Odachi Retrosynthesis Engine: Graph Convolutional Networks for Template Free, Topologically-Guided Organic Retrosynthesis

Rushil Mallarapu[1†]
[1]*Fairfield Ludlowe High School, 785 Unquowa Av., Fairfield, CT*
[†]*Email Address: rushil.mallarapu@gmail.com*

TABLE OF CONTENTS

Odachi Retrosynthesis Engine: Graph Convolutional Networks for Template Free, Topologically-Guided Organic Retrosynthesis

Rushil Mallarapu[1†]
[1]*Fairfield Ludlowe High School, 785 Unquowa Av., Fairfield, CT*
[†]*Email Address: rushil.mallarapu@gmail.com*

The development of novel synthetic methodologies is a main driving force in theoretical and practical organic chemistry and has applications in the production of innovative pharmaceuticals or agrochemicals. Retrosynthesis, a synthetic design methodology that systematically identifies bond disconnections in target molecules, is the root of all synthetic planning. Despite this, little has been done on the computational automation of retrosynthesis. Our research asked whether a deep-learning model could be developed to predict retrosynthetic disconnections with no template-based reaction rules. We report the successful development of the Odachi Retrosynthesis Engine, a graph convolutional network that can identify retrosynthetically similar clusters over molecules and find corresponding disconnections. The model uses spectral graph convolutions to identify topological synthetic contexts. We also develop a website (retrosynthesis.com) to host the engine and allow chemists to utilize the model for synthetic design. This work is the first application of graph convolutional networks to the retrosynthesis problem, and enables the development of efficient and advanced synthetic strategies.

Keywords: Synthesis, Deep Learning, Cheminformatics, Graph Networks

I. INTRODUCTION

Finding efficient and effective synthesis of molecules is the underlying problem of organic chemistry. Synthesis has been described as a fine art, due to its traditional need for creative and multilateral thinking by expert chemists.[1] Modern society relies on chemical synthesis for the production of a vast range of chemicals, from life saving medicines to environmentally friendly insecticides to safe cosmetics. Technologies resulting from total synthesis, the synthesis of natural molecules in the laboratory, have led to impressive leaps forward in materials in high-performance computers, cell phones, and space vehicles.[2] Additionally, the top twenty chemical manufacturers spend an annual $10 billion dollars on research, the majority of which goes to developing synthetic processes.[3] However, designing syntheses is time-intensive and complex, often acting as the major bottleneck in producing a chemical product for public or industrial usage. As such, developments in synthetic techniques and synthetic design are a main driving force in advancing chemistry.

The current capability of chemists to synthesize organic compounds stems from the practice of retrosynthetic thinking, which formalizes and conceptually simplifies the synthetic process.[2] Retrosynthetic analysis is the process of tracing a target molecule back to available compounds. Aside from cases that require functional group interconversion, the chemist retrosynthetically disconnects a bond in the target molecule, hoping to find simpler precursor molecules.[4] The disconnection of a bond results in two synthons, or retrosynthetic fragments, which can be traced back to synthetic equivalents with which to perform the forward reaction. This process of turning a complex target molecule into simple starting molecules is repeated recursively until the starting molecules are known to be commercially found and the synthesis can be carried out experimentally.

The question of whether synthetic design and retrosynthesis can be computerized is the subject of active research.[4,5] Algorithms for performing retrosynthesis are highly desirable for increasing the throughput of chemical experimentation and scalable production. Despite recent advancements in automated chemistry platforms, computational synthetic planning still needs large advances before it will be able to design full synthetic procedures with high efficacy and minimal to no human intervention.[6] Computational synthetic design models fall into three types. First, template-based approaches use fixed reaction patterns to directly search for molecular context and decompose a target molecule in a predefined manner.[7] Despite working in limited contexts, template approaches suffer from being highly inflexible and unable to predict novel reactions. Next, template-free approaches use computational chemistry to derive mechanistically inspired synthetic information, which is extremely computationally expensive. Finally, molecular graph models represent molecules as featurized graphs and trains models to learn over these graphs. This method, while less researched than the other two approaches, holds the most promise, due to its extensibility and computational efficacy.

Previous attempts to perform computational synthetic planning have relied mainly on inflexible rule-based systems or computationally expensive iterative algorithms. Moreover, in the vast majority of cases, computational synthesis has focused on predicting the results of forward reactions rather than retrosynthetic decomposition. Wei *et*

FIG. 1. Example reaction from USPTO database. Addition of a isobutyl Grignard to the Weinraub amide of *p*-hydroxybenzoic acid in THF.

al. used an artificial neural network (ANN) and molecular extended connectivity fingerprints (ECFP) to predict both the type of a reaction and product structure given reactants and reagents.[8] This method, while interesting, depends on a set of fixed SMARTS reaction type templates, making it inextensible, and the use of molecular fingerprint hashes reduces the spatial resolution of input molecular structure. Coley *et al.* used reaction templates to enumerate possible product structures from a set of input reactants, and ranked candidates using a neural network, thus arriving at a major product.[9] This method is similarly bottlenecked by the use of hardcoded templates, which prevents the edit-based neural network from expanding easily.

Meanwhile, other previous work encountered such issues when performing the reverse task of predicting retrosynthesis. Segler *et al.* use a deep neural network (DNN) operating on the molecular fingerprint of the target molecule to predict which reaction rules are relevant to finding a retrosynthetic disconnection, and then apply that rule to the target via templates.[10] Again, the use of templates to hardcode all possible reaction types the system can consider limits the scope of the model to known chemistry, preventing it from being used across a variety of contexts, as well as making the predictive model highly inflexible. Furthering this approach, Grzybowski *et al.* developed Chematica, a retrosynthesis planning application, by using branch-and-bound methods to recursively apply over 60,000 reaction templates to a target molecule.[4] However, because of the key dependence on rules and the computationally expensive exponential subspace search, both building and using the model is slow, e.g. it took almost fifteen years for the software to be built and each prediction takes around 15-20 minutes. The use of direct methods, backed up by approaches independent on reaction templates, is necessary for speed and extensibility in synthetic planning tools.

