similarly to reaction prediction, they can be template-based or template-free, and the specificity of the task enables an additional category of semi-template-based methods.

Template-based methods formulate the task as supervised classification of given product molecules, where labels correspond to predefined templates, which may be constructed manually or derived automatically from a dataset of atom-mapped reactions. After a template for a product is selected, the reactants are generated by applying that template to the product. Segler and Waller [Segler and Waller, 2017] trained an MLP and a highway network [Srivastava et al., 2015] for such a classification of molecules, with molecular fingerprints serving as input features. They used a large proprietary Reaxys dataset and demonstrated their method for both single-step retrosynthesis and reaction prediction. Baylon et al. [Baylon et al., 2019] proposed a two-stage classification of products: first, into general reaction groups obtained by template clustering; then, into templates within a corresponding group. Dai et al. [Dai et al., 2019] developed a method called GLN, which modeled the joint distribution of templates and reactants using logic variables. The state-of-the-art LocalRetro method [Chen and Jung, 2021] employed a GNN encoder to classify every atom and bond in a molecule into a reaction template that is then applied to the specific predicted location. RetroKNN [Xie et al., 2023] extended the LocalRetro method by ensembling a linear classifier and a KNN classifier over GNN embeddings of atoms and bonds. The authors of RetroComposer [Yan et al., 2022] aimed to address the fundamental limitation of the template-based methods - the limited set of available transformations - by training an RNN to compose predefined templates into potentially novel transformations before applying them to the query product. The authors of RetroGFN [Gaiński et al., 2025] further developed the template construction approach of RetroComposer and trained a GFlowNet guided by a reaction feasibility model in an effort to increase the diversity and potential usefulness of unique predictions. The team of Prof. Sepp Hochreiter [Seidl et al., 2022] trained a recommender system for products and templates with the goal of zero-shot generalization to templates unseen during training, using an architecture they referred to as a ”Modern Hopfield Network encoder.” [5]

Template-free methods in single-step retrosynthesis have developed similarly to those in reaction prediction. The advantage of template-free methods over template-based methods is their better scalability with respect to training data size and diversity. Template-free methods can be further divided into text-based and graph-based

[5] Hopfield networks should probably be called ”Amari networks”, as argued in https://people.idsia.ch/ juergen/physics-nobel-2024-plagiarism.html

approaches.

Text-based template-free methods treat retrosynthesis as a sequence-to-sequence translation problem. Liu et al. [Liu et al., 2017] first approached the task as direct SMILES-to-SMILES generation with an encoder-decoder Seq2Seq LSTM. The model generated a large proportion of invalid SMILES, and the accuracy of the model did not surpass that of the machine learning-free method RetroSim [Coley, Rogers, Green and Jensen, 2017]. The team of Dr. Igor Tetko [Karpov et al., 2019b] was the first to train a transformer model similar to the Molecular Transformer to significantly improve the accuracy of single-step retrosynthesis prediction compared to Seq2Seq LSTM. Later, Tetko et al. [Tetko et al., 2020a] further boosted the performance of such a SMILES-to-SMILES transformer by training the model for single-step retrosynthesis and reaction prediction simultaneously and by applying SMILES augmentations to the training data. Since SMILES are based on a spanning tree in a molecular graph, any atom in the molecule can be the root of the tree, allowing multiple different SMILES strings to represent the same molecule. Zheng et al. [Zheng et al., 2019] presented SCROP—an encoder-decoder transformer with a separate model for correcting invalid generated SMILES. Kim et al. [Kim et al., 2021] trained the transformer with two tied decoders—one for reactant prediction and the other for forward prediction—to verify that the generated reactants would indeed yield the desired product. Chemformer [Irwin et al., 2022] aimed to improve transformer accuracy by scaling up model size relative to previous SMILES-to-SMILES methods. The authors also incorporated self-supervised pretraining on masked SMILES reconstruction, following the BART pretraining paradigm from natural language processing. Retroformer [Wan et al., 2022] modified the transformer with architectural changes intended to better align it with the molecular generation task. The model's inputs are the SMILES string, the adjacency matrix, and bond features of a molecule, and some of the attention heads in the multihead attention layers are "local," with an attention mask that restricts the receptive field of an individual token to its one-hop neighborhood based on molecular connectivity. Zhong et al. [Zhong et al., 2022] introduced root-aligned SMILES (RSMILES), an improved SMILES augmentation method for the basic SMILES-to-SMILES transformer. In RSMILES, the starting atom for SMILES generation is chosen to minimize the edit distance between the product and reactant strings, thereby simplifying model training. Such a preprocessing step dramatically improved transformer performance, achieving state-of-the-art results and demonstrating that thoughtful data preprocessing can be more valuable than architectural sophistication for single-step retrosynthesis. Several notable template-free methods diverge from the main line of autoregressive SMILES-to-SMILES generation: EditRetro [Han et al., 2024] describes a non-autoregressive encoder-decoder transformer that learns to edit a source SMILES string "in-place" to turn it into a target sequence. A closely related DiffER model [Current et al., 2025]

trains to transform source SMILES into target SMILES via categorical diffusion. Retro-TRAE [Ucak et al., 2022] replaced SMILES encoding and generation with the encoding and generation of a set of indices corresponding to SMARTS patterns of molecular substructures, essentially generating molecular fingerprints. The final predictions from Retro-TRAE are obtained by retrieving the molecules with the closest fingerprints to the prediction from a large database such as PubChem. Graph2SMILES [Tu and Coley, 2022] replaces a text encoder with a graph encoder in a SMILES generator, supporting both reaction prediction and single-step retrosynthesis. Shee et al. [Shee et al., 2024] blurred the boundary between template-based and template-free approaches by training a transformer to generate reaction templates autoregressively instead of retrieving them from a fixed collection. Xuan-Vu et al. [Xuan-Vu et al., 2025] demonstrated the benefits of the same template-generative approach for improved generalization to input SMILES of increasing length.

Graph-based template-free methods operate directly on molecular graph representations rather than linearized SMILES strings. MEGAN [Sacha et al., 2021] works in both reaction prediction and single-step retrosynthesis, training an autoregressive graph-to-graph model to predict a sequence of graph edits. G2GT [Lin et al., 2023] uses a Graphormer encoder and generates the output graph as a sequence of nodes and edges instead of applying transformations to the input graph. Graph2Edits [Zhong et al., 2023] is a model inspired by MEGAN that proposes constructing the basis set of graph edits by using reaction templates extracted from the training data. RetroBridge [Igashov et al., 2023] proposes a Markov Bridge model, an alternative to discrete diffusion and a general framework for learning the dependency between two intractable discrete-valued distributions, and applies it to molecular graphs in single-step retrosynthesis.

Semi-template-based models form the third major family of single-step retrosynthesis predictors and formulate the task as a two-stage problem. In the first stage, called reaction center identification, the model must identify the reaction center in the query product molecule, i.e., all pairs of atoms that will change connectivity. A deterministic program then applies the predicted change to the query molecule, generally generating incomplete molecular fragments called synthons. In the second stage, called synthon completion, the model must complete the synthons to produce valid reactant molecules. The first stage is most often formulated as a supervised edge classification problem, and the second stage may be approached in different ways. The first semi-template-based method was G2Gs [Shi et al., 2020] . G2Gs approached the first stage by learning atom representations in a molecule with a relational GCN architecture, obtaining edge representations by concatenating the embeddings of the two participating atoms, and classifying the representations of all edges into binary labels with a logistic regression head. The model selects the prediction with the largest

probability over a threshold, and the corresponding bond is broken, producing the synthons. A set of isolated atoms of all elements is added to the synthons, and the resulting graph with multiple connected components is passed to the synthon completion module. The latter is implemented as a conditional VAE with a decoder that autoregressively predicts connectivity changes for the input graph until terminated by a stop-prediction. The major limitation of G2Gs is that it does not support transformations in which several bonds break simultaneously. RetroXpert [Yan et al., 2020] used an edge-enhanced GAT in the center identification stage to learn edge representations directly, supported breaking multiple bonds in a reaction center, and treated synthon completion as SMILES-to-SMILES translation with a transformer. GraphRetro [Somnath et al., 2020] combined a molecular graph with its dual graph to learn richer representations of atoms and bonds with a message-passing neural network, and posed synthon completion as a supervised classification task, matching synthons with leaving groups from a predefined set. Unlike G2Gs and RetroXpert, GraphRetro is more template-based than template-free. RetroPrime [Wang et al., 2021] trained separate SMILES-to-SMILES transformers for both generating synthons from the product and for completing synthons to reactants. RetroExplainer [Wang, Pang, Wang, Jin, Zhang, Zeng, Su, Zou and Wei, 2023] replaced a graph neural network with a transformer for learning representations of atoms and bonds in a molecular graph for reaction center identification, and approached synthon completion similarly to GraphRetro, while changing the classification objective to a contrastive learning objective. G^2Retro [Chen et al., 2023] added nuance to reaction center identification by distinguishing three different types of reaction centers; for synthon completion, it sequentially adds substructures from a predefined vocabulary onto the synthons until complete reactants are formed. Retro-MTGR [Zhao et al., 2025] improves the learning of molecular representations with a contrastive objective: the embeddings of molecules and their synthons are trained to be close together in the embedding space. GDiffRetro [Sun et al., 2025] proposed a 3D diffusion model for synthon completion.

Despite the significant differences in general approaches and implementation details, there is no clearly dominant method family among the three in terms of prediction accuracy, and the best-performing template-based, semi-template-based, and template-free methods all perform comparably. Careful data curation seems to have more impact than sophisticated neural architectures, as indicated, for example, by the substantial benefits of SMILES alignment for SMILES-to-SMILES translation methods.

2.4.5 Multi-step synthesis planning

In 2018, Segler et al. outlined the general design of a machine learning-based CASP system in their seminal work [Segler, Preuss and Waller, 2018]: a neural network for

single-step retrosynthesis prediction and a planning algorithm. The single-step model predicts several sets of precursors for every input, and the planning algorithm builds the retrosynthesis tree, where the given product is the root node and the leaf nodes are building blocks, i.e., molecules from a predefined collection that are considered readily available. At every iteration, the planning algorithm selects the single-step expansions that are the most promising. Segler and coauthors themselves proposed Monte Carlo tree search (MCTS) as the planning algorithm. However, the general framework allows other options, such as depth-first proof-number search, breadth-first search, A*, or variants of A* designed specifically for chemical synthesis planning, such as Retro* [Chen et al., 2020] or Tango* [Armstrong et al., 2025]. Soon, various open-source implementations of machine learning-based CASP emerged, such as AiZynthFinder [Genheden et al., 2020; Saigiridharan et al., 2024], Syntheseus [Maziarz et al., 2025], ASCCOS [Tu et al., 2025], and SynPlanner [Akhmetshin et al., 2025]. Additionally, the need to advance individual aspects of neural reaction models towards creating useful CASP systems prompted the search for the best combinations of single-step models and planning algorithms [Torren-Peraire et al., 2024; Choe et al., 2025; Maziarz et al., 2025]. Such research, however, remains scarce and inconclusive, and it is further complicated by the interactions of the large number of settings in every single-step model and every planning algorithm. Some researchers are trying to move away from Segler's framework and generate the entire synthesis tree with a text generator [Shee et al., 2025; Granqvist et al., 2025; Song et al., 2025]. However, the advantages of such an approach are not fully clear yet.

The described general framework for ML-based CASP is simplified, prototypical, and insufficient as a basis for a production-ready CASP system. The latter would have to include several important subroutines, the descriptions of which are scattered across numerous isolated works. One example of such a subroutine is a product prediction model that verifies the feasibility of transformations proposed by the single-step retrosynthesis model. For a given query product, the single-step retrosynthesis model may often predict precursors that would yield a different product, invalidating the entire synthesis plan. Possible solutions include filtering the reactions in synthesis plans with a regioselectivity predictor [Sigmund et al., 2025] or choosing single-step models based on their round-trip accuracy [Schwaller et al., 2020] - the percentage of reactions in which a separate product prediction model predicts the query product for the precursors suggested by the single-step retrosynthesis model. Another example of a useful subroutine is a yield prediction for reactions in the synthesis tree, which would be useful for guiding the function that selects promising expansions. However, yield prediction is a hard problem that typically requires more information than is contained in a reaction scheme alone [Voinarovska et al., 2023]. The next example of a necessary CASP subroutine, one we find particularly important, is a model for predicting

conditions for reactions in retrosynthetic trees. In real-life organic syntheses, every organic reaction is only enabled under specific combinations of temperature, pressure, experiment length, and other conditions. Additionally, reactions often require specific solvents, catalysts, and other reagents that are not predicted by single-step retrosynthesis models. If reagents are not predicted in a CASP system, its user will have to choose them based on their own expertise. A CASP system that assists chemists in deciding on reagents and conditions is more useful than one that does not. Additionally, the speed of synthesis planning is an important consideration. Besides assisting chemists with synthesis plans for individual molecules, a CASP system should ideally serve as a synthesizability filter for a stream of *de novo* generated molecules. For example, the early drug candidates produced by molecular generators must be synthesizable, and the most reliable way of assessing a molecule's synthesizability is to try to construct its synthesis plan. Current synthesis planning methods take too long to find a synthesis route for a single molecule, which prompts palliative methods such as molecular complexity scores [Ertl and Schuffenhauer, 2009; Thakkar et al., 2021] and molecular generators with synthesizability priors [Gao et al., 2021; Gao et al., 2025], to separate synthesizable molecules from all others. Nonetheless, fast automated synthesis planning for large amounts of molecules is in principle the preferred method for synthesizability screening. Therefore, low latency is a requirement for an ideal CASP system.

In the following chapters, we present the results of our work aimed at improving modern deep learning-based synthesis planning from two angles: retrosynthesis speed and reagent prediction.

Chapter 3

Faster SMILES generation for single-step retrosynthesis

Early drug candidates, obtained, e.g., using a molecular generative model, always undergo extensive filtering, one aspect of which is filtering for synthesizability. Synthesizability, i.e., the existence of a valid synthesis route from a given molecule to the available building blocks, may depend on factors such as route length, yield, cost, the available stock of building blocks, the guidelines for allowed reaction types, and so on [Stanley and Segler, 2023]. Constructing a complete retrosynthetic tree with a Computer-Aided Synthesis Planning (CASP) system provides the most rigorous and flexible assessment of synthesizability. Often, in practical workflows [Liu et al., 2022; Hassen et al., 2025], full synthesis planning may be replaced with filtering of generated molecules by molecular complexity scores such as SA score [Ertl and Schuffenhauer, 2009] or SYBA [Voršilák et al., 2020], or by machine-learned retrosynthesis prediction scores such as RA score [Thakkar et al., 2021], which estimate the likelihood that a CASP system will identify a synthetic route. However, these surrogate metrics provide only approximate assessments and do not replace explicit retrosynthetic tree construction; they may miss feasible routes or overestimate synthetic accessibility. Therefore, accelerating full retrosynthesis planning remains necessary for reliable synthesizability estimation of large amounts of novel molecules.

Computer-aided synthesis planning (CASP) systems, described in detail in Section 2.4, exist to reduce the cognitive burden of developing synthesis plans for chemists, and have a long tradition dating back to the 1960s. While their primary purpose is usually to assist chemists in planning individual experiments, they can potentially quickly plan

This chapter is based on our paper "Accelerating the inference of string generation-based chemical reaction models for industrial applications" [Andronov et al., 2025a]

syntheses for large collections of molecules, providing a reliable filter for the synthesizability of early drug candidates.

While CASP systems leveraging template-based single-step models proved to be effective [Genheden et al., 2020], there is an interest in building CASP with template-free models instead, as they demonstrate state-of-the-art accuracy in both single-step retrosynthesis and reaction product prediction [Zhong et al., 2022; Meng et al., 2023b]. The most accurate template-free models are currently conditional autoregressive SMILES generators based on the transformer architecture [Vaswani et al., 2017], which also serves as the backbone for Large Language Models (LLMs) [Brown et al., 2020; Zhao et al., 2023]. Unfortunately, the high accuracy of such autoregressive models like Chemformer [Irwin et al., 2022] comes at the cost of a slow inference speed [Torren-Peraire et al., 2024], which poses a problem that hinders the successful adoption of those models as part of industrial CASP systems, especially when it comes to high-throughput synthesizability screening. Therefore, we sought to improve the speed of SMILES-to-SMILES transformers for the benefit of CASP.

Our main contributions: We propose a method of inference acceleration for template-free reaction models that are trained to autoregressively generate output SMILES conditioned on input SMILES. We leverage the common overlap between SMILES strings on both sides of any organic chemical reaction and employ speculative decoding, a general technique for LLM inference acceleration [Leviathan et al., 2023; Xia et al., 2023], to perform SMILES-to-SMILES translation more efficiently. We design a chemically specific heuristic drafting scheme for speculative decoding of SMILES in reaction prediction and single-step retrosynthesis. Moreover, we propose an extension of speculative decoding to beam search, enabling the accelerated generation of multiple output sequences for a single input SMILES, which is necessary for multi-step synthesis planning. In our experiments, we demonstrate the acceleration of the Molecular Transformer by up to three times in reaction prediction and single-step retrosynthesis without changing the model architecture and training procedure, and without compromising accuracy.

3.1 Method

3.1.1 Speculative decoding

Autoregressive sequence models, such as variants of the Transformer, generate sequences token by token, and every prediction of the next token requires a forward pass of the model. Such a process may have high latency, especially for models with billions of parameters, which may be critical in certain use cases. Therefore, an in-

triguing question arises as to whether one could generate several tokens in a single forward pass of the model, thus completing the output more quickly. Speculative decoding [Xia et al., 2023; Leviathan et al., 2023; Hu et al., 2025] answers positively. Recently proposed as a method for accelerating inference in Large Language Models, it is based on the draft-and-verify idea. At every generation step, one can append some draft sequence to the sequence generated by the model so far and see if the model "accepts" the draft tokens (Fig. 3.1).

If the draft sequence has length N, in the best case, the model adds $N+1$ tokens to the generated sequence in one forward pass, and in the worst case, it adds one token, as in standard autoregressive generation. The acceptance rate for one generated sequence is the number of accepted draft tokens divided by the total number of tokens in the generated sequence. One can also test multiple draft sequences in one forward pass, taking advantage of parallelization, and choose the best one. Speculative decoding does not affect the content of the predicted sequence in any way compared to token-by-token decoding.

One can freely choose a way of generating draft sequences. For LLMs, one would usually use another smaller language model that performs its forward pass faster than the main LLM [Leviathan et al., 2023] or additional generation heads on top of the LLM's backbone [Cai et al., 2024]. However, one can also construct draft sequences without calling any learned functions. For example, generate random draft sequences, even though their acceptance rate will be minimal. Speculative decoding may imply the training of a separate model for drafting, but it does not require any changes to the architecture or training procedure of the main autoregressive model that generates the output.

3.1.2 Heuristic drafting strategy for SMILES

In reaction product prediction and single-step retrosynthesis posed as SMILES-to-SMILES conversion problems, the source and target sequences are often relatively similar. In a chemical reaction, large fragments of reactants typically remain unchanged, meaning that the SMILES of products and reactants share many common substrings. It is especially true if reactant and product SMILES are aligned to minimize the edit distance between them [Zhong et al., 2022]. This characteristic of chemical reaction data enables a convenient heuristic for speculative decoding. Specifically, one can extract multiple substrings of a chosen length D from the query SMILES and use them as draft sequences with a high acceptance rate. Figure 3.2 demonstrates our drafting method in product prediction. Before generating the target string, we can assemble a list of token subsequences from the source sequence (reactant tokens) using a sliding window of a desired length (4 in this example) and a stride of 1. Then, at every generation step,

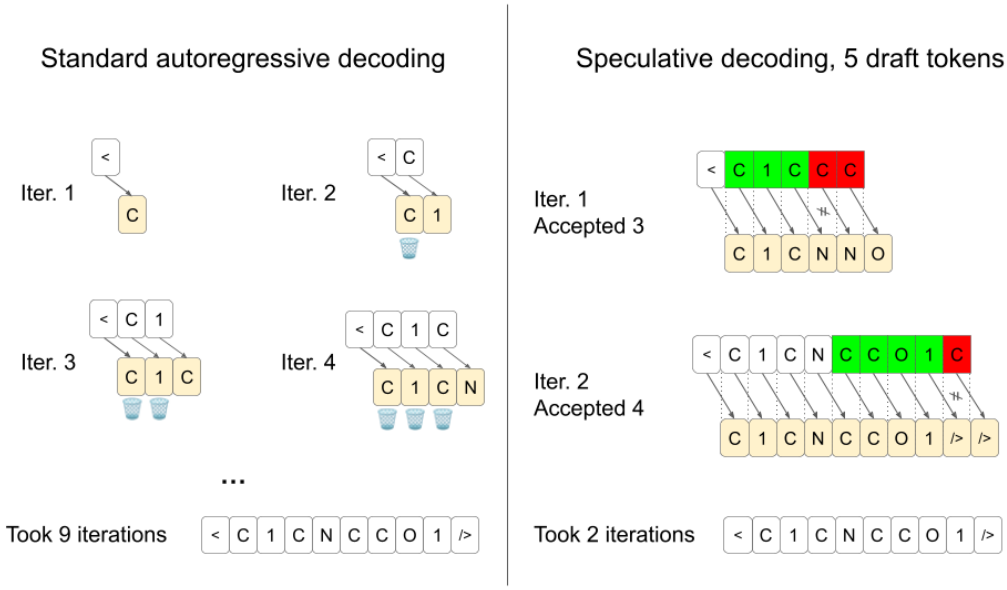

Figure 3.1. The principle of speculative decoding compared to the standard autoregressive decoding from a SMILES-generating transformer. The transformer is generating a string ″C1CNCCO1″. The start-of-sequence token is ″>″, and the end-of-sequence token is ″/>″. The white tiles are the tokens generated so far, the light yellow tiles are the most probable predicted tokens, the green tiles are the accepted draft tokens, and the red tiles are the rejected tokens. Here, the transformer predicts the next token for every token in the sequence (greedy sampling from the predicted distributions). In the case of standard decoding, the prediction for the last token is concatenated to the sequence, and the predictions for all other tokens are discarded. Speculative decoding takes advantage of the predictions for tokens other than the last one. A draft sequence is attached to the sequence that is already generated. If the first token of the draft sequence coincides with the token predicted for the last token so far, the draft token is accepted. The same is done for all draft tokens until the first mismatch. This method allows generating more than one token at a time.

we can try all draft sequences in one forward pass of the model to see if the model can copy up to 4 tokens from one of them. The draft token acceptance rate in this example reaches 78%. We can also make the drafting strategy ”smarter” by only considering the substrings of the query SMILES that start with the token currently concluding the generated output and stripping the first token in those substrings (Figure 3.3). This strategy would partially address the haunting problem of throughput-latency trade-off

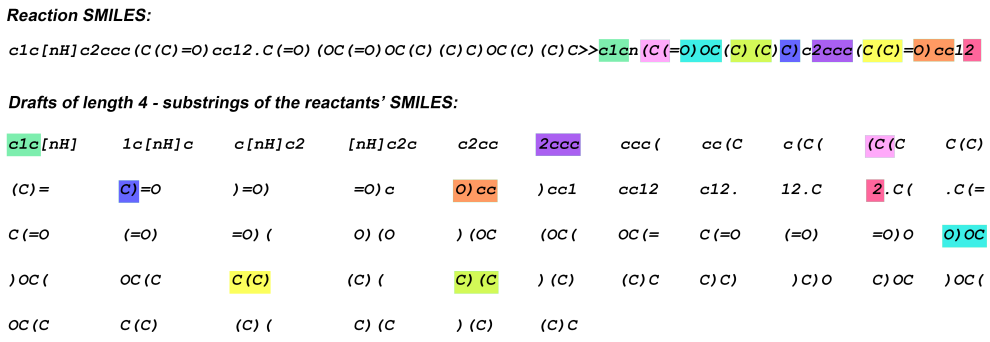

Figure 3.2. Speculative decoding accelerates product prediction with the Molecular Transformer or a similar autoregressive SMILES generator. Before generating an output sequence, we prepare a list of subsequences of a desired length, e.g., four, of the tokenized query SMILES of reactants. Then, at every generation step, the model can copy up to four tokens from one of the draft sequences to the output, thus generating from one to five tokens in one forward pass. Colors highlight the best draft at every decoding step.

(Section 3.3.1), and we use this strategy in our experiments as well (e.g., in Section 3.3.3).

3.1.3 Speculative Beam Search

Speculative decoding was initially formulated for the accelerated generation of a single sequence for a given query, e.g., with greedy decoding or nucleus sampling. However, generating one sequence per query has limited utility in the context of reaction modeling. In CASP, the single-step retrosynthesis model must suggest multiple different reactant sets for every query product, allowing the planning algorithm to choose from them. Additionally, in product prediction, a reaction may yield a mixture of products. Therefore, the real problem is to accelerate the generation of multiple SMILES for a single given SMILES. When using a SMILES-to-SMILES transformer, beam search is the standard choice for generating a set of outputs rather than a single one.

We found a way to accelerate beam search with speculative decoding. The main idea behind it is that at every decoding step, we use a draft sequence to generate multiple candidate sequences of different lengths in a single forward pass, which we then order by probability and retain the best K of them. We refer to our algorithm as "speculative beam search" (SBS). The detailed description of the speculative beam

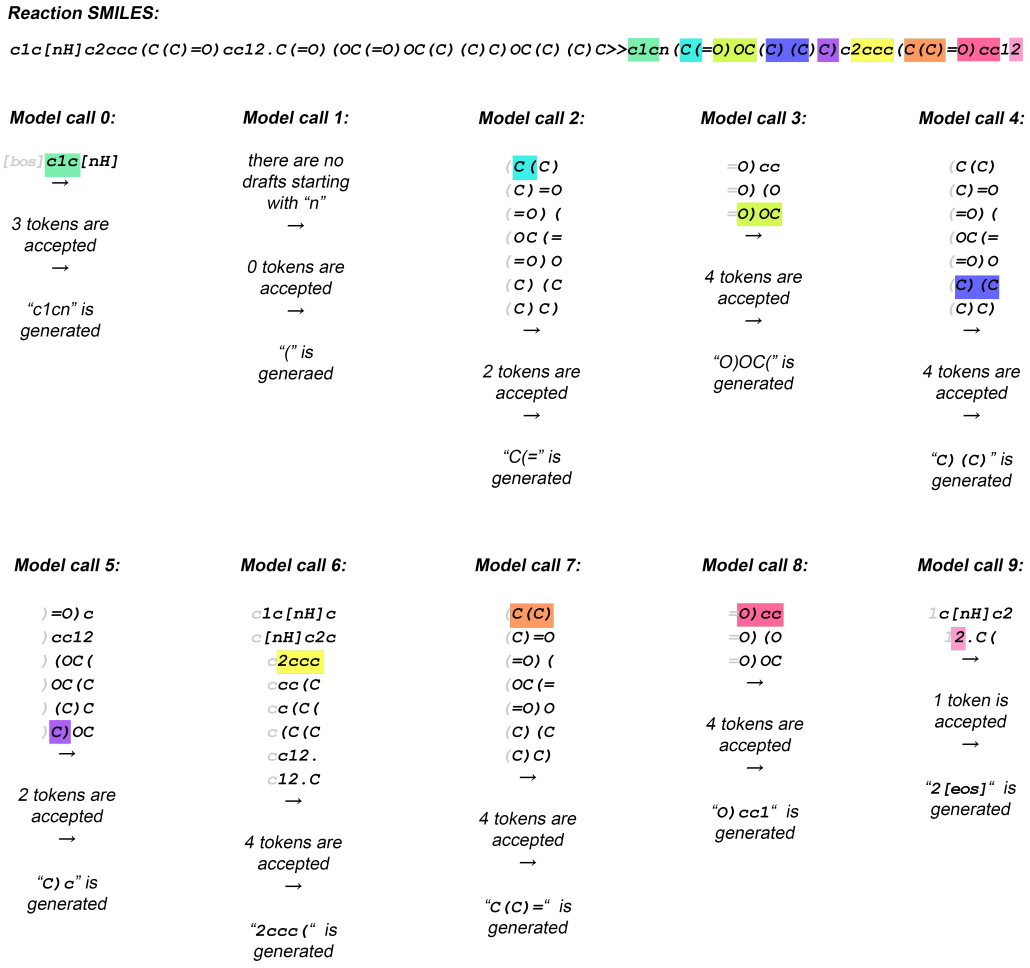

Figure 3.3. One improvement over the naive drafting strategy in Figure 3.2 is to only consider drafts that start with the token that currently concludes the generated output. For example, for the first call of the model we consider only those substrings of the source sequence that start with the [BOS] token, and there is only one such subsequence. We strip the first token (in gray) from the chosen subsequence and use the remainder as a draft. After two calls, we consider only the subsequences that start with ”(” and strip this token to finalize drafts, and so on.

search procedure we implement for our experiments is in Section A.1.

