Machine Learning Quantum Reaction Rate Constants

Evan Komp¹ and Stéphanie Valleau,¹

¹Department of Chemical Engineering, University of Washington, Seattle, Washington 98195, United States

ABSTRACT

The ab initio calculation of exact quantum reaction rate constants come at a high cost due to the required dynamics of reactants on multidimensional potential energy surfaces. In turn, this impedes the rapid design of the kinetics for large sets of coupled reactions. In an effort to overcome this overbearing set-back, a deep neural network (DNN) was trained to predict the logarithm of quantum reaction rate constants multiplied by their reactant partition function – rate products. The training dataset was generated in-house and contains ~1.3 million quantum reaction rate constants for single, double, symmetric and asymmetric one-dimensional potentials computed over a broad range of reactant masses and temperatures. The DNN was able to predict the logarithm of the rate product with a relative error of 1.1%. Further, when comparing the difference between the DNN prediction and classical transition state theory at temperatures below 300K a relative percent error of 38% was found with respect to the exact difference. Systems beyond the test set were also studied, these included the H + H₂ reaction and the diffusion of hydrogen on Ni(100). For these reactions the DNN predictions were accurate at high temperatures and in qualitative agreement with the exact rates at lower temperatures. This work shows that one can take advantage of a DNN to gain insight on reactivity in the quantum regime.

INTRODUCTION

Understanding, evaluating and predicting chemical reactivity computationally is of great interest as it is critical for the design of new materials,¹² to tune the production of compounds in industry,³ for drug design⁴ and so forth. For the last century, theories and their application to the calculation of reaction rate constants have been developed and explored extensively.⁵–⁷ Classical, semi-classical and quantum mechanical approaches all require some knowledge of the potential energy surface, ranging from an activation energy, to a minimum energy path, to a full potential energy surface. Classical static theories such as transition state theory (TST) or variational transition state theory⁸ are frequently employed due to their low computational cost. However, they come with the need to find a minimum energy path ab-initio and are limited by being valid mostly at high temperatures where quantum effects such as tunneling are less significant. At lower temperatures⁹ quantum corrections¹⁰ or approaches such as the Quantum Instanton,¹¹,¹² the flux-flux or flux-side correlation functions take tunneling into account.¹³ Unfortunately, fully quantum approaches cannot be applied to systems with many degrees of freedom due to the exponential growth in computational cost with system size.

To overcome these bottlenecks, in the last decade, there has been much interest in using AI machine learning types of approaches to predict chemical properties¹⁴–²¹ and more recently reactivity.²²–²⁸ The prediction of reaction rate constants using supervised machine learning comes with the requirement of large datasets of reaction rate constants. While some datasets exist for gas phase reactions, e.g. Reaxys²⁹ or Scifinder³⁰, the data is largely inhomogeneous and often does not include reaction rate
constants. Nonetheless, these datasets have been used to train supervised machine learning algorithms to predict e.g. suitable temperatures for reactions. Recent work has also shown promise for the AI prediction of activation energies.

Bowman et al. applied gaussian process regression to a small data set of single barrier reactions and were able to obtain good estimates of the reaction rate constants computed using multidimensional time-dependent hartree (MCTDH), the Quantum Instanton or ring polymer molecular dynamics (RPMD).

In this work we trained and validated a supervised multi-layer perceptron (MLP) deep neural network (DNN) to predict the product of the exact quantum reaction rate constant with the reactant partition function for one-dimensional barriers. The motivation for choosing one-dimensional paths was twofold; on one hand many theoretical methods approximate the overall kinetics with a single most traveled one-dimensional reactive pathway, on the other, given our need for a large dataset, the calculation of rate constants on multidimensional surfaces was unfeasible. Single, double, symmetric and asymmetric barriers were considered. The presence of resonant tunneling for double barrier potentials increases the difficulty of computing the quantum reaction rate constant accurately due to the need to resolve resonant peaks. In this context, a machine learning alternative is more appealing. As shown in Figure 1, we first generated a database of over six million reaction rates by numerical integration of the transmission probability for a broad range of reactant masses, barrier heights, barrier widths, barrier distances, barrier shapes and temperatures. This dataset was then employed to find the optimal hyperparameters for a DNN via grid search, to train the DNN, and ultimately, to predict quantum reaction rate constants.

**COMPUTATIONAL METHODS**

**Database Generation**

A database of ~6.9 million quantum reaction rate constant products, \( k(T)Q_{\text{react}}(T) \), was generated for one-dimensional potentials. The quantum reaction rate constant products were computed in the energy domain from the general equation (1)

\[
k_Q(T) \equiv k(T)Q_{\text{react}}(T) = \frac{1}{2\pi \hbar} \int_{-\infty}^{\infty} e^{-\beta E} T(E) dE
\]
where \( T(E) \) is the transmission coefficient and \( \beta = 1/k_B T \).

