doi.org/10.26434/chemrxiv.12302084.v2

Generating Transition States of Isomerization Reactions with Deep Learning

Lagnajit Pattanaik, John Ingraham, Colin Grambow, William H. Green

Submitted date: 15/05/2020 • Posted date: 18/05/2020

Citation information: Pattanaik, Lagnajit; Ingraham, John; Grambow, Colin; Green, William H. (2020): Generating Transition States of Isomerization Reactions with Deep Learning. ChemRxiv. Preprint. https://doi.org/10.26434/chemrxiv.12302084.v2

Lack of quality data and difficulty generating these data hinder quantitative understanding of reaction kinetics. Specifically, conventional methods to generate transition state structures are deficient in speed, accuracy, or scope. We describe a novel method to generate three-dimensional transition state structures for isomerization reactions using reactant and product geometries. Our approach relies on a graph neural network to predict the transition state distance matrix and a least squares optimization to reconstruct the coordinates based on which entries of the distance matrix the model perceives to be important. We feed the structures generated by our algorithm through a rigorous quantum mechanics workflow to ensure the predicted transition state corresponds to the ground truth reactant and product. In both generating viable geometries and predicting accurate transition states, our method achieves excellent results. We envision workflows like this, which combine neural networks and quantum chemistry calculations, will become the preferred methods for computing chemical reactions.

GENERATING TRANSITION STATES OF ISOMERIZATION REACTIONS WITH DEEP LEARNING

A PREPRINT

Lagnajit Pattanaik
Department of Chemical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139
lagnajit@mit.edu

John B. Ingraham
Department of Computer Science
and Artificial Intelligence
Massachusetts Institute of Technology
Cambridge, MA 02139
ingraham@csail.mit.edu

Colin A. Grambow
Department of Chemical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139
cgrambow@mit.edu

William H. Green*
Department of Chemical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139
whgreen@mit.edu

May 14, 2020

ABSTRACT

Lack of quality data and difficulty generating these data hinder quantitative understanding of reaction kinetics. Specifically, conventional methods to generate transition state structures are deficient in speed, accuracy, or scope. We describe a novel method to generate three-dimensional transition state structures for isomerization reactions using reactant and product geometries. Our approach relies on a graph neural network to predict the transition state distance matrix and a least squares optimization to reconstruct the coordinates based on which entries of the distance matrix the model perceives to be important. We feed the structures generated by our algorithm through a rigorous quantum mechanics workflow to ensure the predicted transition state corresponds to the ground truth reactant and product. In both generating viable geometries and predicting accurate transition states, our method achieves excellent results. We envision workflows like this, which combine neural networks and quantum chemistry calculations, will become the preferred methods for computing chemical reactions.

1 Introduction

Computational methods that provide detailed knowledge about the reactivity of chemical species are becoming increasingly reliant on large datasets. For example, quantitative modeling of gas-phase systems such as combustion and pyrolysis require thousands of thermodynamics and kinetics data points to produce a mechanism used to predict just a few observables [1–3]. Spaces where quantitative data are lacking, such as organic reaction prediction and retrosynthesis, use databases populated with millions of qualitative reaction examples (i.e., major product only) [4, 5]. Because there are insufficient high-quality quantitative reaction data, these studies are typically either supplemented by experimental studies [6, 7] and/or are severely limited in scope [8, 9]. Indeed, creating and expanding databases with reaction rates and energetic barriers remains challenging, especially since such a task necessitates the generation of transition state (TS) structures for each reaction.

A breadth of literature describes TS structure generation as a bottleneck issue in kinetic modeling. Optimization methods to converge initial guesses for TS structures to saddle points on a potential energy surface (PES) are well-established, but producing viable initial guesses remains difficult [10–13]. Although researchers continue to make such guesses manually by using expert-guided decisions to position atoms by hand, automated workflows do exist. Linear or quadratic interpolations between reaction and product structures are quick but error-prone methods that provide an automated

method to produce a viable TS guess [14, 15]. Advanced double-ended methods, which begin with optimized reactants and products, attempt to connect these ground state structures through snapshots of atomic configurations, which are then reoptimized to converge to a TS [16–21]. Single-ended methods, which begin with only the reactant structures, iteratively or systematically alter the input to reach a TS [22–29]. These algorithms, which rely on expensive ab initio or density functional theory (DFT) calculations, require significant computation to arrive at an initial guess, which often fails with subsequent optimization. Heuristic approaches exist to avoid this computational expense, such as KinBot [30, 31] and AutoTST [32, 33], which reposition reactant atoms based on expert-defined templates or molecular group contributions. While these heuristic methods address TS generation in a high-throughput sense, they are limited in scope; users must explicitly add transformation classes before the algorithm can make predictions for new chemistries. Advantages and drawbacks of these various methods are discussed in depth elsewhere [34–38].

Recently, deep learning methods have advanced workflows in chemistry relating to organic synthesis planning [39–43], forward reaction prediction [44–48], property prediction [49–54], and generative optimization [55–61]. However, few approaches tackle three-dimensional structure generation. Most research in this space focuses on protein structure prediction, which benefits from their linear and constrained arrangement [62–65]. The few studies that investigate small molecules target equilibrium geometries and conformer generation [66–70]. Notably, several groups use machine learning (ML) to train force fields at ab initio accuracy [71–73]. Training an ML force field to generate TS structures is an enticing approach but requires significant amounts of off-equilibrium data for generalizablity. Gerrits et al. used this approach to study the surface reactivity of CHD_3 on Cu(111) and required 38000 data points to train their neural network potential [74]. Thus, here we aim to directly predict elements of the TS structure rather than trying to predict the full PES.

Specifically, we describe a novel method to generate 3D TS structures in a data-driven fashion. Our automated method requires optimized reactants and products as inputs and produces coordinates of the TS as the output. The key innovations in our method are its end-to-end differentiability combined with its power to both learn the TS distance matrix and weight the importance of specific distances during structure generation. Our goal is to produce viable initial guesses for subsequent eigenvector-following optimization methods faster than traditional quantum chemical methods and in a less-restricted manner than the available heuristic options.

2 Methods

The workflow to generate transition state geometries involves several steps, which we roughly divide into featurization, prediction, and optimization. We first generate attributed graph structures of the reactant and product and effectively average them to create a graph structure representative of the TS. We feed this TS graph through a fully-connected graph neural network (GNN), which generates an updated graph representation of the TS [75, 76]. An additional dense layer receives the updated TS graph and predicts values for two matrices necessary for a nonlinear least squares (NLS) optimization. The first matrix, D_{init}, is the network prediction of the TS distance matrix, which represents Euclidean distances between all pairs of atoms. The second matrix, W, is a set of weights used for the NLS optimization. Each entry in W corresponds to a pairwise distance in D_{init} and denotes the importance of each distance during reconstruction of the TS coordinates. The final coordinates of the TS, X, are the result of the NLS optimization, which minimizes the weighted residuals between D_{init} and pairwise distances in X. Figure 1 highlights the key steps in the method, while the appendices provide further details. Our code is available on GitHub at `https://github.com/PattanaikL/ts_gen`.