3.1.4 Model

We demonstrate our method using the Molecular Transformer (MT) architecture, which we re-implemented in Pytorch Lightning (see Section A.2). We replaced the standard decoding procedures for the MT with our methods based on speculative decoding and achieved a significant speed-up in product prediction and single-step retrosynthesis. We report our experiments conducted on a single NVIDIA Tesla V100 GPU with 32GB of memory.

3.1.5 Data

We employ the USPTO dataset (see Section 2.3.2) for training and testing our models. For product prediction, we use the USPTO MIT subset [Jin et al., 2017], which comprises approximately 410k reactions for training and 40k for testing. For single-step retrosynthesis, we use the USPTO 50K subset [Schneider et al.]. The test set of USPTO 50K consists of 5007 reactions, and the training set consists of approximately 40k reactions, which we turn into 800k reactions by 20-fold root-aligned SMILES augmentation [Zhong et al., 2022].

3.2 Related work

The research directly related to the present work remains scarce. Hartog et al. [Hartog et al., 2024] approached the acceleration of SMILES-to-SMILES transformers for single-step retrosynthesis with knowledge distillation [Schmidhuber, 1991; Schmidhuber, 1992a; Hinton et al., 2015] and thoroughly studied its effects. The authors demonstrated the acceleration of Chemformer by about two times at the cost of a small loss in accuracy. They also studied several different transformer architectures and the benefits of hyperparameter tuning. The work of Lin et al. [Lin et al., 2024] is related to our speculative beam search, even though it differs significantly. The main idea of their method, applied in the context of generative recommendation, is to perfectly guess all parallel sequences that an LLM would generate with beam search by using a smaller draft model.

Table 3.1. Wall time of the model's inference on the USPTO MIT test set in reaction product prediction with standard and speculative greedy decoding. "B" stands for batch size. The average time and the standard deviation are estimated based on five runs. The number of drafts and the draft length in the speculative greedy algorithm are different for every batch size to ensure the maximum speed-up by balancing the throughput-latency trade-off: 23 drafts of length 17 for B=1, 15 drafts of length 14 for B=4, 7 drafts of length 7 for B=16, 3 drafts of length 5 for B=32.

Decoding time, min	B=1	B=4	B=16	B=32
Greedy	190 ± 36	82 ± 12	28.7 ± 2.3	16.8 ± 1.8
Speculative Greedy	53.9 ± 3.6	28.8 ± 3.2	18.2 ± 1.0	14.2 ± 0.3

3.3 Experiments

3.3.1 Throughput-latency trade-off

Before discussing the application of speculative decoding to SMILES generation, we should emphasize the existence of the *throughput-latency trade-off* in the transformer. When processing the test dataset, one can increase the batch size to improve throughput, as the processing will require fewer model calls. A larger batch size typically reduces the time needed to process the entire dataset. However, this improvement comes at the cost of increased latency per batch. The forward pass becomes slower as larger batches demand more computational resources and memory. At a certain point, the increase in latency may outweigh the benefits of higher throughput, ultimately leading to slower overall decoding. Simultaneously improving latency and throughput is challenging, and one typically must carefully balance the two.

3.3.2 Product prediction

We tested our MT with speculative decoding for product prediction on the USPTO MIT mixed dataset, i.e., without an explicit separation between reactants and reagents. The test dataset in this benchmark comprises 40 thousand reactions.

Greedy decoding acceleration

When serving a transformer as an AI assistant that predicts reaction products, one could start by using greedy decoding for inference. Table 3.1 summarizes our experiments with greedy decoding from MT on the test set of USPTO MIT. On the V100 GPU, our

Table 3.2. Wall time of the model's inference on the USPTO MIT test set in reaction product prediction with beam search (BS) and speculative beam search (SBS). The number of generated sequences (denoted by "K") is 5. "B" stands for batch size. The number of drafts and the draft length in the speculative beam search is different for every batch size. The average time and the standard deviation are estimated based on five runs.

Decoding time, min	B=1	B=2	B=3	B=4
Beam search (K=5)	298 ± 20	166 ± 32	152 ± 19	88 ± 14
SBS (K=5)	140.4 ± 7.2	109.1 ± 3.6	93.3 ± 1.3	82.1 ± 0.6

greedy speculative decoding is faster than the greedy decoding by 3.53, 2.84, 1.57, and 1.18 times on average at batch sizes 1, 4, 16, and 32, respectively. Here, the throughput-latency trade-off comes into play. With our heuristic drafting, we achieve a high acceptance rate by simultaneously evaluating multiple drafts for each sequence in the batch. However, it can significantly inflate the effective batch size of the model's input. As a result, the latency of the forward pass will worsen, reversing all benefits of speculative decoding at larger batch sizes. We find the balance between throughput and latency by doing a grid search over the number of drafts and draft length on a fraction (500 reactions) of the test dataset (Fig. A.2 in Appendix A.3). For batch size 1, 23 drafts of 17 tokens give the best speed-up to greedy decoding. For batch size 4, 15 drafts of 14 tokens for every sequence work best. For batch size 16, it is 7 drafts of 7 tokens. For batch size 32, it is 3 drafts of 5 tokens (Fig. A.2A). The results in Table 3.1 are given for the corresponding drafting parameters. At even larger batch sizes, the effect of speculative greedy decoding reverses, and it becomes slower than standard greedy decoding.

Even accelerating greedy decoding inference with the batch size of 1 would be sufficient for an improved user experience with reaction prediction assistants: chemists would usually enter one query at a time in a user interface of a reaction model like IBM RXN.

The model's accuracy is 88.3% with greedy decoding, and it is not affected when switching to speculative greedy decoding.

Beam search acceleration

We compare the inference time of standard beam search with beam size 5 and our speculative beam search in product prediction (Table 3.2). Standard beam search com-

Table 3.3. Wall time of the model's inference on the USPTO 50K test set (5K reactions) in single-step retrosynthesis with beam search and speculative beam search (SBS). Batch size is 1. "B" stands for batch size, and "K" stands for the number of generated sequences (beam size for beam search). The number of drafts and the draft length in the speculative beam search is different for every K. The average time and the standard deviation are estimated based on five runs.

Decoding time, min	K=5	K=10	K=15	K=20
Beam search (B=1)	67 ± 11	70 ± 13	71 ± 14	72 ± 16
SBS (B=1)	28.9 ± 2.9	39.2 ± 3.4	45.5 ± 1.2	47.2 ± 1.2

pletes in around 298 minutes, 167 minutes, 152 minutes, and 88 minutes on average at batch sizes 1, 2, 3, and 4, respectively. SBS works consistently faster, and gives a speed-up of around 2.12, 1.53, 1.63, and 1.07 times. The speed-up becomes smaller as the batch size increases due to the increased latency of a single forward pass of the model. Before comparing the methods, we determined the optimal drafting parameters for SBS using a grid search as in the greedy speculative decoding experiments. In Table 3.2, SBS uses 23 drafts of 10 tokens for batch size 1, 10 drafts of 14 tokens for batch size 2, 10 drafts of 9 tokens for batch size 3, and 7 drafts of 10 tokens for batch size 4 (Fig. A.2B). The optimal number of simultaneous drafts decreases to combat the increasing forward pass latency as the batch size grows.

While speculative greedy decoding guarantees that the generated outputs will be the same as the outputs generated by standard greedy decoding, our SBS does not guarantee the same outputs as beam search. However, in practice the difference is negligible. The top-1 accuracy of SBS in product prediction is 88.4 %, the top-3 accuracy is 93.7 %, and the top-5 accuracy is 94.7 % (Table 3.5A). There is practically no difference in accuracy compared to standard beam search in product prediction (Table 3.5A): the accuracy percentage differs in the second decimal place in the Top-5 case but not in Top-1 or Top-3. The number of invalid predicted SMILES also differs insignificantly for the fifth prediction, not the first or the third.

3.3.3 Single-step retrosynthesis

We carried out single-step retrosynthesis experiments on USPTO 50K, in which the training dataset was augmented 20 times. The augmentation procedure is to construct alternative root-aligned [Zhong et al., 2022] SMILES for every dataset entry. This augmentation minimizes the edit distance between reactants and products, which simpli-

Table 3.4. Wall time of the model's inference on the USPTO 50K test set (5K reactions) in single-step retrosynthesis with beam search and speculative beam search (SBS). "B" stands for batch size, and "K" stands for the number of generated sequences (beam size for beam search). Here, K is equal to 10. The number of drafts and the draft length in the speculative beam search are different for every B. The average time and the standard deviation are estimated based on five runs.

Decoding time, min	B=1	B=2	B=4	B=8
Beam search (K=10)	70 ± 13	42.1 ± 6.7	26.9 ± 3.5	18.7 ± 1.2
SBS (K=10)	39.2 ± 3.4	30.1 ± 0.8	28.1 ± 0.3	20.1 ± 0.2
SBS (K=10, smart drafts)	22.7 ± 1.3	16.6 ± 0.4	14.8 ± 0.1	13.4 ± 0.1

fies training, pushes the model's accuracy higher, and increases the acceptance rate in our speculative decoding method. We did not augment the test set; it comprises 5007 reactions.

Beam search at different beam widths

Greedy decoding is less relevant to single-step retrosynthesis than to product prediction. Therefore, we describe only the acceleration of beam search with speculative decoding.

The wall time our retrosynthesis model takes to process the USPTO 50K test set with beam search and batch size 1 is around 67 minutes, 70 minutes, 71 minutes, and 72 minutes when generating 5, 10, 15, and 20 predictions for every query SMILES, respectively (Table 3.3), although we notice the spread of the wall time to be above 10 minutes in all cases. The average wall time increases as expected with the number of maintained hypotheses as the effective batch size becomes larger. When we replace the standard beam search with our SBS at the same batch size and the number of generated sequences, the generation finishes in around 29, 39, 46, and 47 minutes, accelerating the inference by around 2.3, 1.8, 1.56, and 1.53 times, respectively. Like in the case of greedy speculative decoding, we run a greed search over the number of drafts and the number of tokens in drafts on 500 reactions from the test set to find the optimal combinations of parameters that give the best acceleration. SBS with B=1 and K=5 uses 15 drafts of length 11, SBS with B=1 and K=10 uses 10 drafts of length 10, with B=1 and K=15 it also uses 10 drafts of length 10, and with B=1 and K=20 it uses 5 drafts of length 14 (Fig. A.2C). To summarize, our SBS works consistently faster than beam search at batch size fixed to one, although the effect becomes less pronounced

as the number of outputs per sequence grows.

Beam search at different batch sizes

We also check the behaviour of beam search and SBS at varying batch size and a fixed number of hypotheses per sequence. We keep the latter to be equal to 10, as it is a realistic setting for synthesis planning which also does not inflate the effective batch size too much. As Table 3.4 shows, the standard beam search finishes in 70, 42, 27, and 19 minutes on average with batch size 1, 2, 4, and 8, respectively. SBS finishes faster: in 39, 30, 28, and 20 minutes, respectively. The speed-up is 1.78 with batch size 1 and 1.4 with batch size 2, and with batch sizes 4 and 8 SBS becomes slightly slower than beam search. The number of drafts and draft length is again optimized beforehand with the help of a grid search. To further combat the latency increase overwhelming the throuthput benefits, we tried changing the drafting strategy. In the variant of SBS, which we here call "SBS with smart drafts", we first select only the drafts that start with the same token as the last token generated by the model so far, and then strip the first draft token (Figure 3.3). This strategy allows to reduce the number of drafts tried at the same time and reduce the effective batch size while maintaining high acceptance rate. The smart drafting allows SBS to consistently outperform the speed of beam search on batch sizes from 1 to 8 (Table 3.4), improving it by 3.1, 2.5, 1.8, and 1.4 times, respectively.

The accuracy when using SBS is the same as with beam search in top-1 and top-5; in top-10, it differs only in the decimals of a percent point (Table 3.5B). The proportion of invalid predictions also changes insignificantly. Thus, our SBS accelerates the MT's generation of multiple reactant sets several times without having to compromise on accuracy. Such a speed-up could make the autoregressive transformer a more attractive basis for single-step models in multi-step synthesis planning.

3.3.4 Limitations

The benefits of speculative decoding in accelerating greedy decoding and beam search decoding of SMILES with our drafting methods become less manifested with the increase of the batch size or the number of generated outputs per sequence. Since our methods rely on applying multiple drafts in parallel to every sequence being generated in the batch at every step, the effective batch size of the model's input inflates from BK to BKN, where B is the batch size, K is the number of generated outputs per sequence, and N is the number of drafts. This inflation leads to the increase in latency of the model's forward pass, which may cancel the benefits of reducing the number of model calls when BKN is large enough. Nonetheless, we are confident that the

Table 3.5. The top-N accuracy of our model and the proportion of invalid SMILES in the N-th prediction with different decoding strategies in product prediction on USPTO MIT (A) and in single-step retrosynthesis on USPTO 50K (B). The difference in accuracy between standard and speculative beam search is negligible.

(A) Product prediction		Top-1, %	Top-3, %	Top-5, %
Accuracy	Beam search	88.425	93.690	94.733
	SBS	88.425	93.690	94.720
		Pred. 1, %	Pred. 3, %	Pred. 5, %
Invalid SMILES	Beam search	0.232	8.270	13.182
	SBS	0.232	8.275	13.285

(B) SSR		Top-1, %	Top-5, %	Top-10, %
Accuracy	Beam search	52.077	82.069	88.918
	SBS	52.077	82.069	89.038
	SBS (smart drafts)	52.077	82.069	88.978
		Pred. 1, %	Pred. 5, %	Pred. 10, %
Invalid SMILES	Beam search	0.799	3.534	8.107
	SBS	0.799	3.534	8.007
	SBS (smart drafts)	0.799	3.534	8.187

proposed speculative decoding methods are promising for building transformer-based CASP systems and reaction prediction assistants. The acceleration will likely extend into larger B and K with better drafting strategies that use fewer drafts (ideally one) with a high acceptance rate, and with other inference acceleration techniques used in conjunction with speculative decoding, such as KV-caching or quantization.

3.4 Conclusion

We combine speculative decoding and chemical insights to accelerate inference in the Molecular Transformer, a SMILES-to-SMILES translation model. Our methods make processing the test set up to three times faster in both single-step retrosynthesis on USPTO 50K and reaction product prediction on USPTO MIT compared to the standard

decoding procedures (greedy decoding and beam search). We propose a novel Speculative Beam Search method to accelerate the generation of multiple output SMILES per query. Our method aims at making state-of-the-art template-free SMILES-generation-based models, such as the Molecular Transformer, more suitable for industrial applications such as computer-aided synthesis planning systems.

Chapter 4

Faster multi-step synthesis planning

The acceleration of inference in single-step retrosynthesis models is a necessary step for the development of modern and future machine learning-based CASP systems. In Chapter 3, we presented our method of accelerating a SMILES-to-SMILES transformer for single-step retrosynthesis using speculative beam search. However, the limitations of the described heuristic drafting scheme motivated us to seek alternatives with better scalability with respect to batch size. Moreover, the acceleration of the single-step retrosynthesis transformer required validation in the multi-step synthesis planning setting. Drawing inspiration from LLM research, we replaced heuristically generated draft sequences in the speculative decoding of SMILES with predicted draft sequences, following the "Medusa" method of speculative decoding [Cai et al., 2024]. We then studied how the acceleration of the single-step transformer with speculative decoding transfers to multistep synthesis planning with AiZynthFinder.

Our main contributions: We demonstrate the acceleration of multi-step retrosynthesis that relies on a SMILES-to-SMILES transformer. Replacing standard beam search with our approach increases solvability in the CASP when planning is constrained by time, and saves hours of computation when synthesis is constrained by the number of iterations. Our method brings AI-based CASP systems closer to meeting the stringent latency requirements of high-throughput screening for synthesizability and improves the overall user experience.

This chapter is based on our manuscript "Fast and scalable retrosynthetic planning with a transformer neural network and speculative beam search" [Andronov et al., 2025b]

4.1 Method

4.1.1 Medusa

SMILES-to-SMILES generation is remarkably well compatible with speculative decoding. In chemical reactions, only some of the reactant atoms typically change their connectivity, whereas large fragments of the reactants remain unchanged and appear identical in the products. Therefore, instead of constructing the target SMILES token by token, the transformer can quickly assemble it from fragments of the query SMILES if they are presented as draft sequences. While such a heuristic drafting strategy in combination with SBS achieves strong inference acceleration in reaction prediction and single-step retrosynthesis (see Chapter 3), heuristic drafting presents a scalability problem. To achieve a high acceptance rate, multiple drafts should be used in parallel. It increases the effective batch size as $O(BKN)$, where B is the basic batch size, K is the number of beams, and N is the number of drafts. The latency of a forward pass of the model increases with batch size, and may quickly outweigh the benefits of the high throughput given by batching.

A recently proposed method, "Medusa" [Cai et al., 2024], offers a simple solution for generating single drafts with a high acceptance rate. The fundamental idea of the method is to add extra subnetworks (decoding heads) to the transformer neural network that predict multiple tokens ahead of the next token in parallel. Instead of the usual transformer logits output of shape (B, L, V), a Medusa model gives (B, L, M, V), where B is the input batch size, L is the decoder input length, V is the vocabulary size, and M is the number of Medusa model heads. While the main prediction head generates the next token as usual, the additional Medusa heads predict the second next token, the third next token, and so on up to the M-th next token.

The tokens predicted by the additional heads are the draft sequences for the main head to verify. The first Medusa call is used to generate a draft. In our experiments, the model has 20 heads, so the draft length is 20. We use greedy decoding to create only one draft per given input sequence to avoid inflating the effective batch size. The second Medusa call uses only the main head's output data to verify draft tokens. At least one draft token will always be approved (since it was generated by the main model head in greedy mode), so the worst case is 2 tokens generated across 2 model calls. Of course, the worst-case scenario is still undesirable, as additional heads necessitate additional weights in the model architecture, which may result in a slightly slower forward pass. In our architecture, adding extra heads increased the number of weights by 7.5%. Thus, a high acceptance rate for draft tokens is important. In the best case, the Medusa model with 20 heads (1 main head and 19 extra heads) produces 21 tokens in 2 model calls.

As a verification procedure, we use one similar to top-p sampling. We sort the predicted probabilities for every token in the vocabulary in descending order and calculate the cumulative probabilities for every token. If the cumulative probability for a given draft token is less than the nucleus parameter (99.75%), then that token is considered probable enough and approved. Additionally, the highest probability token among all vocabulary tokens is always approved.

To train the model, we select the ”joint training, combined loss” recipe from the original Medusa paper [Cai et al., 2024]. To give priority to the accuracy of the main head, we divide each head's contribution to the loss function by the number of that head.

When we replace the heuristic drafting in the Molecular Transformer with the Medusa approach, we observe a significant improvement in SBS scalability at larger batch sizes, expanding the potential of speculative decoding and transformer models for fast synthesis planning.

4.1.2 Multi-step synthesis planning

Since the ultimate goal of building accelerated template-free SMILES-to-SMILES prediction models is to enable fast AI-powered CASP, we evaluate our single-step retrosynthesis model as a component of a multi-step synthesis planning system. We choose AiZynthFinder 4.0 [Genheden et al., 2020; Saigiridharan et al., 2024] because of its straightforward support for arbitrary template-free models and to maintain continuity with prior work that benchmarked various single-step retrosynthesis models as the components of AiZynthFinder [Torren-Peraire et al., 2024]. We mostly focus on retro* [Chen et al., 2020] as the search algorithm for building the synthesis tree. We use only the reactant probability from the single-step model as the guiding probability in tree search, as done previously [Torren-Peraire et al., 2024]. AiZynthFinder allows constraining the planning by both total time and the maximum number of iterations, and these must be set simultaneously. Therefore, those constraints should be chosen carefully, because ultimately, the strictest of the two is the limiting constraint. In our experiments, we used either a time limit (setting a very large maximum number of iterations) or an iteration limit (setting a very large maximum time). This facilitated the analysis of the experiments. AiZynthFinder also features two search modes: either the search stops when at least one route is found, or it continues until the iteration or time limit is reached. When planning is set to build multiple routes until the time limit is reached, the acceleration of the single-step model does not save time; instead, it enables a more thorough exploration of the route space.

It is important to note that AiZynthFinder runs all single-step expansions with a batch size of 1 by design. Therefore, in our multi-step synthesis experiments, the

comparison between single-step models is limited to performance at a batch size of 1. Currently, all ML-based open source CASP systems (AiZynthFinder [Genheden et al., 2020], SynPlanner [Akhmetshin et al., 2025], Syntheseus [Maziarz et al., 2025], and ASKCOS [Tu et al., 2025]) support single-step expansions with only a batch size of 1.

4.1.3 Model

We train a basic and a custom version of the Molecular Transformer [Schwaller et al., 2019] for single-step retrosynthesis. See Section B.1 for details.

4.1.4 Data

For training the single-step retrosynthesis model and its isolated evaluation, we employ the standard USPTO 50K dataset. We apply the 20-fold R-SMILES augmentation [Zhong et al., 2022] to the USPTO 50K training subset. The test set comprises 5007 reactions; we do not augment it. We follow the standard atomwise tokenization procedure [Schwaller et al., 2019] to tokenize SMILES.

For multi-step synthesis planning experiments, we used two sets of building blocks: PaRoutes and ZINC. PaRoutes stock contains 13,414 molecules designated as purchasable in the PaRoutes-n1 dataset [Genheden and Bjerrum, 2022; Torren-Peraire et al., 2024]. ZINC stock contains 17,422,831 molecules.

4.2 Related work

The work related to the acceleration of CASP is not ample and is mentioned in Chapter 3 in Section 3.2. A comprehensive review [Xia et al., 2024] discusses the applications of various speculative decoding approaches in LLMs, including Medusa.

4.3 Experiments

4.3.1 Single-step retrosynthesis

We first test our models in a single-step retrosynthesis setting using the USPTO 50K dataset. We compare the base transformer with beam search decoding, the base transformer with speculative beam search decoding in a ”smart” heuristic drafting variant (Chapter 3), and the Medusa variant of the transformer with speculative beam search that relies on draft tokens predicted by the Medusa heads. We denote the first model

as TBS (Transformer + Beam Search), the second as THSBS (Transformer + Heuristic Speculative Beam Search), and the latter as MSBS (Medusa + Speculative Beam Search). We also include a separate ”optimized” beam search that does not call the model to predict the pad token after the EOS token. This optimization of the beam search does not affect the accuracy or the number of model calls. However, it reduces the effective batch size, thereby reducing calculation time at larger batch sizes. Since THSBS and MSBS do not call their models to produce the pad token after the EOS token, we outline TBSO (Transformer + Beam Search (Optimized)) as a separate position to ensure a more accurate comparison.

As Table 4.1A shows, MSBS significantly outperforms TBS and THSBS at various batch sizes in terms of inference speed. THSBS outperforms TBS and TBSO at smaller batch sizes but suffers from scalability limitations. Due to the *throughput-latency trade-off* inherent to processing multiple draft sequences simultaneously, the heuristic drafting scheme requires careful tuning of draft number and length to achieve optimal performance. At larger batch sizes, the computational overhead of processing multiple drafts negates the acceleration benefits, and the optimal number of drafts becomes 1, making THSBS similar to MSBS, as it also uses only one draft. At the same time, MSBS achieves a higher acceptance rate (Table 4.1D) through its integrated architecture, maintaining consistent acceleration even at batch size 32, which establishes MSBS as the superior acceleration approach for single-step retrosynthesis with transformers. MSBS requires fewer forward passes of the model to complete generation (Table 4.1B) and achieves an acceptance rate of 91%, leaving THSBS far behind.

In terms of accuracy and prediction validity, all three methods demonstrate nearly identical performance (Table 4.2). While our speculative beam search approach does not guarantee output distributions identical to those of standard beam search, the practical differences are negligible. A slightly larger difference in accuracy and SMILES validity between MSBS and THSBS stems from marginal performance differences across model checkpoints rather than algorithmic effects. MSBS implies a custom transformer architecture and requires training a separate model, whereas THSBS is a drop-in replacement for beam search.

4.3.2 Multi-step retrosynthesis

We incorporated our single-step retrosynthesis models into AiZynthFinder to test their performance in multi-step synthesis planning. Since MSBS consistently outperforms THSBS in terms of the speed of single-step prediction generation, we decided to compare MSBS only with the standard Transformer + Optimized Beam Search (TBSO) in multi-step synthesis planning. We will further refer to MSBS as ”Medusa” and to TBSO as ”Transformer”. We selected Caspyrus10k as the benchmark dataset for multi-step

synthesis planning, drawing inspiration from the prior experiments [Torren-Peraire et al. 2024].

Synthesis planning constrained by time

To compare Transformer and Medusa in synthesis planning on Caspyrus10k, we adjusted the original methodology of Torren-Peraire et al. to prioritize inference speed, assuming that chemists would not wait for hours for computation to complete, and constrained the multi-step synthesis to either solve a query molecule within several seconds or consider it unsolved. Solving a molecule means building a synthesis tree in which all leaf nodes are building blocks - molecules that are assumed to be available. Here, we use the PaRoutes stock (13,414 molecules) or the ZINC stock (17,422,831 molecules). We set the models to generate 10 candidate precursor sets with every call of a single-step model, constrained the maximum route length to 5 or 7, and the maximum number of algorithm iterations to 35,000 (enough to ensure the priority of the time constraint). When the algorithm identifies the first route from a query molecule to the building blocks, it stops, and the molecule is considered solved. Table 4.3 summarizes the results of our multi-step retrosynthesis experiments under the time constraints of 5 and 15 seconds for solving a molecule. The results reveal that Medusa consistently outperforms Transformer across all experimental conditions, with improvements in both the number of solved molecules and computational efficiency.

The iteration of the planning algorithm (Retro* or DFPN search) is accelerated with Medusa by approximately 3 times, which results from the acceleration of the single-step model by approximately 4 times (Table 4.1A). Due to the different number of the planning algorithm iterations, the first route is identified by Medusa approximately 2 times faster. It leads to the increase of success rate under the stringent time limit. When using DFPN search with a 5-second limit, Medusa solves 2080 molecules out of 10000, which is 86% more compared to the 1117 solved by Transformer. For the 1017 molecules that both methods successfully solve, Medusa requires on average less than half the time (0.86 seconds vs 1.88 seconds) (Table 4.3A). With the Retro* algorithm, Medusa maintains its advantage, solving 36% more molecules than Transformer within 5 seconds (5287 vs. 3890, Table 4.3B) and 26% more within 15 seconds (6715 vs. 5341, Table 4.3C). When using the ZINC stock with a depth limit of 5 and a 15-second time limit (Table 4.3D), Medusa solves 8343 molecules compared to 7708 for the Transformer, while also reducing the average solution time (1.73 s vs 3.21 s). Finally, with ZINC and the depth limit increased to 7 (Table 4.3E), Medusa solves 8608 molecules versus 7888 for the Transformer and maintains a lower average solution time (1.89 s vs 3.37 s). Across all conditions, Medusa consistently achieves faster average solution times while solving substantially more molecules.

Interestingly, Medusa required more algorithm iterations per commonly solved molecule than Transformer. Similarly, the number of commonly solved molecules under different parameter settings (Table 4.3) never coincides with the Transformer's number of solved molecules. The likely reason for it is a slight difference in probability distributions generated by Transformer and Medusa (see Appendix B.2).