Jin *et al.* were one of the first to approach the computational synthesis problem in a template-free manner by treating molecules as isomorphism-invariant graphs.[7] Their method used Weisfeiler-Lehman Difference Networks (WLDN) to identify candidate atom or bond shifts from a set of reactants and reagents, which were mapped into product molecules. Despite being powerful, this method still depends on candidate enumeration, meaning the model has no ability to contextualize the bond shifts it proposes. In 2019, Coley *et al.* built off of Jin *et al.*'s use of a WLDN by using graph convolutional embeddings of atom features to encode reactivity scores.[11] This is one of the first examples of using graph convolutions in computational synthesis, but their model design is uniquely formulated to tackle the forward synthesis problem. Our research develops the first-ever graph convolutional model for template free retrosynthetic analysis, using a non-iterative approach to increase prediction speeds and contextual generalizability.

II. APPROACH AND METHODS

1. Data

As a source of training and test data, we used reactions from USPTO granted patents collected by Jin *et al.*[7] The training dataset has 400,000 retrosynthesis examples, and the test dataset has 40,000 examples. This data included atom-labelled reaction SMARTS and reaction metadata. As we intended to use this data for retrosynthetic analysis, we only needed to utilize the labelled reaction SMARTS, and were able to ignore the metadata as immaterial to the task at hand. An example reaction from this dataset is visualized in Figure 1.

2. Molecular Graph Retrosynthesis

This work uses a graph convolutional network to perform topologically-guided retrosynthesis, or topological retrosynthesis. In contrast to functional retrosynthesis, which identifies key functional group transforms as synthetic anchors, topological retrosynthesis analyses substructural motifs to identify key disconnection points.[2,3] As such, the use of graphs to represent molecules is a natural translation of the conceptual framework of topological retrosynthesis.

We define a molecule M as a graph consisting of a set of atoms (i.e. vertices) $\{A_0, A_1, \ldots, A_{n-1}\}$ and a set of bonds (i.e. edges)$\{B_0, B_1, \ldots, B_{m-1}\}$. For each atom, we may define a set of f atomic features, or properties specific to that atom, such as atomic symbol, valence, degree, formal charge, hybridization, and so on. Ergo, for each atom A_i we have a $1 \times f$ vector of features. Therefore, we can uniquely represent a molecule M with a $n \times n$ adjacency matrix A and a $n \times f$ feature matrix X. The i-th row of the feature matrix represents the feature vector for atom A_i.

Given a reaction that produces target molecule M from a set of reactant molecules $\{R_0, R_1, \ldots, R_{r-1}\}$, we define a retrosynthesis of M as a coloring of its vertices such that two atoms A_i and A_j have the same color iff $A_i \in R_k$ and $A_j \in R_k$ for some $k < r$. That is, the coloring of M reflects the starting reactant for the component atoms of M. Note that this coloring introduces a notion of clustering over M: clusters formed by this coloring correspond to retrosynthetic synthons of M. An example of this coloring corresponding to the reaction in Figure 1 is shown in Figure 2.

Topologically, between any two atoms A_i and A_j, we can define S_{ij} to be the retrosynthetic similarity of atoms A_i and A_j. In other words, S_{ij} is the probability that atoms A_i and A_j

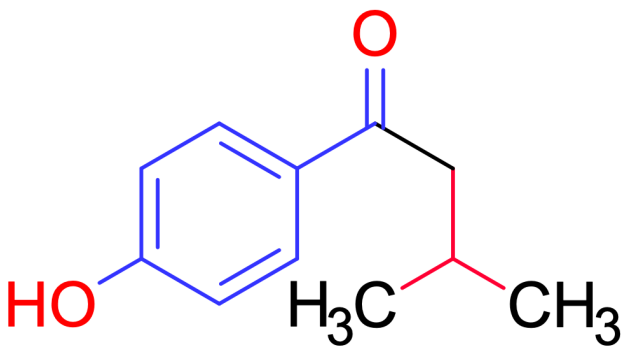

FIG. 2. An example retrosynthetic coloring of the product depicted in Figure 1. The colored clusters here represent synthons, which in turn can be traced back to the synthetic precursors of this molecule.

originate from the same retrosynthetic precursor. Note that this naturally defines a $n \times n$ retrosynthetic similarity matrix S that acts as a weighted adjacency matrix over M. Performing clustering via standard techniques (e.g. spectral clustering) on this similarity matrix preserves components which are retrosynthetically similar and disconnects components which are retrosynthetically dissimilar, thus revealing synthons from the target molecule. As such, we define the molecular graph retrosynthesis problem as creating a retrosynthetic similarity matrix S from a molecule M that reflects synthetic structural coherence as defined above.

3. Network Architecture

At a high level, our graph convolutional network (GCN) uses learnable state transfers over molecular graphs as defined previously to alter atomic feature vectors such that pairs of atoms with low retrosynthetic similarity have classifiably dissimilar convolved feature vectors, and vice versa.

We define spectral graph convolutions as implemented in this work analogously to Kipf *et al.*, as motivated by a first-order approximation of spectral filters over graphs.[12] Given a graph with self-connected $n \times n$ adjacency matrix $\tilde{A} = A + I_n$ and feature matrix X, we define a graph convolutional network $f(A, X)$ with the following propagation rule:

$$H^{(l+1)} = \sigma\left(\hat{A} H^{(l)} W^{(l)}\right).$$

Here, $\hat{A}$ is the normalized matrix $\hat{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$, where $\tilde{D}$ is the degree matrix of $\tilde{A}$. $H^{(l)}$ is the $n \times f$ convolved feature matrix in the l-th layer, such that $H^{(0)} = X$. The $f \times f$ matrix $W^{(l)}$ represents the trainable weights matrix specific to layer l. Finally, $\sigma(\cdot)$ denotes an activation function.

