

Integrating Deep Neural Networks and Symbolic Inference for Organic Reactivity Prediction

Wesley Wei Qian, Nathan T. Russell, Claire L. W. Simons, Yunan Luo, Martin D. Burke, Jian Peng

Submitted date: 19/01/2020 • Posted date: 21/01/2020

Licence: CC BY-NC-ND 4.0

Citation information: Qian, Wesley Wei; Russell, Nathan T.; Simons, Claire L. W.; Luo, Yunan; Burke, Martin D.; Peng, Jian (2020): Integrating Deep Neural Networks and Symbolic Inference for Organic Reactivity Prediction. ChemRxiv. Preprint. <https://doi.org/10.26434/chemrxiv.11659563.v1>

Accurate in silico models for the prediction of novel chemical reaction outcomes can be used to guide the rapid discovery of new reactivity and enable novel synthesis strategies for newly discovered lead compounds. Recent advances in machine learning, driven by deep learning models and data availability, have shown utility throughout synthetic organic chemistry as a data-driven method for reaction prediction. Here we present a machine-intelligence approach to predict the products of an organic reaction by integrating deep neural networks with a probabilistic and symbolic inference that flexibly enforces chemical constraints and accounts for prior chemical knowledge. We first train a graph convolutional neural network to estimate the likelihood of changes in covalent bonds, hydrogen counts, and formal charges. These estimated likelihoods govern a probability distribution over potential products. Integer Linear Programming is then used to infer the most probable products from the probability distribution subject to heuristic rules such as the octet rule and chemical constraints that reflect a user's prior knowledge. Our approach outperforms previous graph-based neural networks by predicting products with more than 90% accuracy, demonstrates intuitive chemical reasoning through a learned attention mechanism, and provides generalizability across various reaction types. Furthermore, we demonstrate the potential for even higher model accuracy when complemented by expert chemists contributing to the system, boosting both machine and expert performance. The results show the advantages of empowering deep learning models with chemical intuition and knowledge to expedite the drug discovery process.

File list (1)

symbolic_rxn.pdf (19.02 MiB)

[view on ChemRxiv](#) • [download file](#)

Integrating Deep Neural Networks and Symbolic Inference for Organic Reactivity Prediction

Wesley Wei Qian^{1,†}, Nathan Russell^{1,†}, Claire L. W. Simons², Yunan Luo¹,
Martin D. Burke², Jian Peng^{1,*}

¹ Department of Computer Science, University of Illinois at Urbana-Champaign

² Department of Chemistry, University of Illinois at Urbana-Champaign

† Equal contribution

* Corresponding author: jianpeng@illinois.edu

Abstract

Accurate *in silico* models for the prediction of novel chemical reaction outcomes can be used to guide the rapid discovery of new reactivity and enable novel synthesis strategies for newly discovered lead compounds. Recent advances in machine learning, driven by deep learning models and data availability, have shown utility throughout synthetic organic chemistry as a data-driven method for reaction prediction. Here we present a machine-intelligence approach to predict the products of an organic reaction by integrating deep neural networks with a probabilistic and symbolic inference that flexibly enforces chemical constraints and accounts for prior chemical knowledge. We first train a graph convolutional neural network to estimate the likelihood of changes in covalent bonds, hydrogen counts, and formal charges. These estimated likelihoods govern a probability distribution over potential products. Integer Linear Programming is then used to infer the most probable products from the probability distribution subject to heuristic rules such as the octet rule and chemical constraints that reflect a user’s prior knowledge. Our approach outperforms previous graph-based neural networks by predicting products with more than 90% accuracy, demonstrates intuitive chemical reasoning through a learned attention mechanism, and provides generalizability across various reaction types. Furthermore, we demonstrate the potential for even higher model accuracy when complemented by expert chemists contributing to the system, boosting both machine and expert performance. The results show the advantages of empowering deep learning models with chemical intuition and knowledge to expedite the drug discovery process.

Introduction

Predicting reaction outcomes is a fundamental problem in synthetic organic chemistry. Accurate *in silico* predictions of product molecules from reactants and reagents have the potential to significantly advance the field of drug discovery and the life sciences as they can displace time-consuming, expensive, and potentially hazardous wet-lab experimentation. Specifically, reactivity prediction models can be used to virtually screen massive libraries of compounds to discover new reactivity needed for the synthetic planning of novel drugs and rapidly explore large substrate scopes for new and existing reactions. *In silico* models that are both accurate and well-calibrated lead to more reliable retrosynthetic planning programs and a model capable of providing rationales for predictions could even prove a valuable educational tool for testing hypotheses, thereby helping scientists further develop their intuition around chemical reactivity.

Some of the earliest approaches for *in silico* reaction prediction construct rule-based expert systems to check chemical reactivity and only cover a relatively narrow range of known reactivity motifs [1, 2]. More recent works such as ref.[3, 4] extend the rule-based approach by using many thousands of reaction templates extracted from known reactions and applying them to reactants at inference time for product prediction. These methods are inherently brittle as no set of rules is exhaustive for all organic chemistry, and are limited in their ability to generalize to unseen mechanisms and molecules. Quantum Mechanical calculations and approximations are also a viable method for predicting reactivity but can have difficulty scaling to multiple interacting reagents and often require significant expertise to use correctly [5–7].

With the recent advancements in machine learning, data availability, and a desire to produce accurate and generalizable models, the task of learning an inductive model for reactivity given a large set of historical reactions has gained increased attention. Kayala and Baldi introduce a model to predict reactivity from a collection of reaction motifs using molecular fingerprints with handcrafted features of reactants [8], and Wei et al. use neural networks to predict reaction types and the corresponding SMARTS transformations from the molecular fingerprints of reactants and reagents [9]. The authors of ref.[10, 11] adapt methods originating from the natural language processing field by representing chemical reactions in SMILES strings [12], and formulating reaction prediction as a translation problem from reactant SMILES to the product SMILES. These black-box end-to-end models provide compelling predictive accuracy but lack interpretable rationales behind bond formations and product predictions as a whole. Furthermore, these models can not be readily augmented with user’s prior knowledge or constraints outside of ensuring chemically valid SMILES. To address these issues with the string representation, the authors of ref.[13–15] model molecules as graphs with atoms as vertices and bonds as edges. The most accurate and relevant method [15] among them makes use of developments in graph neural networks [16–18], and serves as a competitive baseline for this work. Their method operates by 1) predicting the bonds between the atoms (i.e., reaction centers) using a graph network, 2) enumerating valid candidate product molecules from the top K predicted bond changes, 3) ranking candidate products based on the likelihood of their corresponding bond changes, and 4) scoring final candidate products using a second network. The enumeration process constitutes a simple search in a subset of the feasible space of possible products that have been bounded by the chosen set of K bond changes. If a bond change is missing, the correct product will not be found, and expanding the set of K bonds will lead to an exponential growth of candidates to enumerate and rank.

In this work, we combine neural networks and Integer Linear Programming (ILP) to perform a combination of probabilistic and symbolic inference for organic reaction prediction. We introduce a modified graph convolutional neural network to estimate a probability distribution over possible changes in covalent bonds, hydrogen counts, and formal charges, allowing us to describe the resulting

products from a reaction entirely. These predicted distributions of changes govern a probability distribution for all product predictions, and we can efficiently infer the most probable product molecule(s) from such distributions via ILP. The expressive linear constraint system of ILP ensures the generated products are chemically valid and allows us to integrate chemical insight into the prediction system. We show that our approach outperforms previous graph-based neural networks by predicting products with more than 90% accuracy, showing generalizability across different reaction types, and intuitive chemical rationales for bond level predictions.

Results

Overall Framework

In reaction prediction, we want to predict the product molecules from a set of reactants and reagents. Translating molecules into a graph representation, we first predict changes in covalent bonds between heavy atoms, hydrogen count per heavy atom, and formal charges per heavy atom (Figure 1a-c). Message Passing Neural Network (MPNN) [17], a variant of the graph convolutional neural network, is first used to model the reactant molecule graph and generate a neural embedding for each atom. Such graph-based neural networks allow us to learn a more effective embedding representation than traditional handcrafted features and at the same time, models the molecules in a way that follows chemical intuition. A convolution-based co-estimation network is then introduced to predict different kinds of reactivity from these neural embedding. Comparing to a linear independent prediction layer used in previous methods [15, 19], our co-estimation network can capture the dependence structure between the reactivity predictions such as the total number of reaction center or the number of bonding partners for each atom. Modeling the reaction as a combination of bondings, hydrogen, and charge changes encourage the model to learn chemical reactivity from first principles rather than memorizing the common reaction patterns. To infer the most probable product molecule from the estimated reactivity distributions, an ILP formulation is presented (Figure 1c-e), allowing an efficient search of the entire feasible space, unrestricted by the top K reactivity selection as in previous methods [15, 19]. Additionally, the expressive linear constraint system of ILP allows us to enforce chemistry constraints such as the octet rule and infer valid product molecules directly without enumeration or filtering. Combining reactivity modeling and rule-based symbolic inference, our approach also allows the chemists to interact with the prediction model by providing partial solutions or integrate their prior domain knowledge as constraints at inference time.