Synthesis planning constrained by iteration number

To compare our model's performance within AiZynthFinder's planning system with the single-step models compared by Torren-Peraire et al., namely LocalRetro [Chen and Jung, 2021], Chemformer [Irwin et al., 2022] and the default template-based single-step model available in AiZynthFinder, we set the same parameters for the multi-step retrosynthesis search: the maximum route depth of 7, the beam size of 50, ZINC stock (consisting of 17,422,831 molecules) of building blocks, retro* as a search algorithm, and the maximum number of algorithm iterations per molecule of 200. Since we conducted our experiments on a GPU (Tesla V100 32 GB), we set the time limit to 3 minutes instead of the original 480 minutes, which is enough for Transformer and Medusa to be limited only by the iteration limit, and not the time limit, in the case of a complex input molecule. We also use the setting in which the search is stopped as soon as at least one route for a given molecule is found. It does not affect the success rate (the number of molecules for which a full synthesis route is found), but it speeds up experiments by avoiding the calculation of all 200 retro* iterations. According to Table 4.5, Transformer, a model of 17 million parameters, is sufficient to achieve a high percentage of successfully solved molecules, approaching 100%. With the limit of 200 iterations, Transformer reaches 97.13% of solved molecules, spending 0.67 seconds per retro* iteration, while Medusa achieves 95.90% of solved molecules, spending 2 times less, only 0.31 seconds per iteration. This means that if Transformer ran for all 200 iterations, as in the experiments of Torren-Peraire et al., it would spend about 134 seconds per molecule and about 372 hours on the entire dataset. Meanwhile, for Medusa, these values would be 62 seconds and 172 hours, respectively. Thus, Medusa would save approximately 200 hours of GPU calculations while maintaining a 95.9% solved-molecule rate.

We compared the multistep success rate of the models with their round-trip accuracy, diversity, and percentage of invalid SMILES (Table 4.4). Instead of the relative round-trip accuracy, we calculated the number of round-trip feasible reactions, i.e., predicted reactions that lead to the initial target molecule when evaluated with the Molecular Transformer forward model in top-10 mode. Diversity is measured as the number of unique reaction names, assigned by Rxn-INSIGHT [Dobbelaere et al., 2024]. For comparison, we took the default template-based single-step model from

AiZynthFinder 4.0. Despite the template-based model (42554 templates in total) offering 50 templates at a time, some templates cannot be applied, some predictions are filtered out by the default neural filter, and duplicate predictions also occur. As a result, the model, on average, offers only 15 reactions per target molecule during testing. The model runs on a CPU. We assigned it a longer time limit of 5 hours per molecule to ensure the programs' running times would be comparable. It increased the success rate from 72.6% to 82.6%.

According to all the single-step metrics, Transformer and Medusa are very similar. As we trained our models on data without reagents, they achieved relatively low round-trip accuracies of 62.8% and 63.0%, respectively (considering beam size 10 mode). These numbers are close to the template-based model's round-trip accuracy of 62.3%. One can see that in the 3 minutes per molecule time constraint, Medusa speeds up the program by 1.4x. In the iteration constraint, the 4.4x acceleration of the single-step model and, as a result, faster multi-step iteration, leads to 1.7x and 2.9x acceleration of the entire program.

PaRoutes-n1 evaluation

To evaluate the multi-step retrosynthesis performance of our models, we used the PaRoutes-n1 [Genheden and Bjerrum, 2022] benchmark, which is designed to assess the ability of retrosynthesis planners to recover predefined reference routes given a fixed stock of building blocks.

The PaRoutes-n1 dataset contains 10,000 target molecules, each associated with a single reference synthesis route and a predefined stock of building blocks. From this dataset, we randomly selected 1,000 target molecules for evaluation. The search configuration provided with the PaRoutes benchmark was used without modification. Both models successfully completed the full 500-iteration search procedure for all evaluated targets.

The results of this experiment are summarized in Table 4.6. During the search, we collected up to 50 solved synthesis routes per target molecule. Both models demonstrated a high overall solvability, defined as the fraction of targets for which at least one valid synthesis route was found. Medusa solved 937 molecules (93.7%), while the Transformer solved 941 molecules (94.1%). To assess the models' ability to reproduce the reference routes provided in PaRoutes, we evaluated route accuracy, defined as the fraction of targets for which the exact reference route appears among the top-50 predicted synthesis routes. Medusa successfully recovered the reference route for 276 molecules (27.6%) and completed the evaluation in 10 hours. The Transformer recovered the reference route for 286 molecules (28.6%) and required 24 hours of computation. Again, Medusa maintains the Transformer's performance while being

significantly faster.

The sets of molecules solved by both Transformer and Medusa do not fully overlap. Among the molecules for which the reference route was recovered, 248 targets were common to both models. Therefore, Transformer solved 38 molecules that Medusa did not solve, while Medusa solved 28 molecules that Transformer did not solve. This observation is consistent with our earlier results (e.g., Table 4.3) and indicates that the two models explore different regions of the retrosynthetic solution space, despite achieving similar overall performance. For the 38 molecules solved only by the Transformer, Medusa generated routes with exactly matching reference building block sets for 22 molecules, indicating that it was often close to recovering the correct route topology. Conversely, among the 28 molecules solved only by Medusa, the Transformer produced routes with matching building block sets for 8 molecules.

These observations suggest that even when the exact reference route is not recovered, the models frequently identify chemically consistent precursor sets, reflecting partial convergence toward the reference synthesis strategy.

4.3.3 Limitations

Similar to the paper by Torren-Peraire et al. [Torren-Peraire et al., 2024], we focus here solely on the speed of building the retrosynthetic graph. However, Aizynthfinder also spends a significant amount of time splitting the tree into separate routes. The more model calls are made within the time limit, the more nodes are added to the retrosynthesis tree, and the more complex the tree becomes for splitting. We alleviated this problem by allowing AiZynthFinder to extract only the successful routes in which all leaves are building blocks. This approach requires only a small fraction of the time required for the exhaustive process of extracting all retrosynthetic routes in AiZynthFinder. In cases where unsolved routes are required, the tree-splitting algorithm can be easily optimized by first extracting the most probable reactions rather than making a random choice of unsolved reactions, and by decreasing the maximum route number limit.

Table 4.4 shows that the round-trip accuracy of Transformer rapidly decreases from 63% ($\frac{5.8}{9.2}$) to 47% ($\frac{18.5}{39.6}$) when going from top-10 to top-50, although the success rate is noticeably improved from 90% to 97% (with Medusa showing similar behavior). We attribute this to the fact that the quality of the beams decreases with an increase in their number, and the template-free models sometimes return short answers, i.e., SMILES that are valid but meaningless, for instance, ”Cl” as a single precursor for ”ClCc1csc(-c2ccccc2)n1”. At the same time, the metrics show that after the tenth beam, valuable predictions are generated (the absolute number of round-trip feasible reactions is larger in the top-50 than in the top-10), but they are mixed with low-

quality predictions that need to be filtered out. Although it is easy to filter out invalid reactions or those that contain the product itself among the reactants, short answers are more difficult to filter out. The default neural filter of AizynthFinder 4.0 defines "Cl >> ClCc1csc(-c2ccccc2)n1" as a feasible reaction with 99.95% probability. The round-trip filter successfully recognizes this case, but it is heavy and also may erroneously filter out reactions written with missing reagents. Generating reactions with reagents would, in turn, reduce the diversity of autoregressively generated reactions and, consequently, the diversity of routes. Thus, it is necessary to find an elusive balance between all these factors. It may be useful to combine heuristic filters of nonsensical reactions with transformer models in order to avoid formally solved, but in fact incomplete routes.

Currently, AiZynthFinder and its alternatives do not readily support batch sizes other than one for single-step retrosynthesis models in multi-step synthesis planning. Our future work will focus on generalizing multi-step synthesis planning algorithms to support larger batch sizes, which should further reduce the latency in CASP systems, considering the signigicant speed benefits of Medusa at larger batch sizes.

4.4 Conclusion

We demonstrate that a Medusa variant of the Transformer combined with speculative beam search offers significant speed advantage in multi-step synthesis planning in AiZynthFinder compared to the usual combination of SMILES-to-SMILES Transformer and beam search while retaining the same level of accuracy. In isolated single-step retrosynthesis experiments, Medusa and SBS accellerate the generation of 10 expansions per query by approximately 4 times across batch sizes from 1 to 32 without compromising the accuracy of the baseline Transformer. When Transformer is compared against Medusa in multi-step synthesis planning in AiZynthFinder under stringent time constraints, Medusa allows a single iteration of the planning algorithm to run approximately 3 times faster across varying time limits, search algorithms, and building blocks, enabling AiZynthFinder to solve significantly more molecules under the same time constraints compared to the Transformer. When the multi-step search is constrained by a maximum number of iterations instead of maximum time, the search with Medusa completes much faster than with Transformer, saving dozens of hours of compute while demonstrating negligible discrepancy in solvability and route accuracy. Our approach presents a promising alternative for SMILES-to-SMILES transformers in synthesis planning when planning speed is critical and brings the speed of AI-based CASP closer to the level required for high-throughput synthesizability screening in pharmaceutical research.

Table 4.1. Comparison between the inference algorithms for the single-step retrosynthesis model on the USPTO 50K test set (5K reactions). TBS is the transformer with beam search, and TBSO is the transformer with the optimized beam search. "Optimized beam search" means that the finished sequences in a batch are put aside, and the transformer is not called to generate pad tokens after the EOS token (it reduces the average effective batch size). THSBS is the transformer with speculative beam search relying on heuristic drafting; MSBS is the Medusa model with speculative beam search. "B" stands for batch size. The number of drafts and the draft length in THSBS are individual for every B: 10 drafts of length 10 for B=1, 3 drafts of length 10 for B=4, and 1 draft of length 20 for other B (8, 16, 32). Medusa heads always generate 1 draft of length 20. The average time and the standard deviation are estimated based on 5 runs

Beam size 10

A). Decoding wall time, min

Inference	B=1	B=4	B=8	B=16	B=32
TBS	50.0 ± 3.8	26.9 ± 3.5	18.7 ± 1.2	14.9 ± 0.1	16.2 ± 0.1
TBSO	50.0 ± 2.2	16.2 ± 0.3	9.4 ± 0.2	7.3 ± 0.1	5.5 ± 0.1
THSBS	22.7 ± 1.3	10.1 ± 0.2	7.4 ± 0.2	6.1 ± 0.1	5.2 ± 0.0
MSBS	11.4 ± 0.4	4.0 ± 0.2	2.4 ± 0.2	2.1 ± 0.1	1.5 ± 0.1

B).Model calls

Inference	B=1	B=4	B=8	B=16	B=32
TBS	295,947	99,030	54,934	29,941	16,170
TBSO	295,947	99,030	54,934	29,941	16,170
THSBS	92,538	36,960	28,056	15,807	8,817
MSBS	59,502	19,240	10,730	5,906	3,224

C). Average effective batch size

Inference	B=1	B=4	B=8	B=16	B=32
TBS	10	40	80	160	320
TBSO	8	25	45	82	151
THSBS	23	40	29	52	93
MSBS	6	18	32	58	105

D). Acceptance rate, %

Inference	B=1	B=4	B=8	B=16	B=32
THSBS	74	70	64	64	64
MSBS	91	91	91	91	91

Table 4.2. The top-N accuracy of our model in single-step retrosynthesis on USPTO 50K and the proportion of invalid SMILES in the N-th prediction with different decoding strategies: beam search (TBS/TBSO), speculative beam search with heuristic drafting strategy (THSBS), speculative beam search with Medusa heads for drafting (MSBS). The difference in accuracy between all decoding methods is negligible

Single-step retrosynthesis		Top-1	Top-3	Top-5	Top-10
Accuracy, %	TBS/TBSO	52.08	75.16	82.97	89.08
	THSBS	52.08	75.16	82.07	89.12
	MSBS	54.08	75.99	82.92	89.23
		Pred. 1	Pred. 3	Pred. 5	Pred. 10
Invalid SMILES, %	TBS/TBSO	0.8	1.8	3.5	8.1
	THSBS	0.8	1.8	3.5	8.2
	MSBS	0.4	1.6	3.1	9.3

Table 4.3. The comparison between the Transformer and Medusa as a single-step retrosynthesis models within AiZynthFinder on Caspyrus10k under different search algorithms and time limits per molecule. Beam size is 10. Depth-First Proof-Number (DFPN) (A) and Retro* (B-E) are search algorithms (referred as "alg.") that build the synthesis tree. PaRoutes stock (13,414 molecules) (A-C) or ZINC stock (D-E) is used as a building block set. The maximum synthesis route length is 5 (A-C, D) or 7 (E). The search is stopped when at least one route for a given molecule is found

(A) DFPN, PaRoutes stock, depth 5; time limit 5 seconds	Transformer	Medusa
Solved molecules	1117	2080
Common solved molecules	1017	
Avg. time per solved molecule, sec	2.01	1.85
Statistics on common solved molecules::		
avg. time per molecule, sec	1.88	0.86 (x2.)
avg. alg. iterations per molecule	6.52	9.51
avg. alg. iteration time, sec/iteration	0.34	0.10 (x3.)
(B) Retro*, PaRoutes stock, depth 5; time limit 5 seconds	Transformer	Medusa
Solved molecules	3890	5287
Common solved molecules	3628	
Avg. time per solved molecule, sec	2.14	1.41
Statistics on common solved molecules::		
avg. time per molecule, sec	2.06	0.99 (x2.)
avg. alg. iterations per molecule	5.51	7.38
avg. alg. iteration time, sec/iteration	0.43	0.16 (x3.)
(C) Retro*, PaRoutes stock, depth 5; time limit 15 seconds	Transformer	Medusa
Solved molecules	5341	6715
Common solved molecules	5050	
Avg. time per solved molecule, sec	4.25	2.86
Statistics on common solved molecules::		
avg. time per molecule, sec	4.00	1.84 (x2.)
avg. alg. iterations per molecule	12.44	18.99
avg. alg. iteration time, sec/iteration	0.44	0.14 (x3.)
(D) Retro*, ZINC stock, depth 5; time limit 15 seconds	Transformer	Medusa
Solved molecules	7708	8343
Common solved molecules	7439	
Avg. time per solved molecule, sec	3.21	1.73
Statistics on common solved molecules::		
avg. time per molecule, sec	3.06	1.26 (x2.)
avg. alg. iterations per molecule	6.15	10.13
avg. alg. iteration time, sec/iteration	0.60	0.16 (x4.)
(E) Retro*, ZINC stock, depth 7; time limit 15 seconds	Transformer	Medusa
Solved molecules	7888	8608
Common solved molecules	7666	
Avg. time per solved molecule, sec	3.37	1.89
Statistics on common solved molecules::		
avg. time per molecule, sec	3.23	1.41 (x2.)
avg. alg. iterations per molecule	6.5	9.5
avg. alg. iteration time, sec/iteration	0.61	0.19 (x3.)

Table 4.4. Connection between number of beams, multistep success rate of transformer-like models, and their single-step metrics, namely round-trip accuracy, diversity, and percentage of invalid SMILES. The **multistep** experiments are conducted using AiZynthFinder on Caspyrus10k with the retro* search algorithm; ZINC stock (17,422,831 molecules) is used as building blocks; the search is stopped when at least one route for a given molecule is found. Maximum synthesis route length is 7. **Transformer** stands for transformer with beam search, which produces pad-token after EOS-token automatically without extra calls. **Medusa** is a Transformer-like model that uses speculative beam search with Medusa heads as draft source. These models run on a GPU. **AZF** stands for the AiZynthFinder 4.0 default single-step model with a cumulative probability of 99.5% in combination with top-50; it runs on 55 CPU processes in parallel. The multi-step experiments are conducted with a limit of 200 iterations of the planning algorithm and a time limit of 3 minutes (in the case of AZF model, 5 hours per molecule, due to its fast operation). The **single-step** metrics are measured on the USPTO 50K test set. **Eff. N** stands for the number of valid reaction SMILES without repetitions. **R.-t. (round-trip) feasible reactions** stands for the number of reactions that lead to the initial target molecule when evaluated with the Molecular Transformer forward model in top-10 mode. The **diversity** counts the number of unique reaction names, classified by Rxn-INSIGHT [Dobbelaere et al., 2024].

		Multistep			Single-step, average per target molecule			
Model	Top N	Limit	Success rate, %	Total time, h	Invalid SMILES, %	Eff. N	R.-t. feasible reactions	Diversity, classes
Transformer	10	200 it. 180 sec	89.5 94.0	29.6 58.3	4.0	9.2	5.8	5.8
	50	200 it.	97.1	26.4	14.0	39.6	18.5	13.6
Medusa	10	200 it. 180 sec	87.9 94.9	10.3 41.3	4.07	9.4	5.9	5.8
	50	200 it.	95.9	15.4	15.2	39.9	18.5	13.4
AZF (template-based)	50	200 it. 5 h.	72.6 82.6	2.7 24.0	0.	15.1	9.4	7.9

Table 4.5. The comparison between different single-step inference methods within AiZynthFinder on Caspyrus10k under the retro* search algorithm with iteration limit 200 per molecule. ZINC stock (17,422,831 molecules) is used as building blocks. Beam size is 50. Maximum synthesis route length is 7. **Transformer** stands for transformer with beam search, which produces pad-token after EOS-token automatically without extra calls. **Medusa** is a Transformer-like model that uses speculative beam search with Medusa heads as draft source. **AZF** stands for AiZynthFinder single-step model. **Av. retro* iter. time** represents the average time of a single retro* iteration. **GPU** stands for Tesla V100 32 GB. CPU experiments data is taken from [Torren-Peraire et al., 2024]

Model	Time limit per molecule, min	Success rate, %	Av. retro* iter. time, s	Device
Transformer	3	97.1	0.67	GPU
Medusa		95.9	**0.31**	
Local Retro [Torren-Peraire et al., 2024]	480	74.1	0.81	CPU
Chemformer [Torren-Peraire et al., 2024]		62.4	107.6	
AZF [Torren-Peraire et al., 2024]		41.1	0.80	

Table 4.6. Evaluation performed on 1,000 randomly selected targets from the PaRoutes-n1 dataset. For each target, up to 50 synthesis routes were collected during 500 search iterations. Route accuracy is expressed as the absolute number of targets for which the corresponding model retrieved the reference route among the top-50 predicted routes. Building-block accuracy is calculated for molecules solved only by the rival model using the route accuracy metric and is expressed as the number of molecules for which the corresponding model predicted the reference set of building blocks.

	Transformer		**Medusa**
Success rate, %	94.1		93.7
Route accuracy	286		276
Route accuracy overlap		248	
Building block accuracy on unsolved molecules	8/28		22/38
Runtime, h	24		10

Chapter 5

Reagent prediction as SMILES generation

A clever planning algorithm and a fast, accurate, single-step retrosynthesis model are necessary components of a good machine-learning-based CASP system, but these alone are insufficient for the system to be comprehensive and truly helpful to end users. In principle, when modeling a chemical reaction, it is beneficial to take into account as much relevant information as possible. Among the most important aspects of a reaction besides reactants and products are reaction conditions (Fig. 2.3). The conditions comprise temperature, pressure, and other physical parameters, as well as the chemical environment imposed by reagents, i.e., catalysts, solvents, and other molecules necessary for a reaction to occur. The conditions and reagents are integral to any reaction. The same reaction under different conditions can yield different outcomes. When the transformations suggested by a CASP system do not specify the conditions, the user must still determine them, and it is a task that requires significant cognitive load.

The reagent information is commonly ignored in works describing single-step retrosynthesis models (see Section 2.4.2); even though some studies described the prediction of retrosynthetic steps encompassing reagents [Schwaller et al., 2020], most of them only suggest reactants, leaving the chemist wondering about the actual procedure needed to conduct the proposed reaction. Therefore, a specialized reagent prediction model would be a necessary component of an ideal CASP system.

Datasets of known organic reactions that enable machine-learning-based reaction modeling are scarce (see Section 2.3.2). The USPTO dataset is the primary source of reaction data for machine learning among researchers, as it is the only collection

This chapter is based on our manuscript "Reagent prediction with a molecular transformer improves reaction data quality" [Andronov et al., 2023]

of organic reaction data that is openly available and relatively large. Unfortunately, the reagents in this dataset are noisy; they are often unspecified or incorrect, and all temperature or pressure conditions are omitted. Therefore, the models for reaction product prediction or retrosynthesis cannot exactly benefit from full and reliable reaction information when developed using USPTO.

Our main contributions: We propose an encoder-decoder transformer as a model for reagent prediction. Given the SMILES of reactants and products, the model learns to predict the SMILES strings of reagents. Unlike existing approaches designed specifically for reagent prediction, our formulation is not confined to a predefined set of possible reagents and can predict reagents for arbitrary reaction types. Although our model is not the first transformer suitable for reagent prediction in principle [Vaucher, Schwaller and Laino, 2020; Lu and Zhang, 2022], it is the first one to be specifically trained for reagent prediction in a machine translation setting. We also demonstrate that the reagent prediction model can be used to improve a product prediction model trained on the USPTO dataset. First, we predict missing reagents for not well-specified reactions in USPTO. We then train a transformer for product prediction on the corrected USPTO and observe that it improves the accuracy of a basic Molecular Transformer [Schwaller et al., 2019] on the USPTO MIT benchmark.

5.1 Method

We formulate the reagent prediction problem as SMILES-to-SMILES translation, similarly to reaction product prediction and single-step retrosynthesis (see Section 2.4.2), and train an encoder-decoder transformer to solve it. As input, our model takes a reaction SMILES with a ”>>” separator, where to the left of the separator are the SMILES of the reactants separated by dots, and to the right are the SMILES of the products (see Section 2.2.2). The target sequence for each reaction is the SMILES of the reagents. This allows the model to predict reagents for a broad range of reactions without common restrictions: the number of reagents in a reaction and their particular roles are not predetermined.

5.1.1 Model

We use the original OpenNMT [Klein et al., 2018] implementation of the Molecular Transformer (MT) [Schwaller et al., 2019] for reagent prediction and reaction product prediction. For the training details, see Section C.5.

5.1.2 Data

We use the entire USPTO dataset (see Section 2.3.2) to train the reagent prediction model. However, we do not test the model on USPTO because of the unreliability of ground truth reagent sequences in USPTO (see Section C.1). To test the reagent prediction model, we extract a subset of Reaxys that resembles USPTO 50K in its distribution of reaction classes (see Section C.2). We hypothesize that Reaxys has better-curated reagent records. Before training the reagent model, we preprocess the training data (see Section C.3) and the test data (see Section C.4). We randomly choose 5000 reactions from the training set for validation.

For product prediction, we train and evaluate the model on the USPTO MIT dataset [Jin et al., 2017], a common benchmark for this task.

5.2 Related work

Several manuscripts described methods for predicting suitable reaction conditions or chemical contexts across various scenarios. For example, there are reports on using DFT to select suitable solvents for a reaction [Struebing et al., 2013], or thermodynamic calculations to choose heterogeneous catalysts [Toulhoat and Raybaud, 2020]. Some focused on optimizing reaction conditions for particular reaction types using an expert system [Marcou et al., 2015] or machine learning [Maser et al., 2021; Afonina et al., 2022; Walker et al., 2019; Angello et al., 2022]. Gao et al. [Gao et al., 2018] have used a fully connected neural network trained on the Reaxys data in the form of reaction fingerprints to predict reagents in a supervised classification manner. Additionally, Ryou et al. [Ryou et al., 2020] have extended this approach by using a graph network to encode information in reaction graphs rather than reaction fingerprints, and by relying on Reaxys and treating the task as a supervised classification problem.

5.3 Experiments

5.3.1 Model performance

Reagent prediction is not as straightforward as forward reaction prediction. A reaction may be carried out using different sets of reagents. To put it in another way, there may be more than one plausible chemical context for a given reaction: catalysts, redox agents, acids, bases, and solvents in a reaction can be more or less replaceable. Therefore, multiple different sets of molecules might be correct predictions for a given transformation. Having that in mind, we chose the performance measures to be the

Table 5.1. The performance of the transformer for reagent prediction on the test set obtained from Reaxys. All scores are given in percent points

Metric	Top-1	Top-2	Top-3	Top-4	Top-5
Exact match accuracy	17.0	24.7	29.2	31.8	33.5
Full recall	19.2	28.4	35.1	39.3	42.8
Partial match accuracy	70.9	80.5	84.9	87.3	88.9

following:

1. Exact match accuracy: the prediction of the model is considered correct if the symmetric difference between the set of predicted molecules and the set of the ground truth molecules is an empty set.
   Example: $A.C.B$ is an exact match to $A.B.C$.

2. Partial match accuracy: the prediction counts as correct if the ground truth contains at least one of the predicted molecules.
   Example: $A.B$ is a partial match to $A.C.D$.

3. Recall: the number of the correctly predicted molecules divided by the number of molecules in the ground truth.
   Example: $A.B.C$ has 100% recall of $A.C$, $A.D$ has 50 % recall of $A.B.C.D$.

Here, A, B, C, D denote the SMILES strings of some molecules.

We use beam search with a beam size of 5 to obtain predictions from the transformer. Therefore, all performance metrics report the top-N predictions, where N ranges from 1 to 5. A correct top-N prediction means that the correct answer appeared among the first N sequences decoded with beam search. Additionally, our test set contains duplicate reactions with different reported reagent sets. While gathering performance statistics, we group predictions by unique reactions: if the model correctly predicts reagents for one of the duplicates, we count the reaction as correctly predicted.

The model performs quite well on the test dataset (Table 5.1). For each test reaction, each of the top-5 predictions is a valid SMILES string. An exact match of the prediction and the ground truth sequence is observed in 17.0% of the cases for top-1, 29.2% for top-3, and 33.5% for top-5. At the same time, full recall is higher at 19.2%, 28.4%, and 42.8%, respectively. As for the partial match between predictions and ground truth sequences, it is much higher at 70.9%, 84.9%, and 88.9% in the top-1, top-3, and top-5 cases, respectively. Thus, the model cannot correctly predict a single reagent for 11.1% of the reactions in the test dataset. In our evaluation, we do

not employ methods to assess the plausibility of incorrect predictions, such as similarity metrics for interchangeable solvents; therefore, the performance scores should be considered underestimates.

5.3.2 Model confidence

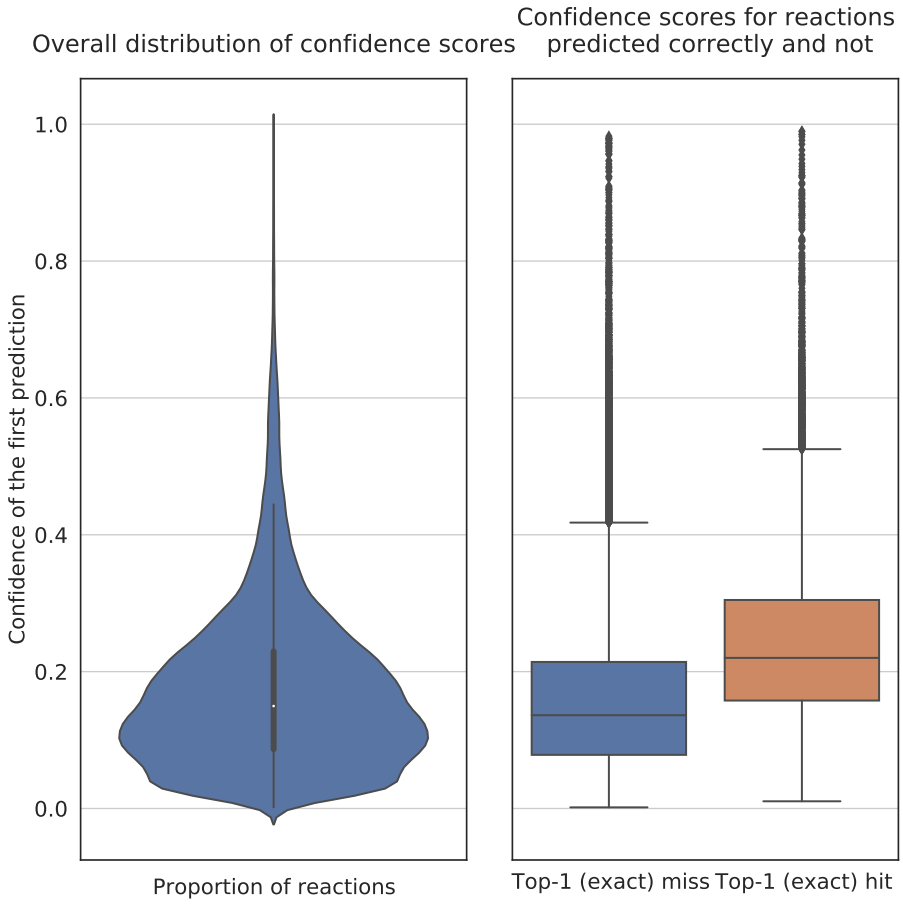

Figure 5.1. Confidence scores of the model predictions. On the left, a violin plot reflecting the distribution of confidence scores across all predictions on the Reaxys test dataset. On the right, separate boxplots of the distributions for correct and incorrect predictions in terms of top-1 exact matches.