We train the model on a recently published dataset of gas phase organic chemistry reactions [77]. These data were generated with the single-ended growing string method, so the reactions correspond to elementary steps through a single saddle point on the PES. All reactions involve anywhere from three to seven heavy atoms including carbon, oxygen, and nitrogen. We filter the data to reactions that contain a single product to include only isomerization reactions. Ref. 77 provides the data at two levels of DFT, but we limit our focus to the high level (ωB97X-D3 with a def2-TZVP basis set). Our final dataset consists of approximately 1000 unique reactants and 8500 reactions, which we divide randomly in an 80:10:10 split for training, validation, and testing.

To train and test the model, we first transform individual log files from ref. 77 into structure-data files (SDFs) for the reactant, product, and TS. Our workflow uses reactant and product data to create an input attributed graph G with vertices (atoms), edges (bonds), and corresponding initial features for atoms and bonds. Note that since both the reactant and product contain the same number of atoms, it is straightforward to build G as an averaged representation for the TS. Importantly, the edge features include the averaged distance between atoms i and j between the reactant and product along with whether or not a bond is broken or formed and if the bond is aromatic. The atom features only include the identity of the atom encoded numerically via atomic number and with a one-hot vector.

The attributed 2D graph representation serves as the input to the GNN, which follows an update procedure for nodes and edges similar to ref. 76. First, we feed both the atom and bond features through dense layers to initialize the network.

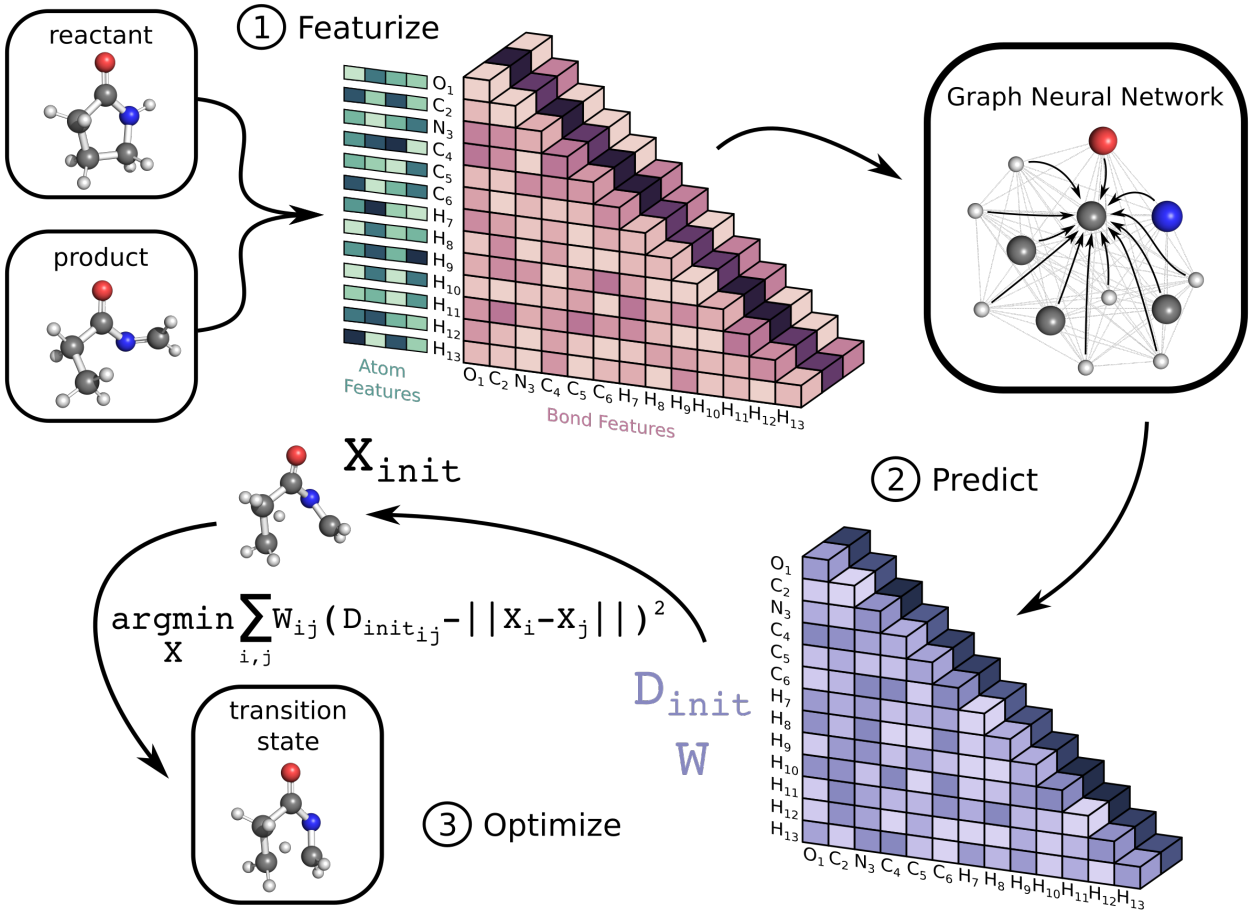

Figure 1: Illustration of workflow used to generate TS structures. Given input 3D reactant and product structures, we generate an initial representation of the TS graph by calculating averaged atom and bond features. We feed this input to a GNN, which generates an updated representation of the TS graph. We then pass this updated representation through a dense layer to predict the TS distance matrix, D_{init}, and the NLS weight matrix, W. Finally, we perform an NLS optimization to recover the TS coordinates, X.

Next, the GNN iteratively updates the bond and atom features, and since this network is fully connected, it updates each edge and vertex with information from all other edges and vertices. After three iterations, we feed the final edge representation to another dense layer and symmetrize the output. The corresponding tensors represent the model's prediction of the TS distance matrix (D_{init}) and a weight matrix (W) for the forthcoming optimization. We detail the mathematical formalization of this network in Appendix A.

We generate an initialization of the TS geometry (X_{init} in Figure 1) for the NLS optimization in two steps using multidimensional scaling theory. First, we calculate the Gram matrix with the predicted D_{init} matrix. Then, we perform an eigendecomposition of the Gram matrix to calculate X_{init} by concatenating the eigenvectors corresponding to the three largest eigenvalues. To recover the final TS coordinates, we optimize this initialization with an NLS algorithm defined by the following:

$$\underset{X}{\text{argmin}} \sum_{ij} W_{ij}(D_{init_{ij}} - ||X_i - X_j||)^2 \tag{1}$$

The desired TS geometry is a result of minimizing the weighted residuals between D_{init} and pairwise distances in X. When training the network, we calculate network loss by first recalculating the distance matrix from X (which we denote as D) and computing the absolute difference between D and the ground truth distance matrix given by ref. 77. We backpropagate this loss through the coordinate recovery procedure and the GNN in an end-to-end manner. This way, we not only learn the TS distance matrix prediction (D_{init}), but we also learn to weight the distance prediction appropriately (W). The end-to-end differentiability of our network combined with its ability to both learn and weight

the TS distance matrix prediction are the crucial aspects of our method that make it successful. Appendix A provides further details regarding the mathematics.

