ARTICLE

doi:10.1038/nature25978

Planning chemical syntheses with deep neural networks and symbolic AI

Marwin H. S. Segler[1,2], Mike Preuss[3] & Mark P. Waller[4]

To plan the syntheses of small organic molecules, chemists use retrosynthesis, a problem-solving technique in which target molecules are recursively transformed into increasingly simpler precursors. Computer-aided retrosynthesis would be a valuable tool but at present it is slow and provides results of unsatisfactory quality. Here we use Monte Carlo tree search and symbolic artificial intelligence (AI) to discover retrosynthetic routes. We combined Monte Carlo tree search with an expansion policy network that guides the search, and a filter network to pre-select the most promising retrosynthetic steps. These deep neural networks were trained on essentially all reactions ever published in organic chemistry. Our system solves for almost twice as many molecules, thirty times faster than the traditional computer-aided search method, which is based on extracted rules and hand-designed heuristics. In a double-blind AB test, chemists on average considered our computer-generated routes to be equivalent to reported literature routes.

Retrosynthetic analysis is the canonical technique used to plan the synthesis of small organic molecules[1,2]. In retrosynthesis, a search tree is built by 'working backwards', analysing molecules recursively and transforming them into simpler precursors until one obtains a set of known or commercially available building-block molecules (Fig. 1)[3,4]. Given that transformations are formally reversed chemical reactions, the plan can be then carried out in the laboratory in the forward direction to synthesize the target compound[3,4]. Transformations are derived from successfully conducted series of similar reactions with analogous starting materials, and are often named after their discoverers ('named reactions')[5]. At each retrosynthetic step, a small set out of hundreds of thousands of transformations known in modern chemistry has to be selected. In a pattern recognition process, chemists intuitively prioritise the most promising transformations, which they then consider, without actively thinking about the less promising ones[6]. However, when a transformation is applied to a new molecule, there is no guarantee that the corresponding reaction will proceed in the expected way[7]. A molecule failing to react as predicted is called 'out of scope'. This can be due to steric or electronic effects, an incomplete understanding of the reaction mechanism, or conflicting reactivity in the molecular context. Predicting which molecules are 'in scope' can be challenging even for the best human chemists[4,7].

Computer-assisted synthesis planning (CASP) could help chemists to find better routes faster, and is a missing component in virtual *de novo* design and robot systems performing molecular design–synthesis–test cycles[8–10]. To perform CASP, the knowledge that humans gain must be transferred into an executable program[11–16]. Despite 60 years of research, attempts to formalize chemistry by manual encoding by experts have not convinced synthetic chemists, and it does not scale to exponentially growing knowledge[15–19]. Methods of algorithmically extracting transformations from reaction datasets[20–22] have been criticized for high noise and lack of 'chemical intelligence'[13,14]. However, we recently showed that deep neural networks can learn to rank extracted symbolic transformations, and to avoid reactivity conflicts, which mimics the expert's intuitive decision-making[23]. To guide the search in promising directions, heuristic best first search (BFS) has been employed, in which hand-designed heuristic functions determine position values[13]. Unfortunately, unlike in chess, it is difficult to define strong heuristics in chemistry for three reasons. First, chemists tend to disagree on what constitutes a good position[24,25]. Second, although it is generally desirable to simplify the molecules, it can be tactically beneficial to temporarily increase complexity by the use of protecting or directing groups. Finally, the position value depends highly on the availability of suitable precursors[13,15]. Even complex molecules can be made in a few steps if precursors are readily available. Therefore, one cannot reliably estimate the value of a synthetic position without completely 'playing the molecules until the end of the game'.

Monte Carlo tree search (MCTS) has emerged as a general search technique for sequential decision problems with large branching factors without strong heuristics, such as games or automated theorem proving[26–28]. MCTS uses rollouts to determine position values. Rollouts are Monte Carlo simulations, in which random search steps are performed without branching until a solution has been found or a maximum depth is reached. These random steps can be sampled from machine-learned policies $p(t|s)$[29], which predict the probability of taking the move (applying the transformation) t in position s, and are trained to predict the winning move by using human games or self-play[30–35].

In this work, we combine three different neural networks together with MCTS to perform chemical synthesis planning (3N-MCTS). The first neural network (the expansion policy) guides the search in promising directions by proposing a restricted number of automatically extracted transformations. A second neural network then predicts whether the proposed reactions are actually feasible (in scope). Finally, to estimate the position value, transformations are sampled from a third neural network during the rollout phase. The neural networks were trained on essentially all reactions published in the history of organic chemistry.

Training the expansion and rollout policies

We extracted transformation rules from 12.4 million single-step reactions from the Reaxys[36] chemistry database[23]. Two sets of rules were extracted. The rollout set comprises rules that contain the atoms and bonds that changed in the course of the reaction (the reaction centre),

[1]Institute of Organic Chemistry and Center for Multiscale Theory and Computation, Westfälische Wilhelms-Universität, Münster, Germany. [2]BenevolentAI, London, UK. [3]European Research Center for Information Systems, Westfälische Wilhelms-Universität Münster, Germany. [4]Department of Physics and International Centre for Quantum and Molecular Structures, Shanghai University, Shanghai, China.

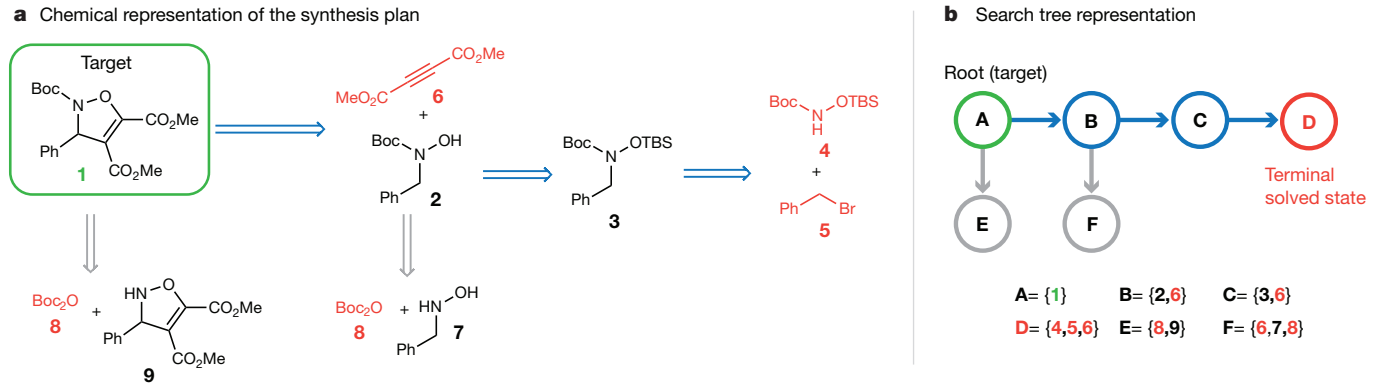


Figure 1 | Translation of the traditional chemists' retrosynthetic route representation to the search tree representation. a, The traditional chemists' retrosynthetic route representation (conditions omitted)[50]. **b**, The search tree representation. The nodes in the tree represent the synthetic position, and contain all precursors needed to make the molecules of the preceding positions all the way down to the tree's root, which contains the target. Branches in the search tree correspond to complete routes. Calculating the value of branches through task-dependent scoring functions allows us to compare and rank different routes. The target molecule can be solved if it can be deconstructed to a set of readily available building blocks (marked red). Ph, phenyl; Boc, *tert*-butyloxycarbonyl; TBS, *tert*-butyldimethylsilyl.

and the first-degree neighbouring atoms. Only rules that occurred at least 50 times in reactions published before 2015 were kept. For the expansion rules, a more general rule definition was employed. Here, only the reaction centre was extracted. Rules occurring at least three times were kept. The two sets encompass 17,134 and 301,671 rules, and cover 52% and 79% of all chemical reactions from 2015 and after, respectively.

Rule extraction associates each reaction, and thus each product, with a transformation rule. This allows us to train neural networks as policies to predict the best transformations given the product, or in other words, the best reactions with which to make the product[23]. Importantly, such neural networks also learn about the context in which the reactions can occur (functional group tolerance)[23]. For the expansion policy, we employed a deep highway network[37] with exponential linear unit nonlinearities[38]. To assess its ability to generalize, we performed a time-split strategy[39]. For training, only reactions published before 2015 were used, whereas for validation and testing, data from 2015 and later were selected.

Extended Data Table 1 shows the metrics for the expansion policy. The neural network predicts the correct solution out of 301,671 transformations with an accuracy of 31%, which is reasonable. It has to be noted that there are almost always many feasible ways to make a molecule. The top 10 and top 50 accuracies of 63.3% and 72.5% indicate that the correct transformations are generally ranked highly. Beyond the top 50 predicted results, the accuracy increases only marginally. This observation allows us to reduce the branching factor drastically, which is 46,175 when rules are applied exhaustively. During search tree expansion, we restrict the possible transformations to a maximum of 50. Additionally, we sum the probabilities of the predicted actions, starting from the highest-ranked transformation. When the cumulative probability reaches 0.995, we stop further expansion, even if fewer than 50 actions have been expanded. This allows the system to focus on highly likely transformations when only a few good options exist, for example in the synthesis of acyl chlorides or Grignard reagents. We observed that the reactions in this reduced top 50 are almost always reasonable and are often variations of the correct prediction. For example, a Heck reaction can often be conducted with bromide, iodide or triflate as the leaving group.

The rollout policy network, which is a neural network with one hidden layer, is trained in the same way as the expansion policy. It uses a set of 17,134 rules, which implies a lower coverage than the expansion policy, yet it needs just 10 ms to make a prediction, in contrast to 90 ms for the expansion policy, owing to the smaller output layer (see Extended Data Table 1). The rationale for using two different rule sets is to use a powerful but slow policy to select the best candidate transformations for expansion, and a fast rollout policy to estimate the position values[35].

Prediction with the in-scope filter network

After the search space has been narrowed down by the expansion policy to the most promising transformations, we need to predict whether the corresponding reactions will actually work for a particular molecule. We trained a deep neural network as a binary classifier to predict whether the reactions corresponding to the transformations selected by the policy network are actually feasible[40]. The classifier has to be trained on successful and failed reactions. Unfortunately, failed reactions are rarely reported and not contained in reaction databases. However, published reactions contain implicit information about reactions that do not occur. For a high-yielding reaction $A + B \rightarrow C$, we can assume that hypothetical products D, E,... are not formed. By applying reaction rules in the forward direction to the reactants of reported reactions, negative reactions, for example, with incorrect regio- and chemoselectivity, can then be generated[41,42]. Here we used the same rule set as for the expansion policy. Additionally, we generated negative examples by shuffling the associated pairs of products and corresponding reactions (see Methods for details). Using these data augmentation strategies, we generated 100 million negative reactions from reactions published before 2015 for training and 10 million published in and after 2015 for testing. As positive cases, all reported reactions from these periods were used. On the test set, the classifier achieves an area under the receiver operation characteristic curve of 0.99, and an area under the precision-recall curve of 0.94, which indicates good performance (see Extended Data Fig. 1)[43]. The false positive rate of the filter (that is, incorrect reactions passing the filter) is 1.5%, whereas the false negative rate (that is, real reactions being filtered out) is 14%. Interestingly, the in-scope filter correlates with basic electronic properties (Hammond parameters and lowest unoccupied molecular orbital (LUMO) energies), even though it is not explicitly trained to do so (see Methods and Extended Data Fig. 5).