Data

For training and evaluation, we follow previous work [11, 15, 19] and use the reactions mined from the United States Patent and Trademark Office (USPTO) by Lowe [20]. The USPTO dataset is the largest and most comprehensive publicly available dataset for organic reaction to our knowledge. Since there are many duplicated and erroneous reactions in the original mined data, we follow the same cleaning process and data splitting published by Coley et al. [15], resulting in 410k training reactions, 30k validation reactions, and 40k testing reactions. Besides, the same data splitting also allows us to make a fairer comparison to our baseline, Coley et al. [15]. Each sample contains an atom mapped SMILES string describing reactants and the products.

Following chemical intuitions, we represent the reactant molecules as a graph with heavy atom vertices and bond edges (and we define heavy atom as any atom that is not hydrogen). The initial atom features v_i and bond features e_{ij} include features such as atom type, bond type,

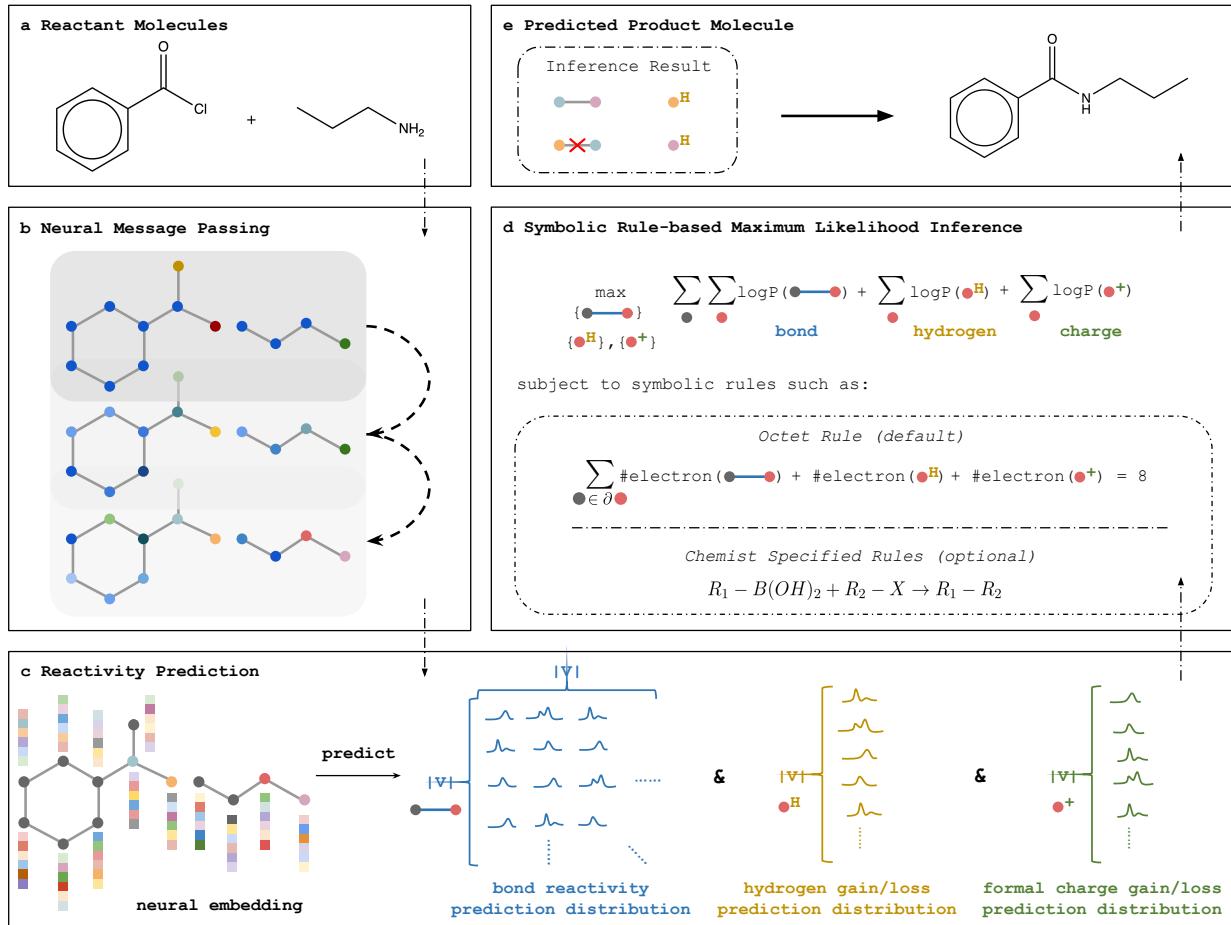


Figure 1: Overview of the framework. **a** Reactant molecules are represented as a graph with disconnected components where the vertices are atoms, and edges are bonds. **b** Neural message passing is used to learn neural representations of atoms from their position in the reactant graph. **c** Chemical reactivity of bonds, hydrogens, and formal charges are predicted from neural embeddings via a convolution-based co-estimation network. **d** Constrained probabilistic inference with Integer Linear Programming allows us to infer the most probable and chemically valid reactivity configuration from the models predicted distribution with a constrained maximum likelihood problem. **e** The predicted product molecule can then be easily constructed from the inferred reactivity configuration.

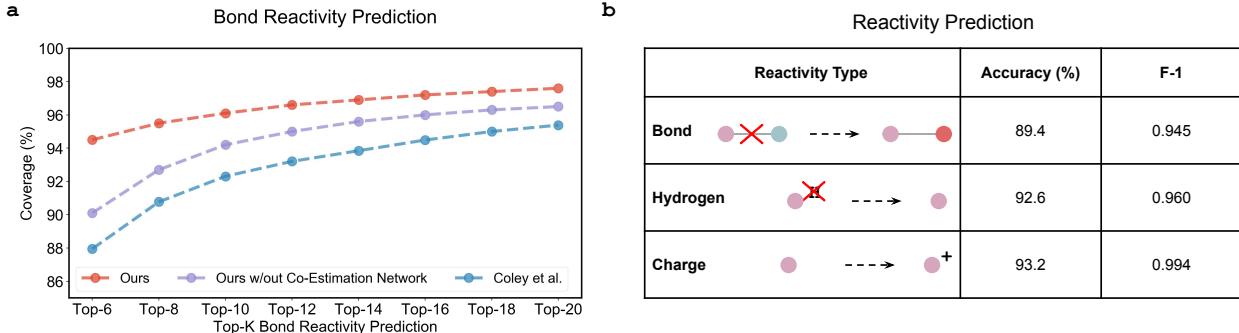


Figure 2: Reactivity prediction performance. **a** Our graph neural network outperforms the baseline method [15] in top-k prediction coverage for bond reactivity, and the bond reactivity is considered recovered only if *all* the bond changes are predicted in the top-k selection. **b** Accuracy and F-1 score for our reactivity prediction. The accuracy is evaluated as a multi-class classification problem for those reactivities model predicts as non-zero. The F-1 score is evaluated as a binary classification problem where we are simply predicting if there is a change in the reactivity. The precision and recall for each class of these reactivities can be found in Table S2, Table S3, and Table S4.

and hybridization (Table S5 and Table S6). Similar to the previous method[11], we differentiate reactants and reagents, adding boolean features to atoms indicating their expected role in the reaction.

Using the atom mapped SMILES reactions, we calculate the *changes* in bonds between heavy atoms, hydrogen counts, and formal charges as measured by electron participation (i.e., a new covalent bond receives the same training label as a single bond changing to a double bond). In a later study, we breakdown the performance by various reaction types, and the reaction type labels are predicted using a very accurate chemical fingerprint-based model [21].

The improved graph neural network makes accurate reactivity predictions

Since knowing the reactivity between heavy atoms is crucial for modeling and understanding the chemical reaction, we first evaluate how well the model predicts the bond changes between heavy atom similar to the baseline method [15] as shown in Figure 2a. Our graph neural network significantly outperforms the baseline model [15], and achieves their Top-20 coverage at Top-6. While their model only predicts bond reactivity, our model also makes accurate predictions for hydrogen and formal charge changes during the reaction allowing us to describe the reaction fully (Figure 2b).