We employ a TS verification workflow similar to ref. 32. The automated method first uses the TS geometry generated by the model as an initial guess in a TS optimization with the Berny method [10] and verifies the presence of a first-order saddle point by checking for exactly one imaginary frequency. It then feeds the optimized transition state to two intrinsic reaction coordinate (IRC) calculations, which link the saddle point to its corresponding reactant and product PES minima by a mass-weighted downhill optimization [78]. Thus, we spawn forward and reverse IRCs on the optimized TS to generate the corresponding reactant and product, which we further verify with a stationary point optimization and frequency calculation, ensuring all frequencies are real. Our workflow feeds these generated reactant and product geometries to OpenBabel, which infers connectivity using its "connect-the-dots" method by adding bonds to atoms closer than their combined covalent radii while maintaining minimum distance and valence constraints [79]. We use these connections to compare graph connectivity between IRC reactants and products and original reactants and products. Matches between both sets for both molecules correspond to a successful TS generation.

3 Results and Discussion

Before submitting the structures generated by the model to the verification workflow, we ensure that our method predicts valid geometries. That is, we evaluate whether the method in Figure 1 independently produces realistic and realizable distance matrices. To check this, we plot the distribution of normalized distances generated by our model before and after the NLS optimization (D_{init} and D, respectively) and compare it to the ground truth distribution of normalized distances reported by ref. 77, shown in Figure 2A. The dashed lines correspond to the model predictions, while the solid line corresponds to the ground truth distribution. Note that we limit these distributions to the reaction core (i.e., bonds that changed during reaction) to further highlight differences. Even before optimization, the D_{init} distribution closely resembles the ground truth distribution; after optimization, the D and ground truth distributions are nearly identical. We additionally compare against a fourth distribution of distances generated by using the average distance between reactant and product, shown as the dotted line in Figure 2A. This distribution is significantly different from the other three, further verifying that our model generates sound geometries to use as initial guesses.

To understand why the model produces valid geometries, we investigate the weight matrices predicted by the model, which emphasize which bonds are considered important during reconstruction of the final TS guess geometry (step 3 in Figure 1). We extract these weights from our network and normalize them by the maximum weight for each molecule. We then bin the weights based on the average topological distance between reactant and product (i.e., the average number of bonds separating the two atoms to which the weight corresponds), shown in Figure 2B, along with counts for each bin. As topological distance between atoms increases, the value of the weight assigned to this distance decreases. That is, the model prioritizes preserving distances of nearby atoms when reconstructing coordinates from the predicted distance matrix. Again, this reinforces that the model is learning which distances are important and focuses on retaining these distances during generation of the TS structure.

We evaluate overall performance on a held-out test set on which the model performs well, achieving an average test loss of 0.11 angstroms (average atomic difference in distance between model and ground truth structures). After investigating several trends, we report a noteworthy correlation, plotted in Figure 2C. To generate this plot, we extract learned embeddings of each reaction as the output of the GNN in both the training and test sets. We calculate pairwise cosine distances between each point in the two sets and plot the minimum cosine distance value for each test data point against its loss from the network. Thus, the plot claims that network loss increases as the similarity between the test data point and training data decreases. While we expect this trend, its proof allows us to add a pseudo-confidence metric to each new TS prediction: any new reaction (within the interpolation space of the training data) that is similar to data in the training set (as defined by the GNN) should generally receive a reliable prediction from the network.

It is important to recognize the multiple success criteria we use to assess our results. While the deep learning method generates TS geometries based on difference in distances from the ground truth (network loss), our true measure of success is defined by the verification workflow. We employ this workflow on the test set of 834 reactions and achieve a 71% success rate, correctly following 595 initial guesses generated by our model to the ground truth reactants and products. Because the dataset contains a number of uncommon reactions, we check performance on the subset of reactions that match reaction families defined in the Reaction Mechanism Generator (RMG) [3]. On this set of well-known gas-phase reaction types, our model achieves a success rate of 89%. We also assess the various failure modes: 41% of failures involve failed TS optimizations, 6% involve failed IRCs, 21% involve failed ground state optimizations, and the final 32% of failures are mismatches between the calculated ground state structures and the reported reactant and product structures. Of the TS optimization failures, 81% exceed the maximum number of steps allowed per optimization, 10% involve symmetry issues related to the generated internal coordinates, 7% identify the

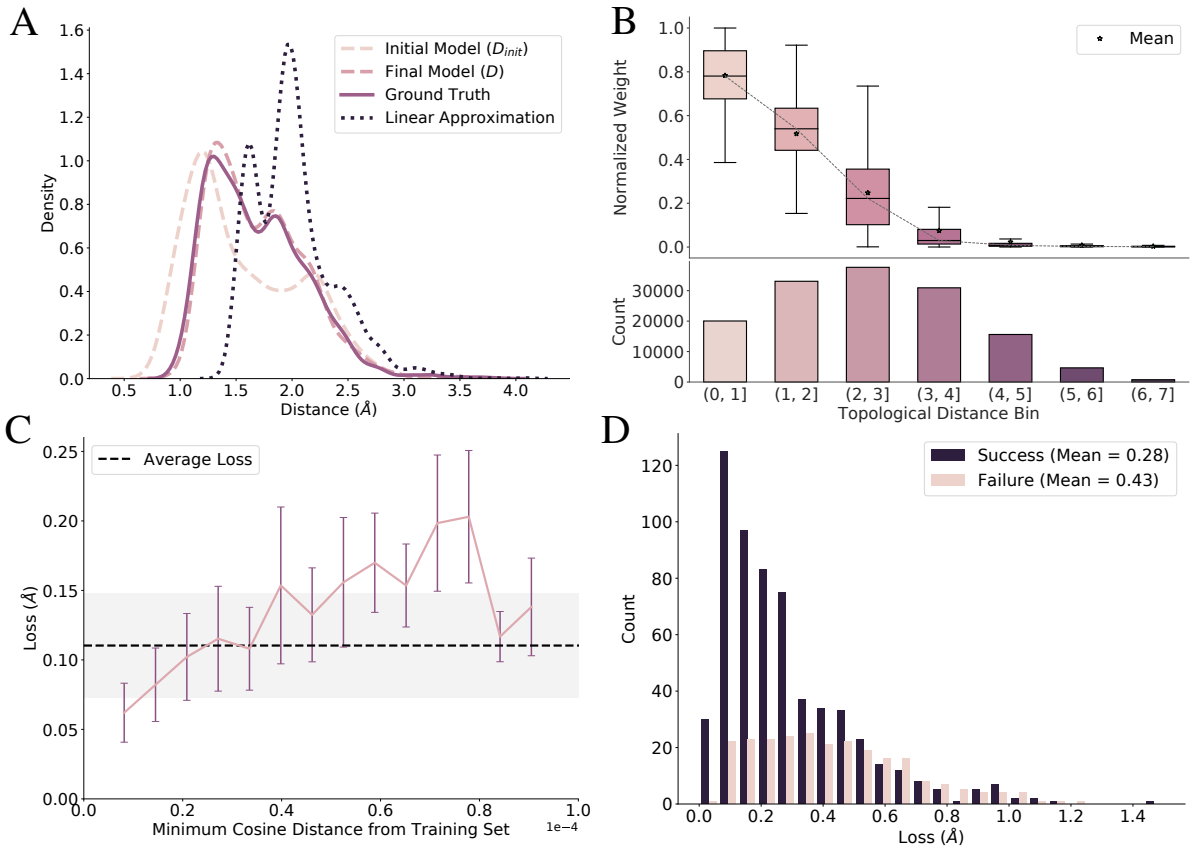

Figure 2: A) Initial model, final model, ground truth, and linear approximation distribution of distances between atoms of the reaction core of TS structures in the test set. The final model and ground truth distributions are nearly identical, while the linear approximation distribution is different. B) Box plots of normalized weight matrices binned by average topological distance between reactant and product of test set reactions. Values of weights decrease as number of bonds between atoms increase. C) Minimum cosine distance of test set from training set using learned embeddings from the GNN. Network loss increases as the distance between the test data point and training data increases. Each full bar corresponds to a single standard deviation. D) Loss distributions of successful and failed reactions in the test set as defined by the verification workflow. The distributions are markedly different but stress room for improvement.