The transmission coefficients were evaluated following the approach developed in Ref. \(^9\) where it was shown that given a coordinate position, \( x^{(1)} \), located far from the reaction barrier, one can write

\[
T(E) = 4 \left| \psi_E(x^{(1)}) + \frac{\hbar}{ip} \partial_x \psi_E(x^{(1)}) \right|^2.
\]  

(2)

In equation (2) the coordinate position, \( x^{(1)} \), must satisfy the condition \( x^{(1)} \gg L \), where \( L \) defines the range \( x \in [-L, +L] \) within which the potential is non-zero. \( \psi_E(x) \) is the eigenfunction of the Hamiltonian for the energy eigenvalue \( E = p^2/2m \) where \( p \) is the linear momentum and \( m \) the reactant mass. This approach was successful in reproducing the full quantum reaction rate constant as could be obtained in the time domain using the flux-flux correlation function approach,\(^{31}\) with the advantage of avoiding the demanding time propagation of the wavefunction. The reactant partition functions where considered as part of the output to be predicted using machine learning. For simplicity, in this paper we will refer to the output product \( k_Q(T) \equiv k(T) Q_{\text{react}}(T) \) as the reaction rate constant product. Equation (1) above was solved for single and double symmetric and asymmetric rectangular, Eckart,\(^{32}\) gaussian and Peskin\(^{33}\) potentials as shown in panel 1) of Figure 1 and defined below in Table 1.

Table 1: Range of potential energy parameters for the symmetric and asymmetric single and double barriers used to compute reaction rate constants for the dataset. For a symmetric barrier, \( \alpha = 1 \).

| Feature             | Values                        |
|---------------------|-------------------------------|
| Reactant mass \(- m\) | 1060 – 1060000 [au]           |
| Barrier heights \(- V_1, V_2\) | 5.0 – 80.7 [kcal / mol]     |
| Barrier widths \(- w_1, w_2\)  | 0.1 – 6 [au]                  |
| Distance \(- d\)          | 0.0 – 5 [au]                  |
| Asymmetry \(- \alpha\)     | 0.0 – 1.0                     |
| Temperature \(- T\)         | 80 – 2500 [K]                 |

For single rectangular and Eckart potentials exact analytical expressions were used for the transmission coefficient.\(^{7,32,34}\) More details on the numerical evaluation of the transmission coefficient can be found in the supporting information.

**Input features and datasets**

A dataset of \(~1.3\) million points was extracted from the original database to train a deep neural network to predict the reaction rate constant products (eq. 1). To obtain the new dataset, some small mass values were randomly dropped so as to avoid having an unbalanced dataset. Plots of the dataset distributions with respect to input features are shown in the supporting information. Of the new dataset, 80% was used to train and 20% to test the DNN.

The input features (Table 1) included reactant mass, reaction temperature, first and second barrier height, first and second barrier widths, distance between barriers, barrier slope and asymmetry. A second set of features was also evaluated, it added the sum of barrier heights, the sum of barrier widths and the number of barriers to the first set. These additional features were not found to provide any advantage to the model’s loss during training and are not considered in the results.

The input features of both the training and testing datasets were normalized respect to the mean and standard deviation of the training dataset. The output label was chosen to be the logarithm of the rate constant product, \( \log k_Q(T) \), to avoid the
problem of values ranging over hundreds of orders of magnitude. The output label was then scaled to the range $[-1, 1]$ by using the training dataset label minimum and maximum values.

Hyperparameters optimization

The network’s optimal architecture and modeling hyperparameters were identified by carrying out grid searches with five-fold cross validation using Scikit-learn’s GridSearchCV\textsuperscript{35} and Keras with TensorFlow backend.\textsuperscript{36} The first search was conducted over one to three hidden layers of $n \in [1, 6, 12, 24, 32, 64, 128]$ neurons per layer with batch size 32, 64 or 128 and epochs in the set $[50, 100, 200, 300]$. For this grid search, we chose the Adam optimizer\textsuperscript{37} with learning rate set to $0.001, \beta_1 = 0.9$ and $\beta_2 = 0.999$, and the sigmoid and tanh activation functions for the hidden and output layer (See Table S2 for details on the first grid search). Subsequently a second and third grid search were carried out on the best model of the first search for each feature set, to identify the optimal activation functions and learning rate (Table S3-5). Finally, a fourth grid search was carried out on the optimal model obtained from the third grid search to choose the final batch size. In all searches, the ranking metric was the mean squared error (MSE). The optimal hyperparameters values are discussed in the next section and shown in Table S6 of the supporting information.

RESULTS AND DISCUSSION

Optimal DNN models

From our grid searches, as shown in Table 2, we found that the optimal network architecture consisted of three hidden layers with 128 neurons in the first layer, 24 in the second and 12 in the third layer.

| Hyperparameter                  | Value            |
|---------------------------------|------------------|
| neuron configuration            | $(128, 24, 12)$  |
| epochs                          | 300              |
| batch size                      | 256              |
| hidden layer activation function| sigmoid          |
| output activation function      | linear           |
| learning rate                   | 0.001            |

The grid search on the activation function indicated the sigmoid function as best for the hidden layers. For the output layer, the top three functions were the softsign, linear and hyperbolic tangent functions. These three combinations had comparable metrics of loss and we chose the linear function as the output function. Finally, the optimal learning rate was found to be 0.001. The best batch size was determined in the fourth grid search by comparing the moving average validation loss of the optimal model and found to be 256. Plots and details of the grid search results are included in the supporting information (Figures S3-5). We would like to note that a grid search for regularization was also carried out but it did not lead to an improvement on our model as overfitting was not observed.
In Figure 2 we show the testing dataset loss (blue solid line) and the training dataset loss (black dashed line) as a function of epochs. The testing loss closely follows the training loss and decreases as training progresses. We do not observe signs of overfitting from the trend of the average testing loss. The fluctuations in the test loss are associated with the batch size in mini batch training. The final value of the training loss expressed in terms of the scaled log ($k_Q$) is equal to $1.55 \cdot 10^{-5}$.