Our ablation study (Figure 2a purple curve) shows that the convolutional co-estimation network makes an essential contribution in boosting the performance. By allowing the model to condition its local predictions on information from distal atoms and bonds, the network can have a global view of different activity and makes better predictions through co-estimation. The models estimate of a bond’s reactivity can be informed by knowing how many other bonds are likely reacting and what nearby reactivity might take precedence. We also note that our multi-task objective provides better regularization than training with bond reactivity alone. Such regularization, along with some slight differences in the graph neural network parameterization (e.g., we do not use the Weisfeiler-Lehman Network), explains the gap between the purple and blue curve in Figure 2a where we observe better performance in our graph neural network even without the co-estimation network.

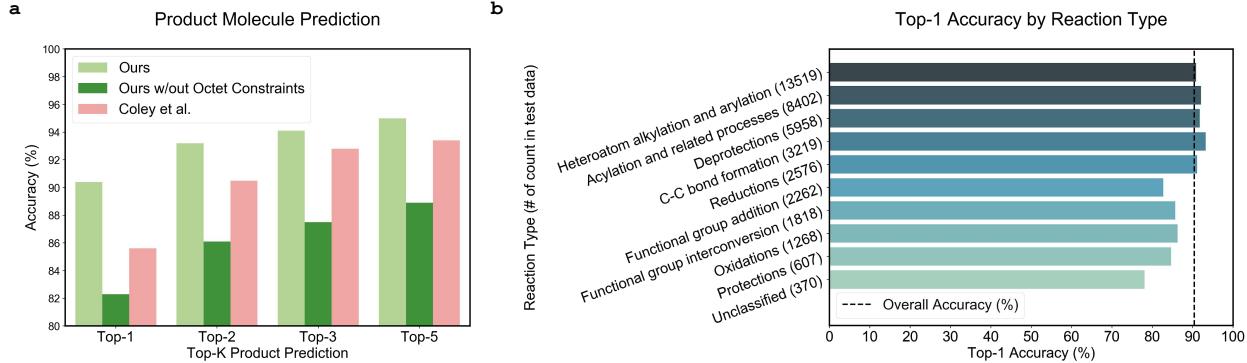


Figure 3: Product prediction performance. **a** Our rule-based symbolic inference achieves more than 90% product prediction accuracy in USPTO outperforming the current state of the art method [15] with the chemistry constraints as a crucial component in the success of our method. The numeric table comparison can be found in Table S1. **b** Our approach generalizes well across various reaction types. The counts of various reaction types in the testing data are also shown after the name of each reaction type. The black dash line is the overall performance of our method.

Symbolic inference with chemical rules predicts the reaction products accurately

To demonstrate the power of symbolic inference with chemistry rules, we compare our approach to the current state of the art graph-based method [15] and an approach where we perform inference without any chemical constraints. As shown in Figure 3a, our method is a significant improvement over the current state of the art and achieves an accuracy of over 90%, which is comparable to the black-box seq2seq approach [11]. We highlight that, despite the better performance, our approach only uses a single neural network for reactivity predictions while [15] uses an additional ranking network to re-rank the candidate products enumerated from their predictions in bond changes. Our success with a single network approach shows that the ranking network might not be necessary if we can make accurate reactivity predictions with the first network. The reduction to a single network prediction also allows us to drastically reduce the complexity of the training procedure and inference pipeline.

The success, however, is not possible without the chemistry rules we impose during the probabilistic inference, as the inference method without the octet constraints still lags in the product prediction accuracy (Figure 3a). The performance improvement we enjoy by incorporating the chemistry constraints are two-fold: 1) the valence constraints of the octet rule ensure only valid solutions are generated, eliminating invalid solutions, and 2) since the bond, hydrogen, and charge predictions are constraining each other as shown in Equation 2, the inference system is less prone to small prediction errors, as other correct predictions can still encourage the optimization algorithm to choose the correct reactivity in order to satisfy the chemistry constraints at lower cost.

Our method generalizes well across various reaction types

A key question for data-driven, learning-based methods is the generalizability: how well does the model perform in a low data regime? does the model still work for out of distribution reactions? To address the concerns around generalizability, Figure 3b shows a performance breakdown by various reaction types. Even for reaction types that are not abundant in our dataset (e.g., oxidation and protections), our model still shows excellent performance with over 80% accuracy. Even in cases where we take out a sub-reaction type entirely from our training data, our approach can still perform reasonably well with accuracy between 70 ~ 80% (Figure S1). Such out-of-scope

predictions would be impossible to achieve by template-based methods as the reaction template must be in the scope of the training set. We believe the generalizability of our approach comes from 1) the explicit modeling of chemical reactivity as low-level inter and intra-atom relations which are shared among many different reaction types as chemical principles, and 2) the chemical rules we impose during symbolic inference, since valence constraints imposed should always be satisfied no matter the reaction type.

Our *in silico* method achieves comparable performance to human experts

To compare our *in silico* method with human experts, we evaluate our approach on the human benchmark dataset introduced by Coley et al. [15]. The dataset contains 80 sample reactions from the USPTO test set, where eleven participants (graduate students, postdocs, and a professor in chemistry and chemical engineering) are asked to write down the likely major products for these reactions. These 80 reactions are further separated into eight bins by the popularity of their corresponding templates in data, and each bin contains ten reactions. If a reaction sample has a small number of corresponding templates (e.g., < 50), the sample might be a rare reaction type, and the model could have a harder time to predict the correct solution due to the lack of training data. To further understand the performance differences between machine learning models and human experts, we also separate these 80 reactions into various reaction types.

As demonstrated in Figure 4, our approach can outperform the best human expert across different template popularity and reaction types consistently in this small benchmark dataset, while the baseline model [15] is only able to achieve average human performance consistently across different categories (Figure S2). While this is only a small benchmark dataset with a limited number of reaction types, our expert-chemist-level performance shows the potential of model-assisted synthesis planning. We also want to point out that the outstanding performance for reactions with high template popularity (≥ 50) demonstrates the power of a data-driven learning-based method: as more and more data become available, our machine learning model can translate the relative cheap data power into the more expensive expert prediction power. On the other hand, if we consider the low data regime where the reaction templates are less popular (< 50), we still observe better performance comparing to the best human. The performance gap indicates that 1) generalization is a challenge for both humans and the models since human experts might not have experience with certain rare reactions, and 2) our model is better in transferring knowledge across different reaction types than human thanks to the representation learning power of the neural network.

Symbolic inference enables the incorporation of chemistry insights

One advantage of our approach comparing to the black box end-to-end approaches is that our explicit modeling of the chemical reactivity allows the chemist to integrate their domain knowledge and insight into the system to further enhance the machine intelligence. For instance, because the USPTO dataset does not always specify conditions, there are cases where a single answer cannot be arrived at with the given information. Figure 5a is one such example. Neither the model nor a chemist could definitely choose between a Chan-Lam coupling product (top) and a Suzuki coupling product (bottom) without more information on the reaction condition. Figure 5b elucidates a similar problem, where missing reagents prevents the algorithm from deducing that the two-step one-pot process of reductive amination (bottom) has occurred rather than the single-step imine condensation (top).

However, if chemists know the potential reaction templates based on inferred conditions for a specific reaction or are looking for a particular reaction type, they can easily encode them as rules

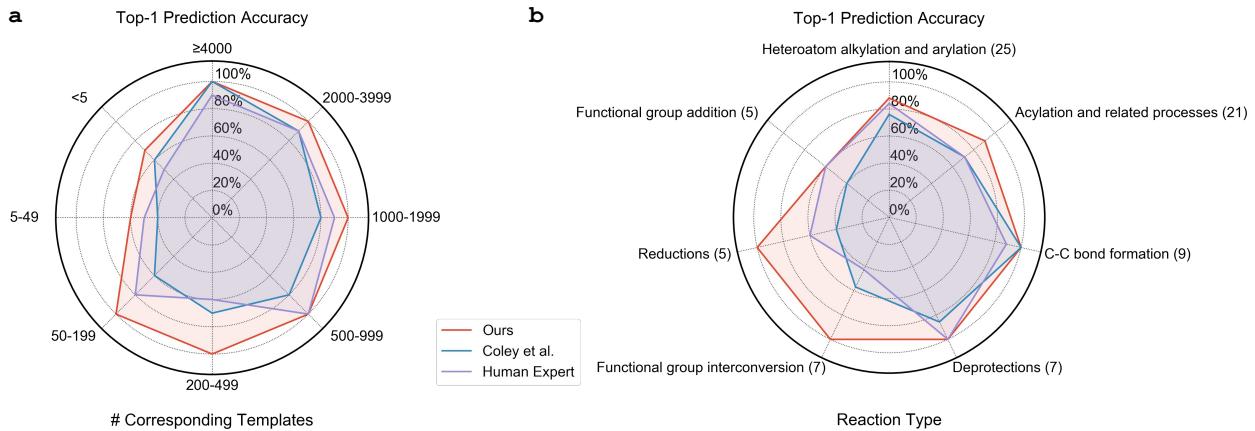


Figure 4: Human benchmark dataset performance. Coley et al. [15] introduces a human benchmark dataset to compare expert human performance to model performance in organic reaction prediction. The dataset contains 80 reactions that are binned by their reaction template popularity. In addition, we also separate the reactions by their reaction types, ignoring the categories that have less than five reactions (i.e., protections and oxidations). **a** Our model can consistently outperform the *best* human expert performance across various template counts, while the baseline method [15] can only achieve similar performance in some of the bins. The label on the x-axis around the circle is the number of corresponding templates indicating the popularity of the reactions in the corresponding bin, and the % on the y-axis is the top-1 prediction accuracy for each bin. **b** The same performances break down by reaction types. The label on the x-axis around the circle is the reaction types, and the % on the y-axis is again the top-1 prediction accuracy for each bin.