(Note that this formulation of graph convolutions differs significantly from other variants of the graph convolution in cheminformatics; specifically, it is unlike the spatial graph convolutions in Altae-Tran *et al.*, which try to parallel the concept of topological aggregation over vertices, as in 2D or 3D image convolutions.[12,13])

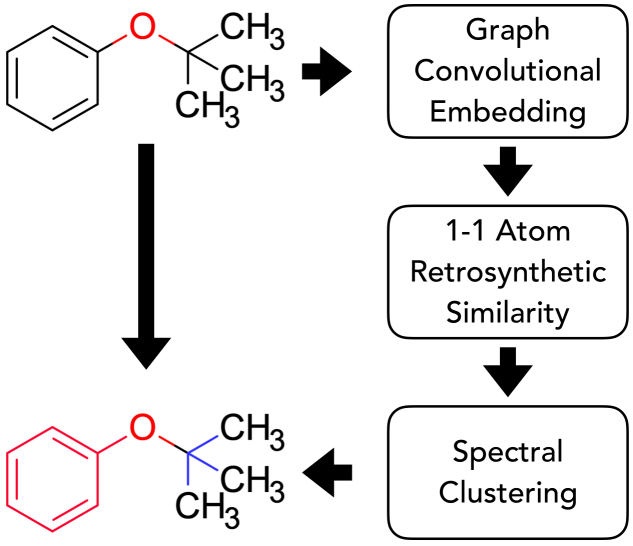

FIG. 3. Network architecture flow diagram depicting stacked phases of model. To the user, the only transformation of interest is the one visualized on the left hand side.

Additionally, considering observations that this formulation of spectral convolutions over molecular graphs had a tendency to overfit, we introduce an additional $f \times f$ stochastic knockdown matrix K, with threshold p. The value of any entry in K is zero with probability p, and one otherwise. This allows for our reformulation of Kipf's propagation rule as

$$H^{(l+1)} = \sigma\left(\hat{A} H^{(l)} \left(W^{(l)} \circ K\right)\right).$$

Here, $\circ$ denotes the element-wise Hadamard product, which is defined as $(A \circ B)_{ij} = A_{ij} \cdot B_{ij}$ for matrices A and B of equal dimensions. Knockdown regularizes the weight matrix to prevent it from learning a false correspondence between the numeric position of an atom in the molecule's node ordering and its retrosynthetic context. The use of convolutional knockdown helped prevent overfitting to a significant degree (see below).

After performing a convolutional embedding of the molecular features via feedforward through a multi-layer GCN, we have a modified features matrix $\tilde{X}$. From here, the network takes every unique pair $(i, j) \in \{0,1,\ldots,n-1\}^2$ where $i \neq j$ and passes the vectors at rows i, j of $\tilde{X}$ through a multilayer ANN, which calculates the retrosynthetic similarity $S_{ij} = S_{ji}$ (note S is symmetric). The diagonal of S has values of one, corresponding to an absolute surety that the same atom will be located in the same retrosynthetic precursor of a target molecule.

Having generated the retrosynthetic similarity matrix S, the final component of the network architecture performs spectral clustering over S. This work uses a modification of

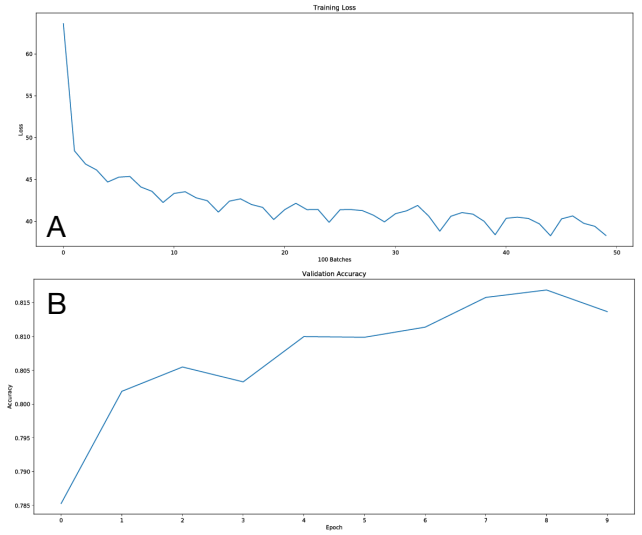

FIG. 4. Odachi model training performance. (A) shows the training loss, measured every 100 batches, or every 100,000 molecular examples. (B) shows the validation accuracy every epoch, or about every 400,000 molecular examples.

the spectral clustering algorithm formulated in Ng *et al.* that operates directly on a given adjacency matrix.[14] Given an adjacency matrix A representing a graph to be clustered into k components, we start by computing the diagonal matrix D where the i-th diagonal element of D is the sum of the i-th row of A and the Laplacian $L = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$. We then form the $n \times k$ matrix $X = [x_1 x_2 \ldots x_k]$, where x_i is the normalized i-th largest eigenvector of L (under the assumption that all x_i in X form an orthonormal basis). Finally, we cluster the graph nodes by clustering rows of X into k clusters via K-means and assigning nodes respectively. The network architecture we propose, depicted visually in Figure 3, uses spectral clustering as defined here to create a coloring over the molecule M's atoms, which directly corresponds to a retrosynthesis of M.

4. Engine Implementation

We implemented the network architecture described above in Python, using Google's Tensorflow framework to build custom graph convolutional layers with convolutional knockdown.[15] We used RDKit for cheminformatics tasks regarding molecular featurization. As a main goal of this research is to develop the niche of graph convolutional retrosynthesis, we make all our code and retrosynthetic engine available in the Odachi Retrosynthetic Engine package for Python, which can be accessed on GitHub (https://github.com/sudo-rushil/odachi) or PyPI (https://pypi.org/project/odachi/).

Our implemented featurization scheme converts a molecule represented as a SMILES string into a convolutional molecule, or Conv object, that stores the molecule's adjacency matrix and atom features. Atom features are represented either as binary values (e.g. part of an aromatic ring, etc.) or one-hot encoded values (e.g. atomic symbol, hybridization, etc.). The current version of the Odachi engine has support for molecules with up to 130 atoms, but future versions will likely expand this to allow for retrosyntheses of larger natural products, such as marine polyethers.