In Figure 3 we show the unscaled predicted DNN values of the log ($k_Q$) respect to the exact test set values (blue circles). The red dashed line indicates the identity, i.e. the case where a model would predict perfectly. The linear fit (black line) to the predicted data is very close to the identity with a mean absolute error (MAE) of 0.27 log (1/ps) and standard deviation of 0.86 log(1/ps). This corresponds to a relative error of 1.1% respect to the mean log ($k_Q$) value. The null hypothesis of always predicting the mean of log ($k_Q$) produces a MAE of 24.5 log(1/ps) with standard deviation 34.7 log(1/ps). The small value of the MAE compared to the null indicates that the model is able to predict accurately within the range of values it was trained on.

Figure 2: Final optimized DNN model training loss (MSE) (black dashed line) and testing loss (blue solid line) for the scaled logarithm of the reaction rate product log ($k_Q$) as a function of epochs.

Figure 3: Predictions of the optimal DNN model on the test set. Blue circles represent the predicted value of log ($k_Q$) compared to the true value in the test set. A linear fit to the data (black line) is very similar to the identity function (red line). The percent MAE rescaled respect to the mean is 1.1%. The standard deviation from the mean is 0.86 log(1/ps).
Quantum rate constant predictions

We investigated the trained DNN’s ability to predict quantum effects by comparing predicted \( \log(k_Q) \) values to transition state theory rate constant results for the entire test set.

In Figure 4 panel a) the average difference between the DNN predicted values and the corresponding transition state rate constant, binned by temperature interval, is shown in grey bars with diagonal lines. Specifically, the \( j \)-th bin’s height \( h_j \) is computed as

\[
h_j = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \left( \log(k_{Q_i}) - \log(k_{Q_i}^{\text{TST}}) \right) / |\log(k_{Q_i})| \quad (3)
\]

The normalization by the absolute value of the quantum \( \log(k_Q) \) is carried out so as to even out changes in rate value due to reactant mass and barrier shape and emphasize the difference respect to the classical rate constant. The same average difference is shown in blue for the test dataset. The DNN values closely follow the test set and for temperatures below 300K, the percent MAE rescaled respect to the exact mean difference is 38%.

\[\text{Figure 4: Panel a) average prediction of quantum effects (equation (3)) as a function of temperature for the test set (light blue) and for the neural network predictions (grey with diagonal lines). As temperature increases the NN correctly predicts a decrease in quantum effects. Panel b) Top plots show, on the left hand side, a double barrier potential taken from the test set with } V_1 = V_2 = 42.89 \text{ kcal/mol, } w_1 = w_2 = 0.5 \text{ au and } d = 0.5 \text{ au and, on the right hand side, the transmission coefficient for reactants of mass } 11.63 \text{ g/mol. The lower plot reports the corresponding predicted } \log(k_Q) \text{ as a function of the inverse temperature (red dashed line), the exact values (blue line) and a transition state theory estimate based on the first barrier height (black line).}
\]

In panel b) of Figure 4 we show an example of the predicted \( \log(k_Q) \) for a symmetric double gaussian barrier taken from the test set. The barrier heights were \( V_1 = V_2 = 42.89 \) kcal/mol, the widths \( w_1 = w_2 = 0.5 \) au and the distance between barriers was \( d = 0.5 \) au. The reactants mass was 11.63 g/mol which is close to the mass of a carbon atom. As can be seen from the transmission coefficient, panel b) top right-hand side plot, the peaks indicate the presence of quasi bound states which lead to resonant tunneling and to a larger rate constant.\textsuperscript{33,38} The presence of quasi bound states increases the duration of the tunneling process and contemporarily the length of time-dependent calculations required to compute the rate constant ab initio. In this context, it is promising to see that the predicted DNN \( \log(k_Q) \) values (red) closely follow the exact quantum values (blue) and only start underestimating at temperatures below \( \sim 160 \text{ K.} \)
Predicting the H+H₂ reaction and hydrogen diffusion on Ni(100) rate constants

To evaluate the ability of the DNN model to predict kinetics beyond the test dataset, two additional reactions were considered. The first is the extensively studied H + H₂ reaction.³⁹ We employed the one-dimensional Eckart barrier approximation to the potential energy surface¹¹ with \( V_1 = 9.8 \text{ kcal/mol} \), \( w_1 = 2.31 \text{ au} \) and \( m = 1060 \text{ au} \) to solve for the exact kinetics. The potential is shown in Figure 5 panel a) upper plot. In the lower plot, we show the exact quantum results (blue with circles), the DNN predictions (red with triangles) and transition state theory (black line). The DNN predictions follow the trend of the exact results and recovers most of the quantum effects.