The confidence scores of the model are much lower than those [Schwaller et al., 2019] of the Molecular Transformer for product prediction (Fig. 5.1). The probability of a decoded sequence is the product of the probabilities of all predicted tokens in it.

In particular, their average value is between 0.1 and 0.2, whereas in reaction prediction most scores exceeded 0.9. The reason appears to be the nature of the problem: several plausible sets of reagents may be proposed for a reaction, and the answer is more ambiguous than in product prediction. Nonetheless, the model's confidence is, on average, noticeably higher for correct predictions than for incorrect ones.

5.3.3 Performance across publication years

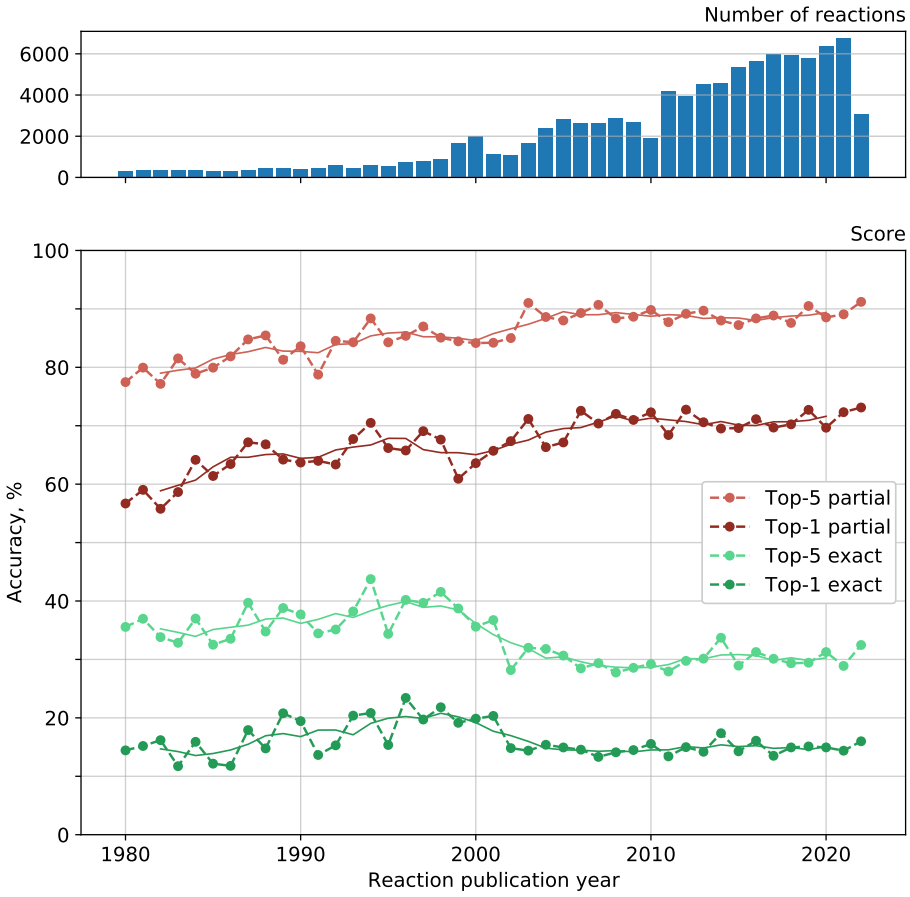

Figure 5.2. On the top, the number of test set reactions published in each year. At the bottom, the dependence of the reagent model's performance on the year of reaction publication. Solid lines represent the five-year moving average.

We examined the model's dependence on reaction publication date. As chemists discover new reaction types and new possible reagents for known reactions, the statistical knowledge gathered by a reagent model can become outdated.

Fig. 5.2 shows both the exact and partial accuracy of our reagent model on reactions published in every single year between 1980 and 2022. Solid lines represent the moving average of accuracy over five years, using a centered window. The top-1 exact match accuracy tends to increase on average from 1980 to 1998, then decreases until 2005, and plateaus after that. The picture is similar for top-5 exact accuracy and top-2 to top-4 as well. The dependence for total recall is similar. The top-1 partial match accuracy tends to increase on average from 1980 to 2008 and then stagnate. Top-2 to top-5 accuracies demonstrate similar behavior as well.

5.3.4 Performance across reaction classes

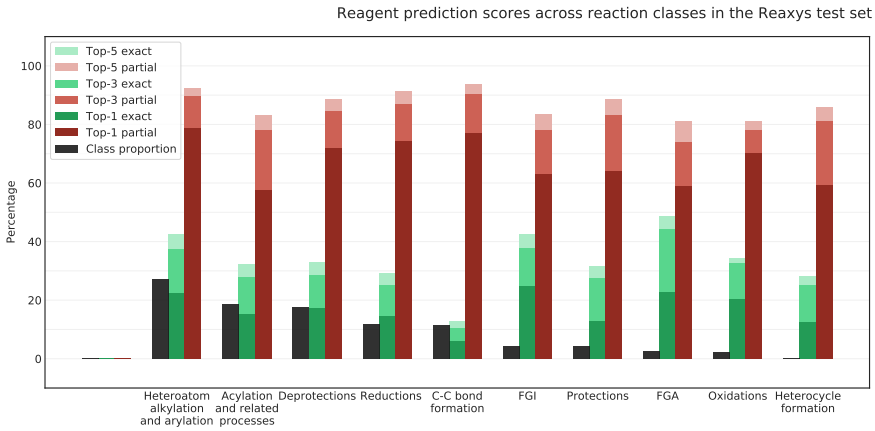

Figure 5.3. Percentage of partial (rightmost, in red) and perfect matches (middle, in green) between the target sequence and the predicted sequence across ten reaction classes in the Reaxys test set. The class proportions are leftmost, in black. All values are grouped by unique reactions.

We investigated the model's performance across various reaction classes in our test dataset. The middle and rightmost bars for each reaction class in Fig.5.3 show the rate of exact and partial matches of the model prediction and ground truth. The leftmost bars reflect the relative proportion of each reaction class in the test dataset. We observe that the quality of the model predictions varies noticeably across classes. This performance difference is most likely due to differences in the data distributions between the training and test sets across reaction classes. The model demonstrates the best top-5

exact match accuracy for FGA, FGI, heteroatom alkylation and arylation, and the best top-5 partial match accuracy for C-C bond formation. At the same time, C-C bond formation has the lowest exact match accuracy. The reason for that must be the wide variety of interchangeable reagents in this reaction class, especially metal-based catalysts. As we use USPTO 50K as a proxy for the training set, we assume that the least represented reaction classes in the former are also the least represented in the latter. The four least represented classes in USPTO 50K are FGA, oxidations, protections, and heterocycle formations. Notably, the model generalizes well across all these classes.

5.3.5 Performance across reagent roles

We also examined the quality of the model predictions for each reagent role (Fig. 5.4).

The first and third columns of the table in the figure indicate that, in both the top-1 and top-5 cases, the solvents are the most difficult to predict. This is most likely because they are often the most interchangeable. However, reactions need not involve reagents in every possible role, and the picture is somewhat different when the roles in the ground-truth sequence were strictly nonempty SMILES strings. In this case, predicting oxidizing agents was the most difficult task. This effect is likely related to flawed heuristics for reagent classification or to the large difference between the typical oxidizing agents in the training and test sets. The more detailed performance summary across both reagent roles and reaction classes is shown in Figure C.2 in Section C.6.

5.3.6 Analysis of prevalence of reaction types

In the test set, there are 684 unique reaction types determined by NameRXN. These types can be split into five ”bins” by occurrence frequency. The summary of those bins is given in table 5.2. The most common types are those which occur more than a thousand times in the test set. There are only 20 such types. Among others, they include Suzuki coupling, Williamson ether synthesis, aldehyde reductive amination, N-Boc protection, and nitrile reduction. Heterocycle formation, Oxidations, FGI, and FGA are not present among the common types. Out of all unique types, 245 are singular, meaning that they are represented in the test set by only one instance. Frequent types have 101 to 1000 instances, rare types have 11 to 100 instances and very rare types have from 2 to 10 reaction examples. The performance of the reagent model decreases with the decrease in type prevalence, which is expected.

In the ”common” and ”frequent” bins, there are no reaction types with only a single possible reagent in any role. In the ”rare” and ”very rare” bins, there is a small number of types in which it is the case. However, the analysis is limited by NameRXN and by the heuristics of role classification. If the type label is a name reaction, which is an

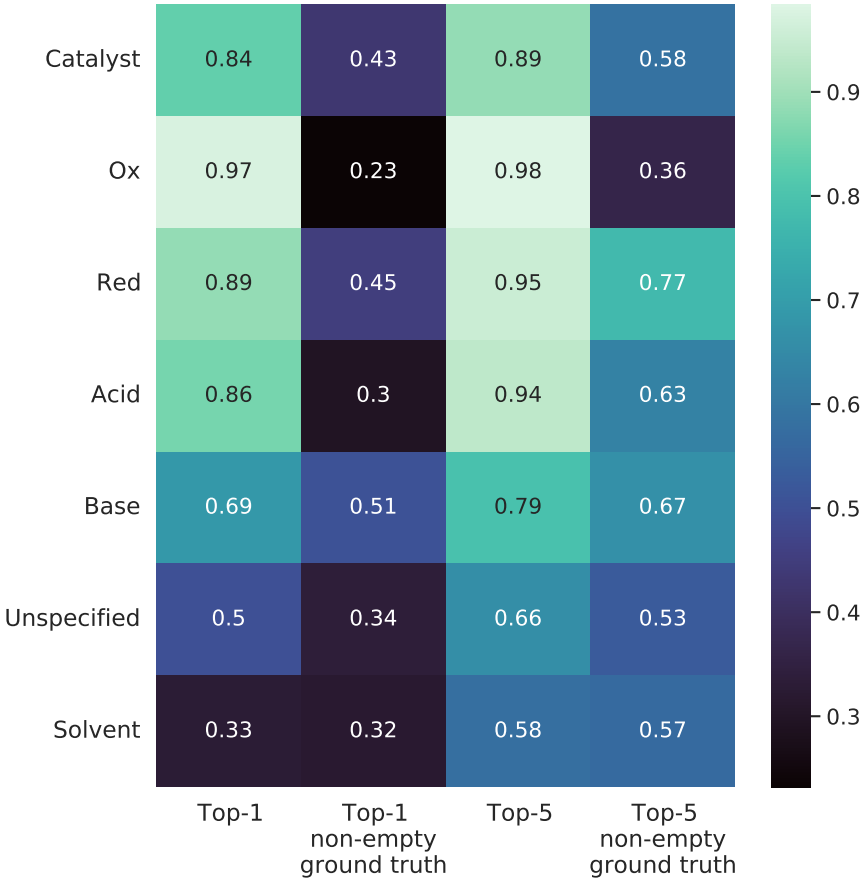

Figure 5.4. Comparison of the proportion of test examples on which the prediction matches the ground truth exactly in each reagent role. The comparison is given for the top-1 and top-5 predictions, both in general and when the ground-truth (GT) sequence is strictly not an empty string.

infrequent case, then the reactions with that label may have a single option for one of the roles. For example, all instances of the rare ”8.1.24 Ketone Swern oxidation” type have an oxidizing agent which is the same for all, and the instances of the rare ”9.1.2 Appel chlorination” and the very rare ”1.7.8 Ullmann condensation” types all have their own specific catalyst. Additionally, some ”very rare” types may have only one option, even for solvents in the test set, but this may be an artifact of under-representation. The reagent may or may not give a perfect top-1 prediction for all reactions of such types.

Table 5.2. The statistics of the reaction types in the Reaxys test set. The types are determined using NameRXN.

Occurence	N	Criterion	Reactions	Top-1 exact acc., %	Top-1 partial acc., %
Common	20	$N > 1000$	67633	17.0	73.5
Frequent	67	$100 < N \leq 1000$	23685	16.2	65.1
Rare	120	$10 < N \leq 100$	4219	13.5	60.3
Very rare	232	$1 < N \leq 10$	947	14.2	53.1
Singular	245	$N = 1$	245	13.5	62.4

5.3.7 Improving product prediction

In addition to predicting reagents *per se*, our model can be used to augment the USPTO with additional reagents for reactions that lack them, and to train a product prediction model on this augmented dataset. As noted above, many reactions in the USPTO contain reagents that are under-specified, yet many reactions in USPTO contain the full set of reagents at the same time. This allows us to use a trained reagent prediction model to recover missing reagents in some reactions. We apply the model trained on the entire USPTO to the USPTO MIT subset. During training, we make sure that the USPTO MIT test set does not overlap with the training set for the reagent model.

There can be various strategies for reagent string replacement. We apply the following rule: if the top-1 prediction of the model contains more molecules than the original string, then the prediction replaces that string. With that, we can improve the reactions with missing reagents without corrupting the good ones. We are aware, however, that this strategy is not ideal, and we are convinced that better ones are possible. Some examples of the reactions with reagents improved after the reagent model inference are shown in Figure 5.5.

The first reaction is an example of peptide coupling. The typical reagents in this case comprise HOBt or its analogs (e.g., HATU), usually used together with Hünig's base (DIPEA). The reagent model reintroduces the missing reagents to the reaction. The second one is an example of reductive amination, and the information about the solvent is not enough. The model proposed a suitable reducing agent, sodium triacetoxyborohydride. The last two reactions are the Suzuki coupling and the Sonogashira reaction, respectively. The model suggests the standard reagents that define these reaction types.

We compared two models, both standard Molecular Transformers with identical hyperparameters, but trained on different datasets. The first model, which we denote as "MT base", was trained on the standard USPTO MIT. This is the model from the original Molecular Transformer paper. The second model, which we denote as "MT new",

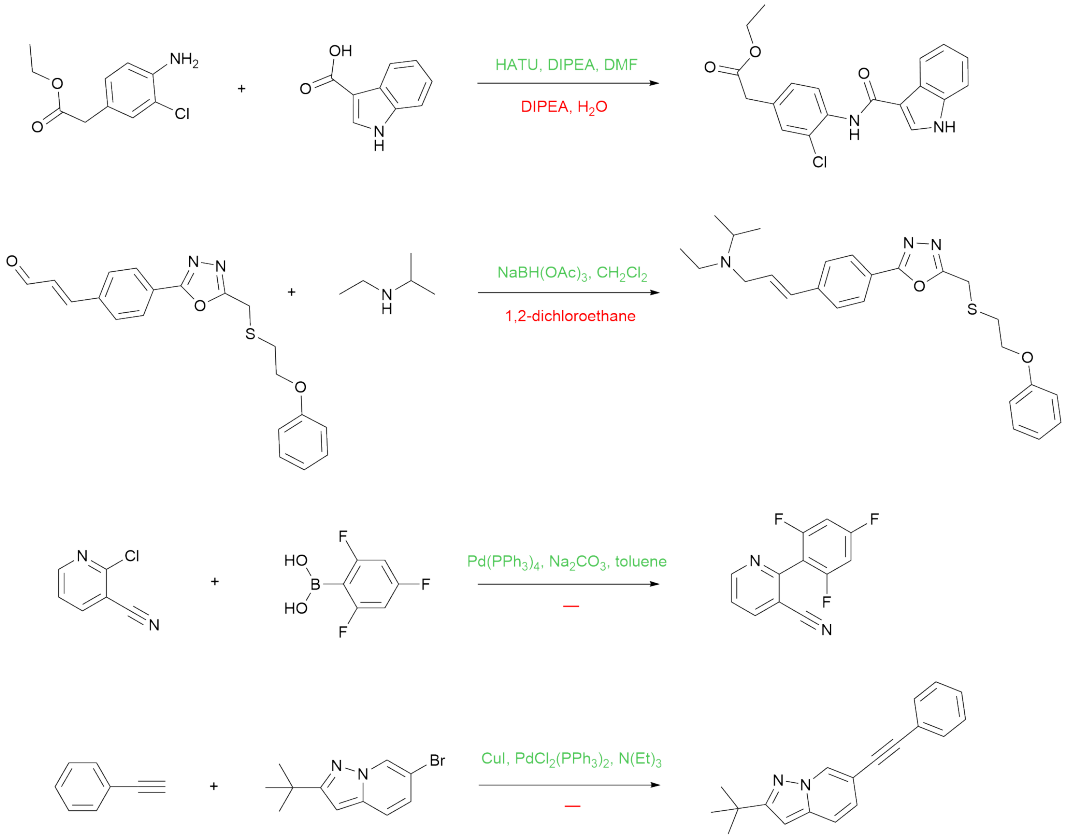

Figure 5.5. Examples of reactions in the USPTO MIT training set for which the reagent model successfully improves reagents. Model predictions are above the arrows (in green), and original reagents are below the arrows (in red).

Table 5.3. The top-1 exact match accuracy (%) of reaction product prediction, both for the Molecular Transformer trained on the default USPTO MIT (MT base) and the Molecular Transformer trained on the USPTO MIT, where in some of the reactions, reagents were augmented by the reagent prediction model (MT new). The models were compared on both the USPTO MIT and Reaxys test sets in the separated, mixed, and no-reagents settings. There is no difference between the old and the new model in the latter case

	Reaxys	USPTO MIT
MT, no reagents	77.3	84.0
MT base, mixed	82.0	87.7
MT new, mixed	**83.0**	**88.3**
MT base, separated	84.3	89.2
MT new, separated	**84.6**	**89.6**

was trained on USPTO MIT, in which some of the reactions had reagents replaced according to the procedure described above. We chose top-1 exact match accuracy as the quality metric. Before testing on Reaxys, we reassign the reactant-reagent partition in every test reaction with the role assignment algorithm [Schneider et al., 2016]. This is done to be consistent with the inference procedure, in which the reagents to replace are the reagents determined by this algorithm. Additionally, we train another Molecular Transformer for product prediction without any reagents in the source sequences. The performance summary of the models is presented in Table 5.3.

The results of the base model are reproduced as described by Schwaller et al. [Schwaller et al., 2019] without SMILES augmentations, checkpoint averaging, or model ensembling. The new model performs better than the old model on both Reaxys and USPTO in both separated and mixed settings. This performance improvement is statistically significant. To prove the statistical significance, we employed McNemar's test [Dietterich, 1998]. The details are provided in Section C.7. The performance on Reaxys is worse than on USPTO because the distribution of data in Reaxys differs more from the distribution in the training set than in the USPTO test set. However, it is important to emphasize that the USPTO test set also underwent a reagent change to test the new model. The performance in the mixed setting is slightly worse than in the separated setting, which is expected [Schwaller et al., 2019]. The score of the model trained with no reagents is expectedly the lowest on both Reaxys and USPTO. However, surprisingly, it is only 4.7 percent points below the base model's score on Reaxys and 3.7 percent points below the base model's score on USPTO. Therefore we can conclude that even though the reagent information helps the product models trained on USPTO,

which it should from a chemical perspective, the effect is not that drastic. However, we conjecture that such improvement will be much more noticeable on larger scale, e.g. if all the models are trained on the entire Reaxys. We suggest that this is a manifestation of USPTO's flaws: the dataset does not contain many reactions in which the same reactant transforms into different products under different conditions.

5.4 Conclusion

A transformer neural network, which is one of the models to achieve state-of-the-art results in reaction prediction, can also learn to successfully predict reagents for organic reactions, which is important for recommending reaction conditions. The reagent prediction model receives an atom-mapping-free reaction SMILES string with no reagents and suggests multiple possible sets of reagents for it. Our work is the first to use the strategy of training a reagent prediction model on USPTO and testing it on a Reaxys subset, demonstrating its generalization capabilities. We also use the reagent prediction model to improve the performance of a product prediction model on USPTO MIT in a self-supervised fashion. In order to do that, we use the reagent model to reconstruct the missing reagents to the reaction data before training the product prediction model on it. Since reagent information is important to predict reaction products, our approach allows to outperform a state-of-the-art model for reaction prediction while being model-agnostic. In our work, in particular, we improve upon the score of the Molecular Transformer on the USPTO MIT (USPTO 480K) benchmark dataset.

Chapter 6

Curation of reagent data with a reagent space map

A reagent prediction model (see Chapter 5) is a necessary part of a useful CASP system. However, before training a reagent prediction model, it is essential to carefully prepare the corresponding data. In organic reaction datasets, the reagents typically require particular attention. Reactant-reagent separation may be inadequate, the same reagent molecule may be represented in different ways in different reactions, and valuable information about the detailed roles of reagents is often missing. While the creators of popular reaction databases attempt to address these issues to some extent, the data often requires additional curation. USPTO, the primary source of open reaction data for research, notoriously suffers from issues with reagent records. For example, reactant-reagent separation in USPTO is derived from an algorithmically inferred atom-atom mapping, which is often inaccurate. The imperfection of this mapping leads to imperfect separation. Additionally, reagent roles are often not detailed enough: both Reaxys and USPTO feature only three roles: ”catalysts,” ”solvents,” and ”reagents”, with the latter being a broad category covering reagents with various purposes. When building reagent prediction models, their detailed performance analysis may require access to richer reagent role attribution. Therefore, we decided to address the problem of reagent data curation.

Our main contributions: We present a simple visual tool for the inspection and curation of reagents in chemical reaction data. The tool is an interactive reagent space map based on distributed vector representations of reagents, served through a web application. We obtain reagent representations using an algorithm equivalent to the

This chapter is based on our manuscript ”Curating reagents in chemical reaction data with an interactive reagent space map” [Andronov et al., 2024]

famous word2vec [Mikolov et al., 2013] algorithm from Natural Language Processing, and they reflect the statistics of reagent co-occurrences and interaction in a given reaction data corpus: representations of reagents of the same role tend to cluster together. We demonstrate the application of our tool to the USPTO dataset. Using our tool, we label around six hundred of the most common reagents in USPTO into ten detailed roles, detect reactants erroneously listed as reagents, and ensure the uniqueness of unique reagents' names. Our reagent space mapping and the web application work with any reaction dataset, and we are confident that they will benefit organic chemists and cheminformatics specialists working with their own reaction data.

6.1 Method

6.1.1 Distributional hypothesis for reagent curation

We develop a natural way to cluster reagents into detailed roles automatically by applying the distributional hypothesis from natural language processing: words that occur in similar contexts tend to have similar meanings. With reaction data, it can be reformulated as follows: if two reagents tend to occur in similar contexts, i.e., with similar other reagents, then it is likely that these two reagents are alternatives and have the same role in reactions. For example, we do not expect two different palladium catalysts for Suzuki coupling to occur simultaneously in the same reaction, but one can use the same bases and solvents with both (Fig. 6.1). Therefore, if we obtain distributed vector representations of reagents based on their co-occurrence data, we can expect the vector space of reagent representations (embeddings) to feature role clusters, such as a cluster of specific catalysts, bases, acids, and others.

6.1.2 Reagent embeddings

To calculate the embeddings of reagents useful for data curation, we take inspiration from the word2vec algorithm. The original word2vec [Mikolov et al., 2013] is a foundational method in machine learning for natural language processing that has sparked widespread interest in this field and inspired numerous increasingly complex successors and improvements, which have gradually led to modern LLMs. The goal of the word2vec model is to obtain learned, distributed vector representations of words in natural language for subsequent use in downstream tasks, such as text classification [Lilleberg et al., 2015]. It has been demonstrated that embeddings similar to those of word2vec enable highly effective text compression [Schmidhuber and Heil, 1996]. The idea behind word2vec is based on the distributional hypothesis: words that occur in similar contexts are likely to have similar meanings and, therefore, should receive

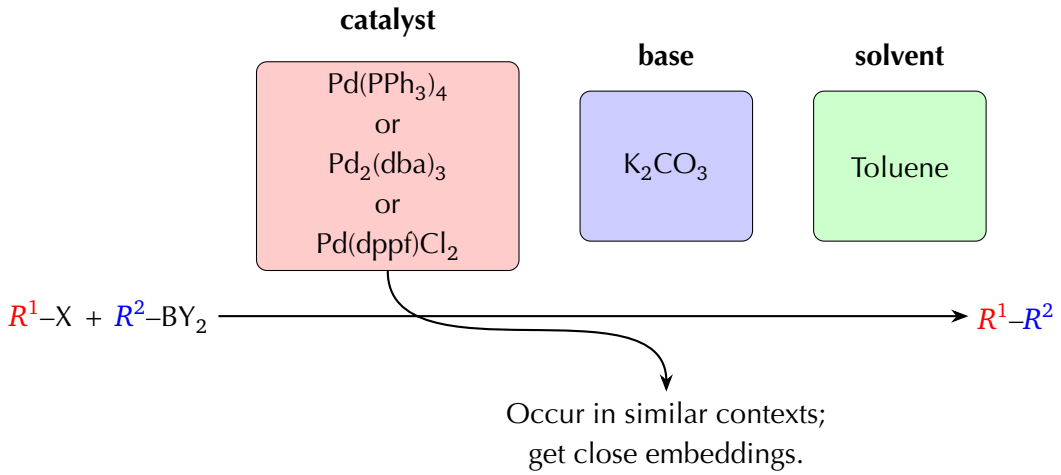

Figure 6.1. The general scheme of the Suzuki coupling reaction. Suzuki coupling is enabled by a palladium catalyst and base in a suitable solvent. Different palladium catalysts can work with the same base.

similar vector representations (embeddings). The algorithm iteratively trains those initially random embeddings by solving a classification task: identifying the words that are likely to fit the context of a given word. The context of a word consists of other words within a window centered on that word, and the data pairs for training are obtained using a sliding window over a large text corpus.

Beyond NLP, word2vec has gained widespread adoption in the fields of cheminformatics and bioinformatics. For example, researchers have adapted it for constructing universal feature vectors for small molecules [Jaeger et al., 2018; Shao et al., 2022]. Additionally, it has been utilized to generate meaningful representations of nucleic acids for phylogenetic analysis [Ren et al., 2022], predict drug–miRNA associations [Guan et al., 2022], and predict RNA degradation [Krishna et al., 2022]. Additionally, word2vec embeddings have been utilized for proteins in tasks such as drug-target interaction [Wang, Zhou and Chen, 2023], drug-target affinity [Xia et al., 2022], protein-protein interaction [Wang et al., 2019], and others. See [Öztürk et al., 2020] for a broad survey of the diverse applications of word2vec in bioinformatics and cheminformatics.

While word2vec is an iterative algorithm, Levy and Goldberg [Levy and Goldberg, 2014] derived a proof that a formulation of word2vec called SGNS (skipgram with negative sampling) is equivalent to factorizing the matrix of point-wise mutual information (PMI) scores with singular value decomposition (SVD). PMI scores can be obtained from co-occurrence counts (Eq. C.1). Although learning word embeddings is better approached with the iterative algorithm, as there are hundreds of thousands of unique

words and their co-occurrence matrix would be too large, the PMI matrix factorization is more efficient when the number of entities to embed is relatively low. This is exactly our case: there are only a few hundred unique reagents to consider in a dataset such as USPTO, and we can resort to PMI matrix factorization when building reagent embeddings.

We obtain reagent embeddings by factorizing the point-wise mutual information (PMI) scores matrix with singular value decomposition (SVD). We can easily derive the table of PMI scores from the table of reagent counts by equation C.1.

$$\mathrm{PMI}(x, y) = \log_2 \frac{P(x, y)}{P(x)P(y)} \tag{6.1}$$

where x and y are reagents, $P(x, y)$ is a relative frequency of a reagent pair among all reagent pairs, and $P(x)$ and $P(y)$ are relative frequencies of individual reagents. Reagent vector representations obtained in this manner lie close in the vector space for entities with similar ”meanings”, which are determined by the ”companions” of those entities.