The Odachi Python package provides two novel Tensorflow objects: one GCN layer, and one model that acts as a stacked set of GCN layers. The GCN layer, GraphConv, performs a single convolutional step over an input adjacency matrix and atomic features matrix as described previously, and exposes a parameter for setting convolutional knockdown to mitigate overfitting. The stacked GCN model, ConvEmbed, wraps a stack of GraphConv layers for convenience, performing multiple convolutions over an input adjacency matrix and atomic features matrix to return the convolutional embedding of the atomic features matrix, subvectors of which can be classified with a standard binary classifier implemented in Tensorflow's Keras API.

Additionally, for chemists who want to utilize the Odachi Engine in designing retrosynthetic routes, we have the retrosynthesis.com website, which queries a Flask application exposing the Odachi Engine to serve users - who may have little to no understanding of the computational background of this research - retrosynthetic predictions. Details of the website's usage follows in later sections.

III. RESULTS AND DISCUSSION

1. Performance on Validation Dataset

We implemented the Odachi model as described in the section above in Python using Google's Tensorflow deep learning framework. As the spectral clustering algorithm does not require training, only the graph convolutional network and retrosynthetic similarity classifier were trained. Both were trained over 400,000 examples per epoch, with 4,000 examples per validation set per epoch. The network training goal was to perform binary classification between every pair of atoms in the molecule as to whether they were part of the same retrosynthetic synthons.

The network was trained with the Adam optimizer over ten epochs. It was trained for 2 weeks on an Intel 4790K. The training loss and validation accuracy over training is shown in Figure 4. The final validation accuracy of the model was 81.3%, averaged across 4,000 validation examples.

The use of convolutional knockdown, implemented as described above, was key to reducing overfitting. Figures 5 and 6 shows the effect of different rates of knockdown on the divergence of training and validation accuracy and loss. Increasing the level of knockdown, while slowing the training process down, allows This novel development allows the model to learn over molecules without developing strong biases with respect to the arbitrary ordering of atoms in the molecule's data representation.

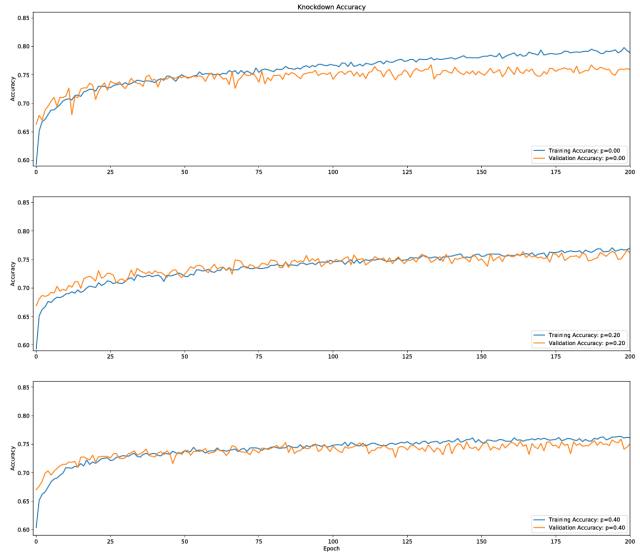

FIG. 5. Effect of knockdown on model accuracy. The horizontal axis shows epoch and the vertical axis shows accuracy. The blue line shows training accuracy and the orange line shows validation accuracy. The top, middle, and bottom correspond to knockdowns of 0, 0.2, and 0.4 respectively. Experiments ran with modified training framework.

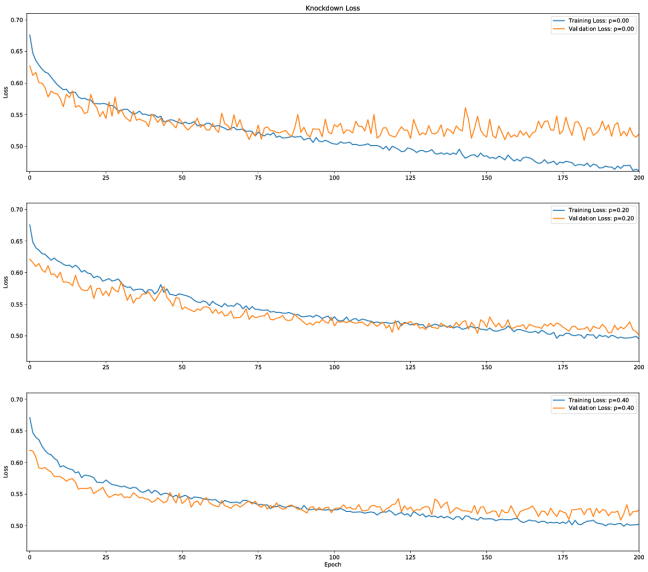

FIG. 6. Effect of knockdown on model loss. The horizontal axis shows epoch and the vertical axis shows loss. The blue line shows training loss and the orange line shows validation loss. The top, middle, and bottom correspond to knockdowns of 0, 0.2, and 0.4 respectively. Experiments ran with modified training framework.

It is important to note that the numerical retrosynthetic similarity classification accuracy is only a simplified metric of model performance. Due to the sensitivity of the spectral clustering process and the inherent stochasticity of the model, the retrosynthetic classification accuracy is not the only determining factor in the predicted retrosynthetic decomposition. However, as seen below, in most cases, the model is able to arrive at a valid synthesis in the first try, with some more complex molecules requiring a small number of reruns.

2. Performance on Prediction Examples

Individual analysis of the Odachi model's performance provides a better insight of the network's behavior than oversimplified numerical metrics. Figures 7 and 8 show the model's correct and incorrect predictions respectively on a random set of examples. These examples are drawn from Coley *et al.,* Wei *et al.*, and papers on natural product syntheses for their structural and functional variety.[8,9,11,16,17]

Figure 7a depicts the retrosynthesis of a diaryl molecule formed by acid-catalyzed dimerization of styrene. Aware of the chemical environment formed by the tertiary carbon and alkene, the suggested disconnection predicts the most homogenous retrosynthesis.