Integrating neural networks and MCTS

The expansion policy network and the in-scope filter network are combined into a pipeline (Fig. 2b). When a position s_i is to be analysed, each molecule of the position is fed into the policy network. Then, the transformations with the highest scores are applied to the molecule, which yields the possible precursors and thus full reactions. These reactions are submitted to the in-scope filter, where only transformations and precursors corresponding to positively classified reactions are kept. They represent the 'legal moves' available in position s_i.

The expansion procedure and the rollout policy are then incorporated in the respective phases of an MCTS algorithm to form 3N-MCTS. The four MCTS phases are then iterated to build the search tree.

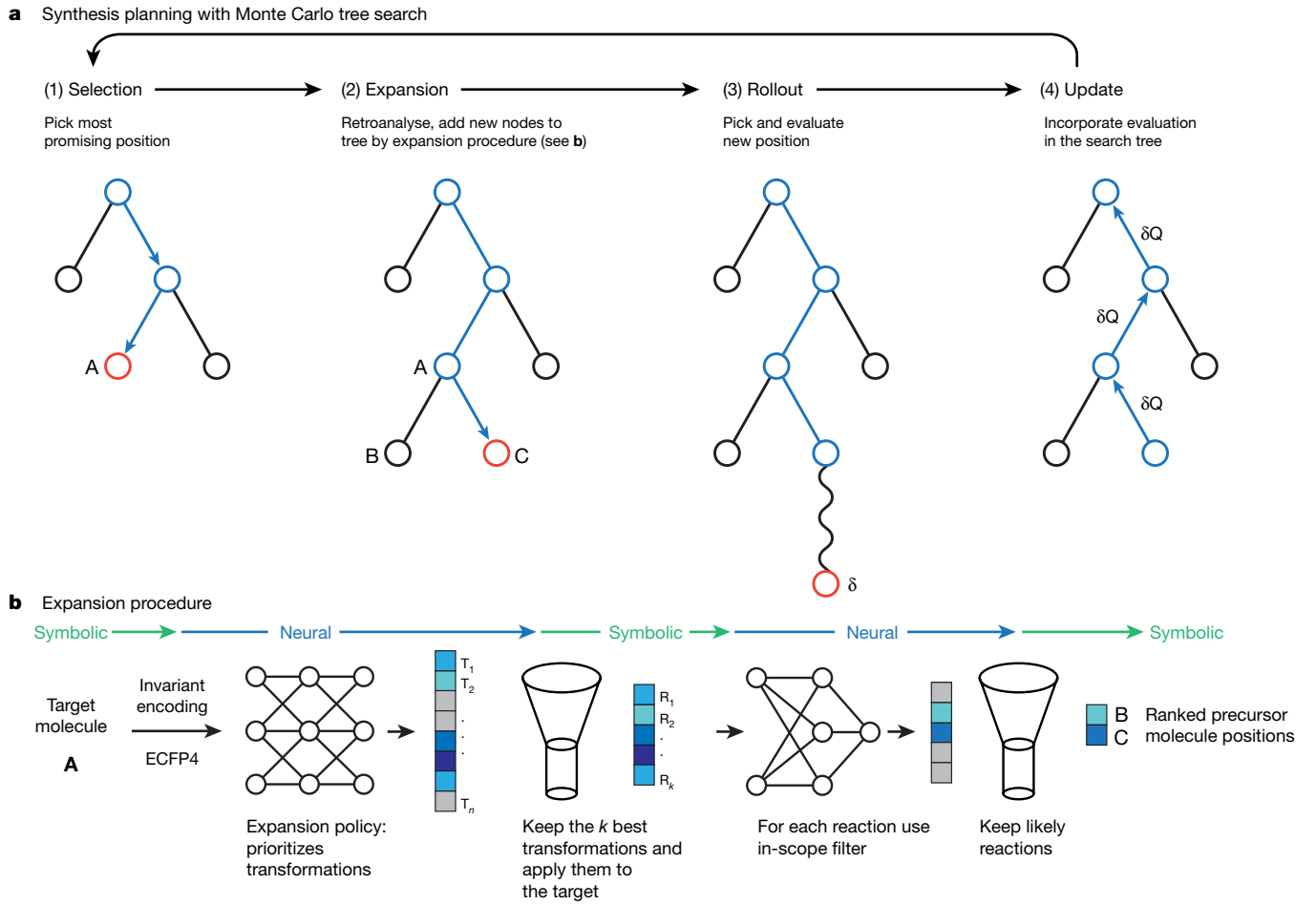


Figure 2 | Schematic of MCTS methodology. a, MCTS searches by iterating over four phases. In the selection phase (1), the most urgent node for analysis is chosen on the basis of the current position values. In phase (2) this node may be expanded by processing the molecules of the position A with the expansion procedure (**b**), which leads to new positions B and C, which are added to the tree. Then, the most promising new position is chosen, and a rollout phase (3) is performed by randomly sampling transformations from the rollout policy until all molecules are solved or a certain depth is exceeded. In the update phase (4), the position values are updated in the current branch to reflect the result of the rollout. **b**, Expansion procedure. First, the molecule (**A**) to retroanalyse is converted to a fingerprint and fed into the policy network, which returns a probability distribution over all possible transformations (T_1 to T_n). Then, only the k most probable transformations are applied to molecule **A**. This yields the reactants necessary to make **A**, and thus complete reactions R_1 to R_k. For each reaction, the reaction prediction is performed using the in-scope filter, returning a probablity score. Improbable reactions are then filtered out, which leads to the list of admissible actions and corresponding precursor positions B and C.

(1) Selection. In the first 3N-MCTS phase, starting at the root node (the target molecule) of the search tree, the algorithm sequentially selects the most promising next position within the tree until a leaf node is reached (Fig. 2a). The algorithm balances the selection of high-value positions and unexplored positions. If a leaf node is visited for the first time, it is directly evaluated by a rollout. If it is visited for the second time, it is expanded by processing via the expansion policy.

(2) Expansion. Now, the possible transformations determining the follow-up positions of the current position are selected by applying the expansion procedure. The predicted follow-up positions are added to the tree as children of the leaf node, and the most promising position is selected for rollout.

(3) Rollout. This phase starts with checking the status of the position. If it is already solved, the algorithm directly receives a reward greater than 1 to encourage exploitation. Non-terminal states are subjected to a rollout, where actions are sampled from the rollout network recursively, until the state has been deconstructed into building blocks or a maximal depth is reached.

(4) Update. If a solution has been found during rollout, a reward of 1 is received. Partial rewards are given if some, but not all, molecules in the state are solved. If no solution was found, a reward of –1 is received. Here, bespoke scoring functions for the problem at hand, such as process chemistry or small-scale medicinal chemistry, can also be supplied. Eventually, the tree is updated to incorporate the achieved reward by updating the position values.

These four phases of 3N-MCTS are iterated until a time budget or maximal iteration count is exceeded. Finally, to obtain the synthesis plan, we repeatedly select the retrosynthetic step with the highest value until a solved position is reached, or a maximum depth has been exceeded, in which case the problem is unsolved.

Evaluating the performance characteristics of 3N-MCTS

To evaluate the performance of 3N-MCTS, we compare our algorithm to the state-of-the-art search method, which is BFS with the hand-coded SMILES$^{3/2}$ heuristic cost function ('heuristic BFS')[13]. This function assigns the lowest cost to steps that split up the molecule into equally sized parts. Additionally, we perform BFS with the cost calculated by the policy network ('neural BFS'). All algorithms use the same set of automatically extracted transformations. The evaluation is again time-split, as follows. Models were trained only on data published before 2015. As test data, only molecules first reported in or after 2015 were considered (which were not contained in the training dataset). Provided with the target molecules, the algorithms then had to find a synthesis route to given building blocks.

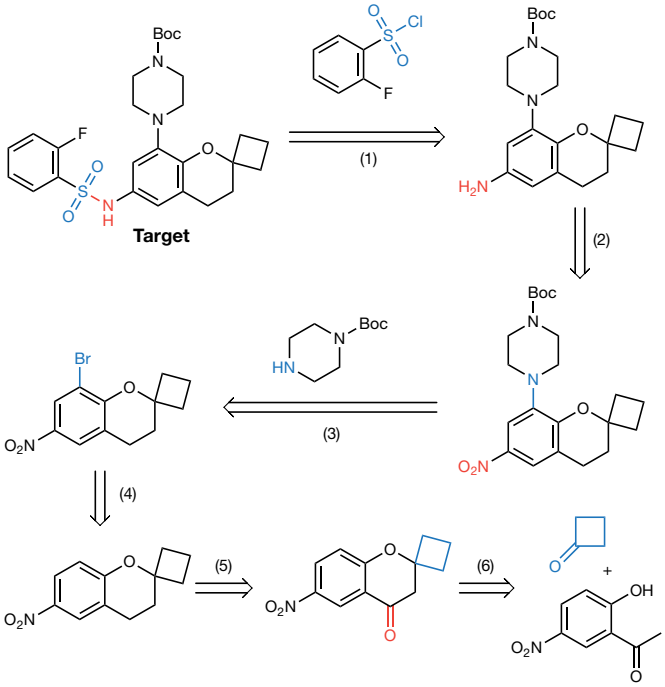


Figure 3 | An exemplary six-step synthesis route for an intermediate in a drug candidate synthesis. This route is identical to the published one[44] and was found by our algorithm autonomously within 5.4 s. The affected functional groups in each step are marked blue or red.

Figure 3 shows an exemplary six-step route for an intermediate of a drug candidate synthesis reported in 2015, which was found by our algorithm in 5.4 s. It matches the published route[44]. Several hundred additional exemplary retrosynthetic routes found by the MCTS algorithm for molecules first synthesized in or after 2015 are deposited in Supplementary Information (see also Extended Data Fig. 2).

Quantitative evaluation

Surprisingly, in the past, neither hand-coded nor automatically extracted retrosynthetic systems have been validated at scale in a statistical way. We quantitatively assessed the performance characteristics of the different search algorithms by finding synthesis routes for 497 diverse molecules first reported in or after 2015 to known building blocks (see Fig. 4).