incorrect number of imaginary frequencies, and 2% involve explicit self-consistent field (SCF) optimization failures. Clearly, a plurality of the failures involve inadequate initial guesses, motivating further optimization of our network. We ensure the deep learning method is compatible with our true measure of success by plotting the loss distributions of the successes and failures as defined by the verification workflow, shown in Figure 2D. While the distributions are clearly different, the overlap suggests that the naïve network loss function (difference in distances) we use could be improved. We further comment on this issue after investigating specific examples.

Figure 3 illustrates several examples of reactions fed to our model; each illustration positions the reactant and product at the beginning and end of the arrow, respectively, and indicates the ground truth TS with a blue highlight (left) and the model-generated TS with a green highlight (right). A.1 shows a simple 1,3-migration reaction, predicted correctly by the linear approximation method as well. Our network correctly predicts many similar simple reactions. A.2 shows a more complicated example, where the geometry of the migrating group changes during the transformation. While the linear approximation method does not correctly predict this TS structure, our method effectively learns the geometric changes necessary to generate this TS. Note that, while our method slightly rotates the oxygenated group, the guess geometry is adequate enough to converge to the saddle point structure with the subsequent Berny optimization. In other words, many of the guesses generated by our method are “close enough” to the target geometry to pass the subsequent verification workflow.

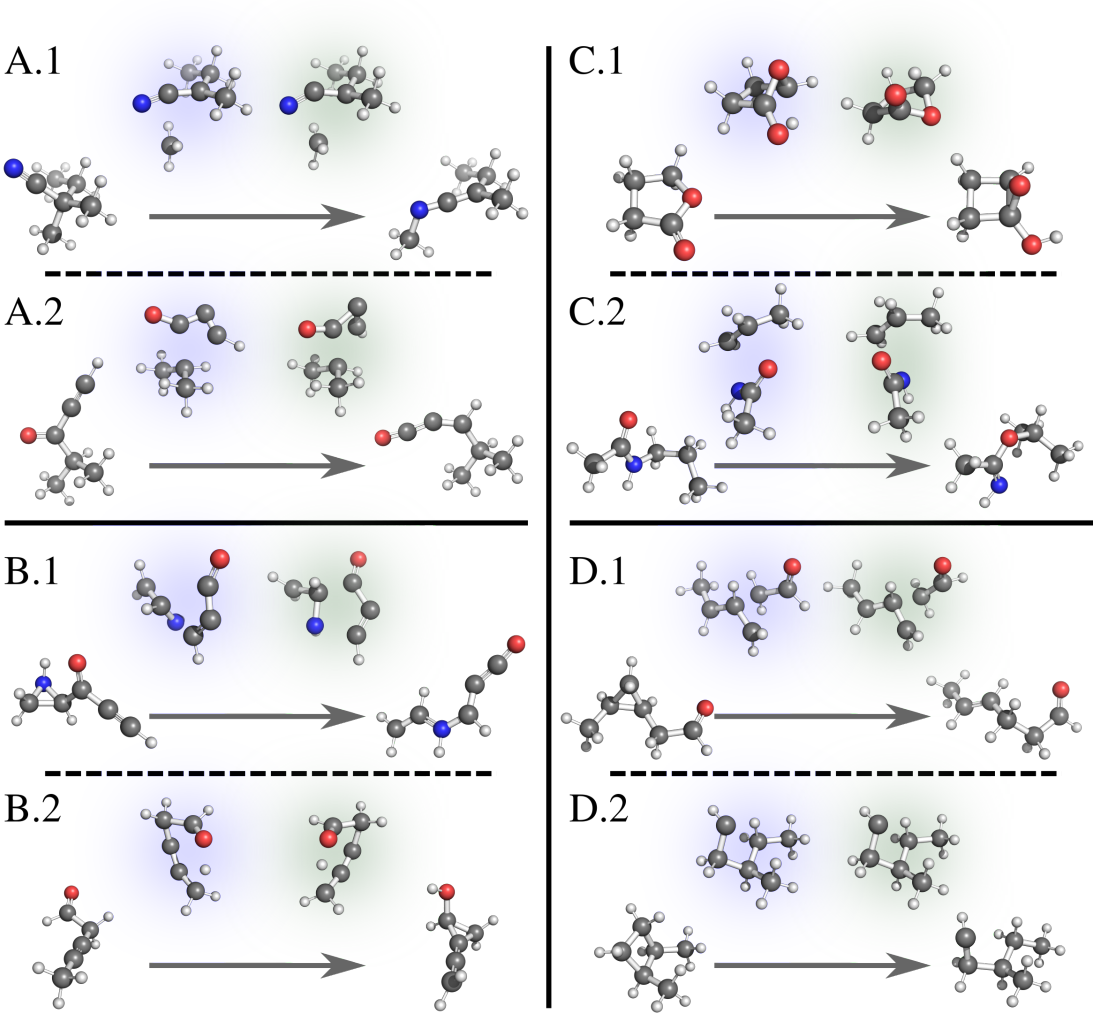

Figure 3: Example reactions with reactant and product at the beginning and end of the arrow, respectively, showing the ground truth TS with a blue highlight (left) and the model-generated TS with a green highlight (right). A.1) A simple reaction predicted correctly by the model. A.2) A complicated reaction predicted correctly by the model. B.1) A complex reaction predicted incorrectly by the model. B.2) A TS where the model predicts a mirror image of the correct structure but fails the verification workflow. C.1 and C.2) TSs where the model predicts a mirror image of the correct structure, and the TSs pass the verification workflow. D.1) A reaction with a deceptively large neural network loss that passes the verification workflow. D.2) A reaction with a near zero neural network loss that surprisingly fails the verification workflow. We render all 3D structures with PyMol [80].

Of the reactions that fail verification, many mirror example B.1. Here, several bonds break and form at once, as the linear oxocarbon group rearranges its connection to the aziridine group, which also simultaneously breaks apart. The model does not capture the orientation between the two groups well enough to pass a Berny optimization. Similar complicated reactions with large distance changes between reactant and product structures compose the failures (Figure 4). B.2 shows an example of a different but more significant failure mode. While the model TS structure correctly captures the transformation, it is nearly a mirror image of the reported transition state. Because we use the distance matrix to generate a TS structure, the target geometry is not entirely unique, and in a few cases, this fact results in a converged TS guess that contains one or more planes of symmetry with the ground truth geometry. Since our goal is to generate TS geometries to calculate rate constants, this is not a significant issue; a mirror image-converged TS produces the same numerical values for a rate constant. B.2, however, is an example where the mirror TS fails subsequent Berny optimization, which is problematic. Examples C.1 and C.2 show similar TS structures generated by our model that are near mirror images of the reported ground truth but pass the verification workflow. Again, while this is not always an issue, we designate these examples as "false positives" and are actively working on modifications to relieve this issue. For database construction, we suggest using the IRC-optimized reactants and products.