Figure 7b depicts the retrosynthesis of 1,4-dinitro-imidazole, in which the model recognizes that the N-nitro group is more susceptible to disconnection (i.e. more easily formed) than the C-nitro group.

Figure 7c depicts the retrosynthesis of a p-hydroxybenzoic acid derivative. The model realizes that the ester bond is much stronger than the oxygen-difluorocarbon bond, and recognizes the latter as the best disconnection.

Figure 7d depicts the retrosynthesis of the product of an aldol condensation. As it looks to recognize structural motifs, the model was able to identify the crossing double bond as the best disconnection to separate the synthons into tightly bound precursors.

Figure 7e depicts the retrosynthesis of an indazole. Here, the model correctly finds the two adjacent bonds needed to be disconnected to trace the target back to a suitable precursor.

Figure 7f depicts the retrosynthesis of a pharmaceutical compound via a Vilsmeier-type reaction. The model distinguishes between disconnecting the aryl-N bond and the N double bond, realizing that the aniline precursor is a better reaction candidate.

Figure 7g depicts the retrosynthesis of aspirin, a well-known reaction. The model is easily able to detect the ester bond as the most susceptible to disconnection. Furthermore, the model cleverly rejects alternative approaches to forming aromatic carbonyls, such as a Friedel-Crafts acylation-type disconnection.

Figure 7h depicts the retrosynthesis of a bicyclic hydrocarbon compound. The model is able to correctly identify a Diels-Alder type retrosynthesis, and recognizes which dienophile synthon is more likely to react. This demonstrates the capability of the graph convolutional network to identify structural motifs that can be easily disconnected

Transitioning to examples that the model did not correctly predict, Figure 8a depicts the predicted and reported retrosynthesis of the core ring system of the natural product gelsemoxonine.[16] The model attempts to

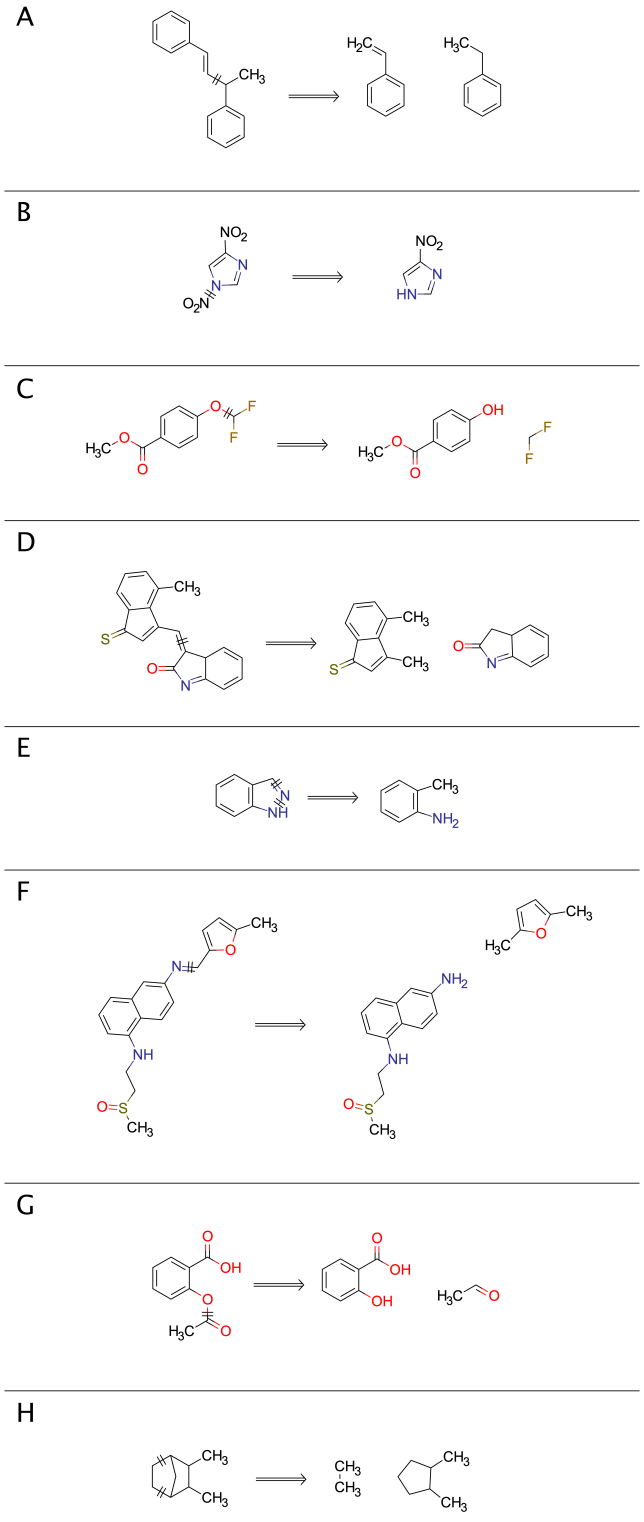

FIG. 7. Retrosynthesis examples where the model predicted the correct retrosynthetic disconnection. Note that the model only detects disconnections; visualizations show synthons, not reactants. Reactions: (a) acid-catalyzed dimerization; (b) nitration; (c) nucleophilic substitution; (d) aldol condensation; (e) indazole formation; (f) Vilsmeier-Haack; (g) esterification; (h) Diels-Alder.