Singular value decomposition is a way of factorizing a matrix in linear algebra. It is a generalization of matrix eigendecomposition to non-square matrices. SVD decomposes a real-valued matrix M of size $m \times n$ into three factors according to Eq. 6.2:

$$M = U\Sigma V^T \tag{6.2}$$

where U and V are orthogonal matrices with sizes $m \times m$ and $n \times n$, respectively, and Σ is an $m \times n$ diagonal matrix with non-negative real numbers on the diagonal called singular values. The number of non-zero singular values is equal to the rank of M, and we can truncate the sizes of Σ to the desired number of singular values, which controls the sizes of U and V. We perform SVD in Python using the SciPy library [Virtanen et al., 2020] and use U as the matrix of reagent embeddings, in which every row is a reagent embedding with the dimensionality equal to the chosen number of singular values. One can freely select the reagent embedding dimensionality, and we choose it to be 50 to achieve information compression that forces reagent embeddings into role clusters but with enough degrees of freedom.

6.2 Experiments

6.2.1 Reagent role attribution with a web application

We extracted 626 unique reagents in their contexts from the USPTO dataset (see Appendix D.1) to identify their roles and standardize them using our method. We obtained the reagent embeddings and placed them onto a UMAP [McInnes et al., 2018]

projection, which we displayed in an interactive web application (Fig. 6.2). When the user hovers over a point on the map, the corresponding reagent structure and its SMILES appear on the screen. The reagents in the map can be filtered by SMARTS patterns and occurrence frequency. The interactivity of our application and the clustering tendency of reagents with the same role on the map enable faster decision-making about reagent roles compared to using only tabular reagent data. Our web application allowed us to quickly attribute all 626 reagents to eleven detailed roles (see Appendix D.2). Figure 6.3 demonstrates the application appearance after roles have been assigned to reagents.

6.2.2 Redundant reagent records detection

The properties of the reagent embedding map enable easy detection of different SMILES that represent the same reagent, since such SMILES are naturally assigned similar embeddings by our method. For example, Figure 6.4 demonstrates a zoom-in on a map region occupied by strong bases. By exploring this region on the interactive map, we can immediately discover that there are sometimes several alternative SMILES representations for the same reagent in the USPTO data. We see two SMILES representations for n-butyllithium, two for lithium diisopropylamide, and three for lithium bis(trimethylsilyl)amide. In other regions of the map in Figure 6.3, there are sometimes mixtures of reagents determined in our preprocessing as one reagent (see Appendix D.1), e.g., two solvents together as a unique reagent. When preparing reaction data for various machine learning experiments, one of the objectives is to ensure that each unique reagent is represented by a unique SMILES string. We revised our reagent map once more, standardizing all redundant SMILES we found in the map in this way and reducing the number of unique reagent entries in our map from 626 to 559.

6.2.3 Alternative reagent determination

At the data preprocessing step (see Appendix D.1), we also experimented with an alternative way of separating reactants from reagents - the fingerprint-based reagent detection procedure described by Schneider et al. [Schneider et al., 2016] and available in RDKit. It does not depend on AAM and is therefore more universal. With this alternative preprocessing, we obtain 558 unique reagents in the dataset after standardizing alternative reagent SMILES. In the reactions where reagent detection occasionally failed and determined all reactants as reagents, we resorted to AAM-based reagent extraction. Among the resulting 558 unique reagents, 25 entries, 14 of which we assign the ”reactant” role (see Appendix D.2), are not among our 559 reagents determined using AAM. At the same time, 26 reagents (19 ”reactants”) determined by AAM are not

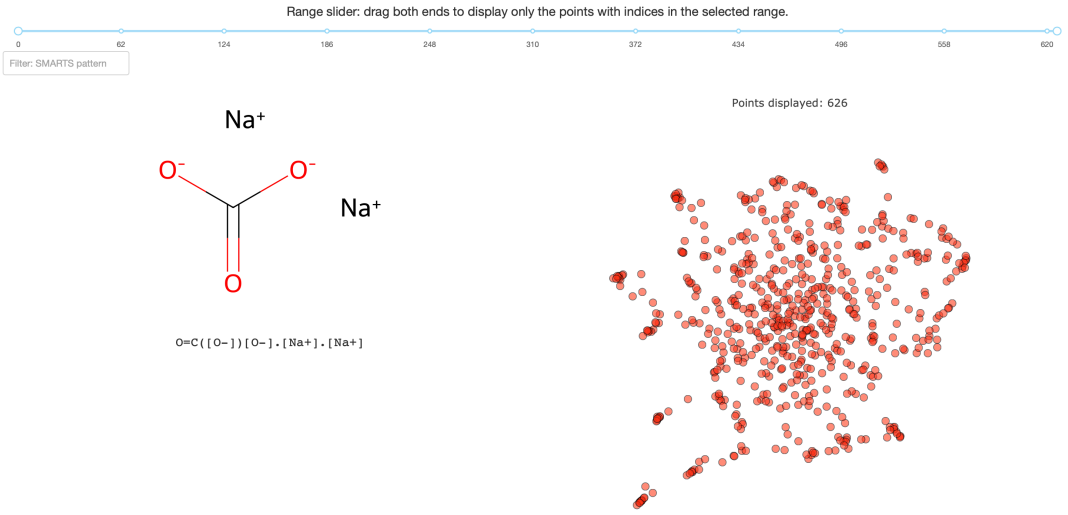

Figure 6.2. The appearance of the web application for the exploration of reagent space. Reagents in our demonstration come from USPTO. On the right, there is an interactive UMAP projection of reagent embeddings. Every point corresponds to a unique reagent. Upon hovering over a point, the corresponding reagent is displayed on the left. Various filters are available for the map. The embeddings are obtained by factorizing the matrix of point-wise mutual information scores between pairs of reagents with singular value decomposition (SVD). The original embedding dimensionality is 50.

in the set of reagents determined by the fingerprint procedure. However, the reagent embedding space maps in both cases do not differ in any significant way. Therefore, we conclude that both reagent determination procedures are alternatives for the user to choose from, depending on the user's confidence in the AAM reliability in their data. In the following analysis, we will consider the set of reagents determined by AAM in USPTO.

6.3 Analysis

After labeling 626 reagents with roles and standardizing all reagent SMILES, reducing the number of reagents to 559, we generated reagent embeddings again. The obtained embedding map demonstrated a slightly different shape but preserved the same clusters united by reagent action. To highlight the regions occupied by similar reactions, we decided to color the map plane using a Voronoi diagram. A Voronoi diagram for a set of two-dimensional points is a partition of the plane into regions defined by the distance to every point in the set. In this case, the points are referred to as seeds, and

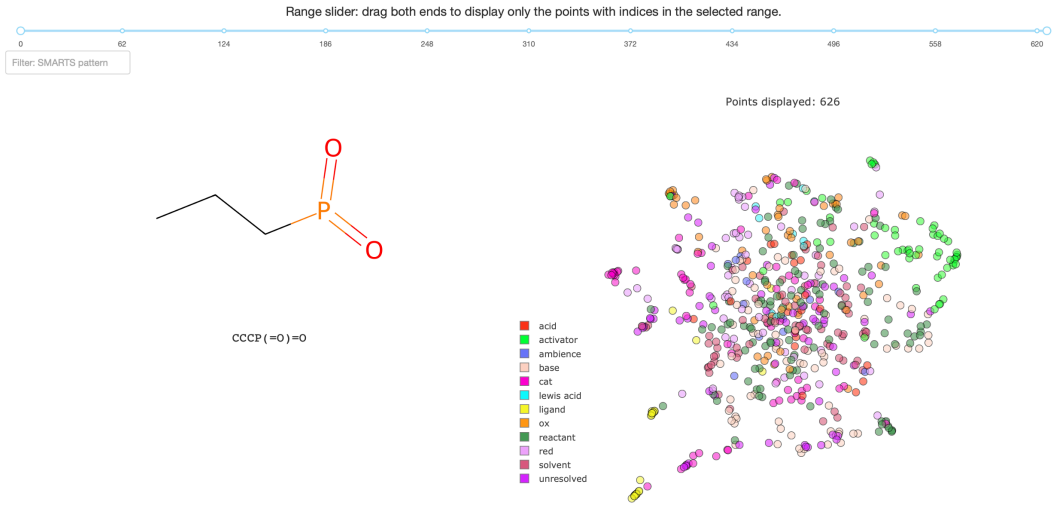

Figure 6.3. The map of embeddings for the subset of 626 most common USPTO reagents colored according to the detailed reagent roles assigned manually.

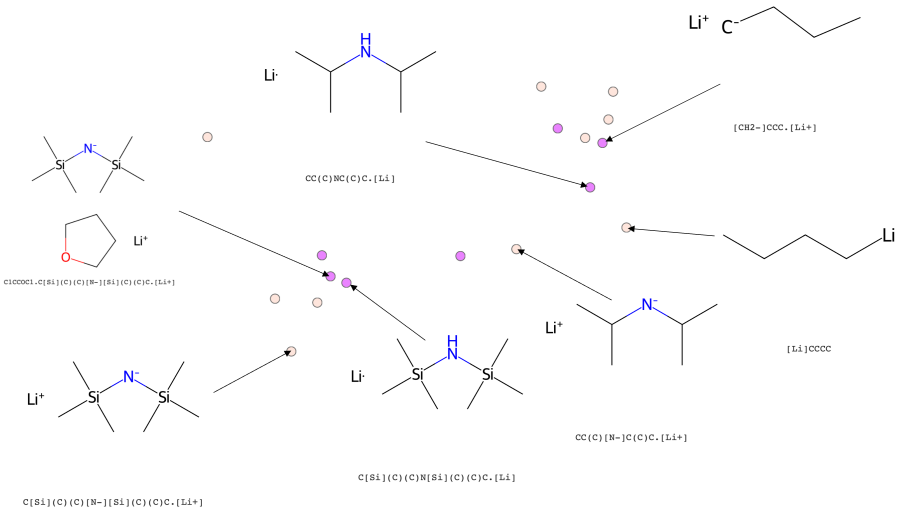

Figure 6.4. A region of the reagent embedding map that reveals unique reagents represented by several different SMILES. In this region of strong bases, the SMILES representations of n-butyllithium, lithium diisopropylamide, and lithium bis(trimethylsilyl)amide require standardization.

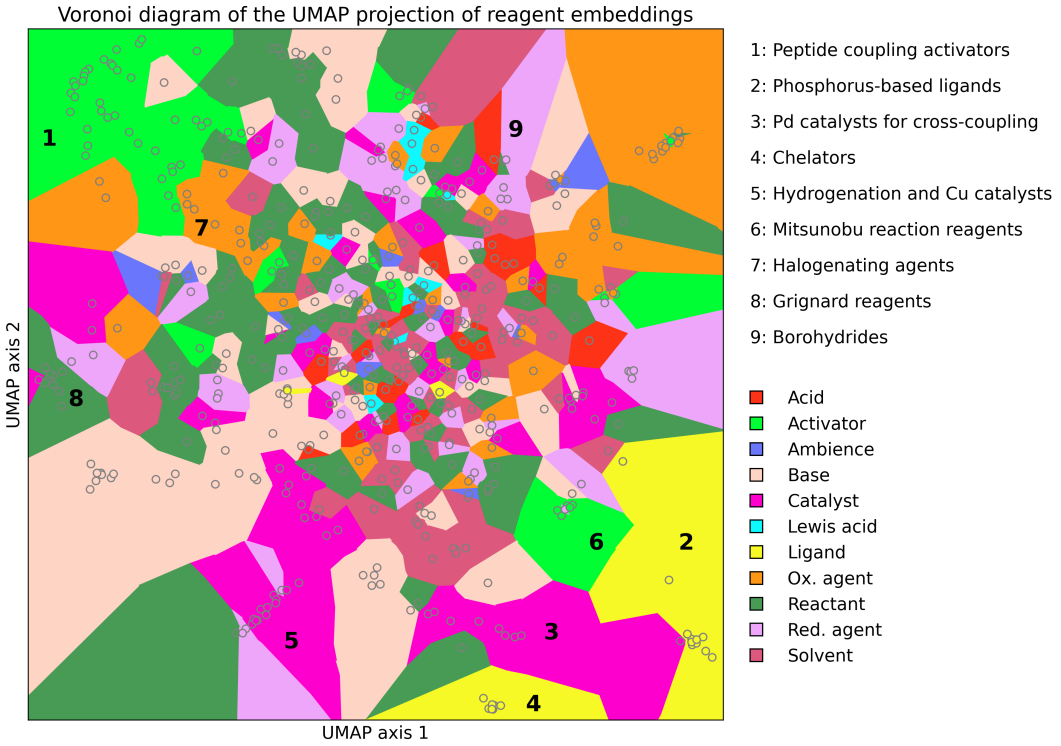

Figure 6.5. Voronoi diagram of the UMAP projection of reagent embeddings. Reagents of the same role tend to form contiguous regions corresponding to the same type of reagent action. Numbers highlight 9 example regions that unite reagents of the same purpose.

the regions are called Voronoi cells. In every cell, the points of the plane are closer to the seed forming the cell than to any other seed. In our case, UMAP projections of reagent embeddings are seeds. The cells formed by reagent embedding projections are colored by the roles of the corresponding reagents, and the touching cells of the same color merge. A Voronoi diagram makes it easy to see regions formed by reagents of the same purpose. Figure 6.5 demonstrates the Voronoi diagram for our reagent map.

We highlight nine example regions in the diagram in Fig. 6.5. The region labeled with 1 comprises various reagents enabling peptide coupling. Among them are HOBt, HOAt, DCC, and their alternatives, many of which are described in the corresponding review [El-Faham and Albericio, 2011]. Region 2 unites organophosphorus ligands for homogeneous metal catalysts, e.g., 1,3-bis(diphenylphosphino)propane (dppp), CyJohnPhos, and BrettPhos. Region 3 is defined by homogeneous palladium catalysts, such as $Pd_2(dba)_3$ or $Pd(PPh_3)_4$. Region 4 is the region of chelating agents, e.g., 8-hydroxyquinoline or phenanthroline. Region 5 features two clusters: one clus-

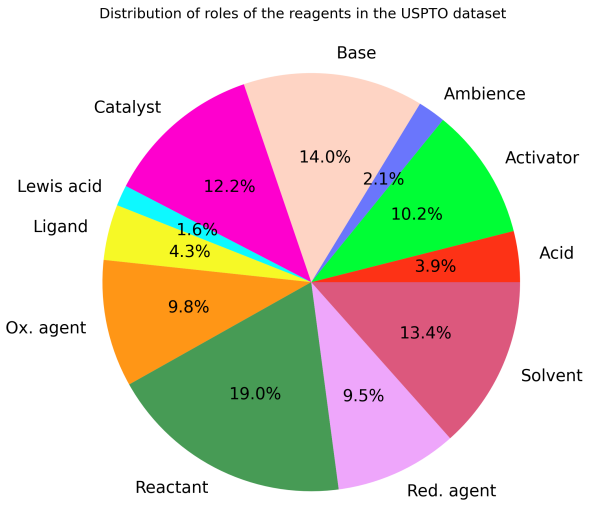

Figure 6.6. Distribution of 559 most common USPTO reagents by detailed roles.

ter is defined by hydrogenation catalysts, such as palladium on carbon and other 10th group metals; the other cluster is formed by catalytic compounds of Cu^{I} and Cu^{II}. Region 6 unites the reagents for Mitsunobu reactions, namely diethyl azodicarboxylate (DEAD) and structurally similar TMAD, DIAD, and DtBAD. Region 7 comprises chlorinating agents such as $SOCl_2$, PCl_5, or cyanuric chloride. Region 8 is the region of Grignard reagents. Region 9 features borohydrides that serve as reducing agents, e.g., $NaBH_4$ or $NaBH(Ac)_3$. These nine regions are only a few examples, and the map contains many more regions that unite reagents of the same action.

Figure 6.6 demonstrates the distribution of reagent roles among our subset of 559 reagents. A valuable insight is that nineteen percent of the reagents are actually reactants. It hints that the atom mapping tool used in USPTO often fails to resolve the noisy reactions for which this dataset is notorious.

We make the list of USPTO reagents and their roles available alongside our codebase. We hope that researchers will find this information useful for their own work involving USPTO.

6.4 Related work

Published reports on curating reagents in reaction data are scarce; we believe the present work is a pioneer in this field. There are, however, several works focusing on reaction data cleaning from other angles. Rxn INSIGHT [Dobbelaere et al., 2024] presents a tool for deterministic classification of reactions into types, and adds reac-

tion type information to the public USPTO reactions. There is work aimed at enhancing USPTO reactions with their mechanistic steps [Chen, Babazade, Kim, Han and Jung, 2024]. Aside from that, there are multiple works on chemical space visualization aimed at helping with data cleaning [] and atom-atom mapping tools.

6.5 Conclusion

We demonstrated a novel approach to facilitate the curation of chemical reaction data, with a focus on reagents. By counting unique reagents in a reaction dataset, turning the table of their pairwise counts into point-wise mutual information scores, and factorizing that table with singular value decomposition, we effectively apply a word2vec algorithm to reagents and obtain their distributed vector representations that capture reagent co-occurrence statistics. By projecting the obtained reagent representations onto the plane using UMAP, we can construct a reagent space map that reveals intriguing clustering patterns among reagents, highlighting that reagents united by a common purpose tend to lie close together and partition into distinct clusters. Based on this map, we present an interactive web application providing a user-friendly platform for researchers to navigate and explore reagent patterns within reaction datasets. We demonstrate the application's use with the USPTO dataset.

Additionally, we systematized and cataloged several hundred of the most commonly used reagents at the USPTO, categorizing them into detailed roles. We believe that such information will be valuable for reagent prediction models trained on USPTO. For example, it can be used to estimate the performance of a reagent prediction model not by the often interchangeable individual molecules, but by the correctness of predicted roles.

The code and data used in our experiments are available at `https://github.com/Academich/reagent_emb_vis`.

Chapter 7

Outlook

This thesis presents a set of methods we developed to improve the existing machine learning-based CASP systems. We focused on three main issues: the latency of synthesis planning, the prediction of reagents in reactions, and the quality of the available reaction data.

In Chapter 3, we presented a method that accelerates template-free SMILES-to-SMILES models for reaction prediction and single-step retrosynthesis, and studied its performance under various settings. We found that our method, while promising, has scalability limitations with respect to batch size, which may limit its applicability in hypothetical CASP systems that leverage batching in single-step retrosynthesis models. Then, in Chapter 4, we described the improvement of our method that addresses this limitation, and studied the influence of the accelerated single-step retrosynthesis transformer on multistep synthesis planning.

In Chapter 5, we presented a study of the transformer as a reagent prediction model, aiming to develop an accurate reagent prediction model necessary for a comprehensive CASP system. We also noticed serious data quality problems with the standard (and to date, the only) open dataset of organic chemical reactions, including missing reagent records. We then demonstrated how the trained reagent model could improve the quality of the reaction data in a bootstrapped manner.

Finally, in Chapter 6, we described a method of reagent data curation prompted by the challenges we faced when training reagent prediction models. Our simple, word2vec-inspired method for grouping reagents by their roles in reactions, wrapped in a web application, enabled us to detect and address serious data issues in the standard reaction data collection and to prepare for future reagent prediction studies.

The presented work will serve as a foundation for future research aimed at further perfecting CASP systems. Here we provide several directions for this future research.

First, a missing piece in CASP systems, such as AiZynthFinder, is the support for

batched inference of the underlying single-step retrosynthesis model. The existing systems do not leverage the model's ability to process multiple inputs simultaneously. Overcoming this limitation would unlock considerable speed gains in multistep synthesis planning, especially with fast single-step models that scale well with batch size, such as our accelerated SMILES-to-SMILES model presented in Chapter 4.

Second, another missing piece in AiZynthFinder and similar systems is the omission of reagent predictions for the generated reactions. Reagents are integral to chemical reactions and should be identified in all cases to conduct the proposed syntheses in a laboratory. We believe that a single-step retrosynthesis model should, in principle, have an integrated architecture capable of predicting reagents alongside reactants. We hypothesize that this can be achieved through appropriate preparation and formatting of the training data and by decoupling reactant and reagent sampling in the single-step retrosynthesis model.

Finally, we believe that the Molecular Transformer, i.e., a small encoder-decoder SMILES-to-SMILES generator, should serve as a basis for modifications aimed at improving various aspects of single-step retrosynthesis, including generalization with respect to input length. We plan to improve it using inference acceleration techniques complementary to speculative decoding, such as KV-caching, and specialized positional embeddings to improve its generalization with respect to input SMILES length.

Additionally, we plan to extend our reagent curation method from Chapter 6 to other aspects of the reaction data, namely, to reaction templates extracted from atom-mapped reactions. The nontrivial deduplication and classification of automatically extracted reaction templates could lead, for example, to the development of better template-based single-step retrosynthesis models.

Appendix A

Acceleration of the inference of string generation-based chemical reaction models for synthesis planning

A.1 Speculative Beam Search Algorithm

The following algorithm provides an outline of the speculative beam search procedure we implement in our experiments.

```
SpeculativeBeamSearch( S_{B×L_s}, K, N, D ):

 Input Parameters
//  S_{B×L_s} : query token sequence (tensor of integers)
//  K : number of best sequences to maintain
//  N : number of drafts
//  D : number of tokens in a draft
//  B : batch size, L_s : source sequence length
//  L_g : length of output generated so far
//  L_gm : maximum requested output length
//  V : vocabulary size, E : embedding dimension

 Initialization
M_{B×L_s×E} := encoder(S);
G_{B×K×1} := [BOS];
D_{B×N×D} := makeDrafts(S, N, D);

 Iterative Decoding
```

```
repeat
    GD_{BKN×(L_g+D)} := concat(G, D);
    L_{BKN×(L_g+D)×V} := decoder(GD, M);
    D*_{B×K×D} := bestDraft(L);
    T_{B×M×(L_g+D+1)} := seqTree(D*, L, G);
    G_{B×K×(L_g+D+1)} := extractTop(T, K);
until L_g = L_gm;

 Return result
return G;
end
```

The algorithm features the following functions:

MAKEDRAFTS: Produces N subsequences of length D from every query sequence to use as drafts according to our heuristic drafting scheme. The subsequences are extracted by a sliding window with a uniform step size. If N is larger than L_s, the extra sequences are padded with the most frequent non-service token in the vocabulary.

ENCODER: Forward pass of the encoder of the Molecular Transformer. Gives the embeddings of the source sequences $\mathcal{M}$ to be used in the decoder.

DECODER: Forward pass of the decoder of the Molecular Transformer. Gives the logits $\mathcal{L}$.

CONCAT: Concatenates every sequence decoded so far with all its corresponding draft sequences.

BESTDRAFT: Uses the logits predicted by the decoder to select the best draft for every resulting sequence and discard the others. The best draft is the one with the largest number of accepted tokens. The model accepts a token at a given position if that token is among the top three most probable tokens that the model predicts at that position. There can also be fewer than the top three contestants for acceptance if the sum of their probabilities exceeds a threshold, such as in nucleus sampling, e.g., 99.75%. For example, if the predicted probability of one of the tokens reaches 100%, as it often happens in our SMILES-to-SMILES experiments, that token will be the only candidate for acceptance.

SEQTREE: The core of the SBS algorithm. Proposes many candidate sequences for the decoding step based on the single accepted draft. The candidate sequences may have different lengths. The candidate sequences are organized in a tree, where every path from the root to a leaf represents a candidate sequence continuation. Figure A.1 demonstrates an example. M is the largest number of candidate continuations in the batch.

EXTRACTTOP: Orders the candidate sequences by probabilities, keeps the K ones

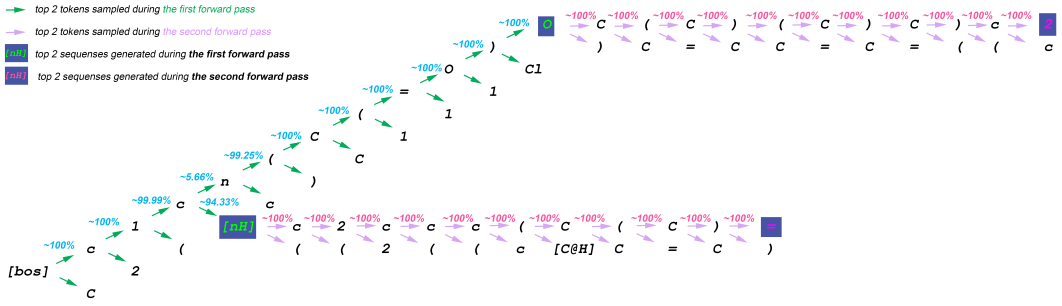

Figure A.1. An example of the first two iterations of the sampling of candidate sequence trees in our speculative beam search with two generated sequences maintained for one query sequence in the batch. Here, we select the two best candidates at each iteration. The draft length D in this example is 10. The best draft for the first iteration is **c1cn(C(=O)**. It gets concatenated with the BOS token. The first forward pass generates 12 candidate sequences. The best sequences in the first iteration are **c1c[nH]** and **c1cn(C(=O)O**, and they become the "beams". In the second iteration, the best draft for the first "beam" is **c2ccc(C(C)**, and for the second one it is **C(C)(C)C)c**. The second forward pass generates 24 sequences overall, which all get sorted by their probabilities. The most probable sequences after the second iteration are **c1c[nH]c2ccc(C(C)=** and **c1cn(C(=O)OC(C)(C)C)c2**, and they become the generated sequences for the next iteration. In this example, after two iterations SBS generates two sequences of lengths 15 and 22, respectively, whereas the standard beam search would have generated two sequences of length 2.

with the highest probabilities, and discards the others.

A.2 Re-implementation of the Molecular Transformer

To demonstrate our method, we chose the Molecular Transformer (MT) [Schwaller et al., 2019] model. MT is an encoder-decoder transformer model for SMILES-conditioned SMILES generation. The original implementation of MT relies on adopting OpenNMT [Klein et al., 2018], a general framework for neural machine translation. We found the code in this framework to be complex and difficult to customize, and therefore decided to reimplement the MT model in PyTorch Lightning, retaining only the necessary code and gaining more design freedom in implementing the model's inference procedure. We decided not to compare decoding speed with the original MT because its under-

Table A.1. The top-5 accuracy of the Molecular Transformer (MT) in predicting reaction products on USPTO MIT (mixed) in the original report and our reimplementation. Both models use beam search with beam size 5 to generate five predictions for every query. Our model successfully reproduces the accuracy of the original model with negligible discrepancy.

Accuracy	Original MT	Our MT	Δ
Top-1, %	88.6	88.4	-0.2
Top-2, %	92.4	92.5	0.1
Top-3, %	93.5	93.7	0.2
Top-5, %	94.5	94.7	0.2

lying OpenNMT inference implementation is optimized with various techniques that we do not use for conceptual simplicity. Those techniques are independent of speculative decoding yet compatible with it, and the OpenNMT implementation would benefit from speculative decoding as much as the implementation in our demonstration.

Our implementation of the Molecular Transformer successfully reproduces the accuracy scores of the original MT [Schwaller et al., 2019] that relies on OpenNMT. Comparing our MT and the original MT, we observe at most 0.2 percentage points discrepancy of top-1 to top-5 accuracy in product prediction with beam search (Table A.1). From that, we conclude that our re-implementation is correct.

A.3 Throughput-latency trade-off in speculative decoding

Speculative beam search significantly increases the effective batch size of the transformer input, leading to the manifestation of the throughput-latency trade-off: the speed advantage of batched processing may become less pronounced as the latency of an individual forward pass becomes larger with the growing input size. Therefore, we should determine the optimal combination of the number of parallel drafts and draft length to maximize the benefits of speculative decoding. We use grid search to select the hyperparameters that demonstrate the best processing speed on the test set (Fig. A.2).