MCTS already solves more than 80% of the test set with a time limit of 5 s per target molecule, compared to 40% with neural BFS and 0% for heuristic BFS. MCTS solved 92% of the test set with a limit of 60 s per molecule, whereas neural BFS solved 71%, and heuristic BFS solved 4%. Even at much longer runtimes of 20 min per molecule, heuristic and neural BFS are not able to compete with MCTS. Provided with infinite runtime, however, the algorithms will converge to the same performance. The molecules that MCTS failed to solve could not be solved by the BFS algorithms either. When looking beyond the first (top 1) retrieved route, MCTS and BFS find similar alternative routes, and do not differ much in terms of route diversity (see Supplementary Information section 2).

To determine which MCTS components are responsible for its superior performance, we compared MCTS against several related search algorithms (see Table 1) at a runtime limit of 3 × 300 s (three restarts). MCTS in conjunction with the expansion policy network solved the highest number of retrosynthetic targets. On average, MCTS required the least amount of time per molecule to find a solution (entry 1). Plain Monte Carlo search randomly selects transformations using the expansion policy network, without building a tree. The Monte Carlo search (entry 2) solved 89.54% of the test set. UCT is an MCTS variant that uses the expansion policy network only to narrow down the possible transformations, but not to guide the search via the predicted probability of the transformation[28]. In this way, 87.12% of the test set is solved. BFS using a cost function based on the expansion policy network solved only 84.24%, highlighting the importance of rollouts. The traditional approach, BFS with a hand-designed heuristic cost function, solves only 45.6% of the test set, and needs 433.4 s on average to find a solution. These results suggest that all components of 3N-MCTS (building a tree, reducing the branching factor via the expansion policy, guiding the search with the expansion policy, and using rollouts) contribute to its superior performance. We also found 3N-MCTS to be robust towards the choice of the MCTS parameters (see Supplementary Information section 1).

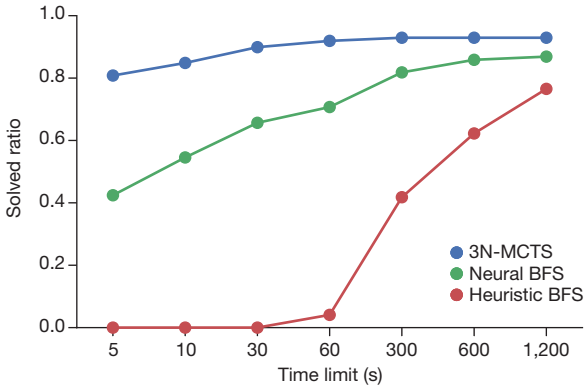


Figure 4 | Influence of the time per query on performance. The maximum number of steps is 100,000.

Assessing route quality via double-blind AB tests

The central criticism of retrosynthesis systems has been that the proposed routes often contain what chemists immediately recognize as chemically unreasonable steps. Therefore, to assess the quality of the solutions we conducted two AB tests, in which 45 graduate-level organic chemists from two world-leading organic chemistry institutes in China and Germany had to choose one of two routes leading to the same molecule on the basis of personal preference and synthetic plausibility. The tests were double-blind, meaning that neither the participants nor the conductors were aware of the origin of the routes. The test molecules were selected randomly from a set of drug-like compounds first published in or after 2015 (see Supplementary Information for the entire list of the targets and routes).

In the first test, the participants had the choice between a route reported by expert chemists in the literature, and a route generated by our 3N-MCTS algorithm for the same target molecule. Routes to nine different target molecules were offered. Routes towards the same molecule were required to have the same number of steps.

Here, one might expect the participants to clearly identify the routes suggested by the machine as inferior. Surprisingly, this is not the case. We found that the experts did not significantly ($P = 0.26$) prefer the literature route (43.0%) over our program's route (57.0%). Figure 5a shows the preference ratios for the individual routes. Here, the preference is generally balanced, with a slight trend towards MCTS. In some cases, the participants have clear preferences (see Fig. 5c and Extended Data Fig. 3 for examples where MCTS was not preferred).

Table 1 | Experimental results

Entry	Search method	Policy*	Percentage solved	Time (seconds per molecule)
1	MCTS	Neural	95.24 ± 0.09	13.0
2	MC	Neural	89.54 ± 0.59	275.7
3	UCT	Neural	87.12 ± 0.29	30.4
4	BFS	Neural	84.24 ± 0.09	39.1
5	BFS	SMILES$^{3/2}$	55.53 ± 2.02	422.1

The time budget was 300 s and 100,000 iterations for MCTS or 300 s and 100,000 expansions for BFS, per molecule. Three restarts were carried out.
*In the BFS, this is the cost function.

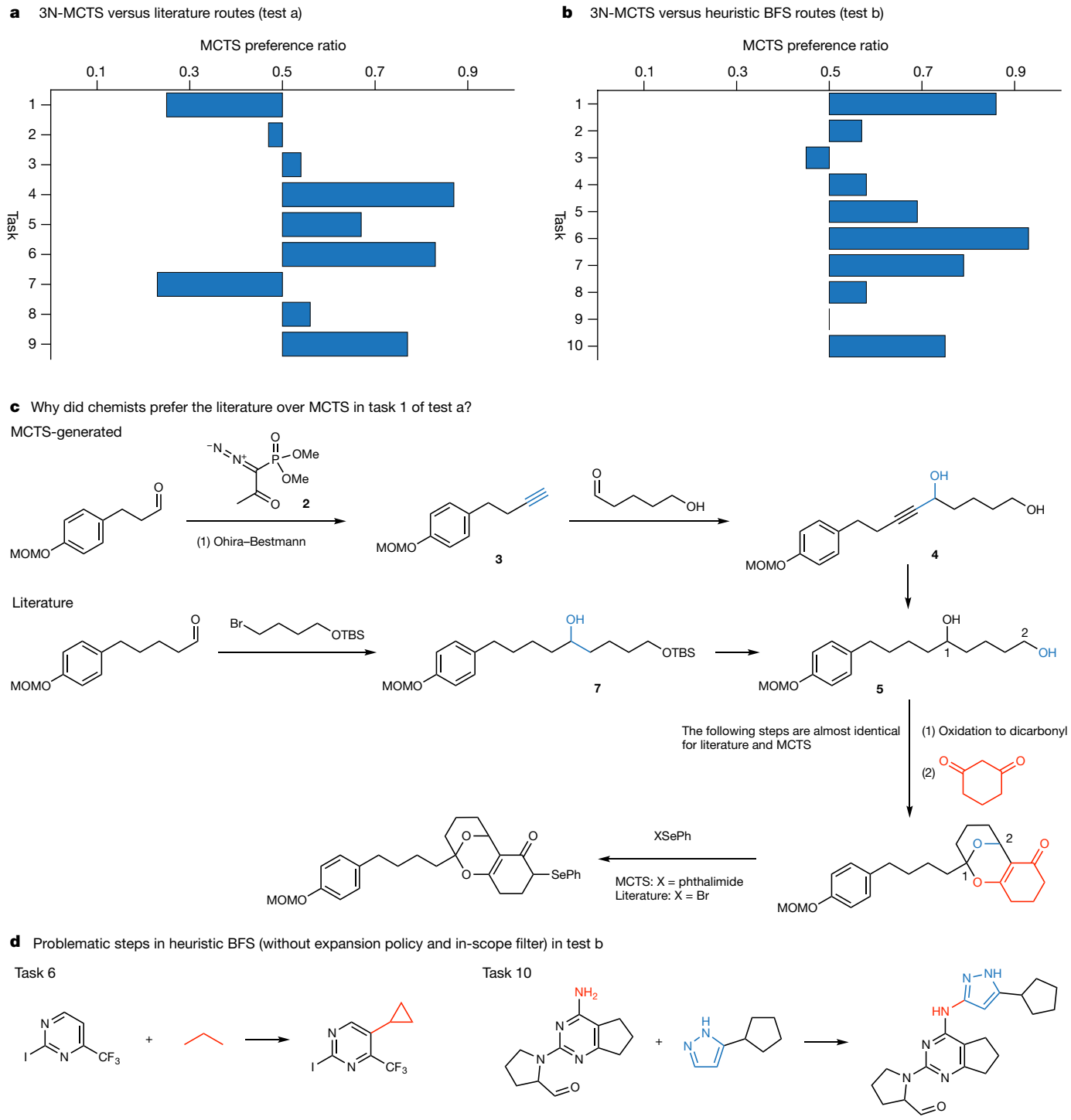


Figure 5 | Double-blind AB testing of MCTS-derived routes against literature and BFS routes. a, Chemists did not significantly prefer literature routes over routes found by MCTS (Wilcoxon signed-rank test, $P = 0.26$). **b**, Chemists significantly prefer routes found by 3N-MCTS over routes generated by heuristic BFS without a policy network and an in-scope filter (Wilcoxon signed-rank test, $P = 0.01$). A ratio above 0.5 indicates that more than 50% of participants preferred the MCTS solution. **c**, In this AB example (task 1 of test a), the chemists preferred the literature route proceeding via a Grignard reaction, in contrast to the MCTS route, which was proposed to proceed via Seyferth–Gilbert homologation with the Ohira–Bestmann reagent. Although the MCTS route is chemically reasonable, it uses less-conventional chemistry in this case. The subsequent key steps to build the annulated cycle are the same for both MCTS and the literature. **d**, Without applying the expansion policy and the in-scope filter to select the best reactions, heuristic BFS produces the typical errors traditionally criticized in retrosynthetic systems. That is, the expert system tries to apply rules that are overgeneral and will not work in this molecular context.

In the second test, the participants had to report their preferences for either routes found by 3N-MCTS or routes generated by a baseline system, which uses heuristic BFS and the same transformation rules as 3N-MCTS. However, it lacks a policy network to preselect promising transformations and an in-scope filter to exclude unlikely steps. Here, the participants significantly ($P = 0.01$) preferred the routes generated by the MCTS algorithm (68.2%) over the baseline system (31.8%). We attribute the preference towards the

3N-MCTS-generated routes to lower frequencies of unreasonable steps (see Fig. 5b and d).

Discussion

We have shown that MCTS combined with deep neural networks and symbolic rules can be used effectively to perform chemical synthesis planning. In contrast to earlier work, our purely data-driven approach can be initially set up within a few days without the need for tedious and biased expert encoding or curation, and is applicable to discipline-scale datasets. Our approach solves more problems and is faster than established search methods. Furthermore, it also performs better qualitatively. In the past, retrosynthetic systems have been criticized for producing more noise than signal. We observed that traditional heuristic BFS without neural network guiding did lead to many unreasonable steps being proposed in the routes, while the 3N-MCTS approach proposed more reasonable routes. This is supported by double-blind AB experiments, where the participating organic chemists showed clear preference towards 3N-MCTS over the traditional approach. Finally, our double-blind AB tests suggest that, for the first time, organic chemists should consider the quality of retrosynthetic routes generated by a machine to be on par with reported routes for molecules of practical relevance.