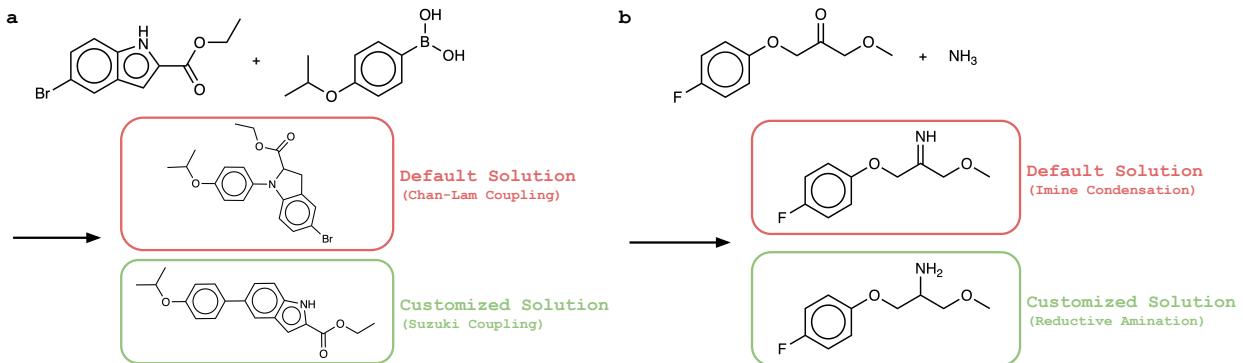


Figure 5: Enhancing machine intelligence by integrating template-based reaction constraints instructed by a human expert. **a** The missing experiment conditions make it impossible to decide whether the reaction is a Cham-Lam coupling or a Suzuki coupling. However, by supporting the natural integration of reaction templates, the ILP can change the default solution of a Cham-Lam coupling to the correct Suzuki coupling. **b** The model predicts the default solution instead of the intended one due to the missing reagent. Again, the ILP can incorporate human instructions at inference time and complete the second step of the reductive amination through the imine intermediate.

in symbolic inference thanks to the flexible linear constraint system of ILP (e.g., Equation 3, 4, and 5). Taking advantage of experts' domain knowledge and insights, our method can make natural adjustments to the inference problem and select the desired final product, as shown in Figure 5. These examples showcase the power and flexibility of our symbolic inference approach, and we are excited to see a world where chemists can interact with a powerful predictive model to explore the possible solution space and come up with an even more accurate prediction together.

To further showcase the power of such 'model with expert' system, we set up another experiment where the model will first flag low confident reactivity prediction (see supplement for details), and experts will then provide their estimation for these low confident predictions as fixed ILP variables in inference time. To represent chemists with different level of expertise, we created simulated experts whose estimations are correct {50%, 60%, 70%, 80%, 90%, 100%} of the time, and evaluated the prediction accuracy after we integrate the expert estimations as rules for symbolic inference.

As shown in Figure S3, a chemist that is only correct 80% of the time could already help improve the model performance even though the chemists are expected to put minimal interventions into the system (Figure S4). The integration improvement also scales with the expert correct rate in a linear fashion. Since the computational model only asks instructions for the low-confident predictions, in the case where we have an amateur chemist that is correct only half of the time, the model can still maintain a relatively high accuracy of 86%. Interestingly, while our model's overall reactivity prediction is well over 80%, an expert with only 80% correct rate can still improve our model performance. The discrepancy indicates that our model predictions are well-calibrated as low confident predictions also have worse accuracy.

Interpretability with the attention mechanism

In drug discovery and organic molecule synthesis, it is crucial for us to interpret models' decisions to gain confidence in its predictions and further our own understanding of reactions. Therefore, instead of building a black-box deep learning model, we introduce an attention mechanism (Methods). While it is designed to capture the global relationship between atoms such that our model can make better predictions, it also allows us to look at how other atoms (red dashed circle) contribute to the reactivity prediction for a particular atom (highlighted blue) as shown in Figure 6. For instance, in Figure 6a, the attention mechanism shows that the model investigates two possible modes of reactivity for the amine: 1) cross-coupling to a carbon-bearing a halogen, and 2) reaction with a carbonyl. Matching chemists' intuition, the algorithm can select the cross-coupling as the mode of reactivity as amides do not readily exchange. For the cross-coupling, the more activated carbon between the two nitrogens in the aromatic ring is selected to give the correct product. Similar phenomena are also observed in other examples where potential reaction sites are contributing to the selected atom showing model rational that chemists can interpret and agree upon. By modeling the reaction explicitly with graph structure that matches our chemistry intuition, our model, with its attention mechanism, allows us to understand its rationale better and be more confident in its predictions.

Methods

Reactivity Prediction with Graph Convolutional Neural Network

For reactivity prediction, the reactant molecules are first represented as a graph $\mathcal{G}_r = (V, E)$ with disconnected components where $V = \{v_i\}$ for all the atom i in the reactants, and $E = \{e_{ij}\}$ for all bonds connecting atoms i and j . To fully describe a reaction, we need to know the bond