While investigating specific examples, we also discover failed structures we designated as "false negatives." These examples are a direct result of our multiple criteria for success discussed earlier: loss as calculated by the network and success as defined by the verification workflow. Example D.1 shows a structure that registers a large loss value from the network but passes the verification workflow. In this example, the model TS methyl and aldehyde groups are rotated from the ground truth. While this difference in distances triggers a large loss, it is easily adjusted during subsequent optimization, further suggesting that using absolute difference of distances as a loss function for the network is a naïve option that should be improved. Conversely, D.2 shows a structure that registers a near-zero network loss but fails the verification workflow. Initial glance at the structure shows little difference between model and ground truth geometries, suggesting an unknown job error. We run each quantum calculation at a set of default parameters without further refinement for failed jobs; we are also implementing an automated strategy to restart failed jobs.

The incapability of the model to capture orientation constraints represents a significant limitation to the current model. Again, while this issue, which manifests itself with symmetric structures from the reported ground truth, does not usually affect any derived rate constants, it may affect generation of complex TS structures in the future, such as those for metal-catalyzed reactions with intricate ligand interactions. The current model design also prevents TS generation for multi-reactant reactions. Such a model would need to orient multiple reactants together before generating a TS structure; we leave this task for future models. Finally, input reactions to the model must indicate the atom-mapping between reactants and products. While this is the case for most double-ended methods, it represents a barrier towards a fully-automated TS search algorithm. Fortunately, automated atom-mapping algorithms do exist and researchers continue to refine and improve them [81–83].

4 Conclusion

The availability of high-quality data and the recent prevalence of deep learning in the natural sciences have combined to usher in a new era of predictive chemistry. Computation especially offers a platform to both generate large quantities of data and extract trends generally applicable to out-of-domain samples. Here, we developed a deep learning model to generate three-dimensional transition state structures given the geometries of the reactant and product trained on a diverse dataset of gas-phase organic reactions. Our method produced viable geometries before QM optimization and achieved excellent results on an external test set via a rigorous verification workflow. We hope that users adopt our method to generate transition state structures relevant for their research and use our model as a basis for architectures involving even more complex chemical transformations.

5 Acknowledgements

We gratefully acknowledge the Machine Learning for Pharmaceutical Discovery and Synthesis consortium for funding this work. We also thank Connor Coley for commenting on the manuscript.

References

[1] Michael R Harper, Kevin M Van Geem, Steven P Pyl, Guy B Marin, and William H Green. Comprehensive reaction mechanism for n-butanol pyrolysis and combustion. *Combustion and Flame*, 158(1):16–41, 2011.

[2] Soumya Gudiyella, Zachary J Buras, Te-Chun Chu, Istvan Lengyel, Sreekanth Pannala, and William H Green. Modeling study of high temperature pyrolysis of natural gas. *Industrial & Engineering Chemistry Research*, 57 (22):7404–7420, 2018.

[3] Connie W Gao, Joshua W Allen, William H Green, and Richard H West. Reaction mechanism generator: Automatic construction of chemical kinetic mechanisms. *Computer Physics Communications*, 203:212–225, 2016.

[4] Alexander J Lawson, Jürgen Swienty-Busch, Thibault Géoui, and David Evans. The making of reaxys—towards unobstructed access to relevant chemistry information. In *The Future of the History of Chemical Information*, pages 127–148. ACS Publications, 2014.

[5] John Mayfield, Daniel Lowe, and Roger Sayle. Pistachio: Search and faceting of large reaction databases. In *ABSTRACTS OF PAPERS OF THE AMERICAN CHEMICAL SOCIETY*, volume 254. AMER CHEMICAL SOC 1155 16TH ST, NW, WASHINGTON, DC 20036 USA, 2017.

[6] RN Straker, Q Peng, A Mekareeya, RS Paton, and EA Anderson. Computational ligand design in enantio-and diastereoselective ynamide [5+ 2] cycloisomerization. *Nature communications*, 7(1):1–9, 2016.

[7] Daniel A DiRocco, Yining Ji, Edward C Sherer, Artis Klapars, Mikhail Reibarkh, James Dropinski, Rose Mathew, Peter Maligres, Alan M Hyde, John Limanto, et al. A multifunctional catalyst that stereoselectively assembles prodrugs. *Science*, 356(6336):426–430, 2017.

[8] Manuel Orlandi, F Dean Toste, and Matthew S Sigman. Multidimensional correlations in asymmetric catalysis through parameterization of uncatalyzed transition states. *Angewandte Chemie*, 129(45):14268–14272, 2017.

[9] Yanfei Guan and Steven E Wheeler. Automated quantum mechanical predictions of enantioselectivity in a rhodium-catalyzed asymmetric hydrogenation. *Angewandte Chemie International Edition*, 56(31):9101–9105, 2017.

[10] H Bernhard Schlegel. Optimization of equilibrium geometries and transition structures. *Journal of Computational Chemistry*, 3(2):214–218, 1982.

[11] H Bernhard Schlegel. Estimating the hessian for gradient-type geometry optimizations. *Theoretica chimica acta*, 66(5):333–340, 1984.

[12] Chunyang Peng, Philippe Y Ayala, H Bernhard Schlegel, and Michael J Frisch. Using redundant internal coordinates to optimize equilibrium geometries and transition states. *Journal of Computational Chemistry*, 17(1): 49–56, 1996.

[13] Jingjing Zheng and Michael J Frisch. Efficient geometry minimization and transition structure optimization using interpolated potential energy surfaces and iteratively updated hessians. *Journal of chemical theory and computation*, 13(12):6424–6432, 2017.

[14] Thomas A Halgren and William N Lipscomb. The synchronous-transit method for determining reaction pathways and locating molecular transition states. *Chemical Physics Letters*, 49(2):225–232, 1977.

[15] Chunyang Peng and H Bernhard Schlegel. Combining synchronous transit and quasi-newton methods to find transition states. *Israel Journal of Chemistry*, 33(4):449–454, 1993.

[16] Graeme Henkelman, Blas P Uberuaga, and Hannes Jónsson. A climbing image nudged elastic band method for finding saddle points and minimum energy paths. *The Journal of chemical physics*, 113(22):9901–9904, 2000.

[17] Baron Peters, Andreas Heyden, Alexis T Bell, and Arup Chakraborty. A growing string method for determining transition states: Comparison to the nudged elastic band and string methods. *The Journal of chemical physics*, 120 (17):7877–7886, 2004.

[18] Anthony Goodrow, Alexis T Bell, and Martin Head-Gordon. Development and application of a hybrid method involving interpolation and ab initio calculations for the determination of transition states. *The Journal of chemical physics*, 129(17):174109, 2008.

[19] Paul Zimmerman. Reliable transition state searches integrated with the growing string method. *Journal of chemical theory and computation*, 9(7):3043–3050, 2013.

[20] Andrew Behn, Paul M Zimmerman, Alexis T Bell, and Martin Head-Gordon. Efficient exploration of reaction paths via a freezing string method. *The Journal of chemical physics*, 135(22):224108, 2011.