Limitations and frontiers

Nevertheless, it would be premature to consider computer-aided synthesis a solved problem, as challenges remain. First, natural product synthesis is currently beyond the capabilities of our method. The sparsity of the training data in this area remains a fundamental challenge for deep learning approaches[45]. However, natural products are also challenging for the best human chemists, as they can behave unpredictably, and often require intense methodology development[46]. Natural product synthesis may be solvable by stronger, but slower-reasoning, algorithms that could be used to invent reactions[41,47].

Another important challenge is the reliable prediction of stereochemical outcomes. While our approach is able to treat stereoinformation, the most important part, predicting enantiomeric or diastereomeric ratios quantitatively, remains an open challenge. Convincing global approaches for the quantitative prediction of enantiomeric or diastereomeric ratios over a wide range of different reactions without recourse to expensive quantum-mechanical calculations[48] have not been reported. However, they could be addressed with stereochemistry-aware descriptors. Furthermore, our system currently does not take reaction mechanisms, equilibria between different forms, such as tautomers, or three-dimensional structures into account, which can be crucial in natural product synthesis. Also, we do not at present perform reaction condition prediction[49].

Outlook

For the past 60 years, experts have been trying to dictate the rules of chemistry to computers via hand-coded heuristics. Instead, we anticipate that equipping machines with strong, general planning algorithms, symbolic representations, and the means to learn autonomously from the rich history of chemistry will be crucial to allowing the machine to become accepted as a valuable assistant in chemical synthesis, which is central to solving humanity's most pressing problems in agriculture, healthcare and material science.

Online Content Methods, along with any additional Extended Data display items and Source Data, are available in the online version of the paper; references unique to these sections appear only in the online paper.

Received 14 August 2017; accepted 31 January 2018.

1. Clayden, J., Greeves, N., Warren, S. & Wothers, P. *Organic Chemistry* 2nd edn (Oxford Univ. Press, 2008).
2. Brückner, R. *Reaktionsmechanismen: Osganische Reaktionen, Stereochemie, Moderne Synthesemethoden* (Springer, 2014).
3. Robinson, R. LXIII. A synthesis of tropinone. *J. Chem. Soc. Trans.* **111,** 762–768 (1917).
4. Corey, E. & Cheng, X. *The Logic of Chemical Synthesis* (Wiley, 1989).
5. Kurti, L. & Czakó, B. *Strategic Applications of Named Reactions in Organic Synthesis* (Elsevier, 2005).
6. Evans, J. in *The Oxford Handbook of Thinking and Reasoning* (eds Holyoak, K. J. & Morrison, R. G.) 115–133 (Oxford Univ. Press, 2012).
7. Collins, K. D. & Glorius, F. A robustness screen for the rapid assessment of chemical reactions. *Nat. Chem.* **5,** 597–601 (2013).
8. Ley, S. V., Fitzpatrick, D. E., Ingham, R. & Myers, R. M. Organic synthesis: march of the machines. *Angew. Chem. Int. Ed.* **54,** 3449–3464 (2015).
9. Schneider, P. & Schneider, G. De novo design at the edge of chaos: miniperspective. *J. Med. Chem.* **59,** 4077–4086 (2016).
10. Segler, M. H., Kogej, T., Tyrchan, C. & Waller, M. P. Generating focussed molecule libraries for drug discovery with recurrent neural networks. *ACS Cent. Sci.* **4,** 120–131 (2018).
11. Vléduts, G. Concerning one system of classification and codification of organic reactions. Inform. *Storage Retrieval* **1,** 117–146 (1963).
12. Todd, M. H. Computer-aided organic synthesis. *Chem. Soc. Rev.* **34,** 247–266 (2005).
13. Szymkuć, S. *et al.* Computer-assisted synthetic planning: the end of the beginning. *Angew. Chem. Int. Ed.* **55,** 5904–5937 (2016).
14. Cook, A. *et al.* Computer-aided synthesis design: 40 years on. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2,** 79–107 (2012).
15. Ihlenfeldt, W.-D. & Gasteiger, J. Computer-assisted planning of organic syntheses: the second generation of programs. *Angew. Chem. Int. Edn Engl.* **34,** 2613–2633 (1996).
16. Fick, R. *Konzepte zur Syntheseplanung: Strukturelle Ähnlichkeit und Strategische Bindungen.* PhD thesis, Friedrich-Alexander-Universität (1996).
17. Ugi, I. *et al.* Models, concepts, theories, and formal languages in chemistry and their use as a basis for computer assistance in chemistry. *J. Chem. Inf. Comput. Sci.* **34,** 3–16 (1994).
18. Kayala, M. A., Azencott, C.-A., Chen, J. H. & Baldi, P. Learning to predict chemical reactions. *J. Chem. Inf. Model.* **51,** 2209–2222 (2011).
19. Minsky, M. *A Framework for Representing Knowledge.* Technical Report (Massachusetts Institute of Technology, 1974).
20. Bøgevig, A. *et al.* Route design in the 21st century: the IC*SYNTH* software tool as an idea generator for synthesis prediction. *Org. Process Res. Dev.* **19,** 357–368 (2015).
21. Law, J. *et al.* Route designer: a retrosynthetic analysis tool utilizing automated retrosynthetic rule generation. *J. Chem. Inf. Model.* **49,** 593–602 (2009).
22. Christ, C. D., Zentgraf, M. & Kriegl, J. M. Mining electronic laboratory notebooks: analysis, retrosynthesis, and reaction based enumeration. *J. Chem. Inf. Model.* **52,** 1745–1756 (2012).
23. Segler, M. H. & Waller, M. P. Neural-symbolic machine learning for retrosynthesis and reaction prediction. *Chemistry* **23,** 5966–5971 (2017).
24. Boda, K., Seidel, T. & Gasteiger, J. Structure and reaction based evaluation of synthetic accessibility. *J. Comput. Aided Mol. Des.* **21,** 311–325 (2007).
25. Ertl, P. & Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *J. Cheminform.* **1,** 8 (2009).
26. Coulom, R. Efficient selectivity and backup operators in Monte-Carlo tree search. In *Int. Conf. on Computers and Games* 72–83 (Springer, 2006).
27. Kocsis, L. & Szepesvári, C. Bandit based Monte-Carlo planning. In *17th Eur. Conf. on Machine Learning* 282–293 (Springer, 2006).
28. Browne, C. B. *et al.* A survey of Monte Carlo tree search methods. IEEE Trans. *Comput. Intell. AI Games* **4,** 1–43 (2012).
29. Sutton, R. S. & Barto, A. G. *Reinforcement Learning: An Introduction* 2nd edn (MIT Press, in the press).
30. Coulom, R. Computing "elo ratings" of move patterns in the game of go. *ICGA J.* **30,** 198–208 (2007).
31. Stern, D., Herbrich, R. & Graepel, T. Bayesian pattern ranking for move prediction in the game of Go. In *Int. Conf. on Machine Learning* 873–880 (Omni Press, 2006).
32. Maddison, C. J., Huang, A., Sutskever, I. & Silver, D. Move evaluation in Go using deep convolutional neural networks. In *3rd Int. Conf. on Learning Representations* (2015); preprint at https://arxiv.org/abs/1412.6564.
33. Clark, C. & Storkey, A. Training deep convolutional neural networks to play Go. In *32nd Int. Conf. on Machine Learning* 1766–1774 (PMLR, 2015); http://proceedings.mlr.press/v37/clark15.html.
34. Winands, M. Neural networks for video game AI. In *Artificial and Computational Intelligence in Games: Integration (Dagstuhl Seminar 15051)* Vol. 5 (eds Lucas, S. M. *et al.*) 224 (2015).
35. Silver, D. *et al.* Mastering the game of Go with deep neural networks and tree search. *Nature* **529,** 484–489 (2016).
36. *Reaxys* http://www.reaxys.com (Elsevier Life Sciences, 2017).
37. Srivastava, R. K., Greff, K. & Schmidhuber, J. Training very deep networks. In *Advances in Neural Information Processing Systems* 2377–2385 (MIT Press, 2015); preprint at https://arxiv.org/abs/1507.06228.
38. Clevert, D.-A., Unterthiner, T. & Hochreiter, S. Fast and accurate deep network learning by exponential linear units (ELUs). In *4th Int. Conf. on Learning Representations* (2016); preprint at https://arxiv.org/abs/1511.07289.
39. Sheridan, R. P. Time-split cross-validation as a method for estimating the goodness of prospective prediction. *J. Chem. Inf. Model.* **53,** 783–790 (2013).

40. Marcou, G. *et al.* Expert system for predicting reaction conditions: the Michael reaction case. *J. Chem. Inf. Model.* **55,** 239–250 (2015).
41. Segler, M. H. & Waller, M. P. Modelling chemical reasoning to predict and invent reactions. *Chemistry* **23,** 6118–6128 (2017).
42. Coley, C. W., Barzilay, R., Jaakkola, T. S., Green, W. H. & Jensen, K. F. Prediction of organic reaction outcomes using machine learning. *ACS Cent. Sci.* **3,** 434–443 (2017).
43. Murphy, K. P. *Machine Learning: a Probabilistic Perspective* (MIT Press, 2012).
44. Nirogi, R. V., Badange, R., Reballi, V. & Khagga, M. Design, synthesis and biological evaluation of novel benzopyran sulfonamide derivatives as 5-HT_6 receptor ligands. *Asian J. Chem.* **27,** 2117–2124 (2015).
45. Lake, B. M., Ullman, T. D., Tenenbaum, J. B. & Gershman, S. J. Building machines that learn and think like people. *Behav. Brain Sci.* **40,** 1–101 (2016).
46. Sierra, M. A. & de la Torre, M. C. Dead ends and detours en route to total syntheses of the 1990s. *Angew. Chem. Int. Ed.* **39,** 1538–1559 (2000).
47. Rocktäschel, T. & Riedel, S. End-to-end differentiable proving. In *Advances of Neural Information Processing Systems* (eds Guyon, I. *et al.*) 3788–3800 (Curran Associates, 2017); https://papers.nips.cc/paper/6969-end-to-end-differentiable-proving.
48. Peng, Q., Duarte, F. & Paton, R. S. Computing organic stereoselectivity—from concepts to quantitative calculations and predictions. *Chem. Soc. Rev.* **45,** 6093–6107 (2016).
49. Lin, A. I. *et al.* Automatized assessment of protective group reactivity: a step toward big reaction data analysis. *J. Chem. Inf. Model.* **56,** 2140–2148 (2016).
50. Gini, A., Segler, M., Kellner, D. & Garcia Mancheno, O. Dehydrogenative tempo-mediated formation of unstable nitrones: easy access to n-carbamoyl isoxazolines. *Chemistry* **21,** 12053–12060 (2015).