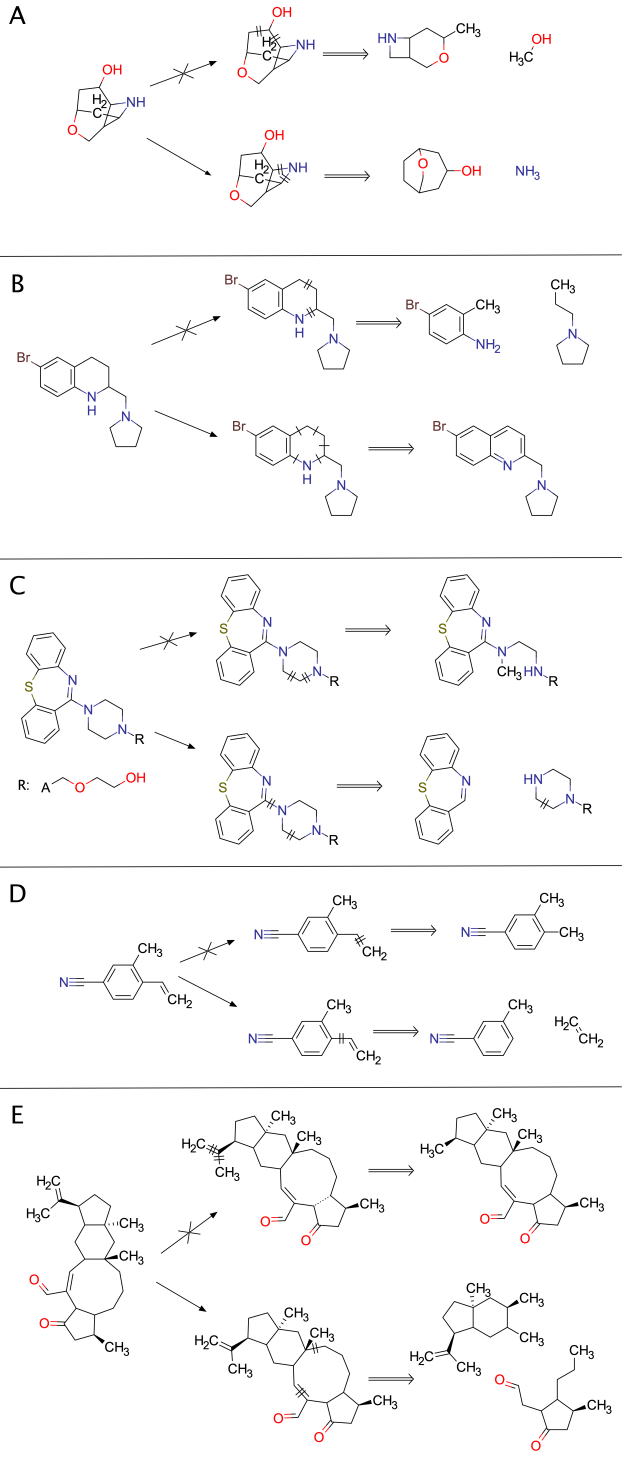

FIG. 8. Retrosynthesis examples where the model predicted a different retrosynthetic disconnection than reported. Note that the model only detects which bonds to disconnect; visualizations here are given for ease of understanding. Predicted [reported] reactions: (a) cyclization [azetidine formation]; (b) aza-cycloaddition [hydrogenation]; (c) ring-closing [Vilsmeier-Haack]; (d) homologation [Stille coupling]; (e) allylation [ring formation].

disconnect the seven-membered ring, not recognizing that the ring strain on the azetidine makes it a better candidate for disconnection. This is attributable to the few examples of strain-driven reactions in the training data, making it harder for the model to identify ring strain as a retrosynthetic marker. However, the model still suggests a disconnection which leads to a major structural and conceptual simplification, as it was designed to do.

Figure 8b depicts the predicted and actual retrosynthesis of a tetrahydroquinoline derivative. The model tried disconnecting the target molecule via an aza-cycloaddition pathway. However, as the model is built to recognize structural, and not functional retrosyntheses, it was unable to recognize the reported hydrogenation pathway as a retrosynthetic route. This demonstrates how the Odachi model is better at finding structural disconnections rather than in-place modifications to functional moieties.

Figure 8c depicts the predicted and actual retrosyntheses of the antidepressant quetiapine. The model tries to disconnect two adjacent bonds in an attempt to simplify the piperazinyl group while preserving the dibenzothiazepine core, but does not recognize the Vilsmeier-type disconnection as reported in the literature. This is likely due to the model's inability to recognize the bond between the two ring systems as a weak-enough disconnection candidate, but can also be affected by clustering stochasticity, as discussed below.

Figure 8d depicts the predicted and actual retrosyntheses of a cross coupling product between a benzonitrile and vinyltributyltin. The model attempted to disconnect the double bond, reflecting an implicit belief that direct aromatic substituents tend to be harder to form compared to other synthetic routes. However, this prediction likely reflects a lack of vinyl couplings in the USPTO data, as such methodologies have less structural intuitivity considering that the model has no information about the vinyl substituent.

Figure 8e depicts the predicted and actual retrosynthesis of the natural product Variecolin. Unlike the reported retrosynthetic methodology, the suggested retrosynthesis predicted removing the isoallyl group, likely as a consequence of recognizing the tight core ring structure. This behavior can also be attributed to clustering stochasticity. As a final disclaimer, it is important to note that the detection of a different disconnection than in the literature does not, as in the case of forward reaction prediction, inherently invalidate the ability of the model to provide insight in a synthetic planning context.

3. Website Interface and Performance

While the Odachi Retrosynthesis Engine is a powerful tool for performing predictive topological retrosynthesis, it is rooted in an incredibly technical codebase outside the scope of expertise of the majority of organic and synthetic chemists. In order to fulfill the project goal of allowing the maximum number of people to benefit from this technology, we developed the retrosynthesis.com website. This website provides a simple and intuitive interface to