The second process we studied was hydrogen diffusion on Ni(100). For this system, we extracted a one-dimensional pathway from the EDIM potential energy surface⁴⁰ with EAM-5 parametrization developed by Truong and Truhlar.⁴¹ The potential energy path was determined by first minimizing the potential energy along the \( z \) coordinate and then by taking a path symmetrically crossing three hollow sites. The one-dimensional path was then fitted to a double gaussian potential to extract features for the DNN. A plot of the two-dimensional surface and of the one-dimensional double gaussian path is shown in Figure 5, panel b) upper section. With this potential we computed the exact logarithm of the reaction rate constant product (Figure 5, panel b) lower plot – blue), the transition state theory estimate for the rate constant (black) and the DNN predicted rate constant (red).

The DNN predictions underestimate the rate constant at low temperatures although they are on the same order of magnitude. This shows that with our simple input feature representation, one can obtain qualitative information on reactivity for systems beyond the test set. To predict quantum rates more accurately and obtain quantitative accuracy for a broader set of reactions, we believe that providing information such as the reactant partition function and input features which include more information on the chemical properties of the reactants could improve the trained DNN’s predictive ability.
CONCLUSIONS

In this work we have shown that a DNN can be trained to successfully predict quantum reaction rate constant for one-dimensional reactive pathways. We generated a database of over 6.9 million quantum reaction rate constant products for smooth and rectangular, single and double, symmetric and asymmetric barriers over a broad range of temperatures and reactant masses. The database was employed to identify the optimal DNN architecture via grid search and finally to train a predictive model. The results showed a low error on predicting the test set rate constants with a relative MAE on the logarithm of $k_D$ of 1.1%. Furthermore, at temperatures lower than 300K the trained DNN was able to predict quantum effects such as tunneling with a relative error of 38%. Beyond the testing dataset we also looked at new cases such as the H+H$_2$ reaction and the diffusion of H on Ni. We found that the predicted values were on the same order of magnitude as the exact results. We believe that these predictions could improve when including input features related to specific molecular reactants and their chemical properties, as well as the reactant partition function. While approximating the kinetics with a single one-dimensional path simplifies the multidimensional dynamics, its quantum mechanical solution remains challenging. In this context our work’s trained DNN model provides a quicker alternative for the prediction of quantum rate constants and helps open the road for the long-term prospect of modeling networks of coupled reactions.

SUPPORTING INFORMATION

The supporting information includes information on the dataset generation, the dataset features, its distribution and cross correlations. It also contains details on the grid search spaces, visualizations of the grid search optimization, and final model hyperparameters.

ACKNOWLEDGMENTS

The authors would like the acknowledge the Hyak supercomputer system at the University of Washington for support of this research.