[21] Shaama Mallikarjun Sharada, Paul M Zimmerman, Alexis T Bell, and Martin Head-Gordon. Automated transition state searches without evaluating the hessian. *Journal of chemical theory and computation*, 8(12):5166–5174, 2012.

[22] H Bernhard Schlegel. Following gradient extremal paths. *Theoretica chimica acta*, 83(1-2):15–20, 1992.

[23] Karl K Irikura and Russell D Johnson. Predicting unexpected chemical reactions by isopotential searching. *The Journal of Physical Chemistry A*, 104(11):2191–2194, 2000.

[24] Alessandro Laio and Michele Parrinello. Escaping free-energy minima. *Proceedings of the National Academy of Sciences*, 99(20):12562–12566, 2002.

[25] Lee-Ping Wang, Alexey Titov, Robert McGibbon, Fang Liu, Vijay S Pande, and Todd J Martínez. Discovering chemistry with an ab initio nanoreactor. *Nature chemistry*, 6(12):1044, 2014.

[26] Manyi Yang, Jingxiang Zou, Guoqiang Wang, and Shuhua Li. Automatic reaction pathway search via combined molecular dynamics and coordinate driving method. *The Journal of Physical Chemistry A*, 121(6):1351–1361, 2017.

[27] Satoshi Maeda, Tetsuya Taketsugu, and Keiji Morokuma. Exploring transition state structures for intramolecular pathways by the artificial force induced reaction method. *Journal of computational chemistry*, 35(2):166–173, 2014.

[28] Paul M Zimmerman. Automated discovery of chemically reasonable elementary reaction steps. *Journal of computational chemistry*, 34(16):1385–1392, 2013.

[29] Paul M Zimmerman. Single-ended transition state finding with the growing string method. *Journal of computational chemistry*, 36(9):601–611, 2015.

[30] Judit Zádor and Habib N Najm. Kinbot 1.0: A code for automatic pes exploration. Technical report, Sandia National Lab.(SNL-CA), Livermore, CA (United States), 2013.

[31] Ruben Van de Vijver and Judit Zádor. Kinbot: Automated stationary point search on potential energy surfaces. *Computer Physics Communications*, 248:106947, 2020.

[32] Pierre L Bhoorasingh and Richard H West. Transition state geometry prediction using molecular group contributions. *Physical Chemistry Chemical Physics*, 17(48):32173–32182, 2015.

[33] Pierre L Bhoorasingh, Belinda L Slakman, Fariba Seyedzadeh Khanshan, Jason Y Cain, and Richard H West. Automated transition state theory calculations for high-throughput kinetics. *The Journal of Physical Chemistry A*, 121(37):6896–6904, 2017.

[34] Donald G Truhlar, Bruce C Garrett, and Stephen J Klippenstein. Current status of transition-state theory. *The Journal of physical chemistry*, 100(31):12771–12800, 1996.

[35] Daniel Sheppard, Rye Terrell, and Graeme Henkelman. Optimization methods for finding minimum energy paths. *The Journal of chemical physics*, 128(13):134106, 2008.

[36] Colin A Grambow, Adeel Jamal, Yi-Pei Li, William H Green, Judit Zador, and Yury V Suleimanov. Unimolecular reaction pathways of a γ-ketohydroperoxide from combined application of automated reaction discovery methods. *Journal of the American Chemical Society*, 140(3):1035–1048, 2018.

[37] Gregor N Simm, Alain C Vaucher, and Markus Reiher. Exploration of reaction pathways and chemical transformation networks. *The Journal of Physical Chemistry A*, 123(2):385–399, 2018.

[38] Jan P Unsleber and Markus Reiher. The exploration of chemical reaction networks. *arXiv preprint arXiv:1906.10223*, 2019.

[39] Marwin HS Segler, Mike Preuss, and Mark P Waller. Planning chemical syntheses with deep neural networks and symbolic ai. *Nature*, 555(7698):604–610, 2018.

[40] Connor W Coley, William H Green, and Klavs F Jensen. Machine learning in computer-aided synthesis planning. *Accounts of chemical research*, 51(5):1281–1289, 2018.

[41] Philippe Schwaller, Riccardo Petraglia, Valerio Zullo, Vishnu H Nair, Rico Andreas Haeuselmann, Riccardo Pisoni, Costas Bekas, Anna Iuliano, and Teodoro Laino. Predicting retrosynthetic pathways using a combined linguistic model and hyper-graph exploration strategy. *arXiv preprint arXiv:1910.08036*, 2019.

[42] Qingyi Yang, Vishnu Sresht, Peter Bolgar, Xinjun Hou, Jacquelyn L Klug-McLeod, Christopher R Butler, et al. Molecular transformer unifies reaction prediction and retrosynthesis across pharma chemical space. *Chemical Communications*, 55(81):12152–12155, 2019.

[43] Benson Chen, Tianxiao Shen, Tommi S Jaakkola, and Regina Barzilay. Learning to make generalizable and diverse predictions for retrosynthesis. *arXiv preprint arXiv:1910.09688*, 2019.

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

[45] John Bradshaw, Matt J Kusner, Brooks Paige, Marwin HS Segler, and José Miguel Hernández-Lobato. Predicting electron paths. *arXiv preprint arXiv:1805.10970*, 2018.

[46] David Fooshee, Aaron Mood, Eugene Gutman, Mohammadamin Tavakoli, Gregor Urban, Frances Liu, Nancy Huynh, David Van Vranken, and Pierre Baldi. Deep learning for chemical reaction prediction. *Molecular Systems Design & Engineering*, 3(3):442–452, 2018.

[47] Philippe Schwaller, Teodoro Laino, Théophile Gaudin, Peter Bolgar, Christopher A Hunter, Costas Bekas, and Alpha A Lee. Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *ACS central science*, 5(9):1572–1583, 2019.

[48] Connor W Coley, Wengong Jin, Luke Rogers, Timothy F Jamison, Tommi S Jaakkola, William H Green, Regina Barzilay, and Klavs F Jensen. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chemical science*, 10(2):370–377, 2019.

[49] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.

[50] Connor W Coley, Regina Barzilay, William H Green, Tommi S Jaakkola, and Klavs F Jensen. Convolutional embedding of attributed molecular graphs for physical property prediction. *Journal of chemical information and modeling*, 57(8):1757–1772, 2017.

[51] KT Schutt, Pan Kessel, Michael Gastegger, KA Nicoli, Alexandre Tkatchenko, and K-R Muller. Schnetpack: A deep learning toolbox for atomistic systems. *Journal of chemical theory and computation*, 15(1):448–455, 2018.

[52] Justin S Smith, Benjamin T Nebgen, Roman Zubatyuk, Nicholas Lubbers, Christian Devereux, Kipton Barros, Sergei Tretiak, Olexandr Isayev, and Adrian E Roitberg. Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning. *Nature communications*, 10(1):1–8, 2019.

[53] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019.

[54] Colin A. Grambow, Lagnajit Pattanaik, and William H. Green. Deep learning of activation energies. *The Journal of Physical Chemistry Letters*, 11(8):2992–2997, 2020. doi: 10.1021/acs.jpclett.0c00500.