Supplementary Information is available in the online version of the paper.

Acknowledgements M.H.S.S. and M.P.W. thank the Deutsche Forschungsgemeinschaft (SFB858) for funding. M.H.S.S. and M.P.W. also thank D. Evans (RELX Intellectual Properties) and J. Swienty-Busch (Elsevier Information Systems) for the reaction dataset. We thank all AB-test participants in Shanghai and Münster, and J. Guo for assistance in AB testing. M.H.S.S. thanks M. Wiesenfeldt, the Studer group, D. Barton, S. McAnanama-Brereton, R. Vidyadharan and T. Kogej for discussions. M.P. thanks M. Winands and J. Togelius for insights.

Author Contributions M.H.S.S. conceived the project, M.P.W. and M.P. contributed ideas. M.H.S.S., M.P. and M.P.W. designed the experiments. M.H.S.S. implemented the program. M.H.S.S. and M.P.W. conducted the experiments. M.P.W. supervised the project. All authors co-wrote the manuscript.

Author Information Reprints and permissions information is available at www.nature.com/reprints. The authors declare no competing interests. Readers are welcome to comment on the online version of the paper. Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations. Correspondence and requests for materials should be addressed to M.H.S.S. (marwin.segler@wwu.de) or M.P.W. (waller@shu.edu.cn).

Reviewer Information *Nature* thanks D. Duvenaud, W. H. Green and the other anonymous reviewer(s) for their contribution to the peer review of this work.

METHODS

Chemistry. Molecules are stored in the search tree as canonical SMILES strings. If stereoinformation is desired, molecules are stored as canonical, isomeric SMILES. For processing, molecules are translated from SMILES into molecular graphs, which are vertex-labelled and edge-labelled graphs $m = (A, B)$, with atoms $a \in A$ as vertices and bonds $b \in B$ as edges. Retrosynthetic transformation rules are productions on graphs[51]. In chemical terminology, transformations are also referred to as 'named reactions'. The Chemistry Development Kit (CDK)[52] and RDKit[53] chemoinformatics libraries were used for the implementation.

Retrosynthesis as a Markov decision process. Markov decision processes (MDPs) model sequential decision processes of an agent in an environment[29]. An MDP is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R})$, with states (positions) $s \in \mathcal{S}$, actions (transformations) $a \in \mathcal{A}$, a transition model $\mathcal{T}(s, a, s')$ determining the probability $\Pr(s'|a, s)$ of reaching state s' when taking action a in state s, and a reward function $\mathcal{R}(s, a, s')$, which returns the reward when transitioning to s' via action a in state s. A policy $\pi(a|s)$ is a probability distribution over all actions given state s.

Unlike in games, such as chess or Go, where it is trivial to write down the ground truth rules (the model) of the game, querying the 'chemical environment' to find out whether an action actually leads to the desired successor state is expensive. Either a wet-laboratory experiment has to be conducted or a quantum-chemical calculation on a high level of theory has to be run, which usually takes longer than running the laboratory experiment. Learning from millions of episodes of self-play can therefore not be employed.

To avoid these expensive interactions, we therefore need to learn or construct a model of the environment to perform planning. As elaborated in the introduction, this model will be inaccurate[29,54]. Even the best human chemists' predictions can and do fail, which implies that humans also perform synthesis planning with inaccurate mental models of the chemical environment. Here, we use automatic rule extraction to determine the action set (the transformations) and the in-scope filter network to learn a transition model (which is applied in a binary way). The expansion policy network serves as a prior policy.

In this Article, a state (position) $s \in \mathcal{S}$ is a set of molecules $s = \{m_i, m_j, ...\}$. The initial state $s_0 = \{m_0\}$ contains only the target molecule m_0. Actions are then transformations (rules or productions on graphs[51]) applied to one of the molecules m in a state s. When applying a legal action a_l to a state $s_i = \{m_a, m_b, m_c\}$, it will produce a new state, for example, $s_j = \{m_d, m_e, m_b, m_c\}$.

Given a set of building-block molecules $\mathcal{B}$ (specified before the start of the search), state s_k is solved if all molecules m_i in s_k are building blocks $m_i \in \mathcal{B}$. A state is terminal either if it is solved or if no legal actions are available. Ideally, it should be possible to provide the set of building blocks $\mathcal{B}$ dynamically before the search is started. Otherwise, the system could not adapt, for example, when a building block runs out of stock. Also, a researcher may choose different sets of building blocks for each search, for example, by first trying to find the solution to a problem with molecules that are in stock in the laboratory, and afterwards considering additional molecules from chemical suppliers. This makes it challenging to define or learn value functions, because a changed set of building blocks changes the terminality of states and the reward function, which entails a change of the value function[29]. A further challenge is that the initial state (the target molecule) changes. In most games, the reward function is always the same (the rules never change, and a terminal state is always terminal).

The size of the state space can only be roughly estimated. The number of drug-like molecules, which contain a restricted set of elements and functional groups, might already exceed 10^{60} molecules[55]. However, this number excludes synthetic intermediates and organometallic and organo-main group chemistry, which add orders of magnitude to the state space size. The action space is formed by the transformations available to the system, and the legal actions are those actions that can be applied to the molecules in a state via subgraph isomorphism. Unlike for other game artificial intelligence problems, in retrosynthesis we can limit the depth of the tree to a relatively small number (here, 25) and abort the simulation as failed if a viable route cannot be found within this limit. Our trees are thus wider and less deep than for other applications.

MCTS. MCTS is a reinforcement learning approach that combines tree search with learning from simulated episodes of experience, which are obtained from interacting with a model of the environment[28,29]. MCTS has been successfully applied in problems of sequential decision making in many domains, such as games or automated theorem proving[28,56]. MCTS has several desirable features, which makes it particularly well suited for retrosynthesis. It allows the calculation of value functions focused on a particular initial state on the fly[54], and therefore does not depend strongly on heuristics. Each edge (s, a) in the search tree stores the action value $Q(s, a)$, the visit count $N(s, a)$ and a prior probability $P(s, a)$ received from the expansion policy network.

Selection. In the first MCTS phase, starting at the root node, the tree policy (equation (1)) is used to select actions. The simulation descends the tree step by step. At each step t, the next action a_t is selected from all available actions $\mathcal{A}(s_t)$ in s_t by equation (1), where $N(s_{t-1}, a_{t-1})$ is the visit count of the state-action pair that led to the current state, and c the exploration constant.

$$a_t = \underset{a \in \mathcal{A}(s_t)}{\operatorname{argmax}} \left(\frac{Q(s_t, a)}{N(s_t, a)} + cP(s_t, a) \frac{\sqrt{N(s_{t-1}, a_{t-1})}}{1 + N(s_t, a)} \right) \quad (1)$$

The inclusion of the prior probability $P(s, a)$ in the second term of equation (1) allows the system to explore the most promising lines of analysis first[57]. With repeated visits this term decays, allowing for the exploration of other options. Additionally, this allows one to take into account the confidence in the evaluation obtained via the rollout, which is expected to be noisy.

The tree policy is applied until a leaf node or a terminal node is found. If a leaf node is visited for the second time, it is expanded. Then, all non-building-block molecules $m_i \in s_t$ are processed by means of the expansion procedure. The resulting state-action pairs are added to the tree as children of the leaf node, and the most probable action according to the policy network is selected for rollout.

Expansion. During expansion, the state is processed once via the expansion procedure, and the reduced top 50 successor states are directly added to the tree. This trick can be applied because retrosynthesis is a single-player game, and we do not have to fear overlooking 'killer' moves (trap moves in which a small mistake will be exploited directly by the opponent) as much as in two-player games. Using only the reduced top 50 entails that the NP-complete subgraph isomorphism problem, which determines whether the corresponding rule can be applied to a molecule and yields the next molecule(s), needs only to be solved for at most 50 rules, instead of for all rules in the transformation rule set.

Evaluation by rollout. Before starting the rollout, the state is first checked for being terminal. A state can be terminal if it is solved. States within the tree that are already solved are called 'proved'[58]. States can also be terminal if no legal actions are available in that state. Terminal states are directly evaluated with the reward function. If the state is non-terminal, a rollout is started. During rollout, actions are sampled recursively for each molecule in the state from the top 10 actions of the rollout policy until it has been deconstructed into building blocks or a maximal recursive function call depth of d_r is exceeded. For the sake of simplicity, rollouts that completely solve the molecule are currently not stored.

The reward function $r(s)$ returns $z > 1$ if the state is proved to encourage exploitation, a reward $\in [0,1]$ depending on the ratio of molecules solved during rollout, and -1 if the state is terminal and unproven, or unsolved during rollout. Learned value functions are a possible, future alternative to rollouts. Investigations to learn value functions are currently ongoing in our laboratory.

Update. In the update phase, the action values $Q(s, a)$ and visit counts $N(s, a)$ of the edges traversed in the branch from s_t to the root node are updated. The edges gather the mean action value as in equation (2), where the indicator function $I_i(s, a)$ is 1 if the edge was played during the ith simulation and z_i is the reward received during rollout.

$$Q(s, a) = \frac{1}{N(s, a)} \sum_{i=1}^{n} I_i(s, a) z_i W(b_i) \quad (2)$$

Here, it is also possible to inject custom objective functions $W(b_i) \rightarrow \mathbb{R}$ that might assign higher rewards, for example, to shorter, convergent, atom-economic or confident branches b_i. In this work, we adjust the reward by using

$$\xi(b_i) = \text{length}(b_i) - \sum_{j=1}^{J} kP(s_j, a_j) \quad (3)$$

$$W^{L_{\max}}(b_i) = \max\left(0, \frac{L_{\max} - \xi(b_i)}{L_{\max}}\right) \quad (4)$$

where $J = \text{length}(\cdot)$ denotes the length of a branch, $P(s_j, a_j)$ is the probability of the jth action in the branch obtained from the expansion policy, $k = 0.99$ is a damping factor, and the maximal branch length $L_{\max} = 25$. Here, inclusion of the prior policy $P(s_j, a_j)$ allows us to bias the reward also towards more confident branches. An interesting scoring function to investigate further is similarity to reported reactions[59,60].

After either the time or the iteration step budget has been exhausted, the synthesis plan is selected, starting at the root node, by greedily choosing the action with the highest action value until a terminal solved state is reached, or a maximum depth has been exceeded, in which case the problem is unsolved. A maximal rollout depth of 5, an exploration constant of 3 and a reward, when proved, of 10 were employed as the MCTS parameters in the quantitative and qualitative experiments.