REFERENCES

1. H. Larsen, J. & Chorkendorff, I. From fundamental studies of reactivity on single crystals to the design of catalysts. Surf. Sci. Rep. 35, 163–222 (1999).
2. Schmidbauer, S., Hohenleutner, A. & König, B. Chemical degradation in organic light-emitting devices: Mechanisms and implications for the design of new materials. Adv. Mater. 25, 2114–2129 (2013).
3. Farrauto, C. H. & Bartholomew, R. J. Fundamentals of Industrial Catalytic Processes. (John Wiley & Sons, Ltd., 2005).
4. Swinney, D. C. Applications of binding kinetics to drug discovery: Translation of binding mechanisms to clinically differentiated therapeutic responses. Int. J. Pharm. Med. 22, 23–34 (2008).
5. Steinfeld, J., Francisco, J. & Hase, W. Chemical Kinetics and Dynamics. (Prentice Hall Englewood Cliffs (New Jersey), 1989).
6. Hänggi, P., Talkner, P. & Borkovec, M. Reaction-rate theory: Fifty years after Kramers. Rev. Mod. Phys. 62, 251–341 (1990).
7. Peters, B. Reaction Rate Theory and Rare Events. (Elsevier Science, 2017).
8. Truhlar, D. G. & Garrett, B. C. Variational Transition State Theory. Ann. Rev. Phys. Chem. 35, 159–189 (1984).
9. Mandrà, S., Valleau, S. & Ceotto, M. Deep nuclear resonant tunneling thermal rate constant calculations. Int. J. Quantum Chem. 113, 1722–1734 (2013).
10. Wigner, E. On the quantum correction for thermodynamic equilibrium. Phys. Rev. 40, 749–759 (1932).
11. Miller, W. H., Zhao, Y., Ceotto, M. & Yang, S. Quantum instanton approximation for thermal rate constants of chemical reactions. *J. Chem. Phys.* **119**, 1329–1342 (2003).
12. Yamamoto, T. & Miller, W. H. On the efficient path integral evaluation of thermal rate constants within the quantum instanton approximation. *J. Chem. Phys.* **120**, 3086–3099 (2004).
13. Miller, W. H., Schwartz, S. D. & Tromp, J. W. Quantum mechanical rate constants for bimolecular reactions. *J. Chem. Phys.* **79**, 4889–4898 (1983).
14. Rupp, M., Tkatchenko, A., Müller, K. R. & Von Lilienfeld, O. A. Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.* **108**, 1–5 (2012).
15. Montavon, G. *et al.* Machine learning of molecular electronic properties in chemical compound space. *New J. Phys.* **15**, (2013).
16. Häse, F., Valleau, S., Pyzer-Knapp, E. & Aspuru-Guzik, A. Machine learning exciton dynamics. *Chem. Sci.* **7**, 5139–5147 (2016).
17. Afzal, M. A. F., Haghhighatfari, M., Ganesh, S. P., Cheng, C. & Hachmann, J. Accelerated Discovery of High-Refractive-Index Polyimides via First-Principles Molecular Modeling, Virtual High-Throughput Screening, and Data Mining. *J. Phys. Chem. C* **123**, 14610–14618 (2019).
18. Schütz, K. T., Arabzadah, F., Chmiela, S., Müller, K. R. & Tkatchenko, A. Quantum-chemical insights from deep tensor neural networks. *Nat. Commun.* **8**, 6–13 (2017).
19. Winter, R., Montanari, F., Noé, F. & Clevert, D. A. Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chem. Sci.* **10**, 1692–1701 (2019).
20. Gómez-Bombarelli, R. *et al.* Design of efficient molecular organic light-emitting diodes by a high-throughput virtual screening and experimental approach. *Mater. Nat.* **15**, 1120–1127 (2016).
21. Afzal, M. A. F., Cheng, C. & Hachmann, J. Combining first-principles and data modeling for the accurate prediction of the refractive index of organic polymers. *J. Chem. Phys.* **148**, (2018).
22. Amabilino, S. *et al.* Training Neural Nets to Learn Reactive Potential Energy Surfaces Using Interactive Quantum Chemistry in Virtual Reality. *J. Phys. Chem. A* **123**, 4486–4499 (2019).
23. Gao, H. *et al.* Using Machine Learning to Predict Suitable Conditions for Organic Reactions. *ACS Cent. Sci.* **4**, 1465–1476 (2018).
24. Singh, A. R., Rohr, B. A., Gauthier, J. A. & Nørskov, J. K. Predicting Chemical Reaction Barriers with a Machine Learning Model. *Catal. Letters* **149**, 2347–2354 (2019).
25. Ulissi, Z. W., Medford, A. J., Bligaard, T. & Nørskov, J. K. To address surface reaction network complexity using scaling relations machine learning and DFT calculations. *Nat. Commun.* **8**, (2017).
26. Houston, P. L., Nandi, A. & Bowman, J. M. A machine learning approach for prediction of rate constants. *J. Phys. Chem. Lett.* **10**, 5250–5258 (2019).
27. Grambow, C. A., Pattanaik, L. & Green, W. H. Deep Learning of Activation Energies. *J. Phys. Chem. Lett.* **11**, 2992–2997 (2020).
28. Bhoorasingh, P. L., Slakman, B. L., Seyedzadeh Khanshan, F., Cain, J. Y. & West, R. H. Automated Transition State Theory Calculations for High-Throughput Kinetics. *J. Phys. Chem. A* **121**, 6896–6904 (2017).
29. Lawson, A. J., Swienty, A., Geou, T. & Evans, D. The making of reaxys - Towards unobstructed access to relevant chemistry information. in *ACS Symposium Series* (2014). doi:10.1021/bk-2014-1164.ch008
30. SciFinder Scholar. *Choice Rev. Online* (2007). doi:10.5860/choice.44-6577
31. Tromp, J. W. & Miller, W. H. The reactive flux correlation function for collinear reactions H + H2, Cl + HCl and F + H2. *Faraday Discuss. Chem. Soc.* **84**, 441–453 (1987).
32. Eckart, C. The penetration of a potential barrier by electrons. *Phys. Rev.* **35**, 1303–1309 (1930).
33. Berman, L. & Peskin, U. Resonant tunneling probabilities for an N-terminal junction by the flux averaging method. in *International Journal of Quantum Chemistry* (2004). doi:10.1002/qua.20024
34. Ahmed, Z. Tunneling through a one-dimensional potential barrier. *Phys. Rev. A* **47**, 4761–4767 (1993).
35. Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* (2011).
36. Chollet, F. and others. Keras. *GitHub*.
37. Kingma, D. P. & Ba, J. L. Adam: A method for stochastic optimization. *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.* 1–15 (2015).
38. Yamamoto, H. Resonant tunneling condition and transmission coefficient in a symmetrical one-dimensional rectangular double-barrier system. *Appl. Phys. A Solids Surfaces* **42**, 245–248 (1987).
39. Truhlar, D. G. & Wyatt, R. E. History of H3 Kinetics. *Annu. Rev. Phys. Chem.* **27**, 1–43 (1976).
40. Truong, T. N., Truhlar, D. G. & Garrett, B. C. Embedded diatomics-in-molecules: A method to include delocalized electronic interactions in the treatment of covalent chemical reactions at metal surfaces. *J. Phys. Chem.* **93**, 8227–8239 (1989).
41. Wonchoha, S. E., Hu, W. P. & Truhlar, D. G. Surface diffusion of H on Ni(100): Interpretation of the transition temperature. *Phys. Rev. B* **51**, 9985–10002 (1995).
Supporting Information for
“Machine Learning Quantum Reaction Rate Constants”