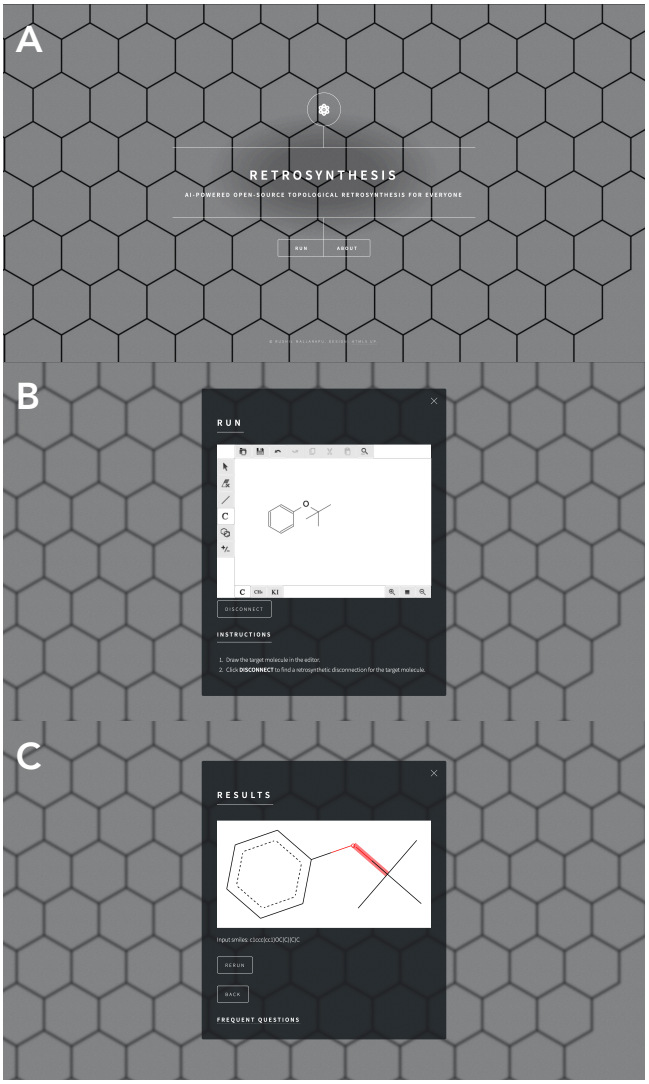

FIG. 9. Interface for retrosynthesis.com. (A) Homepage, with title and description. (B) Run page, with chemical editor and test example. (C) Results page, with rendered prediction and molecular SMILES.

serve retrosynthetic predictions, as shown in the workflow example below. The website was designed in accordance with the following principles:

- *Simplicity*: retrosynthesis.com must be easy to use, especially by chemists with no background in computer science or the details of the retrosynthesis engine.
- *Intuitiveness*: The website interface must be obvious enough that users face little ambiguity when trying to get a prediction.
- *Minimalism*: The website must, in the interest of speed and simplicity, only maintain the features and information necessary, providing links to outside resources if needed.

Workflow Example: Starting at the website homepage, there is a brief explanation of what the website does. There are only two buttons, RUN and ABOUT. The homepage is

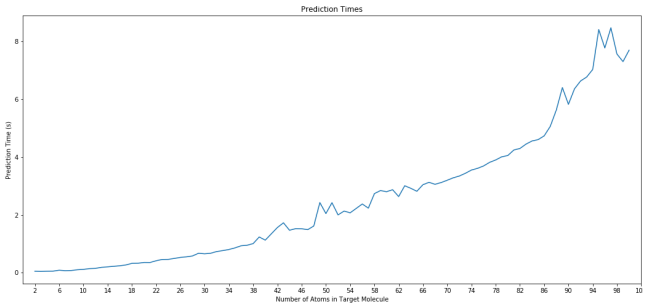

FIG. 10. Graph of prediction timing. Horizontal axis shows molecule size, and vertical axis shows prediction times in seconds.

shown in Figure 9a. This clean layout is a hallmark of the minimalist principle.

Clicking on RUN, an intuitive decision given that it is the only button indicative of performing an action, one is brought to a molecular editor, underneath which is a button labelled DISCONNECT. The run page is shown in Figure 9b. Along with the brief instructions below the editor, the simple layout allows for a straightforward and intuitive user experience.

Clicking on DISCONNECT runs the retrosynthetic prediction, and brings up a results page with the rendered prediction and molecular SMILES. An example results page is shown in Figure 9c. The page has two buttons; BACK leads to the editor page, allowing the user to change the input molecule without having to start from scratch, and RERUN queries the model again, to help solve issues with clustering stochasticity. This is all that needs to be done to get a prediction, and the entire process is extremely straightforward, enabling the ease of both use and widespread adoption of the technology.

Using retrosynthesis.com is also extremely quick. Timing experiments ran on both the engine and render showed the entire process takes approximately 300 milliseconds, which allows for an enjoyable user experience and rapid synthetic prototyping by the user. More details about the model timing and performance are below.

4. Timing and Limitations

Figure 10 shows how the model's prediction time increases with the number of atoms in the query molecule. This behavior is understandable, due to the fact that building the retrosynthetic similarity matrix requires comparing every combination of atoms in the molecule, resulting in a runtime quadratic in the number of atoms.

In comparison with Segler *et al.*'s rule-based retrosynthesis methodology, our model runs around eight times slower (200 ms per molecule on a 2017 MacBook Pro), which is expected given both the lack of referential rules to determine the synthetic pathway and the computational overhead of performing sequential graph convolutions.[10] In the context of use-cases, it is important to note that our model has no dependence on the size of the set of reference rules; it is singularly affected by the size of the input molecule. The prediction times, while a potential point of improvement, are extremely fast with regards to performing such parallelism-intensive computations, having no reliance on reference knowledge, and being pseudo-instant as of the average end-user perspective.

There are certain limitations of the current model that we address. Primary to all these is the fact that there exist few open datasets of examples of pure retrosynthetic problems. This means data for such projects has to be derived from datasets containing reactions known to run in the forward direction. As with all deep learning networks, this introduces an implicit bias in the model towards making retrosynthetic predictions in line with what it was trained on. For instance, if the model had not seen any examples of a Diels-Alder cycloaddition, it would have a higher probability of suggesting other ring-forming reactions than a Diels-Alder.

However, it should be understood that while the training dataset was relatively small compared to Segler *et al.*, it contained a wide variety of examples.[10] Moreover, the model's focus on performing structurally-inspired topological retrosynthesis gives this model a clear advantage when it comes to identifying the most vulnerable bond in a molecule with multiple likely disconnection points.