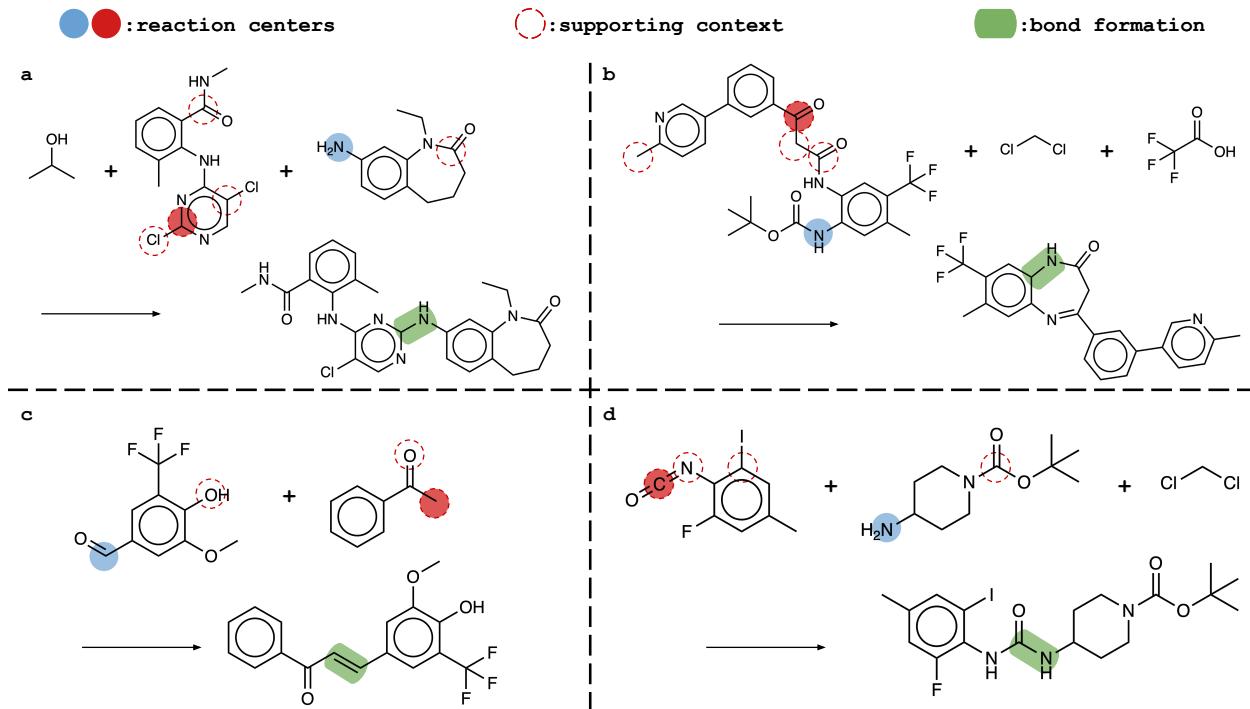


Figure 6: Our models provide chemical interpretability to explain model predictions. For each reaction the model predicts correctly, we select an atom (blue highlighted) whose reactivity is essential for the reaction, and our model shows which atoms (red dash circled) influence the prediction of the selected atom. The reaction partners (red highlighted) of the selected of the four reactions all contribute the most to the selected atom (blue highlighted) as they form a reaction center (green highlighted). **a Buckwald-Hartwig Coupling** Upon the identification of the amine as a reactive functional group, the model investigates two possible modes of reactivity: cross-coupling to a carbon-bearing a halogen, and reaction with a carbonyl. Upon selection of cross-coupling, the model also makes the right decision to react with the more activated carbon between the two nitrogens in the aromatic ring. **b Deprotection and Ring Closing** Upon deprotection of the BOC group with trifluoroacetic acid, a free amine is revealed. The model notes the carbons that bear hydrogens with low pK_a's, but this type of reactivity with an amine requires basic conditions. It is then able to select the more reactive ketone over the amide, to give the 7-membered imine containing ring rather than the 5-membered amidine containing the ring. **c Aldehyde and Alcohol (Aldol) Reaction** The model considers the alcohol since hemiacetal formation via the addition of an alcohol into an aldehyde is a well-precedented reaction, but steric constraints are preventing the reaction. Therefore, the model correctly identifies the enolizable protons of the methyl ketone, allowing an aldol condensation. **d Urea Formation** The model identifies an aryl iodide, an ester, and an isocyanate as possible reactive partners for the amine. As isocyanates are highly reactive, the model again correctly predicts the nucleophilic addition of the amine into the isocyanate to form urea.

changes between heavy atoms Δb , the gain or loss of the hydrogen Δh per atom, and finally, the difference in formal charges Δc of each atom. Since these changes are discrete and bounded in a small range, we treat the reactivity prediction problem as a classification task. Consequently, our reactivity prediction network is a function $F : (V, E) \mapsto (F^{\Delta b}, F^{\Delta h}, F^{\Delta c})$, where F^Δ is the estimated probability distribution for all the reactivities. Modeling the reactants as a graph and the reaction as the combination of all low-level reactivity allows our model to learn implicit chemical knowledge and allows us to integrate chemical knowledge into the system’s decisions better.

To leverage the representation learning power of deep learning, we used the MPNN [17] to learn an embedding representation for each atom. At each message passing step t , we have:

$$\begin{aligned} m_i^{t+1} &= \sum_{j \in \mathcal{N}_i} \tau(\mathbf{U}[v_j^t, e_{ij}]) \\ v_i^{t+1} &= \tau(\mathbf{V}_1 v_i^t + \mathbf{V}_2 m_i^{t+1}) \end{aligned}$$

where \mathcal{N}_i is the set of neighboring atoms connected to the atom i , $[\cdot, \cdot]$ denotes the concatenation of two vectors, and $\tau(\cdot)$ is a ReLU activation function. Moreover, the weights \mathbf{U} , \mathbf{V}_1 , and \mathbf{V}_2 are shared among all steps and v_i^0 is initialized as v_i . After running message passing for T steps, we have an embedding representation for each atom i as v_i^{MPNN} .

While MPNN can capture the local structure, it is not able to capture some of the long dependency, especially when we are interested in capturing the relationship between the atoms in two disconnected components. Therefore, we invoke a global version of the Graph Attention Network (GAT) [18] as the following:

$$\begin{aligned} a_{ij} &= (\mathbf{u}^\top \tau(\mathbf{W}_1 v_i^{\text{MPNN}} + \mathbf{W}_2 v_j^{\text{MPNN}} + \mathbf{W}_2 e_{ij}))r \\ \alpha_{ij} &= \text{softmax}_j(a_{ij}) = \frac{\exp(a_{ij})}{\sum_{k \in V} \exp(a_{ik})} \\ v_i^{\text{GAT}} &= v_i^{\text{MPNN}} + \sum_{j \in V} \alpha_{ij} v_j^{\text{MPNN}} \end{aligned}$$

where \mathbf{u} and \mathbf{W} s are all learned parameters, and $\sum_{i \in V}$ means summing over all atoms in the reactants graph while original GAT will only sum over neighboring atoms. The attention term α_{ij} allows us to model the non-local relationships between atoms, and such global modeling will be especially beneficial for those reactions where the reaction centers are not necessarily local.

Since reactivity is decided by the relationship between pairs of atoms, we then compute an embedding for each pair of atoms i and j as:

$$s_{ij} = \tau(\mathbf{M}_1 v_i^{\text{GAT}} + \mathbf{M}_1 v_j^{\text{GAT}} + \mathbf{M}_2 e_{ij})$$

giving us an embedding for the interaction between each pair of atoms, and \mathbf{M} s are the matrices of learnable weights. We then put s_{ij} together to form a tensor \mathbf{S} with size $|V| \times |V| \times |s|$ where $[\mathbf{S}]_{ij} = s_{ij}$.

In previous methods, s_{ij} is directly used to predict the bond changes after some dense layer [15, 19]. However, there is important global information that these ‘independent’ predictions do not capture, such as the total number of bond changes or the number of other atoms an atom can interact with. Therefore, we introduce a convolution-based co-estimation network to better capture these global dependencies during the reactivity prediction. Since we can index a molecule canonically with a canonical SMILE such that neighboring atoms are almost always indexed together, we use a 2D Convolutional Neural Network (CNN) [22, 23] with a very small filter kernel (3×3)

to capture the dependencies among different positions. We apply L layers of convolution filter as a transformer for the original interaction embedding \mathbf{S} , and in the end, we have a transformed interaction embedding \mathbf{S}^{CNN} that better capture the global dependency.

After the co-estimation network, we can obtain the final reactivity predictions from the interaction embedding \mathbf{S}^{CNN} with the fully connected networks:

$$F_{ij}^{\Delta b} = \text{softmax}(\mathbf{P}_1 s_{ij}^{\text{CNN}}) \quad F_i^{\Delta h} = \text{softmax}(\mathbf{P}_2 \sum_{j \in V} s_{ij}^{\text{CNN}}) \quad F_i^{\Delta c} = \text{softmax}(\mathbf{P}_3 \sum_{j \in V} s_{ij}^{\text{CNN}})$$

where \mathbf{P} s are the parameters for the fully connected output layers, and the softmax layer is normalized over all the possible changes resulting in a valid probability distribution over all the possible reactivity. Once we have the estimated probability distribution, we can train the model with the following objective:

$$\mathcal{L} = \sum_{\mathcal{G}_r \in \mathcal{T}} [\lambda^{\Delta b} \mathcal{L}_n^{\Delta b} + \lambda^{\Delta c} \mathcal{L}_n^{\Delta c} + \lambda^{\Delta h} \mathcal{L}_n^{\Delta h}]$$

where

$$\mathcal{L}_{\mathcal{G}_r}^{\Delta b} = - \sum_{i \in V} \sum_{j \in V} \log F_{ij}^{\Delta b}(\Delta b_{ij} | \mathcal{G}_r) \quad \mathcal{L}_{\mathcal{G}_r}^{\Delta h} = - \sum_{i \in V} \log F_i^{\Delta h}(\Delta h_i | \mathcal{G}_r) \quad \mathcal{L}_{\mathcal{G}_r}^{\Delta c} = - \sum_{i \in V} \log F_i^{\Delta c}(\Delta c_i | \mathcal{G}_r)$$

and \mathcal{T} is all the reactant molecules in the dataset, while Δb_{ij} , Δh_i and Δc_i are the ground truth reactivities for the reactant molecule. $F_{i(j)}^{\Delta}(\cdot | G)$ denotes the predictive probability for selecting that particular changes, and the coefficients λ are introduced to bias the model training since downstream inference requires different levels of accuracy for these different reactivities. Since our objective \mathcal{L} is fully differentiable with respect to the model parameter θ , we can optimize θ via stochastic gradient methods such as Adam [24].

Probabilistic Symbolic Inference for Product Prediction

Given a reactant molecule \mathcal{G}_r and its corresponding reactivity Δ , we can easily infer the product molecule \mathcal{G}_p as shown in Figure 1e. Since our predictive model generates a probability distribution for all the reactivity, the task then becomes inferring the most probable product molecule from such distribution - a likelihood maximization problem. More formally, we have the following objective:

$$\max_{\mathcal{G}_p \text{ or } \Delta} \sum_{i \in V} \sum_{j \in V} \log F_{ij}^{\Delta b}(\Delta b_{ij} | \mathcal{G}_r) + \sum_{i \in V} \log F_i^{\Delta h}(\Delta h_i | \mathcal{G}_r) + \sum_{i \in V} \log F_i^{\Delta c}(\Delta c_i | \mathcal{G}_r) \quad (1)$$

where we want to find the most probable set of reactivity Δ (and the corresponding product \mathcal{G}_p) from the neural network estimated probability distribution F^{Δ} .