Evan Komp¹ and Stéphanie Valleau,¹

¹Department of Chemical Engineering, University of Washington, Seattle, Washington 98115, United States

Table of Contents

I. DATABASE GENERATION .............................................................................................................................. 2

II. DATASET ...................................................................................................................................................... 4

III. GRID SEARCH & HYPERPARAMETER TUNING ......................................................................................... 7
   A. First grid search: neurons, hidden layers, batch size and epochs ................................................................. 7
   B. Second grid search: activation function ....................................................................................................... 8
   C. Third grid search: learning rate ................................................................................................................. 9
   D. Fourth grid search: batch size for optimal model ..................................................................................... 10
   E. Optimal hyperparameters ....................................................................................................................... 11
I. DATABASE GENERATION

The equations for the potentials employed to generate the database are defined below in Table S 1. For the single barriers the exact expressions for the transmission coefficient \( T(E) \) were used.\(^1\)\(^2\)

For the double barriers, the transmission coefficient was evaluated as in equation (2) of the main text\(^3\), with \( x \in [-L, L] \) and \( L = 20[\text{au}] \). The evaluation of the transmission coefficient was carried out using scipy’s \texttt{integrate.solve_ivp} function to solve the initial value problem with the implicit Runge-Kutta method of order 5 as defined by the ‘Radau’ option. The range of energy values over which the transmission coefficient was evaluated was a set of 300 points in the interval \( E_{\text{values}} = \frac{E_{\text{max}}}{1000}; 2E_{\text{max}} \), with \( E_{\text{max}} = 2 \cdot \max[V(x)] \). The maximum value of the potential energy was evaluated over 1000 points in the range \(-L < x < L\).

Table S 1: Equations for the one-dimensional potential energy barriers used to build the database. The value of \( \Delta x \) was 0.1 au for symmetric barriers and 0.2 au for asymmetric barriers.

| Barrier               | Potential energy                                                                 | Parameters                      |
|-----------------------|----------------------------------------------------------------------------------|---------------------------------|
| Single rectangular    | \( V_R(x) = \begin{cases} V_1 & 0.0 < x < w_1 \\ 0.0 & x < 0 \lor x > w_1 \end{cases} \) | \( V_1 \) – barrier height      |
|                       |                                                                                  | \( w_1 \) – barrier width       |
|                       |                                                                                  | \( \alpha = 1.0 \)              |
|                       |                                                                                  | \( \delta = 0.0 \)              |
| Single symmetric Eckart | \( V_{SE}(x) = \frac{V_1}{\cosh^2(\pi x / w_1)} \)                          | \( V_1 \) – barrier height      |
|                       |                                                                                  | \( w_1 \) – barrier width       |
|                       |                                                                                  | \( \alpha = 1.0 \)              |
|                       |                                                                                  | \( \delta = \frac{V(x^*) - V(x^* + \Delta x)}{\Delta x} \) |
|                       |                                                                                  | \( x^* \mid V(x^*) = \max(V(x)) \) |
| Single asymmetric Eckart | \( V_{AE}(x) = \frac{V_1(1 - \alpha)}{1 + e^{-2w_1x}} + \frac{V_1(1 + \sqrt{\alpha})^2}{4 \cosh^2 w_1 x} \) | \( V_1 \) – barrier height      |
|                       |                                                                                  | \( w_1 \) – barrier width       |
|                       |                                                                                  | \( \alpha \) – asymmetry        |
|                       |                                                                                  | \( \delta = \frac{|V(x^* + \Delta x) - V(x^*)| + |V(x^*) - V(x^* - \Delta x)|}{2\Delta x} \) |
|                       |                                                                                  | \( x^* \mid V(x^*) = \max(V(x)) \) |
| Double rectangular    | \( V_{DR}(x) = \begin{cases} V_1 & 0.0 < x < w_1 \\ V_2 & w_1 + d < x < w_1 + d + w_2 \\ 0.0 \end{cases} \) | \( V_i \) – barrier / height    |
|                       |                                                                                  | \( w_i \) – barrier / width     |
|                       |                                                                                  | \( \delta \) – distance between barriers |
|                       |                                                                                  | \( \alpha = \frac{1}{2} \left( \frac{|V_1 - V_2|}{|V_1 + V_2|} + \frac{|w_1 - w_2|}{|w_1 + w_2|} \right) \) |
|                       |                                                                                  | \( \delta = 0.0 \)              |
Double gaussian

\[ V_{DG}(x) = V_1 e^{-\frac{(x-x_1)^2}{2\sigma_1^2}} + V_2 e^{-\frac{(x-x_2)^2}{2\sigma_2^2}} \]

\[ \sigma_1 = w_1/3; \quad \sigma_2 = w_2/3 \]

\[ x_1 = 0.0; \quad x_2 = 3 \cdot (\sigma_1 + \sigma_2) + d \]

\[ V_i \rightarrow \text{ barrier } i \text{ height} \]

\[ w_i \rightarrow \text{ barrier } i \text{ width} \]

\[ d \rightarrow \text{ distance between barriers} \]

\[ \alpha = \frac{1}{2} \left( \frac{|V_1 - V_2|}{|V_1 + V_2|} \right) \left( \frac{|w_1 - w_2|}{|w_1 + w_2|} \right) \]

\[ s = \frac{V(x^*) - V(x^* + \Delta x)}{\Delta x} \]

\[ x^*|V(x^*) = \max(V(x)) \]