In product prediction with greedy speculative decoding, the best combination is 23 drafts of 17 tokens when batch size is 1, 15 drafts of 14 tokens when batch size is 4, 7 drafts of 7 tokens when batch size is 16, and 3 drafts of 5 tokens when batch size is 32. In product prediction with SBS and 5 generated sequences per input sequence, the best

hyperparameters are 23 drafts of 10 tokens with batch size 1, 10 drafts of 14 tokens with batch size 2, 10 drafts of 9 tokens with batch size 3, and 7 drafts of 10 tokens with batch size 4. In single-step retrosynthesis with SBS, the batch size of 1, and varying number of generated sequences per input, the optimal choices are 15 drafts of length 11 at 5 sequences, 10 drafts of length 10 for both 10 and 15 sequences, and 5 drafts of length 14 for 20 sequences. In single-step retrosynthesis and SBS with 10 sequences per input and varying batch size, the best choices are 9 drafts of 14 tokens with batch size 1, 5 drafts of 12 tokens with batch size 2, 3 drafts of 10 tokens with batch size 3, and 2 drafts of 11 tokens with batch size 4.

Overall, as the batch size or the number of generated sequences per input increases, the number of drafts and their lengths should be reduced to maximize the benefits of speculative decoding.

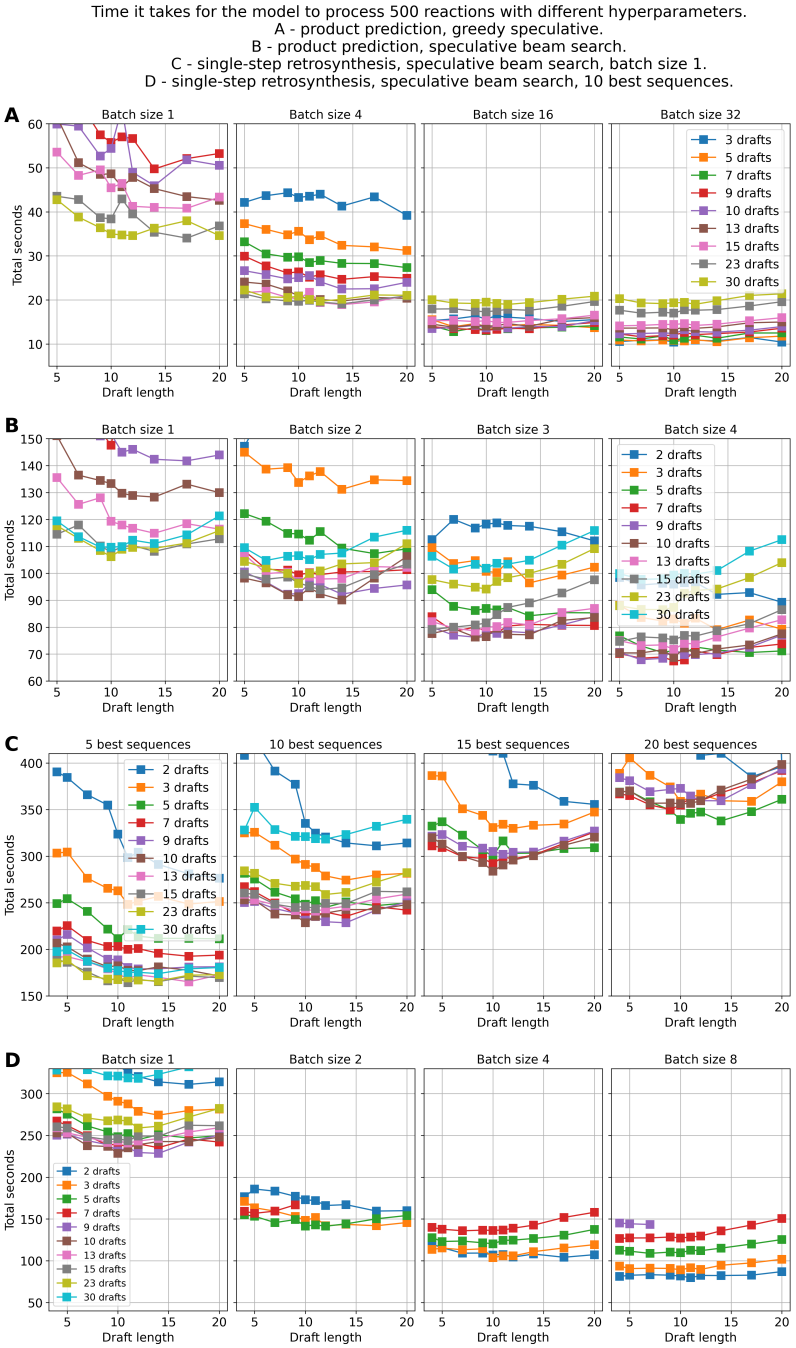

Figure A.2. Results of the grid search over the draft length and the number of drafts: the model needs a different time to give predictions for 500 queries from the full test dataset with different combinations of hyperparameters in reaction prediction and single-step retrosynthesis. We ran the grid search once without estimating the spread of processing time.

Appendix B

Fast multi-step synthesis planning with a Medusa neural network and speculative beam search

B.1 Training details

Our Medusa model is a custom variant of the Molecular Transformer [Schwaller et al., 2019], an encoder-decoder transformer for SMILES-to-SMILES translation. Our model consists of six encoder and decoder layers, eight heads in the multi-head attention mechanism, an embedding dimensionality of 256, a feedforward dimensionality of 2048, and 20 additional Medusa heads to predict tokens at 1-20 positions ahead of the next token. All Medusa heads are implemented as a single MLP with one hidden layer with dimensionality $20 \times 50 = 1000$, followed by a residual connection [Hochreiter, 1991] and layer normalization. The base transformer has 17.4 million parameters, and the Medusa heads have 1.3 million, for a total of 18.7 million. We re-use the basic models described in Chapter 3 (see Section A.2).

We use the Adam [Kingma and Ba, 2015] optimizer for training. We train and test our model on one NVIDIA Tesla V100 GPU with 32GB of memory.

B.2 Comparison of the predicted distributions in Medusa and Transformer

Transformer tends to concentrate probability mass on the top candidate, while Medusa produces more smooth distributions across candidates, leading to more exploratory search behavior. Figure B.1 shows that the top-1 probability of Medusa is lower than

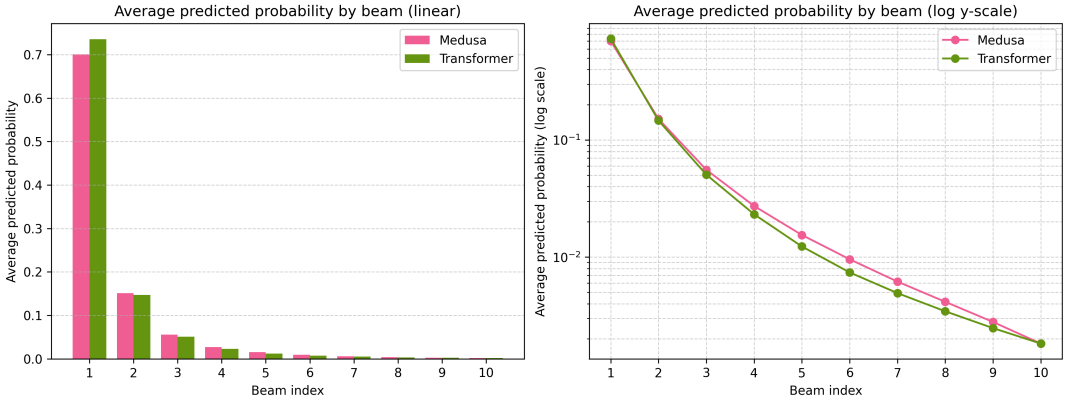

Figure B.1. Average predicted probabilities of the top-10 sequences generated by Transformer (TBSO) and Medusa (MSBS) for every input in the USPTO 50K test set. The probabilities of Medusa outputs are slightly smoother than those of the Transformer.

that of the Transformer. The effect is small, but it is noticeable.

Appendix C

Reagent prediction with a molecular transformer and the improvement of reaction data quality

C.1 USPTO data quality

We do not use any USPTO subsets as a test dataset for the reagent prediction model. The problem with the USPTO dataset is that it is assembled using text mining, so the reactions in it are recorded with significant noise. For example, different instances of the same reaction type may be written with different amounts of detail. Fig. C.1A shows examples of Suzuki coupling with the year and patent number. This reaction type accounts for a significant portion of USPTO reactions [Thakkar et al., 2020]. The Suzuki coupling generally requires a palladium catalyst, a base, and a suitable solvent. However, in many reactions of this type in USPTO the necessary reagents are not specified. This is also observed for other types of reactions. In addition, USPTO may contain reaction SMILES involving nonsensical molecules (Fig. C.1B).

If such noisy reactions end up in the test dataset, it will prevent us from correctly evaluating the performance of the reagent model. Our preliminary experiments with the reagent model on USPTO data showed that the model's predictions often do not match the ground-truth sequence, although the former may be more appropriate than the latter. To overcome this problem, we assembled our test set from the reactions in the Reaxys database. Since Reaxys is comprised of reactions manually extracted from scientific papers, we assume that the quality of reagent information there is ensured by human experts.

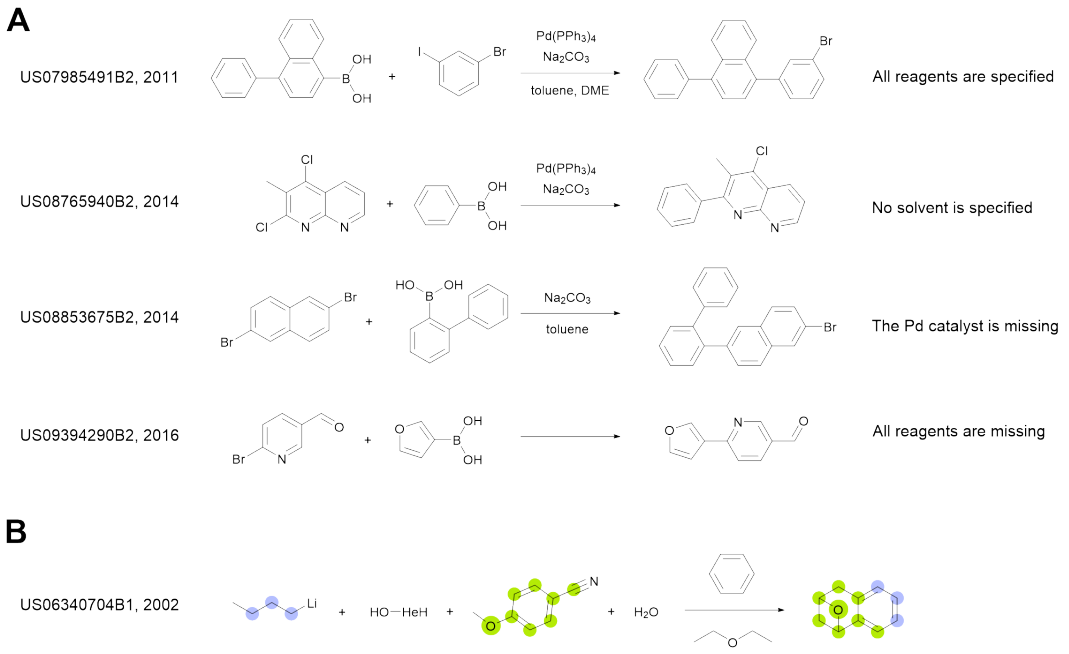

Figure C.1. A: Instances of Suzuki-Miyaura coupling present in the USPTO data and written with a different amount of detail. Generally, this type of reaction requires a palladium catalyst, a base, and a solvent. Any of those species may be missing in the examples of the Suzuki-Myuara reaction found within USPTO. B: An example of a nonsensical entry in the USPTO data. Colored circles represent the original USPTO atom mapping for this reaction. To the left are the number of patents and publication years.

C.2 Similarity between the Reaxys test set and USPTO 50K

While gathering the Reaxys subset, we aimed to make it resemble the USPTO 50K dataset [Schneider et al., 2016] in its distribution of reaction types. The purpose of such a design is to make the test distribution as close as possible to the training distribution. We use USPTO 50K as a proxy for the training set because it is the only open subset of USPTO that contains reaction class labels. The subset of Reaxys used for testing comprises 96,729 reactions across 10 broad classes. Their distribution is displayed in Table C.1. We aimed at making the class proportions in USPTO 50K and the Reaxys test set similar. The classes were determined using NameRXN software [*NameRXN*].

Table C.1. The proportion of reactions belonging to ten broad reaction classes both in USPTO 50K and the Reaxys test set used to test reagent prediction models

Reaction class proportion	USPTO 50K (%)	Reaxys test (%)
Heteroatom alkylation and arylation	28.7	28.1
Acylation and related processes	24.3	18.2
Deprotections	17.0	17.4
C-C bond formation	11.5	12.0
Reductions	9.7	11.1
Protections	1.4	4.5
Functional group interconversion (FGI)	3.9	4.1
Functional group addition (FGA)	0.5	2.5
Oxidations	1.7	2.1
Heterocycle formation	1.3	0.1

C.3 Preprocessing of the training set

Our preprocessing pipeline for reagent prediction is as follows: for each reaction, we first remove all the auxiliary information written in the form of ChemAxon extended SMILES (CXSMILES). Then, we mix together all the molecules that are not products. The procedure for extracting the original USPTO reaction data from patents included detecting catalysts and solvents and placing them in the reagent section. However, we do not place our trust in it, as it is quite imprecise and does not account for other possible reagent types. Therefore, we separate reactants from reagents according to the fingerprint-based algorithm described by Schneider et al. [Schneider et al., 2016] and implemented in RDKit. A small number of all reactions, in this case, end up with no reactants whatsoever. For them, we decide on the separation based on the original atom mapping in USPTO: reactants are molecules with atom mapping labels that also appear in the products. We avoid using this approach for all reactions as the default atom mapping in USPTO is not reliable enough. Then, we canonicalize the SMILES of all molecules in a reaction, drop atom mapping, and remove any isotope information. Finally, we order all molecules in the reaction: the molecules with the longest SMILES strings come first; strings of the same length are ordered alphabetically. We also tried implementing a step to remove all duplicate molecules in the reaction. This would make the reactions unbalanced, but the reaction SMILES would become shorter while preserving the chemical context. However, this step proved unhelpful, and we eventually did not include it in the preprocessing pipeline.

After processing every reaction as described, we remove rare reagents from the data, i.e., those that appear in the training data fewer than 20 times. This lowers the

number of unique reagents in the training set from 37602 (from which 26431 were encountered only once) to 1314. The model is unlikely to learn to predict a reagent from so few examples, even if they are correct. However, the visual analysis shows that such rare reagents are, in fact, rather reactants in not well-specified reactions. For example, if a reaction involves several isomers of a single reactant, only one of which is reported to form a product, the other isomers are treated as reagents. By removing rare reagents, we alleviate this problem. Finally, we drop duplicate reactions and those in which a product appears among the reactants or reagents. In accordance with the common procedure of data augmentation in reaction modeling, we employ SMILES augmentation as implemented in the PySMILESutils Python package [Bjerrum et al., 2021]. Only the SMILES of reactants and products are augmented in reagent prediction. In addition to that, we use "role augmentations": some molecules from the side of the reagent have a chance to move to the reactants in an augmented example.

All molecules in the target sequences are canonicalized and ordered by their detailed roles: catalysts come first, then redox agents, then acids and bases, then any other molecules and ions, but solvents come last. In this case, we utilize the model's autoregressive nature to predict the most important reagents first based solely on the input, and then the more interchangeable reagents based both on the input and the reagents suggested so far. A similar ordering of reagents by role was used by Gao et al. [Gao et al., 2018], and it generally follows the line of thought of a chemist who would suggest reagents for a reaction based on their experience. The roles of the molecules are determined using the following heuristics:

1. Every molecule in a SMILES string of reagents is assigned a role in the following order of decreasing priority: solvent, catalyst, oxidizing agent, reducing agent, acid, base, unspecified role.

2. A molecule is a solvent if it is one of the standard 46 solvent molecules, like THF, hexane, benzene, etc.

3. A molecule is a catalyst if

   (a) It is a free metal.

   (b) It contains a cycle together with a metal or phosphorus atom.

   (c) It is a metal halide.

4. A molecule is an oxidizing agent if

   (a) It contains a peroxide group.

   (b) It contains at least two oxygen atoms and a transition metal or iodine atom.

(c) It is a standard oxidizing agent like free halogens.

(d) It is a standard halogenating agent.

(e) It contains both a positively charged atom and a negatively charged oxygen but is not a nitrate anion.

5. A molecule is a reducing agent if it is one of the standard reducing agents or some hydride of boron, silicon or aluminum.

6. A molecule is an acid if

   (a) It is a derivative of sulphuric, sulfamic or phosphoric acid with the acidic -OH group intact.

   (b) It is a carboxylic acid.

   (c) It is a hydrohalic acid or a common Lewis acid like the aluminium chloride.

7. A molecule is a base if

   (a) It is a tertiary or secondary amine.

   (b) It contains a negatively charged oxygen atom and consists of only C, O, S, and P atoms.

   (c) It consists only of lithium and carbon.

   (d) It is the hydride ion or the hydroxide ion.

To tokenize our sequences, we employ the standard atomwise tokenization scheme for Molecular Transformer [Schwaller et al., 2019]. Additionally, we experimented with the scheme in which entire molecules get their own tokens, namely, all solvents and some common reagents. However, this does not seem to improve the quality of a trained reagent prediction model, so we resort to standard atomwise tokenization in our final model.

For product prediction, we follow the Molecular Transformer procedure. The tokenization is atomwise as well. Some of the current deep learning reaction prediction methods use explicit reagent information to make predictions [Schwaller et al., 2019; Qian et al., 2020; Bi et al., 2021], and some allow mixing all the precursors together, but the separation of reactant and reagent information improves the performance of such models [Schwaller et al., 2019; Sacha et al., 2020; Do et al., 2019]. We trained product prediction models in both the separated setting (reactants and reagents are separated by the token ”>” in the input sequences) and the mixed setting (all molecules are separated by dots in the input sequences). To evaluate the quality of the models, we used both the USPTO MIT test set and our Reaxys test set on which we tested the

reagent prediction model. We did not use SMILES augmentations for product prediction.

C.4 Preprocessing of the test set

We use a subset of Reaxys data for testing purposes. To obtain it, we use the Reaxys web interface. In Reaxys, reaction SMILES do not include reagents; they are enumerated separately by IUPAC name or common name, separated by semicolons. We employ the PubChemPy$^{\alpha}$ package to retrieve SMILES of reagents from PubChem. We drop reactions in which reagents were absent, or their SMILES could not be successfully retrieved. After constructing full reaction SMILES for all reactions, we canonicalize all molecules and order them as we did for the training set. After preprocessing, every reaction in our test set has non-empty SMILES of reactants, reagents, and products. The final test set comprises 96729 reactions.

C.5 Training details

For both reagent and product prediction, we used the same transformer settings and hyperparameters used by Schwaller et al. [Schwaller et al., 2019]: Adam optimizer [Kingma and Ba, 2015], noam learning rate schedule [Vaswani et al., 2017] with 8000 warmup steps, a batch size of around 4096 tokens, accumulation count 4, and dropout rate 0.1. We did not conduct weight averaging across checkpoints. All the models were trained on an Nvidia GeForce GTX TITAN X GPU with 12 GB of memory.

C.6 Reagent prediction performance details

The performance of the reagent prediction model across reaction classes and reagent roles in the Reaxys test set is given in Fig. C.2.

In each reaction class, only a few reagent roles are usually represented. For example, oxidizers rarely appear in C-C bond formation, and catalysts are not very frequent in FGA. At the same time, solvents are usually listed for any class of reactions. The table in the middle of Fig. C.2 takes into account only the performance of the model in correctly predicting the presence of reagents, not the general presence and absence. It can be seen that the performance is pretty low when predicting the presence of rare reagent roles for any type of reaction.

[0α] *https://github.com/mcs07/PubChemPy*

Table C.2. A contingency table is needed to perform McNemar's test. Each entry contains the number of test examples in T for which either F_1 or F_2 give correct or incorrect binary predictions.

	F_1 incorrect	F_1 correct
F_2 incorrect	A	B
F_2 correct	C	D

C.7 Statistical tests

To demonstrate that the improvement in the model trained on the new data relative to the baseline Molecular Transformer is statistically significant, we used McNemar's test. McNemar's test is usually applied to test if one binary classifier performs better than another. If one has some test set T and two classifiers F_1 and F_2, then one can build a 2×2 contingency table (Table C.2):

The McNemar's test statistic is calculated as in Eq. C.1:

$$x = \frac{(|B - C| - 1)^2}{B + C} \tag{C.1}$$

Under the null hypothesis, this statistic has a χ^2-distribution with one degree of freedom if both B and C are large enough, i.e. one hundred and more. The null hypothesis is that the performance of the models F_1 and F_2 is in fact the same and any apparent difference is accidental. Therefore, we can reject the null hypothesis with p-value > 0.05 if x exceeds ~ 3.83, and with p-value > 0.01 if x exceeds ~ 6.6.

In our experiments, we treat the product prediction models like binary classifiers, which are either correct if the generated SMILES sequence of the product is correct, or incorrect otherwise. We denote the baseline Molecular Transformer as "MT base" and the Molecular Transformer trained on the data with altered reagents as "MT new". They are compared both in mixed and separated settings on both USPTO MIT and the Reaxys test set. The corresponding contingency tables are tables C.3-C.6.

The McNemar's test statistic for table C.3 is equal to 15.6.

The McNemar's test statistic for table C.4 is equal to 114.2.

The McNemar's test statistic for table C.5 is equal to 11.3.

The McNemar's test statistic for table C.6 is equal to 22.3.

One can see that for all four tables the value of McNemar's statistic is sufficiently high to reject the null hypothesis with p-values less than 0.01. This allows concluding that the improvement in product prediction performance when the models are trained

Table C.3. The contingency table for the product models tested on the Reaxys test set in the separated setting.

Reaxys test	MT new, separated, fail	MT new, separated, correct
MT base, separated, fail	11751	3444
MT base, separated, correct	3123	78411

Table C.4. The contingency table for the product models tested on the Reaxys test set in the mixed setting.

Reaxys test	MT new, mixed, fail	MT new, mixed, correct
MT base, mixed, fail	12768	4623
MT base, mixed, correct	3650	75688

Table C.5. The contingency table for the product models tested on USPTO MIT in the separated setting.

USPTO MIT	MT new, separated, fail	MT new, separated, correct
MT base, separated, fail	3086	1230
MT base, separated, correct	1068	34616

Table C.6. The contingency table for the product models tested on USPTO MIT in the mixed setting.

USPTO MIT	MT new, mixed, fail	MT new, mixed, correct
MT base, mixed, fail	3448	1460
MT base, mixed, correct	1215	33877

on the data with altered reagents is statistically significant.

Figure C.2. Comparison of the reagent prediction top-1 exact match accuracy across all reaction classes and reagent roles. At the top, the overall comparison of the test set is given. In the middle, for nonempty ground truth only. At the bottom, the table shows the number of nonempty ground-truth strings for each reagent role and reaction type in the Reaxys test set.

Appendix D

Curation of reagents in chemical reaction data using an interactive reagent space map

D.1 Data processing

For our experiments, we used reagents extracted from the full USPTO reaction dataset, which we downloaded in the SMILES format using the `rxnutils` Python package [Kannas and Genheden, 2022].

The initial raw dataset contained 3,748,191 reactions. We first canonicalized all SMILES and removed reactions containing molecules that could not be canonicalized. Then, we stripped stereochemical information from all molecules, eliminated duplicate reactions, and retained only those reactions with fewer than ten precursors (reactants and reagents combined), reducing the dataset to 1,393,677 reactions. In this filtered dataset, we identified the reagents in every reaction using the available atom–atom mapping (AAM) information, and merged the disjoint ionic fragments into single reagent species based on the available CXSMILES metadata. After identifying reagents, we applied more filtering steps. We removed bare ions, species with unbalanced charges, and bound water in hydrates (e.g., $Na_2SO_4 \cdot 10H_2O$). We then excluded reactions containing more than five reagents or no reagents at all, as well as trivial records in which a product appeared among the reactants or reagents. The resulting dataset contained 1,128,297 reactions featuring at least one reagent.

In the remaining reactions, we removed reactants and products, separated the SMILES of reagents by semicolons, and counted unique reagents, identifying 40,556 distinct molecular species. Among all unique reagents in the resulting file, two-thirds (27,100) occurred only once. We removed them, as their attribution to reagents was

mostly the result of erroneous atom-atom mapping. Furthermore, for demonstration purposes, we limited our study to reagents that occur at least 100 times in the final file, as our method works best with reagents that occur frequently in the data and ideally co-occur with various other reagent species. The final filtering step resulted in 626 unique reagent SMILES.

D.2 Reagent roles

Reagent role information in the USPTO is rather limited, only differentiating between catalysts, solvents, and all other substances. To achieve a finer attribution of reagent roles, we decided to categorize reagents into the following eleven roles:

Acids

Acidic compounds that are typically used as catalysts, e.g., HCl, H_2SO_4.

Bases

Basic compounds that are typically used as catalysts, e.g., NaOH, n-butyllithium, or Hünig's base.

Lewis Acids

Catalysts that are Lewis acids, e.g., $AlCl_3$.

Catalysts

Other catalysts, most of which are metal-based. For example, those would comprise homogeneous palladium-based catalysts for cross-coupling reactions, such as Suzuki coupling, or heterogeneous catalysts for hydrogenation, such as metallic nickel.

Ligands

Compounds with the purpose of forming coordination complexes with metal ions, e.g., phosphorus-based ligands for homogeneous palladium catalysts or chelating agents.

Oxidizing Agents

Various oxidizers, including halogenating agents. For example, $KMnO_4$, CrO_3, $SOCl_2$.

Reducing Agents

Various reducing agents, e.g H_2, $SnCl_2$, $LiAlH_4$.

Activators

Reagents that facilitate an overall reaction but are consumed in the process. For example, we refer to activators as the agents that facilitate the formation of active intermediates, which enable a reaction, such as active esters in peptide coupling reactions. A comprehensive review by El-Faham and Albericio [El-Faham and Albericio, 2011] systematizes a number of such reagents, many of which are present in USPTO. Examples of activators include 1-hydroxybenzotriazole and N,N'-dicyclohexylcarbodiimide for peptide coupling, and diethyl azodicarboxylate for the Mitsunobu reaction.

Ambience

Other reagents that do not fit any described role and serve some auxiliary purpose, such as radical reaction inhibitors or nitrogen for an inert atmosphere.

Reactants

Molecules that, in fact, contribute atoms to the product and are mistakenly classified as reagents.

D.3 UMAP details

We use the default parameters of the UMAP class (15 neighbors, Euclidean metric) in the UMAP Python package [McInnes et al., 2018] in the web application.

D.4 Reagent occurrence frequency

Fig. D.1 displays the logarithm of the number of occurrences for every reagent. Around 50 most common reagents dominate the dataset. However, $\sim$ 35 percent of all reactions use reagents other than those 50 most common ones. The less common reagents form a fat tail of the occurrence frequency distribution: the relative occurrence frequency of reagents starting from the 100th falls like $\frac{1}{x^2}$.

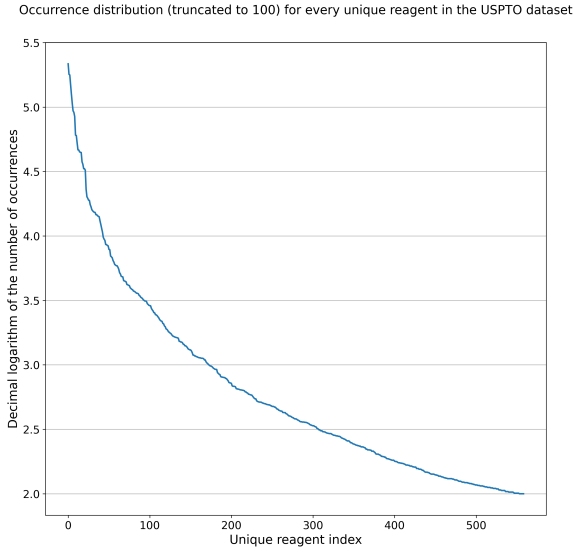

Figure D.1. Decimal logarithm of the number of occurrences of unique reagents in the USPTO dataset. We consider only the reagents that occur at least 100 times. The dataset is dominated by approximately 50 most common reagents, and the others are relatively rare. Unique reagent indices are sorted by occurrence frequency.