Unfortunately, a naive maximization for Equation 1 is not desirable because the optimal reactivity configuration here might not result in a valid molecule due to the physical constraint in the valence shell. In addition to the physical constraints, chemists might also want to impose other chemistry constraints based on their domain knowledge. Therefore, we reformulate the problem as a constrained log-likelihood maximization problem where the solution has to follow certain rules. While previous methods use the bounded enumeration and filtering approach to ensure the validity [15, 19], we employ Integer Linear Programming (ILP) to search the solution space more efficiently, and the expressive linear constraint system in ILP also allows us to impose chemistry constraints to ensure validity and integrate other domain knowledge during optimization. By default, we apply the octet rule, a well-known heuristic, to ensure the chemical validity of the predicted product molecules. Importantly, with the above optimization formulation, it is straightforward to include users' preference or prior chemistry knowledge on specific reactions as constraints to be included in the ILP.

The Octet Rule To make sure the product molecules are valid, we impose a variant of the octet rule, which states that a heavy atom prefers to have eight electrons in its valence shell, so an atom should not gain or loss any electron at the end of the reaction. Translating the rule into the ILP, we have the following constraints for all the heavy atom:

$$\sum_{j \in \mathcal{N}_i} (b_{ij} + \Delta b_{ij}) \cdot 2 + (h_i + \Delta h_i) \cdot 2 + (c_i + \Delta c_i) = 8 \quad \forall i \in V \quad (2)$$

or simply $\sum_{j \in \mathcal{N}_i} \Delta b_{ij} + \Delta h_i + \Delta c_i = 0$ as the condition is also satisfied before the reaction. In addition, modifications to the original octet rule are made for atoms with known hypervalent potential like iodine or conjugated systems where electrons are shared. By incorporating Constraint 2 into Objective 1, the ILP is able to find the most probable and *chemically valid* product from the predicted reactivity distribution. Since we expect most of the product molecules to satisfy the octet rule, Constraint 2 is imposed by default during the probabilistic inference.

Prior Knowledge or Reaction Template as Constraints In the case where the chemists know the reactivities for some of the atoms or want to explore alternative reactions, they can specify partial solutions as simple constraints for the ILP (e.g., $\Delta b_{ij} = x$ and $\Delta h_i = y$). In addition, the expressive linear constraint system also allows us to implement complex chemical constraints such as reaction templates. For instance, to implement a Suzuki coupling reaction with a template like $R_1 - B(OH)_2 + R_2 - X \rightarrow R_1 - R_2$ we can have the following constraints:

$$\sum_{(i-j) \in (R_1 - R_2)} \Delta b_{ij} = +1 \quad \text{and} \quad \Delta b_{ij} \in \{0, +1\} \quad \forall (i-j) \in (R_1 - R_2) \quad (3)$$

$$\sum_{(i-j) \in (R_1 - B)} \Delta b_{ij} = -1 \quad \text{and} \quad \Delta b_{ij} \in \{-1, 0\} \quad \forall (i-j) \in (R_1 - B) \quad (4)$$

$$\sum_{(i-j) \in (R_2 - X)} \Delta b_{ij} = -1 \quad \text{and} \quad \Delta b_{ij} \in \{-1, 0\} \quad \forall (i-j) \in (R_2 - X) \quad (5)$$

where $(i-j) \in (X - Y)$ means the atom pair that matches the template specified by $(X - Y)$, and the ILP will pick one of the best reaction sites to perform a Suzuki coupling under these constraints. We note that this is only one of the many interesting chemistry constraints that can be integrated.

Data availability

The processed USPTO dataset is available at https://github.com/WesleyyC/symbolic_rxn. All of the data and trained models used in this paper are publicly available and can be accessed from the link.

Code availability

Instructions and code for running our method with the processed USPTO dataset are available at https://github.com/WesleyyC/symbolic_rxn.

References

1. Satoh, H. & Funatsu, K. SOPHIA, a knowledge base-guided reaction prediction system-utilization of a knowledge base derived from a reaction database. *Journal of chemical information and computer sciences* **35**, 34–44 (1995).
2. Satoh, H. & Funatsu, K. Further development of a reaction generator in the sophia system for organic reaction prediction. knowledge-guided addition of suitable atoms and/or atomic groups to product skeleton. *Journal of chemical information and computer sciences* **36**, 173–184 (1996).
3. Chen, J. H. & Baldi, P. No electron left behind: a rule-based expert system to predict chemical reactions and reaction mechanisms. *Journal of chemical information and modeling* **49**, 2034–2043 (2009).
4. Coley, C. W., Barzilay, R., Jaakkola, T. S., Green, W. H. & Jensen, K. F. Prediction of organic reaction outcomes using machine learning. *ACS central science* **3**, 434–443 (2017).
5. Cramer, C. Essentials of computational chemistry: theories and models. 2004, Chichester, West Sussex, England; Hoboken. NJ: Wiley **73**, 549–551.
6. Lu, Z. & Yang, W. Reaction path potential for complex systems derived from combined ab initio quantum mechanical and molecular mechanical calculations. *The Journal of chemical physics* **121**, 89–100 (2004).
7. Olsen, R., Kroes, G., Henkelman, G., Arnaldsson, A. & Jónsson, H. Comparison of methods for finding saddle points without knowledge of the final states. *The Journal of chemical physics* **121**, 9776–9792 (2004).
8. Kayala, M. A. & Baldi, P. ReactionPredictor: prediction of complex chemical reactions at the mechanistic level using machine learning. *Journal of chemical information and modeling* **52**, 2526–2540 (2012).
9. Wei, J. N., Duvenaud, D. & Aspuru-Guzik, A. Neural networks for the prediction of organic chemistry reactions. *ACS central science* **2**, 725–732 (2016).
10. Nam, J. & Kim, J. Linking the neural machine translation and the prediction of organic chemistry reactions. *arXiv preprint arXiv:1612.09529* (2016).
11. Schwaller, P. *et al.* Molecular Transformer: A Model for Uncertainty-Calibrated Chemical Reaction Prediction. *ACS central science* **5**, 1572–1583 (2019).
12. Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of chemical information and computer sciences* **28**, 31–36 (1988).
13. Bradshaw, J., Kusner, M. J., Paige, B., Segler, M. H. & Hernández-Lobato, J. M. A Generative Model For Electron Paths. *arXiv preprint arXiv:1805.10970* (2018).
14. Do, K., Tran, T. & Venkatesh, S. *Graph transformation policy network for chemical reaction prediction* in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2019), 750–760.
15. Coley, C. W. *et al.* A graph-convolutional neural network model for the prediction of chemical reactivity. *Chemical science* **10**, 370–377 (2019).
16. Duvenaud, D. K. *et al.* *Convolutional networks on graphs for learning molecular fingerprints* in *Advances in neural information processing systems* (2015), 2224–2232.

17. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. *Neural message passing for quantum chemistry* in *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (2017), 1263–1272.
18. Veličković, P. *et al.* Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
19. Jin, W., Coley, C., Barzilay, R. & Jaakkola, T. *Predicting organic reaction outcomes with Weisfeiler-Lehman network* in *Advances in Neural Information Processing Systems* (2017), 2607–2616.
20. Lowe, D. M. Patent reaction extraction: downloads. <https://bitbucket.org/dan2097/patent-reaction-extraction/downloads> (2014).
21. 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. *Journal of chemical information and modeling* **55**, 39–53 (2015).
22. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., *et al.* Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **86**, 2278–2324 (1998).
23. Krizhevsky, A., Sutskever, I. & Hinton, G. E. *Imagenet classification with deep convolutional neural networks* in *Advances in neural information processing systems* (2012), 1097–1105.
24. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Acknowledgements

We thank Jiaqi Guan, Yang Liu, Yufeng Su, and other group members for their helpful comments and discussion. We also would like to thank the Burke laboratory for many helpful discussions and feedback. This work is supported in part by the Sloan Research Fellowship and NSF CAREER Award (#1652815) to J.P.. Y.L. is supported by the CompGen fellowship.

Author contributions

J.P. conceived the idea of integrating neural networks with symbolic inference for organic reaction prediction. N.R. conceived of and implemented the ILP for symbolic inference. W.W.Q. conceived of and implemented the neural network for reactivity prediction, refined the ILP implementation, and conducted the experiments and analysis. C.L.W.S. and M.D.B. shared their chemistry insights for complex symbolic inference rule and model interpretation. Y.L. made some of the figures. All authors wrote the manuscript.