[55] Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In *Advances in neural information processing systems*, pages 6410–6421, 2018.

[56] Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander Gaunt. Constrained graph variational autoencoders for molecule design. In *Advances in neural information processing systems*, pages 7795–7804, 2018.

[57] Marwin HS Segler, Thierry Kogej, Christian Tyrchan, and Mark P Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS central science*, 4(1):120–131, 2018.

[58] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

[59] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. *arXiv preprint arXiv:1802.04364*, 2018.

[60] John Bradshaw, Brooks Paige, Matt J Kusner, Marwin Segler, and José Miguel Hernández-Lobato. A model to search for synthesizable molecules. In *Advances in Neural Information Processing Systems*, pages 7935–7947, 2019.

[61] Wenhao Gao and Connor W Coley. The synthesizability of molecules proposed by generative models. *arXiv preprint arXiv:2002.07007*, 2020.

[62] Namrata Anand and Possu Huang. Generative modeling for protein structures. In *Advances in Neural Information Processing Systems*, pages 7494–7505, 2018.

[63] Mohammed AlQuraishi. End-to-end differentiable learning of protein structure. *Cell systems*, 8(4):292–301, 2019.

[64] John Ingraham, Adam Riesselman, Chris Sander, and Debora Marks. Learning protein structure with a differentiable simulator. In *International Conference on Learning Representations*, 2019.

[65] Frank Noé, Gianni De Fabritiis, and Cecilia Clementi. Machine learning for protein folding and dynamics. *Current Opinion in Structural Biology*, 60:77–84, 2020.

[66] Niklas WA Gebauer, Michael Gastegger, and Kristof T Schütt. Generating equilibrium molecules with deep neural networks. *arXiv preprint arXiv:1810.11347*, 2018.

[67] Moritz Hoffmann and Frank Noé. Generating valid Euclidean distance matrices. *arXiv preprint arXiv:1910.03131*, 2019. URL `http://arxiv.org/abs/1910.03131`.

[68] Niklas Gebauer, Michael Gastegger, and Kristof Schütt. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. In *Advances in Neural Information Processing Systems*, pages 7564–7576, 2019.

[69] Tobias Lemke and Christine Peter. Encodermap: Dimensionality reduction and generation of molecule conformations. *Journal of chemical theory and computation*, 15(2):1209–1215, 2019.

[70] Gregor NC Simm and José Miguel Hernández-Lobato. A generative model for molecular distance geometry. *arXiv preprint arXiv:1909.11459*, 2019.

[71] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters*, 98(14):146401, 2007.

[72] Stefan Chmiela, Huziel E Sauceda, Klaus-Robert Müller, and Alexandre Tkatchenko. Towards exact molecular dynamics simulations with machine-learned force fields. *Nature communications*, 9(1):1–10, 2018.

[73] Huziel E Sauceda, Stefan Chmiela, Igor Poltavsky, Klaus-Robert Müller, and Alexandre Tkatchenko. Construction of machine learned force fields with quantum chemical accuracy: Applications and chemical insights. *arXiv preprint arXiv:1909.08565*, 2019.

[74] Nick Gerrits, Khosrow Shakouri, Jörg Behler, and Geert-Jan Kroes. Accurate probabilities for highly activated reaction of polyatomic molecules on surfaces using a high-dimensional neural network potential: Chd3+ cu (111). *The journal of physical chemistry letters*, 10(8):1763–1768, 2019.

[75] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30(8):595–608, 2016.

[76] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[77] Colin A. Grambow, Lagnajit Pattanaik, and William H. Green. Reactants, products, and transition states of elementary chemical reactions based on quantum chemistry. *ChemRxiv preprint DOI 10.26434/chemrxiv.11400240.v2*, 2019.

[78] Kenichi Fukui. The path of chemical reactions-the irc approach. *Accounts of chemical research*, 14(12):363–368, 1981.

[79] Noel M O'Boyle, Michael Banck, Craig A James, Chris Morley, Tim Vandermeersch, and Geoffrey R Hutchison. Open babel: An open chemical toolbox. *Journal of cheminformatics*, 3(1):33, 2011.

[80] Schrödinger, LLC. The PyMOL molecular graphics system, version 1.8. 11 2015.

[81] David Fooshee, Alessio Andronico, and Pierre Baldi. Reactionmap: An efficient atom-mapping algorithm for chemical reactions. *Journal of chemical information and modeling*, 53(11):2812–2819, 2013.

[82] Nuno Osório, Paulo Vilaça, and Miguel Rocha. A critical evaluation of automatic atom mapping algorithms and tools. In *International Conference on Practical Applications of Computational Biology & Bioinformatics*, pages 257–264. Springer, 2017.

[83] Wojciech Jaworski, Sara Szymkuć, Barbara Mikulak-Klucznik, Krzysztof Piecuch, Tomasz Klucznik, Michał Kaźmierowski, Jan Rydzewski, Anna Gambin, and Bartosz A Grzybowski. Automatic mapping of atoms across both simple and complex chemical reactions. *Nature communications*, 10(1):1–11, 2019.

[84] G. Landrum. RDKit: Open-Source Cheminformatics. `http://rdkit.org`, 2006.

[85] Gordon M Crippen, Timothy F Havel, et al. *Distance geometry and molecular conformation*, volume 74. Research Studies Press Taunton, 1988.

[86] MJ Frisch, GW Trucks, HB Schlegel, GE Scuseria, MA Robb, JR Cheeseman, G Scalmani, V Barone, GA Petersson, HJRA Nakatsuji, et al. Gaussian 16, 2016.

A Methodology

Overview Our model predicts the 3D coordinates of transition states in an end-to-end manner by combining a graph neural network with a differentiable multidimensional scaling procedure. Specifically, the model takes as input graph features derived from the reactant and product structures, processes these with a graph neural network to predict a target distance matrix with associated i, j weights, and then recovers 3D coordinates consistent with these predictions via an unrolled multidimensional scaling procedure. The model is trained end-to-end via gradient descent with gradients flowing through the coordinate recovery procedure into the neural network.

Graph features To train and test the model, we first transform individual log files from ref. 77 into structure-data files (SDFs) for the reactant, product, and TS. Our workflow uses reactant and product data (TS data is only used to calculate loss during training) to create an input attributed graph $G = (V, E)$ with vertices (atoms) V, edges (bonds) E, and corresponding initial features $\{x_i \in \mathbb{R}^{f_v} | i \in V\}$ for atoms and $\{y_{ij} \in \mathbb{R}^{f_e} | ij \in E\}$ for bonds. Note that since both the reactant and product contain the same number of atoms, it is straightforward to build G as an averaged representation for the TS. We convert all three SDFs to molecule objects with RDKit [84] and build the featurizations. Importantly, the edge features include the averaged distance between atoms i and j between the reactant and product along with whether or not a bond is broken or formed and if the bond is aromatic. The atom features only include the identity of the atom encoded numerically via atomic number and with a one-hot vector.

Graph neural network The neural network architecture is a graph neural network (GNN) with both node and edge updates [76], in which all atoms are considered neighbors. A GNN operates by iteratively passing information between neighbors, creating an updated representation of the input graph. Our network operates on a single input graph, which represents averaged features from the reactant and product graphs. Outputs from the GNN pass through several dense layers to generate the distance and weight predictions, which finally travel to the nonlinear least squares optimization to create the TS structure.