Automatic transformation rule extraction. Formalizing chemical knowledge by hand has been attempted. Even though it sounds simple to write down the rules of chemistry, it takes years to formalize only humble knowledge bases, it is error-prone and biased towards the knowledge of the encoding experts, and in many cases chemical systems are too complex or just not well enough understood to formally write down their limitations and scope. Like rule-based common-sense reasoning[19], this approach is considered to have exhausted its potential[15,17,18]. Additionally, given the exponential growth of chemical knowledge (it doubles roughly every 15 years), manual encoding is a hopeless endeavour.

Following our previously reported procedure[23], and building on previous work[20–22], transformation rules were therefore extracted automatically. The rules are stored using the RDKit reaction SMARTS format[53]. A very general rule definition was employed for the expansion rule set, where only the atoms of the reaction centre (including implicit hydrogen atoms and neighbouring-atom count) were extracted. The rules in the rollout set contain the reaction centre atoms (with implicit hydrogen atoms and neighbouring-atom count) and additionally the directly neighbouring atoms of the atoms in the reaction centre with their implicit hydrogen atom count. Rules were extracted only from single-step reactions with one, two or three reactants and a single product. As this work is a proof-of-concept study with the intent to radically avoid expert encoding and curation, we chose not to exclude reactions based on low yield or extreme reaction conditions, as these are quite subjective criteria. For example, a yield of just a few per cent can be sufficient if the aim is to obtain only a few milligrams of a compound for biological testing, while 90% yield is clearly unsatisfying if quantitative alternatives are available. In the future, more sophisticated approaches based on reaction classification could be employed to extract rules, for example, by grouping together similar leaving groups into a single rule[21,61,62]. Also, further investigation of directly translating from products to reactants using neural networks is called for[63].

The general advantages and limitations of automatic rule extraction have been discussed in detail elsewhere[13–15,20,21,23]. Its main disadvantages (defining the scope of reactions and competing reactivity, incorporating mechanistically needed/activating functional groups, and deciding which rules to apply first) can be addressed by learning supervised policies to predict which rules to apply[23].

The use of symbolic rules has the great advantage that it is deeply rooted in chemists' language. This makes it easy for the model to communicate its results to the human user. Furthermore, because the transformations were extracted from the literature, we can link back directly to literature precedents, which is of crucial importance for chemists.

We note that even when taking all pre-2015 rules without count restriction into account, only 82% of the reactions published in and after 2015 are covered. The missing 18% are novel reaction types. This highlights the success of chemists inventing novel methodologies, but also implies that eventually a retrosynthesis system should also be able to discover novel reactions on its own[23].

Policy networks. The neural policy networks were trained by minimizing the negative log-likelihood of selecting the transformation a that was used in the literature to make molecule m. This is essentially supervised multi-class classification. To evaluate the accuracy, reactions that are not covered in the rule set are excluded. Training was carried out using stochastic gradient descent (ADAM optimizer[64]) within 1–2 days on a single NVIDIA K80 graphics processing unit. The Keras neural network framework was employed, using Theano as the backend[65,66].

Expansion policy network. Molecules are represented by real vectors in the form of counted extended-connectivity fingerprints (ECFP4)[67], which are first modulo-folded to 1,000,000 dimensions, and then $\ln(x+1)$-preprocessed. After that, a variance threshold is applied to remove rare features, leaving 32,681 dimensions. For the machine-learning model, we used a 1+5-layer highway network with exponential linear unit (ELU) nonlinearities[37,38]. A dropout ratio of 0.3 was applied after the first affine transformation to 512 dimensions, and a dropout ratio of 0.1 was applied after each of the five highway layers. The last layer of the neural networks is a softmax, which outputs a probability distribution over all actions (transformations) $p(a|m)$, which forms the policy (see Extended Data Fig. 4).

Rollout policy network. For rollout, molecules are represented by counted ECFP4 fingerprints[67], modulo-folded to 8,192 dimensions, and then $\ln(x+1)$-preprocessed. As the rollout policy, a neural network with a single hidden dense layer with a dimensionality of 512, and ELU nonlinearity and dropout of 0.4 was used.

In-scope filter. The function of the in-scope classifier is to rapidly filter out the nonsensical reactions that plague rule-based systems, such as incorrect regioisomers in electrophilic aromatic substitutions. For this purpose, we chose a binary classifier, which is fast to evaluate. The investigation of more sophisticated, but slower, reaction prediction approaches is left to future work[18,23,41,42,68–70]. For the same reason, only the product and the reaction fingerprint serve as inputs to the classifier, although the exclusion of conditions makes the classifier underdetermined[41,42]. Reactions can selectively lead to different products under different conditions. However, the inclusion of reaction conditions as another input feature would require additional search in condition space at each step, which is not feasible given the time constraints we imposed here. In tasks where search time is not a constraint, such as in process development, reaction condition prediction could be performed at each step of the search, or be performed in a second sweep. One way to predict reaction conditions might be via reaction similarity search, which is related to how chemists use reaction databases.

Our classifier is a neural network with two branches (see Extended Data Fig. 4). The first branch embeds the reaction r_i, represented as a counted ECFP4 reaction fingerprint[40,59,71–73] ρ_i modulo-folded to 2,048 dimensions, via a single dense ELU layer. The second branch embeds the product, represented as a counted ECFP4 fingerprint φ_i, modulo-folded to 16,384 dimensions and $\ln(x+1)$-preprocessed, through a 5-layer ELU-highway network. The cosine proximity of these embeddings is then fed into a sigmoid unit to predict the probability that the reaction gives rise to the expected product.

We used two strategies to obtain negative data, as follows. First, 30 million incorrect reactions were obtained by the application of reaction rules to the reactants of reported reactions, using the same rule set as for the expansion policy[41,42,74]. Here we make the assumption that the reactants can only react in the reported way. Any product generated by rule application not matching the reported product is considered to be a failed product. With this approach, for example, wrong regioisomers can be generated. We note that these 'negative' reactions generated in this way capture the cases where a naive, contextless rule-based system would fail. Furthermore, 70 million negative training data points were generated by perturbing tuples (ρ_i, ϕ_i) to (ρ_i, ϕ_j), where $i \neq j$, by random sampling. Random sampling gives a small performance boost of 0.0025 score points in the area under the receiver operation characteristic curve (AUROC) and 0.0072 points in the area under the precision-recall curve. Training data were generated only from reactions published before 2015, while test data were generated from data published in or after 2015. The classifier was trained by minimizing negative log-likelihood using the ADAM optimizer. Extended Data Fig. 1 shows the receiver operation characteristic curve of the classifier. A value of 0.9 was selected as the decision threshold for our classifier. Here, the classifier has a false negative rate of 14% and a false positive rate of 1.5%

Given that the in-scope filter learns to embed molecules in a vector space close to the reactions that were used to make them, it would be interesting to investigate if this can be directly used for nearest-neighbour search of molecules in reaction-rule space, which can also be described as label embedding.

Rediscovering electronic properties. To study what the in-scope filter has learned, we conducted two experiments. First, we studied Diels–Alder reactions of cyclopentadiene with various dienophiles. The reactions were submitted to the in-scope filter, and the raw logit scores (the output of the neural network before applying the final sigmoid function) were calculated. As a comparison, the structures of the dienophiles were optimized and the energies of the LUMO were calculated via density functional theory (BP86-D3BJ/def2-SVP) to capture the qualitative trend, using the ORCA3 software[75]. Extended Data Fig. 5a shows the correlation of the LUMO energy with the logit score, which has an $r^2 = 0.74$. Additionally, parabrominations via electrophilic aromatic substitution were studied (Extended Data Fig. 5b). Here, the logit scores were correlated to Hammond electronic parameters, and an $r^2 = 0.78$ was found. This indicates that the in-scope filter correlates with basic electronic properties, following the expected behaviour of the respective reaction mechanisms. This is remarkable, as its input features (ECFP4-based molecular and reaction fingerprints) do not contain electronic information.

Performance evaluation studies. *Baselines.* In BFS, each branch is added to a priority queue, which is sorted by cost. In heuristic BFS, this cost function is the SMILES$^{3/2}$ heuristic, which is used as reported[13]. In neural BFS, the cost is calculated as $f(b) = \sum_{i=0}^{s \in b} (1 - P(a_i|s_i))$, where $P(a_i|s_i)$ is the probability of that transformation calculated by the expansion policy. Evaluation studies were performed using the central processing unit of a 24-core commodity cluster node using a single search thread. To provide a more meaningful comparison no graphical processing unit was used for the evaluation studies.

Building block and test molecule selection. The building blocks have to be selected before the search is started, and could be molecules on stock in the laboratory, known in the literature, or commercially available chemicals. Here we use a set of 423,731 molecules, containing 84,253 building blocks from three major chemical suppliers (SigmaAldrich, AlfaAesar and Acros), obtained from the ZINC database (http://zinc15.docking.org/) and 339,478 molecules from the Reaxys database, which have been used as reactants at least five times before 2015. To obtain a set of target molecules for the quantitative evaluation that contains different scaffolds, first all molecules reported after 2014 were clustered using the ECFP6-based Butina algorithm[76]. Then, 497 target molecules were randomly selected from amongst the 82,673 different cluster cores.

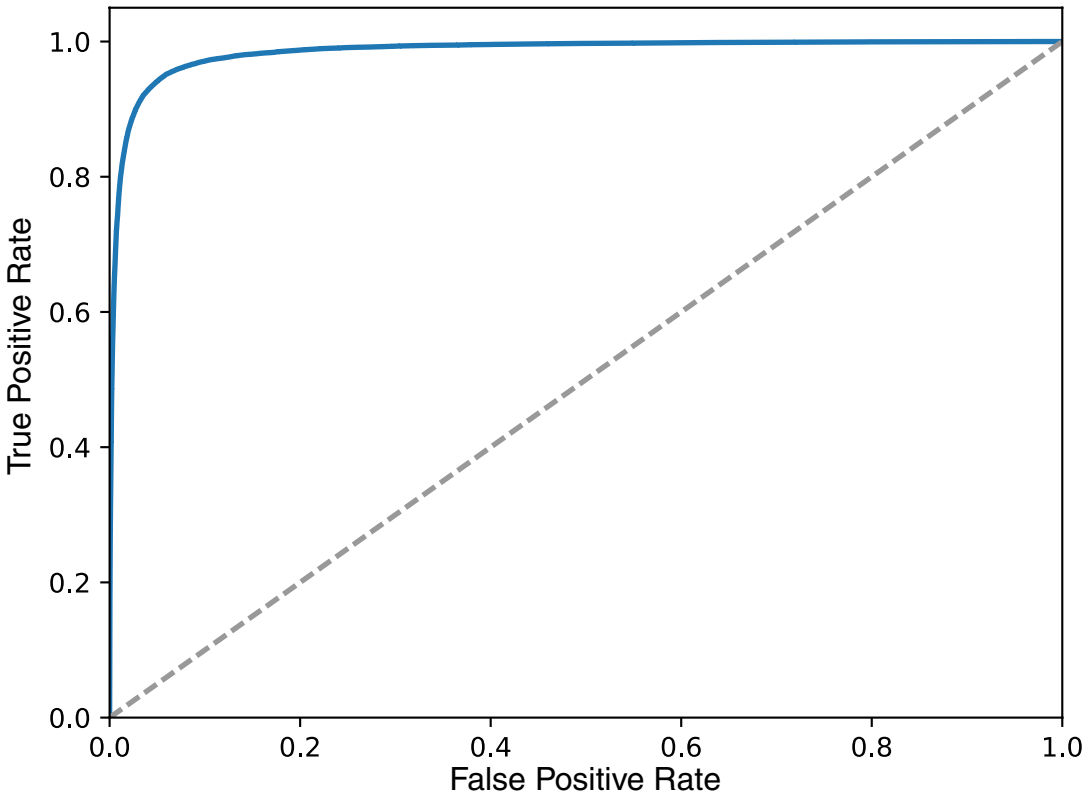
AB testing. The participants in the AB tests were 45 postgraduate students who had specialized in organic chemistry at the Institute of Organic Chemistry at Westfälische Wilhelms-Universität Münster and Shanghai Institute of Organic Chemistry. The study was conducted in a double-blind setup. During the test, neither the participants nor the conductors were aware of the origin of the route. Statistical significance was tested via the Wilcoxon signed-rank test.