Competing interests

The authors declare no competing interests.

Supplementary Information

Implementation Details for Reactivity Prediction Model

The reactivity prediction model is implemented in Tensorflow. For the message passing neural network, we select $T = 4$ and hidden feature size of 800. For the convolution-based co-estimation network, we select $L = 6$ with the following hidden features size for each layer [400, 200, 100, 50, 25, 12] and a kernel size of 3 for each layer. For the training, we used Adam [15] with an initial learning rate of 0.001 where the learning rate decays by a factor of 0.9 every 10,000 iterations. We run a total of 300,000 iterations, where each interaction contains a batch of 20 training samples. In addition, we clip the gradient at a global norm of 5.0 to stabilize the training. Last but not least, we set $\lambda_{\Delta b} = 5$, $\lambda_{\Delta h} = 1$, and $\lambda_{\Delta c} = 1$ as the bond reactivity between heavy atoms plays a more important role in describing the reactions. The hyper-parameters are selected based on our empirical validation. To further improve the performance, we perform model ensembling with eight separated train models and take their average estimation distribution as the ensembled model output.

Implementation Details for Integer Linear Programming

The Integer Linear Programming (ILP) is implemented in Gurobi (www.gurobi.com) where we set `PoolSolutions=20`, `PoolSearchMode=2`, `MIPGap=0.0`, and `MIPFocus=2`. The integer Δ values are also turned into binary variables allows us to formulate the non-linear $\log F(\cdot)$ function in Equation 1 as a linear optimization problem since the value for $\log F(\cdot)$ can be pre-computed.

Running Time and Complexity

Training Following the implementation details listed above, training a single reactivity prediction model takes about 80 hours on a single NVIDIA TITAN X GPU, which translates to an average throughput of 48 ms per training reaction.

Inference Predicting the reactivity for the 40,000 test reactions from a single model takes about 10 minutes on a single NVIDIA TITAN X GPU, translating to an average throughput of 15 ms per reaction. Once we have the reactivity prediction, it takes about 2 hours to inference the most probable product for the 40,000 test reactions by solving the ILP with a single CPU (i.e., `PoolSolutions=1,threads=1`). The total run time translates to an average throughput of 180 ms per reaction. Therefore, the total inference time for a single reaction is about 200 ms.

Complexity of the ILP While the size might vary from reaction to reaction, the ILP, on average, has about 1700 variables and 500 constraints in the model formulation.

Probability Re-calibration

Since the network estimated likelihood might not be perfectly calibrated and to avoid one likelihood score for one reactivity to decide the final product, we re-calibrate the bond reactivity probability with the following calibration function:

$$P(x) \begin{cases} x & \text{if } x \leq 10^{-9} \text{ or } x \geq 1 - 10^{-9} \\ \log \frac{x}{1-x}/50 + 0.5 & \text{otherwise} \end{cases}$$

and the probability is re-normalized after the calibration procedure.

Low-confidence Predictions

For the experiments where the model flags the low-confidence reactivity prediction, we measure the uncertainty of the estimated probability distribution F using entropy:

$$S = - \sum_{\Delta} \Delta \log F(\Delta)$$

where Δ are all the possible values as $\sum_{\Delta} F(\Delta) = 1$, and we consider a reactivity prediction is low-confidence if the entropy $S > 0.5$.

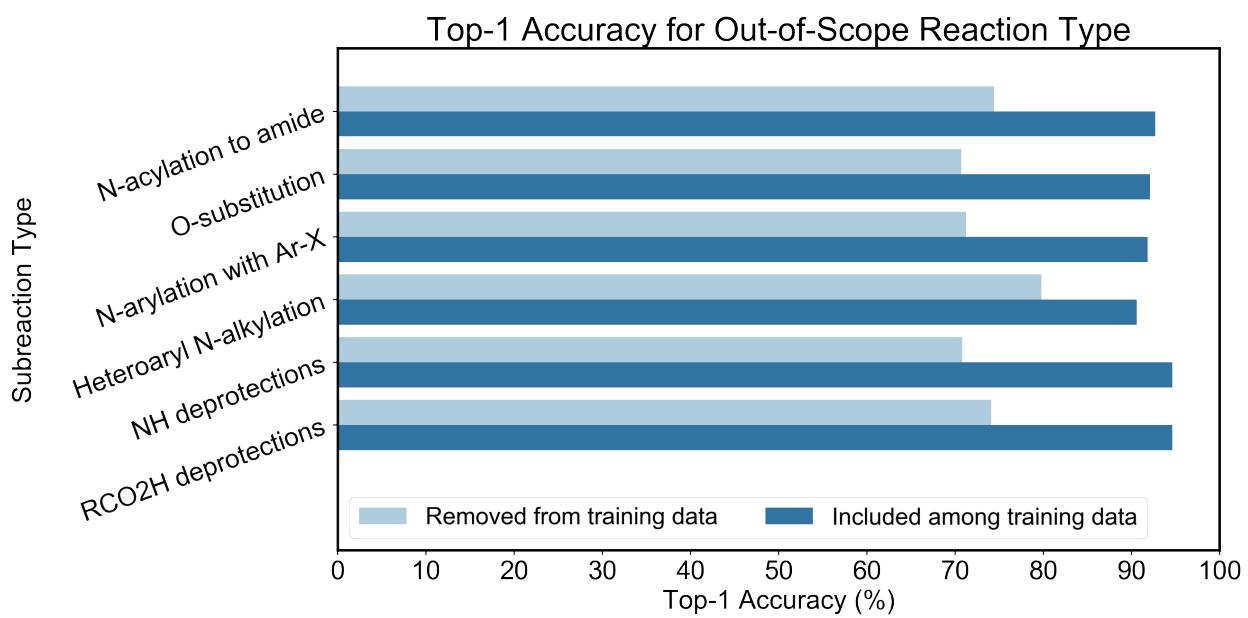


Figure S1: Product prediction accuracy for out-of-scope subreaction type. Here we show the test dataset prediction accuracy for the six most common subreaction types in the data, with the two colors corresponding to two different training settings where we include or remove the corresponding subreaction type from the training dataset.

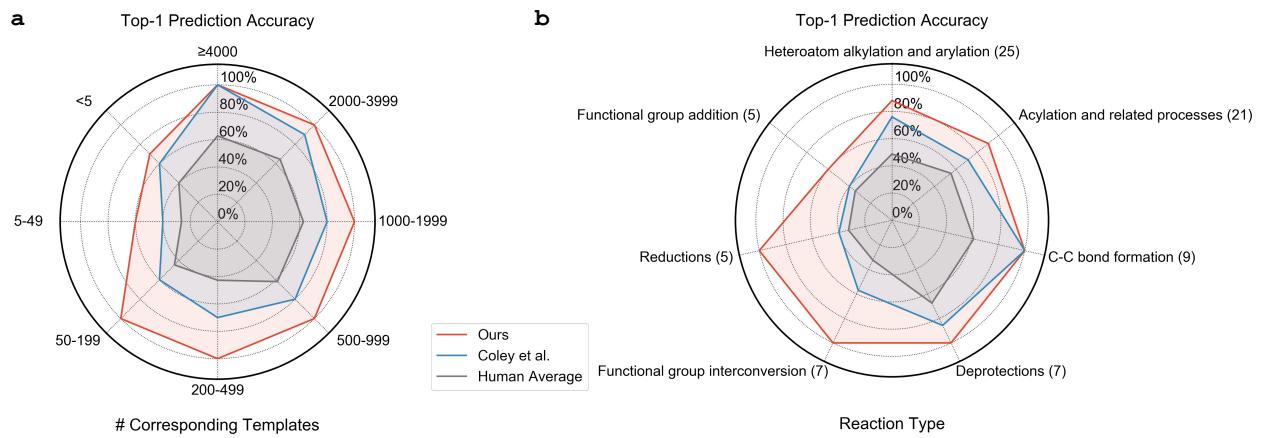


Figure S2: Human benchmark study performance with human average. Here we show the average human performance similar to Figure 4.