Now, we show the graph update procedure. While some layers include bias parameters, we omit them for clarity. Additionally, some transformations involve several layers with multiple activations, but we only show the final layer for clarity. We first obtain an initial hidden representation for both the edge and vertex features:

$$y^0_{ij,h} = \tau(W^0_e y_{ij}) \tag{2}$$

$$x^0_{i,h} = \tau(W^0_v x_i) \tag{3}$$

where $\tau(\cdot)$ is the ReLU activation function and $W^0_e \in \mathbb{R}^{h \times f_e}$ and $W^0_v \in \mathbb{R}^{h \times f_v}$ are learned matrices. We determine an optimal hidden size of $h = 256$. Each GNN iteration consists of an edge update then a node update. First, we pair the node and edge features of the current time step together to create the initial feature representation:

$$f^t_{ij} = \tau(W^t_e y^t_{ij,h} + W^t_{v1} x^t_{i,h} + W^t_{v2} x^t_{j,h}) \tag{4}$$

where W^t_e, W^t_{v1}, $W^t_{v2} \in \mathbb{R}^{h \times h}$ are all learned matrices, different for each iteration. This operation occurs for all combinations of atoms, i.e., all atoms' representations update with information from all other atoms; this is why we refer to the network as fully-connected. We next generate the updated bond features:

$$y^{t+1}_{ij,h} = y^t_{ij,h} + \tau(W^t_f f^t_{ij}) \tag{5}$$

with learned $W^t_f \in \mathbb{R}^{h \times h}$. From the updated bond features, we generate the updated atom features:

$$x^{t+1}_{i,h} = x^t_{i,h} + \tau\left(W^t_{f3} \sum_{k \in V} \tau(W^t_{f2} y^{t+1}_{ik,h})\right) \tag{6}$$

where W^t_{f2} and $W^t_{f3} \in \mathbb{R}^{h \times h}$ are additional learned matrices. Equations 4-6 represent the update steps, which repeat for $t \in \{1, \ldots, T\}$, and we optimally determine $T = 3$.

Distance prediction with weights To prepare for the prediction of the distance matrix and the least squares weight matrix, the final edge output of the GNN passes through an additional series of dense layers:

$$p_{ij} = W_{p2}\tau(W_{p1}y_{ij,h}^T) \tag{7}$$

where $W_{p1} \in \mathbb{R}^{h\times h}$ and $W_{p2} \in \mathbb{R}^{2\times h}$ represent additional learned matrices. We next symmetrize the predictions and enforce positivity:

$$\tilde{p}_{ij} = \sigma(p_{ij} + p_{ji}) \tag{8}$$

where $\sigma(\cdot)$ is the softplus activation function. We additionally set the diagonal entries to zero. $\tilde{p}_{ij}$ is a vector consisting of two values. The first is the prediction of the distance between atoms i and j, $D_{init_{ij}}$, while the second is the value of the weight matrix for the least squares optimization, W_{ij}.

Coordinate recovery To reconstruct coordinates from the predicted distance and weight matrices, we use a nonlinear least squares optimization defined by the following:

$$\underset{X}{\operatorname{argmin}} \sum_{ij} W_{ij}(D_{init_{ij}} - ||X_i - X_j||)^2 \tag{9}$$

where $X \in \mathbb{R}^{N\times 3}$ is the array of the x, y, and z positions of each of the N atoms in the TS structure.

Coordinate initialization We follow the algorithm described in ref. 85 to construct an initialization X_{init} for this nonlinear optimization with the the Gram matrix B:

$$B_{ij} = -\frac{1}{2}\left(D_{init_{ij}} - \frac{1}{N}\sum_{k\in V} D_{init_{ik}} - \frac{1}{N}\sum_{k\in V} D_{init_{kj}} + \frac{1}{N^2}\sum_{km\in E} D_{init_{km}}\right) \tag{10}$$

Because B is real and symmetric, we can write the following decomposition:

$$B = X^T X = Q\Lambda Q^T = \sum_{i}^{N} u_i u_i^T \lambda_i \tag{11}$$

We use a matrix deflation technique to identify the the eigenvectors corresponding to the three largest eigenvalues, which, after scaling, we concatenate to generate an Nx3 array of Cartesian coordinates X_{init}.

Coordinate optimization After obtaining an initialization, we solve Equation 9 via gradient descent to obtain X. In this optimization, we initialize the values for X with X_{init}.

Loss function We calculate network loss by first calculating D, the set of pairwise distances derived from X, the final TS geometry. We compare this value to the set of pairwise distances generated from the ground truth TS structure reported in ref. 77, which we denote as D_{GT}:

$$loss = \frac{1}{N^2}\sum_{ij}(D_{ij} - D_{GT_{ij}})^2 \tag{12}$$

Training We train the model for a fixed number of epochs or until validation loss no longer decreases. We use the Adam optimizer with default parameters.

B Quantum Chemistry Methods

We performed all optimization and IRC calculations with Gaussian 16 [86] using the ωB97X-D level of theory with a def2-TZVP basis set. Since ref. 77 used a different software package to optimize structures, we re-optimized these structures with Gaussian 16 for a fair ground truth comparison.

C Reaction Complexity

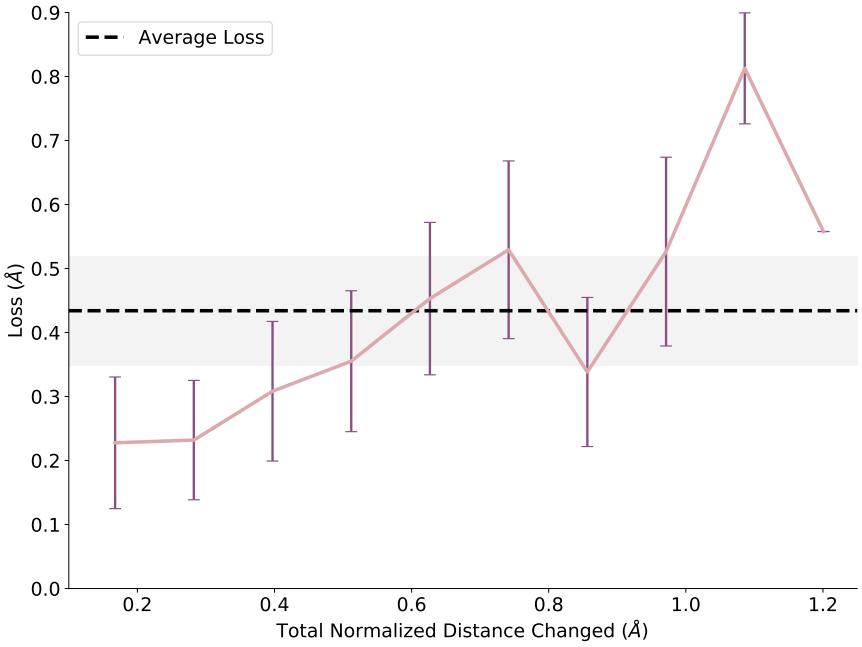

Figure 4: Distance changed between reactant and product normalized by number of atoms vs. network loss. More complex reactions as described by greater distance changed show higher loss values. Each full bar corresponds to a single standard deviation.