Appendix E

Summary of methods in AI-based CASP

Table E.1. Summary of methods in machine learning-based reaction prediction methods.

Advance	Type	Year	Author
Mechanistic approach: prediction at the level of elementary mechanistic steps for explainability	Mechanistic	2012	Kayala and Baldi
Template-based MLP classifier using molecular fingerprints on a small manually curated dataset	Template-based	2016	Wei et al.
Template-based MLP and highway network scaled to 3.5M reactions from Reaxys; applied to both reaction prediction and retrosynthesis	Template-based	2017	Segler and Waller
Template enumeration with neural ranking of candidate products using a dense network	Template-based	2017	Coley, Barzilay, Jaakkola, Green and Jensen
SMILES-to-SMILES Seq2Seq LSTM for reaction prediction	Template-free (seq)	2018	Schwaller et al.

Continued on next page

Table E.1 — continued

Advance	Type	Year	Author
Graph-edit-based method using a gated GNN to iteratively add or remove bonds in the reactant graph	Template-free (graph)	2018	Bradshaw et al.
Graph convolutional WLDN scoring bond changes for all atom pairs, producing a ranked list of candidate products	Template-free (graph)	2019	Coley, Jin, Rogers, Jamison, Jaakkola, Green, Barzilay and Jensen
Molecular Transformer: encoder-decoder transformer achieving state-of-the-art accuracy; established template-free SMILES translation as the dominant paradigm	Template-free (seq)	2019	Schwaller et al.
MEGAN: encoder-decoder graph attention network supporting both reaction prediction and retrosynthesis via graph editing	Template-free (graph)	2021	Sacha et al.
Non-autoregressive electron redistribution: predicts the full bond-change matrix in one forward pass, enabling mass-balanced and fast predictions	Template-free (graph)	2021	Bi et al.
Augmentation of atom embeddings with quantum-chemical descriptors to improve bond-change scoring	Template-free (graph)	2022	Stuyver and Coley

Table E.2. Summary of methods in machine learning-based single-step retrosynthesis prediction.

Advance	Type	Year	Author
RetroSim: similarity-based non-ML baseline combining product and reactant similarity for template retrieval and ranking	Similarity-based	2017	Coley, Rogers, Green and Jensen
MLP/highway network classifier over molecular fingerprints trained on large proprietary data	Template-based	2017	Segler and Waller
SMILES-to-SMILES Seq2Seq LSTM for retrosynthesis; accuracy did not surpass similarity baseline	Template-free (seq)	2017	Liu et al.
Two-stage template classification: coarse reaction group then fine-grained template within group	Template-based	2019	Baylon et al.
First transformer-based retrosynthesis, significantly outperforming Seq2Seq LSTM	Template-free (seq)	2019b	Karpov et al.
SCROP: transformer with a separate correction module for fixing invalid generated SMILES	Template-free (seq)	2019	Zheng et al.
GLN: joint probabilistic modeling of templates and reactants using logic variables	Template-based	2019	Dai et al.
G2Gs: first semi-template method; GCN reaction center identification followed by conditional VAE synthon completion	Semi-template	2020	Shi et al.

Continued on next page

Table E.2 — continued

Advance	Type	Year	Author
RetroXpert: edge-enhanced GAT supporting multi-bond breaking; transformer-based synthon completion	Semi-template	2020	Yan et al.
Multitask training and SMILES augmentation boost transformer accuracy substantially	Template-free (seq)	2020a	Tetko et al.
GraphRetro: dual-graph message passing for richer atom/bond representations; synthon completion as leaving-group classification	Semi-template	2020	Somnath et al.
RetroPrime: separate SMILES transformers for both synthon generation and synthon completion stages	Semi-template	2021	Wang et al.
Dual tied decoders for simultaneous forward and backward verification of generated reactants	Template-free (seq)	2021	Kim et al.
LocalRetro: GNN classifying every atom and bond into a local reaction template, achieving state-of-the-art template-based accuracy	Template-based	2021	Chen and Jung
MEGAN: autoregressive graph-to-graph editing supporting both reaction prediction and retrosynthesis	Template-free (graph)	2021	Sacha et al.
Chemformer: larger model with BART-style masked pretraining to improve generalisation	Template-free (seq)	2022	Irwin et al.

Continued on next page

Table E.2 — continued

Advance	Type	Year	Author
Retroformer: local attention heads constrained by molecular connectivity for better molecular alignment	Template-free (seq)	2022	Wan et al.
RetroComposer: RNN composing novel templates beyond a fixed library to overcome coverage limitations	Template-based	2022	Yan et al.
Modern Hopfield Network encoder for zero-shot generalisation to templates unseen during training	Template-based	2022	Seidl et al.
Retro-TRAE: generates molecular fingerprint indices and retrieves reactants from a large database	Template-free (seq)	2022	Ucak et al.
Graph2SMILES: graph encoder replacing text encoder in SMILES generation for both tasks	Template-free (graph)	2022	Tu and Coley
RSMILES: root-aligned SMILES augmentation minimising edit distance between product and reactant strings; dramatic accuracy gain from data preprocessing alone	Template-free (seq)	2022	Zhong et al.
G2GT: Graphormer encoder with sequential node-and-edge graph generation	Template-free (graph)	2023	Lin et al.
RetroExplainer: transformer for center identification with contrastive learning for synthon completion	Semi-template	2023	Wang, Pang, Wang, Jin, Zhang, Zeng, Su, Zou and Wei

Continued on next page

Table E.2 — continued

Advance	Type	Year	Author
Graph2Edits: template-guided basis set of graph edits, combining graph-edit and template ideas	Template-free (graph)	2023	Zhong et al.
G^{2}Retro: fine-grained three-type reaction center identification; substructure-vocabulary synthon completion	Semi-template	2023	Chen et al.
RetroKNN: KNN classifier over GNN embeddings ensembled with linear classifier for improved template retrieval	Template-based	2023	Xie et al.
RetroBridge: Markov Bridge model as alternative to diffusion for learning dependencies between molecular graph distributions	Template-free (graph)	2023	Igashov et al.
EditRetro: non-autoregressive in-place SMILES editing transformer	Template-free (seq)	2024	Han et al.
Autoregressive template generation transcending a fixed template collection	Template-free (seq)	2024	Shee et al.
RetroGFN: GFlowNet guided by feasibility model for diverse and novel template composition	Template-based	2025	Gaiński et al.
Retro-MTGR: contrastive learning pulling molecule and synthon embeddings together	Semi-template	2025	Zhao et al.
GDiffRetro: 3D diffusion model for synthon completion incorporating geometric information	Semi-template	2025	Sun et al.

Continued on next page

Table E.2 — continued

Advance	Type	Year	Author
DiffER: categorical diffusion for SMILES-to-SMILES transformation	Template-free (seq)	2025	Current et al.
TempRe: template-generative transformer with improved generalisation to longer input SMILES	Template-free (seq)	2025	Xuan-Vu et al.

Bibliography

[1] Afonina, V. A., Mazitov, D. A., Nurmukhametova, A., Shevelev, M. D., Khasanova, D. A., Nugmanov, R. I., Burilov, V. A., Madzhidov, T. I. and Varnek, A. [2022]. Prediction of optimal conditions of hydrogenation reaction using the likelihood ranking approach, *International Journal of Molecular Sciences* **23**(1).
URL: *https://www.mdpi.com/1422-0067/23/1/248*

[2] Akhmetshin, T., Zankov, D., Gantzer, P., Babadeev, D., Pinigina, A., Madzhidov, T. and Varnek, A. [2025]. Synplanner: An end-to-end tool for synthesis planning, *Journal of Chemical Information and Modeling* **65**(1): 15–21. PMID: 39739735.
URL: *https://doi.org/10.1021/acs.jcim.4c02004*

[3] Alampara, N., Aneesh, A., Ríos-García, M., Mirza, A., Schilling-Wilhelmi, M., Aghajani, A. A., Sun, M., Prastalo, G. and Jablonka, K. M. [2025]. General purpose models for the chemical sciences, *arXiv preprint arXiv:2507.07456* .

[4] Andronov, M., Andronova, N., Wand, M., Schmidhuber, J. and Clevert, D.-A. [2024]. Curating reagents in chemical reaction data with an interactive reagent space map, *International Workshop on AI in Drug Discovery*, Springer, pp. 21–35.

[5] Andronov, M., Andronova, N., Wand, M., Schmidhuber, J. and Clevert, D.-A. [2025a]. Accelerating the inference of string generation-based chemical reaction models for industrial applications, *Journal of Cheminformatics* **17**(1): 31.

[6] Andronov, M., Andronova, N., Wand, M., Schmidhuber, J. and Clevert, D.-A. [2025b]. Fast and scalable retrosynthetic planning with a transformer neural network and speculative beam search, *arXiv preprint arXiv:2508.01459* .

[7] Andronov, M., Voinarovska, V., Andronova, N., Wand, M., Clevert, D.-A. and Schmidhuber, J. [2023]. Reagent prediction with a molecular transformer improves reaction data quality, *Chemical Science* **14**(12): 3235–3246.

[8] Angello, N. H., Rathore, V., Beker, W., Wołos, A., Jira, E. R., Roszak, R., Wu, T. C., Schroeder, C. M., Aspuru-Guzik, A., Grzybowski, B. A. and Burke, M. D.

[2022]. Closed-loop optimization of general reaction conditions for heteroaryl suzuki-miyaura coupling, *Science* **378**(6618): 399–405.
URL: *https://www.science.org/doi/abs/10.1126/science.adc8743*

[9] Armstrong, D., Jončev, Z., Guo‡, J. and Schwaller, P. [2025]. Tango*: constrained synthesis planning using chemically informed value functions, *Digital Discovery* **4**: 2570–2578.

[10] Bahdanau, D. [2014]. Neural machine translation by jointly learning to align and translate, *arXiv preprint arXiv:1409.0473* .

[11] Baylon, J. L., Cilfone, N. A., Gulcher, J. R. and Chittenden, T. W. [2019]. Enhancing retrosynthetic reaction prediction with deep learning using multiscale reaction classification, *Journal of chemical information and modeling* **59**(2): 673–688.

[12] Beck, A. G., Muhoberac, M., Randolph, C. E., Beveridge, C. H., Wijewardhane, P. R., Kenttamaa, H. I. and Chopra, G. [2024]. Recent developments in machine learning for mass spectrometry, *ACS Measurement Science Au* **4**(3): 233–246.

[13] Bhattacharya, D., Cassady, H. J., Hickner, M. A. and Reinhart, W. F. [2024]. Large language models as molecular design engines, *Journal of Chemical Information and Modeling* **64**(18): 7086–7096.

[14] Bi, H., Wang, H., Shi, C., Coley, C., Tang, J. and Guo, H. [2021]. Non-autoregressive electron redistribution modeling for reaction prediction, *International Conference on Machine Learning*, PMLR, pp. 904–913.

[15] Bishop, C. M. and Bishop, H. [2023]. *Deep learning: Foundations and concepts*, Springer Cham, Cham, Switzerland.

[16] Bishop, C. M. and Nasrabadi, N. M. [2006]. *Pattern recognition and machine learning*, Vol. 4, Springer.

[17] Bjerrum, E., Rastemo, T., Irwin, R., Kannas, C. and Genheden, S. [2021]. PySMILESUtils – Enabling deep learning with the SMILES chemical language.
URL: *https://chemrxiv.org/engage/chemrxiv/article-details/60d99ad2fca490cddfc97083*

[18] Bohde, M., Manjrekar, M., Wang, R., Ji, S. and Coley, C. W. [2025]. Diffms: Diffusion generation of molecules conditioned on mass spectra, *arXiv preprint arXiv:2502.09571* .

[19] Bradshaw, J., Kusner, M. J., Paige, B., Segler, M. H. and Hernández-Lobato, J. M. [2018]. A generative model for electron paths, *arXiv preprint arXiv:1805.10970* .

[20] Brown, F. K. [1998]. Chemoinformatics: what is it and how does it impact drug discovery., *Annual reports in medicinal chemistry* **33**: 375–384.

[21] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. et al. [2020]. Language models are few-shot learners, *Advances in Neural Information Processing Systems* **33**: 1877–1901.

[22] Cai, T., Li, Y., Geng, Z., Peng, H., Lee, J. D., Chen, D. and Dao, T. [2024]. Medusa: Simple LLM inference acceleration framework with multiple decoding heads, *arXiv preprint arXiv:2401.10774* .

[23] Capela, F., Nouchi, V., Van Deursen, R., Tetko, I. V. and Godin, G. [2019]. Multitask learning on graph neural networks applied to molecular property predictions, *arXiv preprint arXiv:1910.13124* .

[24] *Chemical reactions from US patents (1976-Sep2016) dataset. Accessed 23 March 2024*. [n.d.]. https://figshare.com/articles/dataset/Chemical_reactions_from_US_patents_1976-Sep2016_/5104873.

[25] Chen, B., Li, C., Dai, H. and Song, L. [2020]. Retro*: learning retrosynthetic planning with neural guided a* search, *International conference on machine learning*, PMLR, pp. 1608–1616.

[26] Chen, L.-Y. and Li, Y.-P. [2024]. AutoTemplate: enhancing chemical reaction datasets for machine learning applications in organic chemistry, *Journal of Cheminformatics* **16**(1): 74.

[27] Chen, S., An, S., Babazade, R. and Jung, Y. [2024]. Precise atom-to-atom mapping for organic reactions via human-in-the-loop machine learning, *Nature Communications* **15**(1): 2250.

[28] Chen, S., Babazade, R., Kim, T., Han, S. and Jung, Y. [2024]. A large-scale reaction dataset of mechanistic pathways of organic reactions, *Scientific Data* **11**(1): 863.

[29] Chen, S. and Jung, Y. [2021]. Deep retrosynthetic reaction prediction using local reactivity and global attention, *JACS Au* **1**(10): 1612–1620.

[30] Chen, Z., Ayinde, O. R., Fuchs, J. R., Sun, H. and Ning, X. [2023]. G 2 retro as a two-step graph generative models for retrosynthesis prediction, *Communications Chemistry* **6**(1): 102.

[31] Cherkasov, A., Muratov, E. N., Fourches, D., Varnek, A., Baskin, I. I., Cronin, M., Dearden, J., Gramatica, P., Martin, Y. C., Todeschini, R. et al. [2014]. QSAR modeling: where have you been? Where are you going to?, *Journal of medicinal chemistry* **57**(12): 4977–5010.

[32] Choe, J., Kim, H., Chok, Y. T., Gim, M. and Kang, J. [2025]. Retrosynthetic crosstalk between single-step reaction and multi-step planning, *Journal of cheminformatics* **17**(1): 130.

[33] Cireşan, D. C., Meier, U., Masci, J., Maria Gambardella, L. and Schmidhuber, J. [2011]. Flexible, high performance convolutional neural networks for image classification, *IJCAI proceedings-international joint conference on artificial intelligence*, Vol. 22, Barcelona, Spain:, p. 1237.

[34] Cireşan, D. C., Meier, U., Masci, J. and Schmidhuber, J. [2012]. Multi-column deep neural network for traffic sign classification, *Neural networks* **32**(1): 333–338.

[35] Coley, C. W., Barzilay, R., Jaakkola, T. S., Green, W. H. and Jensen, K. F. [2017]. Prediction of organic reaction outcomes using machine learning, *ACS central science* **3**(5): 434–443.

[36] Coley, C. W., Green, W. H. and Jensen, K. F. [2019]. Rdchiral: An rdkit wrapper for handling stereochemistry in retrosynthetic template extraction and application, *Journal of chemical information and modeling* **59**(6): 2529–2537.

[37] Coley, C. W., Jin, W., Rogers, L., Jamison, T. F., Jaakkola, T. S., Green, W. H., Barzilay, R. and Jensen, K. F. [2019]. A graph-convolutional neural network model for the prediction of chemical reactivity, *Chemical science* **10**(2): 370–377.

[38] Coley, C. W., Rogers, L., Green, W. H. and Jensen, K. F. [2017]. Computer-assisted retrosynthesis based on molecular similarity, *ACS central science* **3**(12): 1237–1245.

[39] Corey, E. J. [1967]. General methods for the construction of complex molecules, *Pure and Applied chemistry* **14**(1): 19–38.

[40] Corey, E. J. and Cheng, X.-M. [1989]. *The Logic of Chemical Synthesis*, John Wiley & Sons, New York.

[41] Corey, E. J. and Wipke, W. T. [1969]. Computer-assisted design of complex organic syntheses, *Science* **166**(3902): 178–192.

[42] Corey, E., Wipke, W. T., Cramer III, R. D. and Howe, W. J. [1972]. Computer-assisted synthetic analysis. facile man-machine communication of chemical structure by interactive computer graphics, *Journal of the American Chemical Society* **94**(2): 421–430.

[43] Cremer, J., Le, T., Ghahremanpour, M. M., Sługocka, E., Menezes, F. and Clevert, D.-A. [2025]. FLOWR. root: A flow matching based foundation model for joint multi-purpose structure-aware 3D ligand generation and affinity prediction, *arXiv preprint arXiv:2510.02578* .

[44] Cremer, J., Le, T., Noé, F., Clevert, D.-A. and Schütt, K. T. [2024]. Pilot: equivariant diffusion for pocket-conditioned de novo ligand generation with multi-objective guidance via importance sampling, *Chemical Science* **15**(36): 14954–14967.

[45] Current, S., Chen, Z., Adu-Ampratwum, D., Ning, X. and Parthasarathy, S. [2025]. Differ: categorical diffusion ensembles for single-step chemical retrosynthesis, *Journal of Cheminformatics* **17**(1): 112.

[46] Dai, H., Li, C., Coley, C., Dai, B. and Song, L. [2019]. Retrosynthesis prediction with conditional graph logic network, *Advances in Neural Information Processing Systems* **32**.

[47] Daylight Chemical Information Systems, Inc. [n.d.]. Daylight Theory: SMIRKS – A Reaction Transform Language, https://www.daylight.com/dayhtml/doc/theory/theory.smirks.html. Accessed: 2025-11-19.

[48] del Rio, B. G., Phan, B. and Ramprasad, R. [2023]. A deep learning framework to emulate density functional theory, *npj Computational Materials* **9**(1): 158.

[49] Deng, Y., Zhao, X., Sun, H., Chen, Y., Wang, X., Xue, X., Li, L., Song, J., Hsieh, C.-Y., Hou, T. et al. [2025]. Rsgpt: a generative transformer model for retrosynthesis planning pre-trained on ten billion datapoints, *Nature Communications* **16**(1): 7012.

[50] Dietterich, T. G. [1998]. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms, *Neural Computation* **10**(7): 1895–1923.
URL: *https://doi.org/10.1162/089976698300017197*

[51] Do, K., Tran, T. and Venkatesh, S. [2019]. Graph transformation policy network for chemical reaction prediction, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 750–760.

[52] Dobbelaere, M. R., Lengyel, I., Stevens, C. V. and Van Geem, K. M. [2024]. Rxninsight: fast chemical reaction analysis using bond-electron matrices, *Journal of Cheminformatics* **16**(1): 37.

[53] Dugundji, J. and Ugi, I. [1973]. An algebraic model of constitutional chemistry as a basis for chemical computer programs, *Fortschritte der Chemischen Forschung* pp. 19–64.

[54] El-Faham, A. and Albericio, F. [2011]. Peptide coupling reagents, more than a letter soup, *Chem Rev* **111**(11): 6557–6602.

[55] Ertl, P. and Schuffenhauer, A. [2009]. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions, *Journal of cheminformatics* **1**: 1–11.

[56] Fooshee, D., Mood, A., Gutman, E., Tavakoli, M., Urban, G., Liu, F., Huynh, N., Van Vranken, D. and Baldi, P. [2018]. Deep learning for chemical reaction prediction, *Molecular Systems Design & Engineering* **3**(3): 442–452.

[57] Forcada, M. L. and Ñeco, R. P. [1997]. Recursive hetero-associative memories for translation, *International Work-Conference on Artificial Neural Networks*, Springer, pp. 453–462.

[58] Gaiński, P., Koziarski, M., Maziarz, K., Segler, M., Tabor, J. and Śmieja, M. [2025]. Diverse and feasible retrosynthesis using gflownets, *Information Sciences* **714**: 122194.

[59] 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.
URL: *https://doi.org/10.1021/acscentsci.8b00357*

[60] Gao, W., Luo, S. and Coley, C. W. [2025]. Generative ai for navigating synthesizable chemical space, *Proceedings of the National Academy of Sciences* **122**(41): e2415665122.

[61] Gao, W., Mercado, R. and Coley, C. W. [2021]. Amortized tree generation for bottom-up synthesis planning and synthesizable molecular design, *arXiv preprint arXiv:2110.06389* .

[62] Gasteiger, J. and Engel, T. [2003]. *Chemoinformatics: A Textbook*, Wiley-VCH Verlag GmbH Co. KGaA.

[63] Gasteiger, J. and Jochum, C. [1978]. Eros a computer program for generating sequences of reactions, *Organic Compunds*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 93–126.

[64] Gelernter, H., Sanders, A., Larsen, D., Agarwal, K., Boivie, R., Spritzer, G. and Searleman, J. [1977]. Empirical explorations of synchem: The methods of artificial intelligence are applied to the problem of organic synthesis route discovery., *Science* **197**(4308): 1041–1049.

[65] Genheden, S. and Bjerrum, E. [2022]. Paroutes: towards a framework for benchmarking retrosynthesis route predictions, *Digital Discovery* **1**(4): 527–539.

[66] Genheden, S., Thakkar, A., Chadimová, V., Reymond, J.-L., Engkvist, O. and Bjerrum, E. [2020]. Aizynthfinder: a fast, robust and flexible open-source software for retrosynthetic planning, *Journal of Cheminformatics* **12**(1): 70.

[67] Ghiandoni, G. M., Evertsson, E., Riley, D. J., Tyrchan, C. and Rathi, P. C. [2024]. Augmenting dmta using predictive ai modelling at astrazeneca, *Drug discovery today* **29**(4): 103945.

[68] Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. and Dahl, G. E. [2017]. Neural message passing for quantum chemistry, *International conference on machine learning*, Pmlr, pp. 1263–1272.

[69] Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P. and Aspuru-Guzik, A. [2018]. Automatic chemical design using a data-driven continuous representation of molecules, *ACS central science* **4**(2): 268–276.

[70] Granqvist, E., Mercado, R. and Genheden, S. [2025]. Retrosynformer: Planning multi-step chemical synthesis routes via a decision transformer, *ChemRxiv* . Preprint. This content has not been peer-reviewed.

[71] Graves, A. [2012]. Sequence transduction with recurrent neural networks, *arXiv preprint arXiv:1211.3711* .

[72] Grzybowski, B. A., Szymkuć, S., Gajewska, E. P., Molga, K., Dittwald, P., Wołos, A. and Klucznik, T. [2018]. Chematica: a story of computer code that started to think like a chemist, *Chem* **4**(3): 390–398.

[73] Guan, Y.-J., Yu, C.-Q., Qiao, Y., Li, L.-P., You, Z.-H., Ren, Z.-H., Li, Y.-C. and Pan, J. [2022]. MFIDMA: A multiple information integration model for the prediction of drug–mirna associations, *Biology* **12**(1): 41.

[74] Han, Y., Xu, X., Hsieh, C.-Y., Ding, K., Xu, H., Xu, R., Hou, T., Zhang, Q. and Chen, H. [2024]. Retrosynthesis prediction with an iterative string editing model, *Nature Communications* **15**(1): 6404.

[75] Hansch, C., Maloney, P. P., Fujita, T. and Muir, R. M. [1962]. Correlation of biological activity of phenoxyacetic acids with hammett substituent constants and partition coefficients, *Nature* **194**(4824): 178–180.

[76] Hartog, P. B., Westerlund, A. M., Tetko, I. V. and Genheden, S. [2024]. Investigations into the efficiency of computer-aided synthesis planning, *Journal of Chemical Information and Modeling* .

[77] Hassen, A. K., Šícho, M., van Aalst, Y. J., Huizenga, M. C., Reynolds, D. N., Luukkonen, S., Bernatavicius, A., Clevert, D.-A., Janssen, A. P., van Westen, G. J. et al. [2025]. Generate what you can make: achieving in-house synthesizability with readily available resources in de novo drug design, *Journal of Cheminformatics* **17**(1): 41.

[78] Hendrickson, J. B. and Toczko, A. G. [1989]. Syngen program for synthesis design: basic computing techniques, *Journal of chemical information and computer sciences* **29**(3): 137–145.

[79] Hinton, G., Vinyals, O. and Dean, J. [2015]. Distilling the knowledge in a neural network, *arXiv preprint arXiv:1503.02531* .

[80] Hochreiter, S. [1991]. Untersuchungen zu dynamischen neuronalen netzen, *Diploma, Technische Universität München* **91**(1): 31.

[81] Hochreiter, S. and Schmidhuber, J. [1997]. Long short-term memory, *Neural Computation MIT-Press* .

[82] Hoogeboom, E., Satorras, V. G., Vignac, C. and Welling, M. [2022]. Equivariant diffusion for molecule generation in 3d, *International conference on machine learning*, PMLR, pp. 8867–8887.

[83] Hoskins, J. C. and Himmelblau, D. M. [1988]. Artificial neural network models of knowledge representation in chemical engineering, *Computers & Chemical Engineering* **12**(9-10): 881–890.

[84] Hu, Y., Liu, Z., Dong, Z., Peng, T., McDanel, B. and Zhang, S. Q. [2025]. Speculative decoding and beyond: An in-depth survey of techniques, *arXiv preprint arXiv:2502.19732* .

[85] Huang, B. and Von Lilienfeld, O. A. [2021]. Ab initio machine learning in chemical compound space, *Chemical reviews* **121**(16): 10001–10036.

[86] Igashov, I., Schneuing, A., Segler, M., Bronstein, M. and Correia, B. [2023]. Retrobridge: Modeling retrosynthesis with Markov bridges, *arXiv preprint arXiv:2308.16212* .

[87] Ihlenfeldt, W.-D. and Gasteiger, J. [1996]. Computer-assisted planning of organic syntheses: the second generation of programs, *Angewandte Chemie International Edition in English* **34**(23-24): 2613–2633.

[88] Irwin, R., Dimitriadis, S., He, J. and Bjerrum, E. J. [2022]. Chemformer: a pre-trained transformer for computational chemistry, *Machine Learning: Science and Technology* **3**(1): 015022.

[89] Ivanenkov, Y., Zagribelnyy, B., Malyshev, A., Evteev, S., Terentiev, V., Kamya, P., Bezrukov, D., Aliper, A., Ren, F. and Zhavoronkov, A. [2023]. The hitchhiker's guide to deep learning driven generative chemistry, *ACS Medicinal Chemistry Letters* **14**(7): 901–915.

[90] Jaeger, S., Fulle, S. and Turk, S. [2018]. Mol2vec: unsupervised machine learning approach with chemical intuition, *J Chem Inf Model* **58**(1): 27–35.

[91] Jiang, D., Wu, Z., Hsieh, C.-Y., Chen, G., Liao, B., Wang, Z., Shen, C., Cao, D., Wu, J. and Hou, T. [2021]. Could graph neural networks learn better molecular representation for drug discovery? a comparison study of descriptor-based and graph-based models, *Journal of cheminformatics* **13**(1): 12.

[92] Jin, W., Coley, C., Barzilay, R. and Jaakkola, T. [2017]. Predicting organic reaction outcomes with Weisfeiler-Lehman network, *Advances in neural information processing systems* **30**.

[93] Kannas, C. and Genheden, S. [2022]. Rxnutils–a cheminformatics python library for manipulating chemical reaction data.

[94] Karpov, P., Godin, G. and Tetko, I. V. [2019a]. Transformer-cnn: Fast and reliable tool for qsar, *arXiv preprint arXiv:1911.06603* .

[95] Karpov, P., Godin, G. and Tetko, I. V. [2019b]. A transformer model for retrosynthesis, *International conference on artificial neural networks*, Springer, pp. 817–830.

[96] Kayala, M. A., Azencott, C.-A., Chen, J. H. and Baldi, P. [2011]. Learning to predict chemical reactions, *Journal of chemical information and modeling* **51**(9): 2209–2222.