Finally, due to the use of spatial graph convolutions, there is an upper bound on the maximum number of atoms in a query molecule. Currently, the maximum query size is 130 atoms. This is because all the molecules in our dataset had a maximum size of 122 atoms, and the vast majority had a size under 60 atoms. As such expanding the max atom size farther reduces the training efficacy, as the model has no comparable examples of such size. Future iterations will be able to process larger molecules, dependent on having more comprehensive training data.

IV. CONCLUSION

In summary, we have developed the Odachi Retrosynthesis Engine, a novel graph convolutional model to predict topologically-informed retrosyntheses of molecules without the use of hardcoded reaction rules or templates. The prediction examples demonstrate that graph convolutional networks can learn to recognize structural motifs as indicators of potential retrosynthetic disconnections. This is a major step forward, as graph convolutional architectures are powerful tools for learning over structural connectivity, allowing for more advanced retrosynthesis models. The codebase for this project has been open-sourced to allow for public usage, and an intuitive website has been implemented to allow chemists to utilize intelligent retrosynthetic predictions in their work. Future work will focus on refining the clustering algorithm to detect viable synthons quickly, increasing the scope of retrosynthesis data, and optimizing the model for performance and speed.

ACKNOWLEDGEMENTS

The first author thanks their father for his domain expertise. They thank Dr. Bruce Lipshutz, Dr. Balaram Takale, Dr. Ruchita Thakore, Dr. Tomislav Rovis, Dr. Nicholas Tay, and Dr. Jessica Davis-Peineke for their invaluable teachings on what being a chemist is. They also thank Cecil Yang, for the inspiration to pursue this research. They finally thank their friends and family for their timeless support.

REFERENCES

[**1**] Nicolaou, K. C. *Proc. R. Soc. A,* **2014**, 470, 20130690.
[**2**] Corey, E. J. *Chem. Soc. Rev.*, **1988**, 17, 111-133.
[**3**] Reisch, M, S. Chemical Engineering and News. Research spending continues on an upward trajectory. https://cen.acs.org/business/investment/Research-spending-continues-upward-trajectory/97/i23 (accessed Feb 20, 2020).
[**4**] Klucznik, T.; Mikulak-Klucznik, B.; McCormack, M. P.; Lima, H.; Szymkuc, S.; Bhowmick, M.; Molga, K.; Zhou, Y.; Rickershauser, L.; Gajewska, E. P.; Toutchkine, A.; Dittwald, P.; Startek, M. P.; Kirkovits, G. J.; Roszak, R.; Adamski, A.; Sieredzinska, B.; Mrksich, M.; Trice, S. L. J.; Grzybowski, B. A. *Chem*, **2018**, 4, 522-532.
[**5**] Wan, W. A. *Mol. Inf.*, **2014**, 33, 469-476.
[**6**] Coley, C. W.; Thomas, D. A.; Lummiss, J. A. M.; Jaworski, J. N.; Breen, C. P.; Shultz, V.; Hart, T.; Fishman, J. S.; Rogers, L.; Gao, H.; Hicklin, R. W.; Plehlers, P. P.; Byington, J.; Plotti, J. S.; Green, W. H.; Hart, A. J.; Jamison, T. F.; Jensen, K. F. *Science*, **2019**, 365, eaax1566.
[**7**] Jin, W.; Coley, C. W.; Barzilay, R.; Jaakkola, T. 2017, arXiv:1709.0455v3. arXiv.org e-Print archive. https://arxiv.org/abs/1709.04555 (accessed Oct 27, 2019).
[**8**] Wei, J. N.; Duvenaud, D.; Aspuru-Guzik, A. *ACS Cent. Sci*., **2016**, 2, 725-732.
[**9**] Coley, C. W.; Barzilay, R.; Jaakkola, T. S.; Green, W. H.; Jensen, K. F. *ACS Cent. Sci.*, **2017**, 3, 434-443.
[**10**] Segler, M. H. W.; Waller, M. P. *Chem. Eur. J.*, **2017**, 23, 5966-5971.
[**11**] Coley, C. W.; Jin, W.; Rogers, L.; Jamison, T. F.; Jaakkola, T. S.; Green, W. H.; Barzilay, R.; Jensen, K. F. *Chem. Sci.*, **2019**, 10, 370-377.
[**12**] Kipf. T. N.; Welling, M. arXiv:1609.02907. ArXiv.org e-Print archive. https://arxiv.org/abs/1609.02907 (accessed Oct 27, 2019).
[**13**] Altae-Tran, H; Ramsundar, B.; Pappu, A. S.; Pande, V. *ACS Cent. Sci.*, **2017**, 3, 283-293.
[**14**] Ng, A. Y.; Jordan, M. I.; Weiss, Y. *On Spectral Clustering: Analysis and an algorithm*, Advances in Neural Information Processing Systems, 2001, Dietterich, T. G.; Becker, S.; Ghahramani, Z., Eds.; MIT Press:Cambridge, 2001.
[**15**] Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Goodfellow, I.; Harp, A.; Irving, G.; Isard, M.; Jozefowicz, R.; Jia, Y.; Kaiser, L.; Kudlur, M.; Levenberg, J.; Mané, D.; Schuster, M.; Monga, R.; Moore, S.; Murray, D.; Olah, C.; Shlens, J.; Steiner, B.; Sutskever, I.; Talwar, K.; Tucker, P.; Vanhoucke, V.; Vasudevan, V.; Viégas, F.; Vinyals, O.; Warden, P.; Wattenberg, M.; Wicke, M.; Yu, Y.; Zheng, X. TensorFlow: Large-scale machine learning on heterogenous systems. Whitepaper, 2015.
[**16**] Shimokawa, J; Harada, T; Yokoshima, S; Fukuyama, T. *J. Am. Chem. Soc.*, **2011**, 133, 17643-17637.
[**17**] Molander, G. A.; Quirmbach, M, S.;Silva Jr, L. F.; Spencer, K. C.; Balsells. *Org. Lett.*, **2001**, 3, 15, 22572260.

APPENDIX

1. Code access

The source code for the Odachi retrosynthesis engine can be found on GitHub (https://github.com/sudo-rushil/odachi). The packaged engine can be found on PyPI (https://pypi.org/project/odachi/). All code is released under an MIT license.