3N-MCTS versus literature. In the comparison of 3N-MCTS with the literature, the expectation would be that experts prefer the literature option. In 128 AB tests, the experts preferred the literature route in 43.0% and MCTS in 57.0% (Wilcoxon signed-rank test on paired data, $P = 0.26$). The null hypothesis that both datasets stem from the same source cannot be rejected.

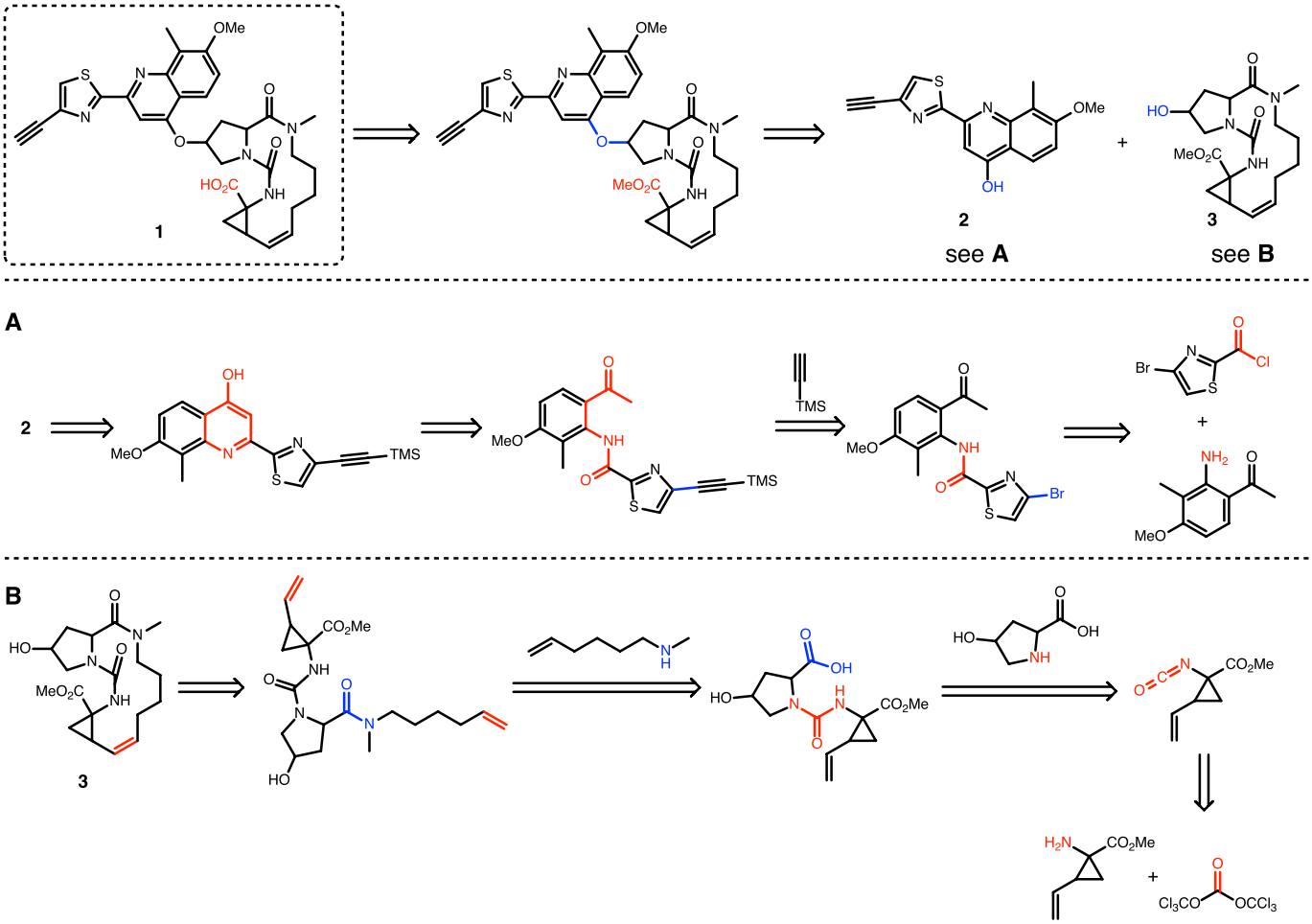
3N-MCTS versus heuristic BFS. Here, 68.2% of the participants preferred 3N-MCTS generated solutions, whereas only 31.8% preferred heuristic-BFS-generated solutions in 129 submitted tests. The experts strongly favour MCTS, the null hypothesis of indistinguishable sources (50% preference for each) can clearly be rejected (Wilcoxon signed-rank test on paired data, $P = 0.01277$).

Data availability. The reaction dataset used in this study is provided by Elsevier Information Systems GmbH under licence.

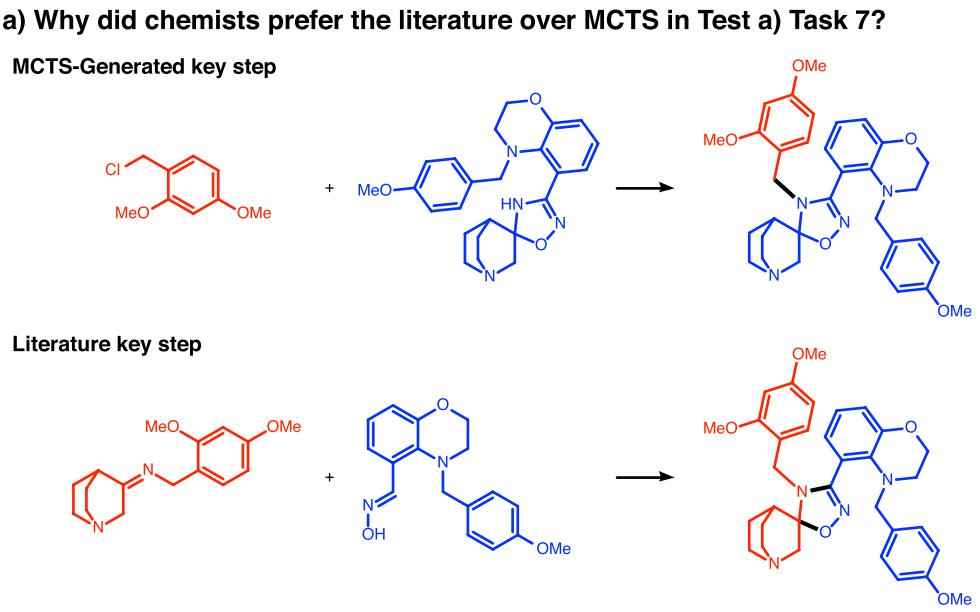
51. Andersen, J. L., Flamm, C., Merkle, D. & Stadler, P. F. Generic strategies for chemical space exploration. *Int. J. Comput. Biol. Drug Des.* **7,** 225–258 (2014).
52. Steinbeck, C. *et al.* Recent developments of the chemistry development kit (CDK)-an open-source Java library for chemo- and bioinformatics. *Curr. Pharm. Des.* **12,** 2111–2120 (2006).
53. Landrum, G. *RDKit: Open-Source Cheminformatics* http://www.rdkit.org.
54. Silver, D. *Reinforcement Learning and Simulation-Based Search.* PhD thesis, Univ. Alberta (2009).
55. Reymond, J.-L., Ruddigkeit, L., Blum, L. & van Deursen, R. The enumeration of chemical space. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2,** 717–733 (2012).
56. Färber, M., Kaliszyk, C. & Urban, J. Monte Carlo connection prover. Preprint at https://arxiv.org/abs/1611.05990 (2016).
57. Rosin, C. D. Multi-armed bandits with episode context. *Ann. Math. Artif. Intell.* **61,** 203–230 (2011).
58. Winands, M. H., Björnsson, Y. & Saito, J.-T. Monte-Carlo tree search solver. In *Int. Conf. on Computers and Games* 25–36 (Springer, 2008).
59. Schneider, N., Lowe, D. M., Sayle, R. A. & Landrum, G. A. Development of a novel fingerprint for chemical reactions and its application to large-scale reaction classification and similarity. *J. Chem. Inf. Mod.* **55,** (2015).
60. Coley, C. W., Rogers, L., Green, W. H. & Jensen, K. F. Computer-assisted retrosynthesis based on molecular similarity. *ACS Cent. Sci.* **3,** 1237–1245 (2017).
61. Gelernter, H., Rose, J. R. & Chen, C. Building and refining a knowledge base for synthetic organic chemistry via the methodology of inductive and deductive machine learning. *J. Chem. Inf. Comput. Sci.* **30,** 492–504 (1990).
62. Rose, J. R. & Gasteiger, J. Horace: an automatic system for the hierarchical classification of chemical reactions. *J. Chem. Inf. Comput. Sci.* **34,** 74–90 (1994).
63. Liu, B. *et al.* Retrosynthetic reaction prediction using neural sequence-to-sequence models. *ACS Cent. Sci.* **3,** 1103–1113 (2017).
64. Kingma, D.P. & Ba, J. ADAM: a method for stochastic optimization. In *3rd Int. Conf. for Learning Representations*; preprint at https://arxiv.org/abs/1412.6980 (2015).
65. Chollet, F. *et al. Keras* https://github.com/fchollet/keras (2015).
66. The Theano Development Team Theano: a Python framework for fast computation of mathematical expressions. Preprint at https://arxiv.org/abs/1605.02688 (2016).
67. Rogers, D. & Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **50,** 742–754 (2010).
68. Wei, J. N., Duvenaud, D. & Aspuru-Guzik, A. Neural networks for the prediction of organic chemistry reactions. *ACS Cent. Sci.* **2,** 725–732 (2016).
69. Socorro, I. M. & Goodman, J. M. The ROBIA program for predicting organic reactivity. *J. Chem. Inf. Model.* **46,** 606–614 (2006).
70. Satoh, H. & Funatsu, K. Sophia, a knowledge base-guided reaction prediction system—utilization of a knowledge base derived from a reaction database. *J. Chem. Inf. Comput. Sci.* **35,** 34–44 (1995).
71. Patel, H., Bodkin, M. J., Chen, B. & Gillet, V. J. Knowledge-based approach to de novo design using reaction vectors. *J. Chem. Inf. Model.* **49,** 1163–1184 (2009).
72. Zhang, Q.-Y. & Aires-de Sousa, J. Structure-based classification of chemical reactions without assignment of reaction centers. *J. Chem. Inf. Model.* **45,** 1775–1783 (2005).
73. Polishchuk, P. *et al.* Structure–reactivity modeling using mixture-based representation of chemical reactions. *J. Comput. Aided Mol. Des.* **31,** 829–839 (2017).
74. Carrera, G. V., Gupta, S. & Aires-de Sousa, J. Machine learning of chemical reactivity from databases of organic reactions. *J. Comput. Aided Mol. Des.* **23,** 419–429 (2009).
75. Neese, F. The ORCA program system. *WIREs Comput. Mol. Sci.* **2,** 73–78 (2012).
76. Butina, D. Unsupervised data base clustering based on Daylight's fingerprint and Tanimoto similarity: a fast and automated way to cluster small and large data sets. *J. Chem. Inf. Comput. Sci.* **39,** 747–750 (1999).
77. Parsy, C. C. *et al.* Discovery and structural diversity of the hepatitis C virus NS3/4a serine protease inhibitor series leading to clinical candidate IDX320. *Bioorg. Med. Chem. Lett.* **25,** 5427–5436 (2015).