Peskin barrier

\[ V_{Pesk}(x) = V_1 \left( \frac{1}{\cosh^2(x)} - \frac{1}{\cosh^2(w_1,x)} \right) \]

\[ V_1 \rightarrow \text{ barrier height} \]

\[ w_1 \rightarrow \text{ barrier width} \]

\[ \alpha = 1.0 \]

\[ s = \frac{V(x^*) - V(x^* + \Delta x)}{\Delta x} \]

\[ x^*|V(x^*) = \max(V(x)) \]
II. DATASET

A visualization of the distribution of the training and testing dataset respect to its input features and output label is shown in Figures S1a-c. The first plot (a) displays scattering of pairwise feature distributions with kernel density estimate (KDE) diagonals for the entire dataset. The following plots (b-c) show the pairwise distribution of subsets of features and labels for the test and train sets through KDE. Figure S2 maps the correlation between variables through the Pearson correlation coefficient.

Figure S1a: Database pair-wise distribution for all input features and for the output label. Points are datum. Diagonals are gaussian KDE of single features.
Figure S1b: Pair-wise distribution of a subset of the database features related to the barrier for the train and test sets. Blue surfaces are train set KDE, while red lines are test set KDE. KDE is normalized to 1.0. A KDE of 1.0 then indicates the region of two-feature space with the highest density of data.

When looking at the diagonal distribution of weights and heights in Figure S1b, we see a large number of points in the first bin. This comes from the fact that for single barriers the distance between barriers, the second barrier width and height are all equal to zero. Features which included the sum of the heights and the sum of the widths were considered to reduce this bias, however no significant improvement was observed in the trained neural network.
Figure S1c: Pair-wise distribution of a subset of the database features for the train and test sets. Blue surfaces are train set KDE, while red lines are test set KDE. KDE is normalized to 1.0. A KDE of 1.0 then indicates the region of two-feature space with the highest density of data.

Figure S2: Correlation matrix of the Pearson correlation coefficient for input features of the database. Values close to 1.0 indicate a strong positive correlation between variables, while -1.0 indicates a strong negative correlation. Features are perfectly correlated to themselves, as can be seen by the dark red diagonal. Most features have a small correlation, close to zero which indicates little bias.
III. GRID SEARCH & HYPERPARAMETER TUNING

As described in the main text, four subsequent grid searches were carried out. The hyperparameter search spaces for each are shown in Table S 2-5, and optimal hyperparameters are shown in Table S 6. Figure S 3 indicates how the logarithm of the mean absolute error, MSE, averaged over all runs, changes with the number of neurons for each single hidden layer. Figure S 4 shows how the loss changes with the learning rate. Figure S 5 shows validation loss and moving average validation loss during training for a range of batch sizes, highlighting the optima.

A. First grid search: neurons, hidden layers, batch size and epochs

For the first grid search we searched over number of hidden layers, number of neurons per layer, batch size, and number of epochs. The range of parameters is shown in Table S 2 below and the results in the following Figure S 3. We found that the optimal model was of neuron configuration (128, 24, 12).

Table S 2: First grid search space values for number of hidden layers, number of neurons per layer, batch size and epochs.

| Hyperparameter                 | Search space       |
|--------------------------------|--------------------|
| number of hidden layers        | 1, 2, 3            |
| number of neurons in hidden layer | 1, 6, 12, 24, 32, 64, 128 |
| epochs                         | 50, 100, 200, 300  |
| batch size                     | 32, 64, 128        |

| Fixed hyperparameters          | Default values     |
|--------------------------------|--------------------|
| hidden layer activation function| sigmoid            |
| output activation function     | tanh               |
| optimizer                      | Adam               |
| - learning rate                | 0.001              |
| - beta 1                       | 0.9                |
| - beta 2                       | 0.9999             |
| weight initialization          | random uniform     |
| - range                        | [0, 1]             |
| loss function                  | mean squared error (MSE) |

| Optimal hyperparameters        | Values              |
|--------------------------------|---------------------|
| neuron configuration           | (128, 24, 12)       |
| epochs                         | 300                 |
| batch size                     | 32                  |
B. Second grid search: activation function

For the second grid search, the choice of activation functions applied to each neuron in the hidden and output layers was explored. The activation functions tested are seen below in Table S 3. The optimum configuration was found to be sigmoid, linear for hidden layers and output layer, respectively. Other activations such as tanh also performed well in the output layer but only the linear function allows for predictions outside the testing the range.