[97] Kayala, M. A. and Baldi, P. [2012]. Reactionpredictor: prediction of complex chemical reactions at the mechanistic level using machine learning, *Journal of chemical information and modeling* **52**(10): 2526–2540.

[98] Kayala, M. and Baldi, P. [2011]. A machine learning approach to predict chemical reactions, *Advances in Neural Information Processing Systems* **24**.

[99] 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.

[100] Kim, E., Lee, D., Kwon, Y., Park, M. S. and Choi, Y.-S. [2021]. Valid, plausible, and diverse retrosynthesis using tied two-way transformers with latent variables, *Journal of Chemical Information and Modeling* **61**(1): 123–133.

[101] Kim, S., Chen, J., Cheng, T., Gindulyte, A., He, J., He, S., Li, Q., Shoemaker, B. A., Thiessen, P. A., Yu, B. et al. [2025]. Pubchem 2025 update, *Nucleic acids research* **53**(D1): D1516–D1525.

[102] Kingma, D. P. and Ba, J. [2015]. Adam: A method for stochastic optimization, *CoRR* **abs/1412.6980**.

[103] Klein, G., Kim, Y., Deng, Y., Nguyen, V., Senellart, J. and Rush, A. M. [2018]. OpenNMT: Neural Machine Translation Toolkit.

[104] Krenn, M., Ai, Q., Barthel, S., Carson, N., Frei, A., Frey, N. C., Friederich, P., Gaudin, T., Gayle, A. A., Jablonka, K. M. et al. [2022]. Selfies and the future of molecular string representations, *Patterns* **3**(10).

[105] Krishna, U. V., Premjith, B. and Soman, K. [2022]. A comparative study of pretrained gene embeddings for COVID-19 mRNA vaccine degradation prediction, *Proceedings of the Seventh International Conference on Mathematics and Computing: ICMC 2021*, Springer, pp. 301–308.

[106] Krizhevsky, A., Sutskever, I. and Hinton, G. E. [2012]. Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems* **25**.

[107] Lederberg, J., Sutherland, G. L., Buchanan, B. G., Feigenbaum, E. A., Robertson, A. V., Duffield, A. M. and Djerassi, C. [1969]. Applications of artificial intelligence for chemical inference. i. number of possible organic compounds. acyclic structures containing carbon, hydrogen, oxygen, and nitrogen, *Journal of the American Chemical Society* **91**(11): 2973–2976.

[108] Leviathan, Y., Kalman, M. and Matias, Y. [2023]. Fast inference from transformers via speculative decoding, *International Conference on Machine Learning*, PMLR, pp. 19274–19286.

[109] Levy, O. and Goldberg, Y. [2014]. Neural word embedding as implicit matrix factorization, *Advances in neural information processing systems* **27**.

[110] Li, J., Fang, L. and Lou, J.-G. [2023]. Retroranker: leveraging reaction changes to improve retrosynthesis prediction through re-ranking, *Journal of Cheminformatics* **15**(1): 58.

[111] Li, J. J. [2006]. *Name Reactions: A Collection of Detailed Mechanisms and Synthetic Applications*, Springer Berlin, Heidelberg, Germany.

[112] Lilleberg, J., Zhu, Y. and Zhang, Y. [2015]. Support vector machines and word2vec for text classification with semantic features, *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, IEEE, pp. 136–140.

[113] Lin, A., Dyubankova, N., Madzhidov, T. I., Nugmanov, R. I., Verhoeven, J., Gimadiev, T. R., Afonina, V. A., Ibragimova, Z., Rakhimbekova, A., Sidorov, P. et al. [2022]. Atom-to-atom mapping: a benchmarking study of popular mapping algorithms and consensus strategies, *Molecular Informatics* **41**(4): 2100138.

[114] Lin, M. H., Tu, Z. and Coley, C. W. [2022]. Improving the performance of models for one-step retrosynthesis through re-ranking, *Journal of cheminformatics* **14**(1): 15.

[115] Lin, X., Yang, C., Wang, W., Li, Y., Du, C., Feng, F., Ng, S.-K. and Chua, T.-S. [2024]. Efficient inference for large language model-based generative recommendation, *arXiv preprint arXiv:2410.05165* .

[116] Lin, Z., Yin, S., Shi, L., Zhou, W. and Zhang, Y. J. [2023]. G2gt: retrosynthesis prediction with graph-to-graph attention neural network and self-training, *Journal of Chemical Information and Modeling* **63**(7): 1894–1905.

[117] Litsa, E. E., Chenthamarakshan, V., Das, P. and Kavraki, L. E. [2023]. An end-to-end deep learning framework for translating mass spectra to de-novo molecules, *Communications Chemistry* **6**(1): 132.

[118] Liu, B., Ramsundar, B., Kawthekar, P., Shi, J., Gomes, J., Luu Nguyen, Q., Ho, S., Sloane, J., Wender, P. and Pande, V. [2017]. Retrosynthetic reaction prediction using neural sequence-to-sequence models, *ACS central science* **3**(10): 1103–1113.

[119] Liu, C.-H., Korablyov, M., Jastrzebski, S., Włodarczyk-Pruszynski, P., Bengio, Y. and Segler, M. [2022]. Retrognn: fast estimation of synthesizability for virtual screening and de novo design by learning from slow retrosynthesis software, *Journal of Chemical Information and Modeling* **62**(10): 2293–2300.

[120] Liu, X., Guo, Y., Li, H., Liu, J., Huang, S., Ke, B. and Lv, J. [2024]. Drugllm: Open large language model for few-shot molecule generation, *arXiv preprint arXiv:2405.06690* .

[121] Lowe, D. M. [2012]. Extraction of chemical structures and reactions from the literature. Ph.D. dissertation, University of Cambridge, Cambridge, UK., https://doi.org/10.17863/CAM.16293.

[122] Lu, J. and Zhang, Y. [2022]. Unified deep learning model for multitask reaction predictions with explanation, *Journal of Chemical Information and Modeling* **62**(6): 1376–1387.

[123] Marcou, G., Aires de Sousa, J., Latino, D. A. R. S., de Luca, A., Horvath, D., Rietsch, V. and Varnek, A. [2015]. Expert system for predicting reaction conditions: The michael reaction case, *Journal of Chemical Information and Modeling* **55**(2): 239–250. PMID: 25588070.
URL: *https://doi.org/10.1021/ci500698a*

[124] Maser, M. R., Cui, A. Y., Ryou, S., DeLano, T. J., Yue, Y. and Reisman, S. E. [2021]. Multilabel classification models for the prediction of cross-coupling reaction conditions, *Journal of Chemical Information and Modeling* **61**(1): 156–166. PMID: 33417449.
URL: *https://doi.org/10.1021/acs.jcim.0c01234*

[125] Maziarz, K., Tripp, A., Liu, G., Stanley, M., Xie, S., Gaiński, P., Seidl, P. and Segler, M. H. [2025]. Re-evaluating retrosynthesis algorithms with syntheseus, *Faraday Discussions* **256**: 568–586.

[126] McInnes, L., Healy, J. and Melville, J. [2018]. UMAP: Uniform manifold approximation and projection for dimension reduction, *arXiv preprint arXiv:1802.03426* .

[127] Meng, Z., Zhao, P., Yu, Y. and King, I. [2023a]. Doubly stochastic graph-based non-autoregressive reaction prediction, *arXiv preprint arXiv:2306.06119* .

[128] Meng, Z., Zhao, P., Yu, Y. and King, I. [2023b]. A unified view of deep learning for reaction and retrosynthesis prediction: current status and future challenges, *arXiv preprint arXiv:2306.15890* .

[129] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. and Dean, J. [2013]. Distributed representations of words and phrases and their compositionality, *Advances in neural information processing systems* **26**.

[130] Mikolov, T. and Zweig, G. [2012]. Context-dependent recurrent neural network language model, *2012 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, pp. 234–239.

[131] Miyaura, N. and Suzuki, A. [1995]. Palladium-catalyzed cross-coupling reactions of organoboron compounds, *Chemical reviews* **95**(7): 2457–2483.

[132] Nam, J. and Kim, J. [2016]. Linking the neural machine translation and the prediction of organic chemistry reactions, *arXiv preprint arXiv:1612.09529* .

[133] *NameRXN* [n.d.]. https://www.nextmovesoftware.com/namerxn.html.

[134] Nugmanov, R., Dyubankova, N., Gedich, A. and Wegner, J. K. [2022]. Bidirectional graphormer for reactivity understanding: neural network trained to reaction atom-to-atom mapping task, *Journal of Chemical Information and Modeling* **62**(14): 3307–3315.

[135] O'Boyle, N. and Dalke, A. [2018]. Deepsmiles: An adaptation of smiles for use in machine-learning of chemical structures, *ChemRxiv* . Preprint.

[136] Öztürk, H., Özgür, A., Schwaller, P., Laino, T. and Ozkirimli, E. [2020]. Exploring chemical space using natural language processing methodologies for drug discovery, *Drug Discov Today* **25**(4): 689–705.

[137] Qian, W. W., Russell, N. T., Simons, C. L. W., Luo, Y., Burke, M. D. and Peng, J. [2020]. Integrating Deep Neural Networks and Symbolic Inference for Organic Reactivity Prediction.
URL: *https://chemrxiv.org/engage/chemrxiv/article-details/60c7476dbb8c1a4fad3daa77*

[138] *RDKit: Open-source cheminformatics* [n.d.]. https://www.rdkit.org. [Online; accessed 2025-08-27].

[139] *Reaxys database. Accessed 23 March 2024.* [n.d.]. https://www.reaxys.com.

[140] Ren, R., Yin, C. and S.-T. Yau, S. [2022]. kmer2vec: A novel method for comparing dna sequences by word2vec embedding, *J Comput Biol* **29**(9): 1001–1021.

[141] Ross, J., Belgodere, B., Chenthamarakshan, V., Padhi, I., Mroueh, Y. and Das, P. [2022]. Large-scale chemical language representations capture molecular structure and properties, *Nature Machine Intelligence* **4**(12): 1256–1264.

[142] Ruddigkeit, L., Van Deursen, R., Blum, L. C. and Reymond, J.-L. [2012]. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17, *Journal of chemical information and modeling* **52**(11): 2864–2875.

[143] Ryou, S., Maser, M. R., Cui, A. Y., DeLano, T. J., Yue, Y. and Reisman, S. E. [2020]. Graph neural networks for the prediction of substrate-specific organic reaction conditions, *arXiv preprint arXiv:2007.04275* .

[144] Sacha, M., Błaz, M., Byrski, P., Dabrowski-Tumanski, P., Chrominski, M., Loska, R., Włodarczyk-Pruszynski, P. and Jastrzebski, S. [2021]. Molecule edit graph attention network: modeling chemical reactions as sequences of graph edits, *Journal of Chemical Information and Modeling* **61**(7): 3273–3284.

[145] Saigiridharan, L., Hassen, A. K., Lai, H., Torren-Peraire, P., Engkvist, O. and Genheden, S. [2024]. Aizynthfinder 4.0: developments based on learnings from 3 years of industrial application, *Journal of cheminformatics* **16**(1): 57.

[146] Schlag, I., Irie, K. and Schmidhuber, J. [2021]. Linear transformers are secretly fast weight programmers, *International conference on machine learning*, PMLR, pp. 9355–9366.

[147] Schmidhuber, J. [1991]. Neural sequence chunkers, *Technical Report FKI-148-91*, Institut für Informatik, Technische Universität München.

[148] Schmidhuber, J. [1992a]. Learning complex, extended sequences using the principle of history compression, *Neural Computation* **4**(2): 234–242.

[149] Schmidhuber, J. [1992b]. Learning to control fast-weight memories: An alternative to recurrent nets, *Neural Computation* **4**(1): 131–139.

[150] Schmidhuber, J. and Heil, S. [1996]. Sequential neural text compression, *IEEE Transactions on Neural Networks* **7**(1): 142–146.

[151] Schneider, N., Stiefl, N. and Landrum, G. A. [2016]. What's what: The (nearly) definitive guide to reaction role assignment, *J Chem Inf Model* **56**(12): 2336–2346.

[152] Schwaller, P., Gaudin, T., Lanyi, D., Bekas, C. and Laino, T. [2018]. "found in translation": predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models, *Chemical science* **9**(28): 6091–6098.

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

[154] 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.

[155] 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.

[156] Segler, M. H., Kogej, T., Tyrchan, C. and Waller, M. P. [2018]. Generating focused molecule libraries for drug discovery with recurrent neural networks, *ACS central science* **4**(1): 120–131.

[157] 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.

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

[159] Seidl, P., Renz, P., Dyubankova, N., Neves, P., Verhoeven, J., Wegner, J. K., Segler, M., Hochreiter, S. and Klambauer, G. [2022]. Improving few-and zero-shot reaction template prediction using modern Hopfield networks, *Journal of Chemical Information and Modeling* **62**(9): 2111–2120.

[160] Shao, J., Gong, Q., Yin, Z., Pan, W., Pandiyan, S. and Wang, L. [2022]. S2DV: converting SMILES to a drug vector for predicting the activity of anti-HBV small molecules, *Brief Bioinform* **23**(2).

[161] Shee, Y., Li, H., Zhang, P., Nikolic, A. M., Lu, W., Kelly, H. R., Manee, V., Sreekumar, S., Buono, F. G., Song, J. J. et al. [2024]. Site-specific template generative approach for retrosynthetic planning, *Nature Communications* **15**(1): 7818.

[162] Shee, Y., Morgunov, A., Li, H. and Batista, V. S. [2025]. Directmultistep: Direct route generation for multistep retrosynthesis, *Journal of Chemical Information and Modeling* **65**(8): 3903–3914. PMID: 40197023.
URL: *https://doi.org/10.1021/acs.jcim.4c01982*

[163] Shi, C., Xu, M., Guo, H., Zhang, M. and Tang, J. [2020]. A graph to graphs framework for retrosynthesis prediction, *International conference on machine learning*, PMLR, pp. 8818–8827.

[164] Shrivastava, A. D., Swainston, N., Samanta, S., Roberts, I., Wright Muelas, M. and Kell, D. B. [2021]. Massgenie: a transformer-based deep learning method for identifying small molecules from their mass spectra, *Biomolecules* **11**(12): 1793.

[165] Sigmund, L. M., Seifert, T., Halder, R., Bergonzini, G., Johansson, M. J., Norrby, P.-O., Jorner, K. and Kabeshov, M. [2025]. Predicting reaction feasibility and selectivity of aromatic c□ h thianthrenation with a qm–ml hybrid approach, *Angewandte Chemie* p. e202510533.

[166] Smith, J. S., Isayev, O. and Roitberg, A. E. [2017]. Ani-1: an extensible neural network potential with dft accuracy at force field computational cost, *Chemical science* **8**(4): 3192–3203.

[167] Somnath, V. R., Bunne, C., Coley, C. W., Krause, A. and Barzilay, R. [2020]. Learning graph models for template-free retrosynthesis, *arXiv preprint arXiv:2006.07038* .

[168] Song, X., Pan, X., Zhao, X., Ye, H., Zhang, S., Tang, J. and Yu, T. [2025]. Aot*: Efficient synthesis planning via llm-empowered and-or tree search, *arXiv preprint arXiv:2509.20988* .

[169] Srivastava, R. K., Greff, K. and Schmidhuber, J. [2015]. Highway networks, *arXiv preprint arXiv:1505.00387* .

[170] Stanley, M. and Segler, M. [2023]. Fake it until you make it? generative de novo design and virtual screening of synthesizable molecules, *Current Opinion in Structural Biology* **82**: 102658.

[171] Struebing, H., Ganase, Z., Karamertzanis, P. G., Siougkrou, E., Haycock, P., Piccione, P. M., Armstrong, A., Galindo, A. and Adjiman, C. S. [2013]. Computer-aided molecular design of solvents for accelerated reaction kinetics, *Nature Chemistry* **5**: 952–957.
URL: *https://doi.org/10.1038/nchem.1755*

[172] Stuyver, T. and Coley, C. W. [2022]. Quantum chemistry-augmented neural networks for reactivity prediction: Performance, generalizability, and explainability, *The Journal of Chemical Physics* **156**(8): 084104.

[173] Sun, R., Dai, H., Li, L., Kearnes, S. and Dai, B. [2021]. Towards understanding retrosynthesis by energy-based models, *Advances in Neural Information Processing Systems* **34**: 10186–10194.

[174] Sun, S., Yu, W., Ren, Y., Du, W., Liu, L., Zhang, X., Hu, Y. and Ma, C. [2025]. Gdiffretro: Retrosynthesis prediction with dual graph enhanced molecular representation and diffusion generation, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39, pp. 12595–12603.

[175] Sutskever, I., Vinyals, O. and Le, Q. V. [2014]. Sequence to sequence learning with neural networks, *Advances in neural information processing systems* **27**.

[176] Tan, X. [2025]. A transformer-based generative chemical language ai model for structural elucidation of organic compounds, *Journal of cheminformatics* **17**(1): 103.

[177] Tetko, I. V., Karpov, P., Van Deursen, R. and Godin, G. [2020a]. State-of-the-art augmented nlp transformer models for direct and single-step retrosynthesis, *Nature communications* **11**(1): 5575.

[178] Tetko, I. V., Karpov, P., Van Deursen, R. and Godin, G. [2020b]. State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis, *Nature Communications* **11**(1): 5575.

[179] Thakkar, A., Chadimová, V., Bjerrum, E. J., Engkvist, O. and Reymond, J.-L. [2021]. Retrosynthetic accessibility score (rascore)–rapid machine learned synthesizability classification from ai driven retrosynthetic planning, *Chemical science* **12**(9): 3339–3349.

[180] 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.

[181] Tingle, B. I., Tang, K. G., Castanon, M., Gutierrez, J. J., Khurelbaatar, M., Dandarchuluun, C., Moroz, Y. S. and Irwin, J. J. [2023]. Zinc-22□ a free multi-billion-scale database of tangible compounds for ligand discovery, *Journal of chemical information and modeling* **63**(4): 1166–1176.

[182] Toniato, A., Schwaller, P., Cardinale, A., Geluykens, J. and Laino, T. [2021]. Unassisted noise reduction of chemical reaction datasets, *Nature Machine Intelligence* **3**(6): 485–494.

[183] Torren-Peraire, P., Hassen, A. K., Genheden, S., Verhoeven, J., Clevert, D.-A., Preuss, M. and Tetko, I. V. [2024]. Models matter: the impact of single-step retrosynthesis on synthesis planning, *Digital Discovery* **3**(3): 558–572.

[184] Torren-Peraire, P., Verhoeven, J., Herman, D., Ceulemans, H., Tetko, I. V. and Wegner, J. K. [2025]. Improving route development using convergent retrosynthesis planning, *Journal of Cheminformatics* **17**(1): 26.

[185] Toulhoat, H. and Raybaud, P. [2020]. Prediction of optimal catalysts for a given chemical reaction, *Catal. Sci. Technol.* **10**: 2069–2081.
URL: *http://dx.doi.org/10.1039/C9CY02196E*

[186] Tu, Z., Choure, S. J., Fong, M. H., Roh, J., Levin, I., Yu, K., Joung, J. F., Morgan, N., Li, S.-C., Sun, X. et al. [2025]. ASKCOS: an open source software suite for synthesis planning, *arXiv preprint arXiv:2501.01835* .

[187] Tu, Z. and Coley, C. W. [2022]. Permutation invariant graph-to-sequence model for template-free retrosynthesis and reaction prediction, *Journal of chemical information and modeling* **62**(15): 3503–3513.

[188] Ucak, U. V., Ashyrmamatov, I., Ko, J. and Lee, J. [2022]. Retrosynthetic reaction pathway prediction through neural machine translation of atomic environments, *Nature communications* **13**(1): 1186.

[189] 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**.

[190] Vaucher, A. C., Schwaller, P. and Laino, T. [2020]. Completion of partial reaction equations, *ChemRxiv* .

[191] Vaucher, A. C., Zipoli, F., Geluykens, J., Nair, V. H., Schwaller, P. and Laino, T. [2020]. Automated extraction of chemical synthesis actions from experimental procedures, *Nature communications* **11**(1): 3601.

[192] Vijayan, R., Kihlberg, J., Cross, J. B. and Poongavanam, V. [2022]. Enhancing preclinical drug discovery with artificial intelligence, *Drug discovery today* **27**(4): 967–984.

[193] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P. and SciPy 1.0 Contributors [2020]. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, *Nat Methods* **17**: 261–272.

[194] Vleduts, G. [1963]. Concerning one system of classification and codification of organic reactions, *Information Storage and Retrieval* **1**(2-3): 117–146.

[195] Voinarovska, V., Kabeshov, M., Dudenko, D., Genheden, S. and Tetko, I. V. [2023]. When yield prediction does not yield prediction: an overview of the current challenges, *Journal of Chemical Information and Modeling* **64**(1): 42–56.

[196] Voršilák, M., Kolář, M., Čmelo, I. and Svozil, D. [2020]. Syba: Bayesian estimation of synthetic accessibility of organic compounds, *Journal of cheminformatics* **12**(1): 35.

[197] 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. PMID: 31381340.
URL: *https://doi.org/10.1021/acs.jcim.9b00313*

[198] Wan, Y., Hsieh, C.-Y., Liao, B. and Zhang, S. [2022]. Retroformer: Pushing the limits of end-to-end retrosynthesis transformer, *International Conference on Machine Learning*, PMLR, pp. 22475–22490.

[199] Wang, L., Song, C., Liu, Z., Rong, Y., Liu, Q., Wu, S. and Wang, L. [2025]. Diffusion models for molecules: A survey of methods and tasks. arxiv 2025, *arXiv preprint arXiv:2502.09511* .

[200] Wang, L., Zhou, Y. and Chen, Q. [2023]. AMMVF-DTI: A novel model predicting drug–target interactions based on attention mechanism and multi-view fusion, *Int J Mol Sci* **24**(18): 14142.

[201] Wang, X., Li, Y., Qiu, J., Chen, G., Liu, H., Liao, B., Hsieh, C.-Y. and Yao, X. [2021]. Retroprime: A diverse, plausible and transformer-based method for single-step retrosynthesis predictions, *Chemical Engineering Journal* **420**: 129845.

[202] Wang, Y., Pang, C., Wang, Y., Jin, J., Zhang, J., Zeng, X., Su, R., Zou, Q. and Wei, L. [2023]. Retrosynthesis prediction with an interpretable deep-learning framework based on molecular assembly tasks, *Nature Communications* **14**(1): 6155.

[203] Wang, Y., You, Z.-H., Yang, S., Li, X., Jiang, T.-H. and Zhou, X. [2019]. A high efficient biological language model for predicting protein–protein interactions, *Cells* **8**(2): 122.

[204] Wei, J. N., Duvenaud, D. and Aspuru-Guzik, A. [2016]. Neural networks for the prediction of organic chemistry reactions, *ACS central science* **2**(10): 725–732.

[205] 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.

[206] Wigh, D. S., Arrowsmith, J., Pomberger, A., Felton, K. C. and Lapkin, A. A. [2024]. Orderly: data sets and benchmarks for chemical reaction data, *Journal of Chemical Information and Modeling* **64**(9): 3790–3798.

[207] Willett, P. [2008]. From chemical documentation to chemoinformatics: 50 years of chemical information science, *Journal of Information Science* **34**(4): 477–499.

[208] Willighagen, E. L., Mayfield, J. W., Alvarsson, J., Berg, A., Carlsson, L., Jeliazkova, N., Kuhn, S., Pluskal, T., Rojas-Chertó, M., Spjuth, O., Torrance, G., Evelo, C. T., Guha, R. and Steinbeck, C. [2017]. The chemistry development kit (cdk) v2.0: atom typing, depiction, molecular formulas, and substructure searching, *Journal of Cheminformatics* **9**(1): 33.

[209] Winter, R., Montanari, F., Noé, F. and Clevert, D.-A. [2019]. Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations, *Chemical science* **10**(6): 1692–1701.

[210] Wipke, W. T., Ouchi, G. I. and Krishnan, S. [1978]. Simulation and evaluation of chemical synthesis—secs: An application of artificial intelligence techniques, *Artificial Intelligence* **11**(1-2): 173–193.

[211] Xia, H., Ge, T., Wang, P., Chen, S.-Q., Wei, F. and Sui, Z. [2023]. Speculative decoding: Exploiting speculative execution for accelerating seq2seq generation, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 3909–3925.

[212] Xia, H., Yang, Z., Dong, Q., Wang, P., Li, Y., Ge, T., Liu, T., Li, W. and Sui, Z. [2024]. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding, *arXiv preprint arXiv:2401.07851* .

[213] Xia, M., Hu, J., Zhang, X. and Lin, X. [2022]. Drug-target binding affinity prediction based on graph neural networks and word2vec, *International Conference on Intelligent Computing*, Springer, pp. 496–506.

[214] Xie, S., Yan, R., Guo, J., Xia, Y., Wu, L. and Qin, T. [2023]. Retrosynthesis prediction with local template retrieval, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37, pp. 5330–5338.

[215] Xuan-Vu, N., Armstrong, D. P., Jončev, Z. and Schwaller, P. [2025]. Tempre: Template generation for single and direct multi-step retrosynthesis, *arXiv preprint arXiv:2507.21762* .

[216] Yan, C., Ding, Q., Zhao, P., Zheng, S., Yang, J., Yu, Y. and Huang, J. [2020]. Retroxpert: Decompose retrosynthesis prediction like a chemist, *Advances in Neural Information Processing Systems* **33**: 11248–11258.

[217] Yan, C., Zhao, P., Lu, C., Yu, Y. and Huang, J. [2022]. Retrocomposer: composing templates for template-based retrosynthesis prediction, *Biomolecules* **12**(9): 1325.

[218] Yao, L., Guo, W., Wang, Z., Xiang, S., Liu, W. and Ke, G. [2024]. Node-aligned graph-to-graph: elevating template-free deep learning approaches in single-step retrosynthesis, *JACS Au* **4**(3): 992–1003.

[219] Zdrazil, B., Felix, E., Hunter, F., Manners, E. J., Blackshaw, J., Corbett, S., De Veij, M., Ioannidis, H., Lopez, D. M., Mosquera, J. F. et al. [2024]. The chembl database in 2023: a drug discovery platform spanning multiple bioactivity data types and time periods, *Nucleic acids research* **52**(D1): D1180–D1192.

[220] Zhang, W., Wang, Q., Kong, X., Xiong, J., Ni, S., Cao, D., Niu, B., Chen, M., Li, Y., Zhang, R. et al. [2024]. Fine-tuning large language models for chemical text mining, *Chemical science* **15**(27): 10600–10611.

[221] Zhang, X., Ling, T., Jin, Z., Xu, S., Gao, Z., Sun, B., Qiu, Z., Wei, J., Dong, N., Wang, G. et al. [2025]. π-primenovo: an accurate and efficient non-autoregressive deep learning model for de novo peptide sequencing, *Nature Communications* **16**(1): 267.

[222] Zhao, P.-C., Wei, X.-X., Wang, Q., Wang, Q.-H., Li, J.-N., Shang, J., Lu, C. and Shi, J.-Y. [2025]. Single-step retrosynthesis prediction via multitask graph representation learning, *Nature Communications* **16**(1): 814.

[223] Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z. et al. [2023]. A survey of large language models, *arXiv preprint arXiv:2303.18223* .

[224] Zheng, S., Rao, J., Zhang, Z., Xu, J. and Yang, Y. [2019]. Predicting retrosynthetic reactions using self-corrected transformer neural networks, *Journal of chemical information and modeling* **60**(1): 47–55.

[225] Zhong, W., Yang, Z. and Chen, C. Y.-C. [2023]. Retrosynthesis prediction using an end-to-end graph generative architecture for molecular graph editing, *Nature Communications* **14**(1): 3009.

[226] Zhong, Z., Song, J., Feng, Z., Liu, T., Jia, L., Yao, S., Wu, M., Hou, T. and Song, M. [2022]. Root-aligned SMILES: a tight representation for chemical reaction prediction, *Chemical Science* **13**(31): 9023–9034.

[227] Zhuang, J., Shi, Y., Hou, J., He, Y., Ye, M., Xu, M., Su, Y., Zhang, L., Ke, G. and Cai, H. [2025]. Reasoning-enhanced large language models for molecular property prediction, *arXiv preprint arXiv:2510.10248* .