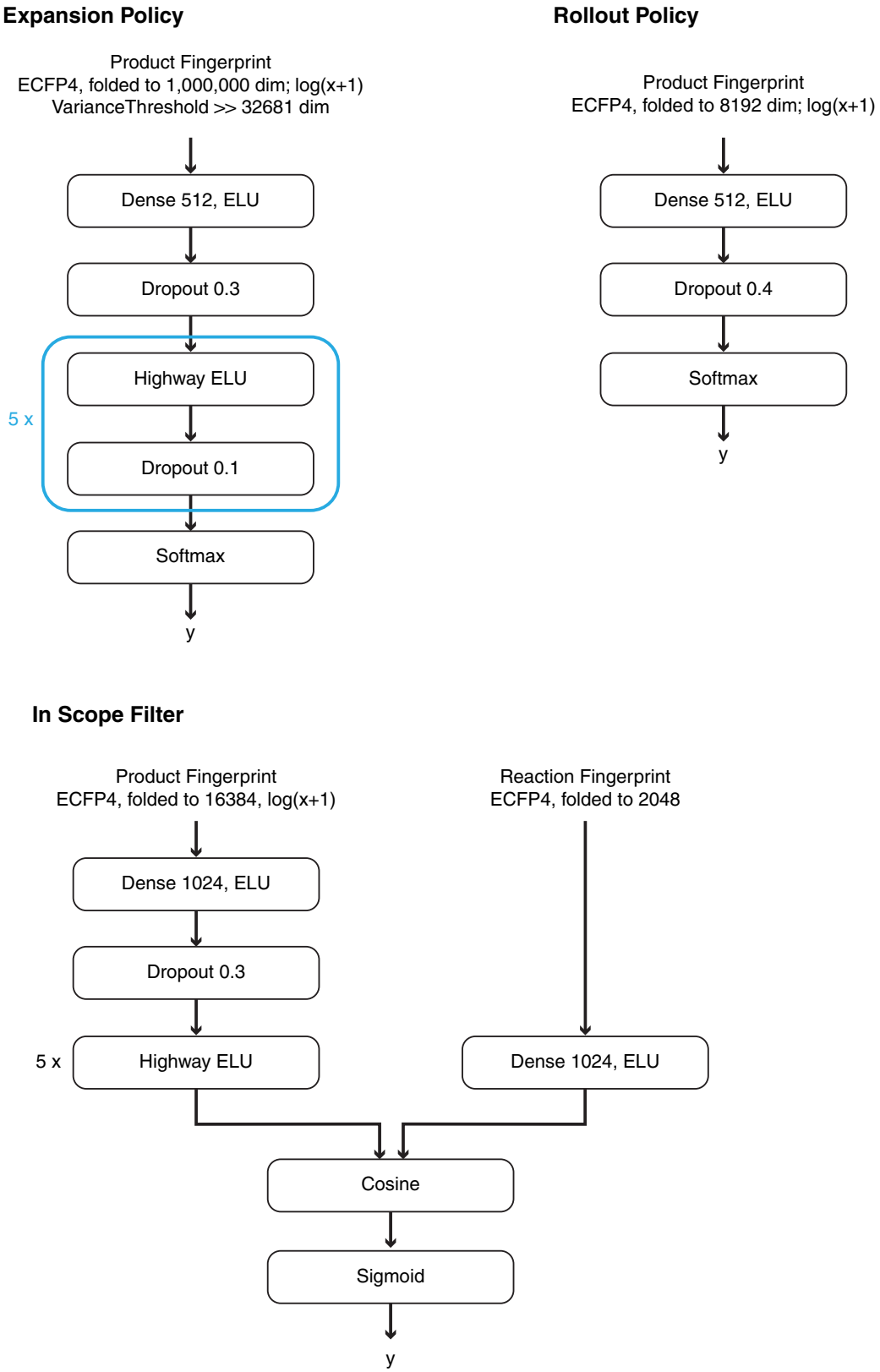
Extended Data Figure 1 | Receiver operation characteristic curve for the in-scope filter. The area under the curve is 0.99.



Extended Data Figure 2 | An exemplary 10-step synthesis route for a complex intermediate in a drug synthesis. It resembles the published route[77] (with intermediates **A** and **B**) and was found by our algorithm autonomously within 30 s. The target was not contained in the training set.

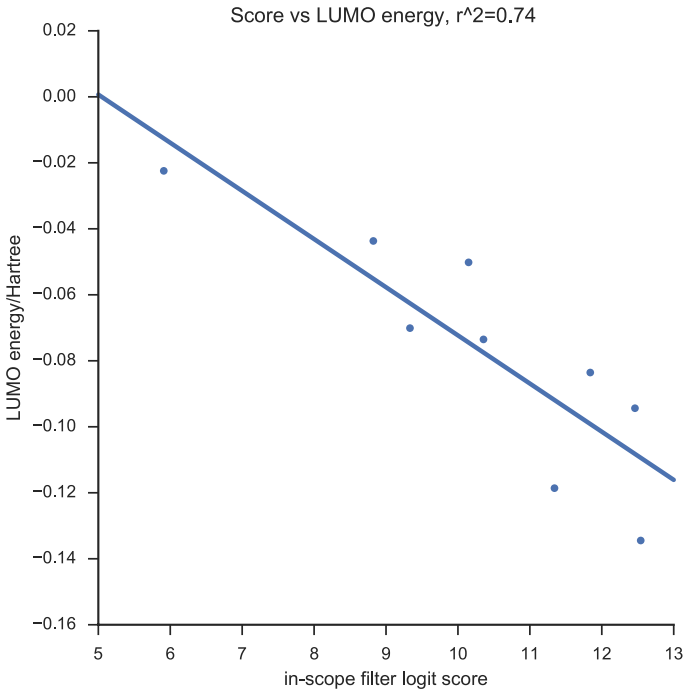


Extended Data Figure 3 | Example of reaction used in the AB testing, where the MCTS-derived route was less favoured. In this task, the participants preferred the literature solution, as its key step was presumably perceived to be more convergent.

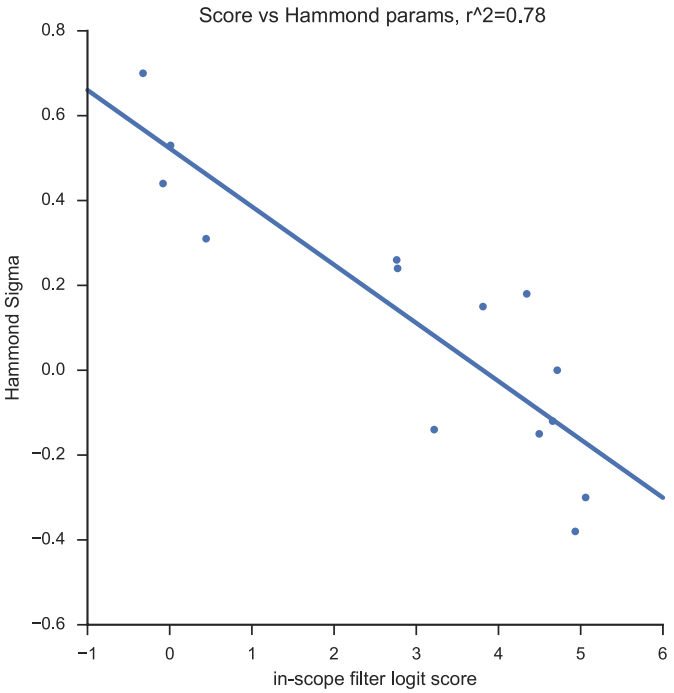


Extended Data Figure 4 | Architectures of the employed neural networks. ('dim', dimensions.)

a) Diels-Alder reactions with Cyclopentadiene



b) para-Bromination of benzenes



Extended Data Figure 5 | Rediscovering physicochemical properties with the in-scope filter. The output logit score of the neural network correlates surprisingly well with calculated quantum-mechanical properties (LUMO energies, in Hartree) in Diels–Alder reactions ($r^2 = 0.74$) (**a**) and with empirically measured Hammond parameters in electrophilic brominations ($r^2 = 0.78$) (**b**), even though the input features (ECFP4 fingerprints) do not contain electronic information.

Extended Data Table 1 | Metrics for the supervised neural network policies

Policy	# rules	Coverage	Matching rules/mol[b]	Accuracy[a]	top10Acc[a]	top50Acc[a]
Expansion	301,671	0.79	46,175	0.310	0.633	0.725
Rollout	17,134	0.52	321	0.501	0.891	0.964

Top10Acc/top50Acc is the ratio of correct/incorrect predictions if we allow the system to make 10 or 50 predictions.
[a]Accuracy is calculated on the molecules covered by the respective rulebase.
[b]Matching rules/mol corresponds to the branching factor.