Table S 3: Second grid search space values for activation function in both hidden and output layers.

| Hyperparameter          | Search space                                      |
|-------------------------|---------------------------------------------------|
| activation function     | softmax, softplus, softsign, relu, tanh, sigmoid, hard sigmoid linear |
| Fixed hyperparameters   | Default values                                    |
| neuron configurations   | (128, 24, 12)                                     |
| epochs                  | 300                                               |
| batch size              | 32                                                |
| optimizer               | Adam                                              |
| - learning rate         | 0.001                                             |
| - beta 1                | 0.9                                               |
| - beta 2                | 0.9999                                            |
| weight initialization   | random uniform                                    |
| - range                 | [0, 1]                                            |
| loss function           | mean squared error (MSE)                          |
| Optimal hyperparameters | Values                                            |
| hidden layer activation | sigmoid                                           |
| output layer activation | linear                                            |
The third grid search optimized the value of learning rate for the Adam optimizer. The range of values explored were on the logarithmic scale to best search hyperparameter space and are shown below in Table S 4. This was conducted on a randomly selected portion of 30% of the training set as well as on 100% of the training set. In both cases, the Adam optimizer’s default value of 0.001 was the top performer. The results are also shown in Figure S 4.

| Hyperparameter       | Search space                  |
|----------------------|-------------------------------|
| learning rate        | 1E-5, 1E-4, 1E-3, 1E-2, 1E-1, 2E-1 |

| Fixed Hyperparameters | Default values |
|-----------------------|----------------|
| neuron configuration  | (128, 24, 12)  |
| epochs                | 300            |
| batch size            | 32             |
| hidden layer activation function | sigmoid     |
| output activation function | linear        |
| optimizer             | Adam           |
| - beta 1              | 0.9            |
| - beta 2              | 0.9999         |
| weight initialization | random uniform |
| - range               | [0, 1]         |
| loss function         | mean squared error (MSE) |

**Optimal hyperparameters**

| Values |
|--------|
| learning rate | 0.001 |

Figure S 4: Validation loss as a function of learning rate in the third grid search. The black and blue lines correspond to grid search on 100% and 30% of the training dataset. Both searches indicate that 0.001 is the optimal learning rate value.
D. **Fourth grid search: batch size for optimal model**

A final grid search on the batch size was conducted to evaluate how fluctuations on the validation loss related to mini batch training batch sizes. Details of the search are described in Table S 5. A moving average of the validation loss was used as a metric, to average the impact of the fluctuations. It was found that 128, and 256 performed best in terms of reduction in fluctuation and final validation loss magnitude and given that 256 had a lower final rolling average loss it was chosen as the best batch size. This result is shown in Figure S 5.

Table S 5: Fourth grid search space for batch size.

| Hyperparameter               | Search space          |
|------------------------------|-----------------------|
| batch size                   | 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192 |

| Fixed Hyperparameters        | Default values        |
|------------------------------|-----------------------|
| neuron configuration         | (128, 24, 12)        |
| epochs                       | 300                   |
| hidden layer activation function | sigmoid              |
| output activation function   | linear                |
| optimizer                    | Adam                  |
| - learning rate              | 0.001                 |
| - beta 1                     | 0.9                   |
| - beta 2                     | 0.9999                |
| weight initialization        | random uniform        |
| - range                      | [0, 1]                |
| loss function                | mean squared error (MSE) |

| Optimal hyperparameters      | Values                |
|------------------------------|-----------------------|
| batch size                   | 256                   |

Figure S 5: Left panel) Validation loss and moving average of validation loss over 20 points for models trained with a range of batch sizes as a function of epochs. Batch size is seen to impact both the fluctuations in the validation loss, as well as the final training and validation loss value. Right panel) Moving average validation loss for the last 100 epochs of training. Batch sizes 128, and 256 perform best with final average losses of 2.0E-5 and 1.9E-5 respectively.
E. Optimal hyperparameters

Table S6: Optimal hyperparameters identified from grid searches for the final training.

| Hyperparameter                        | Value                      |
|---------------------------------------|----------------------------|
| neuron configuration                  | (128, 24, 12)              |
| epochs                                | 300                        |
| batch size                            | 256                        |
| hidden layer activation function      | sigmoid                    |
| output activation function            | linear                     |
| optimizer                             | Adam                       |
| - learning rate                       | 0.001                      |
| - beta 1                              | 0.9                        |
| - beta 2                              | 0.9999                     |
| weight initialization                 | random uniform             |
| - range                               | [0, 1]                     |
| loss function                         | mean squared error (MSE)   |

Performance

|                        |                              |
|------------------------|------------------------------|
| training loss          | 1.55E-5                      |
| test loss              | 1.57E-5                      |
| test MAE [log(1/ps)]   | 0.27                         |

REFERENCES

1. Eckart, C. The penetration of a potential barrier by electrons. *Phys. Rev.* **35**, 1303–1309 (1930).
2. Ahmed, Z. Tunneling through a one-dimensional potential barrier. *Phys. Rev. A* **47**, 4761–4767 (1993).
3. Mandrà, S., Valleau, S. & Ceotto, M. Deep nuclear resonant tunneling thermal rate constant calculations. *Int. J. Quantum Chem.* **113**, 1722–1734 (2013).
4. Virtanen, P. *et al.* SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **17**, 261–272 (2020).