Model with Expert Involvement

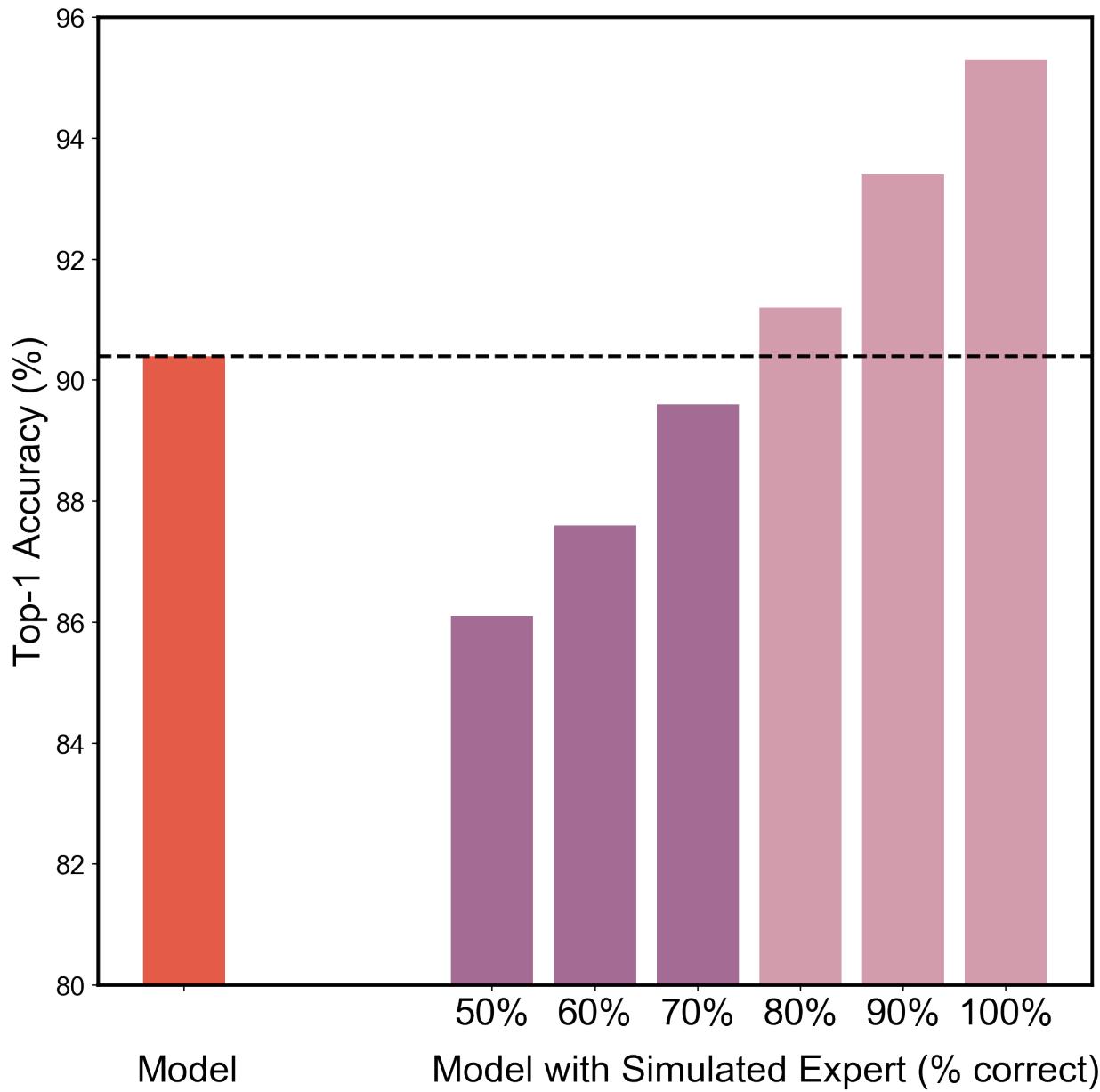


Figure S3: Integrating expert insight with our predictive model through rule-based symbolic inference. Simulated experts with varying levels of correct rate provide their estimation for low-confidence model prediction as partial solutions. We show that if an expert, when presented with a low confidence prediction, can provide the correct bond change at least 80% of the time, their interventions will improve our method's performance.

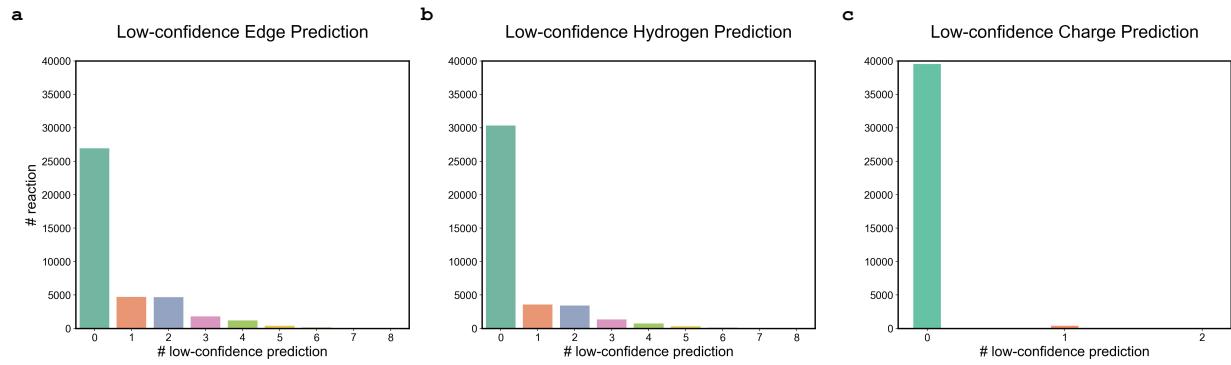


Figure S4: Distribution for the number of low-confidence predictions in a reaction. Most of the reactions have 0 low-confidence predictions, and therefore human expert would have minimum interventions with the system in the study shown in Figure S3.

Product Prediction Accuracy (%)							
Method	Top-1	Top-2	Top-3	Top-5	Interpretability	Knowledge Integration	Candidate Generation
Ours	90.4	93.2	94.1	95.0	Yes	Yes	ILP Optimization
Coley et al. [15]	86.5	90.5	92.8	93.4	Yes	Yes	Bounded Enumeration
Jin et al. [19]	79.6	-	87.7	89.2	Yes	Yes	Bounded Enumeration
[†] Schwaller et al. [11]	91.0	94.3	95.2	95.8	*No	No	Beam Search

Table S1: USPTO product prediction accuracy compares to other baselines. We outperform other graph-based models that explicitly model the chemical reactivity [13, 15, 19], and have comparable performance to the black-box seq2seq model [11]. [†]Schwaller et al. is only applicable to reactions with a single product, so the performance is only reported for a subset of the USPTO reactions with a single product. *Schwaller et al. does have the ability to rationalize the model’s decision based on token strings, but the model is not able to produce intuitive chemical rationals as ours does, due to the limitation of their string representation.

Prediction Performance for Different Covalent Bond Changes							
$\Delta_b =$	-3	-2	-1	0	+1	+2	+3
precision	0.92	0.92	0.96	0.99	0.94	0.91	-
recall	0.80	0.89	0.92	0.99	0.85	0.77	0.0
# of instances	344	16088	85528	77123604	68932	4746	76

Table S2: Precision and recall for each type of covalent bond changes.

Prediction Performance for Different Hydrogen Count Changes									
$\Delta_h =$	-4	-3	-2	-1	0	+1	+2	+3	+4
precision	-	0.94	0.91	0.95	0.99	0.96	0.93	0.95	-
recall	0	0.52	0.81	0.93	0.99	0.93	0.90	0.81	-
# of instances	3	62	1738	27713	1502645	49238	10579	444	0

Table S3: Precision and recall for each type of hydrogen count changes.

Prediction Performance for Different Charge Changes							
$\Delta_c =$	-3	-2	-1	0	+1	+2	+3
precision	-	-	0.97	0.99	0.98	1	-
recall	-	-	0.93	0.99	0.91	0.55	-
# of instances	0	0	1488	1589244	1677	11	0

Table S4: Precision and recall for each type of formal charge changes.

Vertex Features for Atom	
Feature	Type
Atom Type	One Hot Encoding
Period	One Hot Encoding
Default Valence	One Hot Encoding
Orbitals	One Hot Encoding
IUPAC Series	One Hot Encoding
# Hydrogen	One Hot Encoding
Node Degree	One Hot Encoding
Explicit Valence	One Hot Encoding
Implicit Valence	One Hot Encoding
Hybridization	One Hot Encoding
Centered Electrons	Integer
Formal Charge	Integer
Radical Electrons	Integer
Electron Negativity	Float
Electron Affinity	Float
Atomic Weight	Float
Covalent	Float
Ring?	Boolean
Aromatic?	Boolean
Conjugated?	Boolean
Reagent?	Boolean

Table S5: The chemical features used to initialize v_i for vertices.

Edge Features for Bond	
Feature	Type
Bond Type	One Hot Encoding
Bond Charge	Integer
Ring?	Boolean
Conjugated	Boolean
Reagent?	Boolean

Table S6: The chemical features used to initialize e_{ij} for edges.

symbolic_rxn.pdf (19.02 MiB)

[view on ChemRxiv](#) • [download file](#)
