Deep Learning for Scalable Chemical Kinetics

Alisha J. Sharma*, Ryan F. Johnson†, Adam D. Moses‡, and David A. Kessler§
Laboratories for Computational Physics & Fluid Dynamics
U.S. Naval Research Laboratory, Washington, D.C. 20375

Chemistry is critical to many computational fluid dynamics (CFD) problems, such as propulsion system design, engine diagnostics, and atmospheric modeling. However, many real-world reacting flow problems are impractical due to the high computational cost and poor scalability of stiff chemistry integration. Scientific machine learning is gaining traction in the physical sciences as a way to overcome such bottlenecks; however, this technique is rarely used in real-world chemical kinetics codes. Artificial neural networks (ANNs) are particularly interesting due to their ability to compactly represent highly nonlinear functions. In this work, we present our ANN-based model for evolving a stiff chemical source term. We demonstrate this model with a hydrogen-oxidation reaction, integrate the model into a detailed reacting Navier Stokes CFD code, and finally discuss the potential for machine learning in reacting flow CFD.

I. Introduction

The extreme cost of solving for the chemistry source term is a major challenge in reacting flow simulations. It is impractical to solve the fully coupled convection, diffusion, and reaction terms for stiff, detailed chemistry. The most common alternative is operator splitting[1–3], in which the chemical source terms are solved independently as a system of ordinary differential equations (ODEs). Solving for the chemistry term is the primary bottleneck in these simulations: the chemistry often takes 80-90% of the total required resources, with the relative share increasing as chemical mechanism fidelity increases[4]. This is a problem for modern applications as mechanism size has only increased over the past decade[5].

Solving the chemical kinetics for these large mechanisms can be expensive or even impossible, making methods of reducing the cost of chemical kinetics solvers a critical area of research. One popular avenue of reducing cost is through reduced-order or skeletal models, such as the HyChem technique [6–9], which automatically reduce the size of large mechanisms. This approach has yielded some promising reductions in cost, but still require a large amount of computational time to solve with traditional methods. Since the solvers still use stiff ODE integrators, skeletal models share the load-balancing issues and high-precision requirements of traditional methods. In addition to scalability concerns, making simplifying assumptions and removing less salient dependencies affects the accuracy of the model.

Recently, data-driven approaches to scientific computing have gained popularity as a method of overcoming computational bottlenecks in traditional models. Artificial neural networks (ANNs) are particularly interesting due to their representational power, success in highly nonlinear domains, and ability to learn from large amounts of data [10]. For example, Ling et al. improved RANS stress tensor prediction accuracy with a tensor-basis neural network architecture [11] and Bartók et al. used Gaussian processes to replace an expensive density functional theory operation [12].

Despite its success in other areas of scientific computing, deep learning techniques in chemical kinetics are not widespread; however, early results are promising. A number of researchers have achieved impressive speedups, reduced memory requirements, and reasonable accuracy by modeling reduced-order hydrocarbon mechanisms with shallow ANNs [13–19]. These studies saw a speedup of 1–2 orders of magnitude over stiff chemistry integrators, and while propagating prediction errors caused some stability issues in time series integration, most of these models achieved an acceptable accuracy within their training domain. Other studies have focused on extending the capabilities of ANN-based chemistry solvers through developing ANN ensembles for larger, partitioned parameter ranges [15] and more complex

*Computer Scientist, Laboratory for Advanced Computational Physics, Code 6042, AIAA Member.

†Aerospace Engineer, Laboratory for Propulsion, Energetic, and Dynamic Systems, Code 6041, AIAA Member.

‡Computer Scientist, Laboratory for Advanced Computational Physics, Code 6042.

§Mechanical Engineer, Laboratory for Multiscale Reactive Flow Physics, Code 6043, AIAA Member.

Distribution Statement A: Approved for public release; distribution is unlimited.

fuels [20]. Several recent studies have worked to bring these techniques into mainstream use by integrating ANN-based models into traditional computational fluid dynamics (CFD) codes [21–25]. However, issues such as prediction stability in time series integration, the lack of formalized techniques for modeling new fuels or parameter ranges, and the lack of a standards in integrating trained models into CFD codes has limited widespread adoption.

This work is particularly inspired by a recent study by Peng and Pinkowski, in which they develop an ANN to model hydrogen combustion in a homogenous reactor, resulting in an order-of-magnitude speedup over traditional stiff ODE integrators [13]. We build upon this work by expanding to wider parameter ranges and developing reusable techniques for CFD code integration.

We present a new chemical kinetics ANN architecture designed for a large parameter space and stiff regimes. We use these ANNs to model a hydrogen combustion mechanism composed of eight reacting species and 34 reactions in two different parameter spaces. Finally, we analyze our models predictions and robustness and discuss the path towards using ANNs in production.

II. Methods

In this research, we use a deep feedforward ANN to model an H_2 oxidization reaction. Neural networks are a class of machine learning algorithms capable of acting as universal approximators [26], excelling in highly-nonlinear, noisy, and online domains [10]. They have gained popularity in the past decade due to their breakthrough success in fields such as computer vision [27] and natural language processing [28], their flexibility in both the architecture and loss function, and their scalability on modern architectures.

The basic goal of a deep neural network is to approximate a function $f : x \longrightarrow y$ from data for some input x and output y. After the form of f is chosen, the network weights are adjusted through a process called *stochastic gradient descent*: the model iteratively (a) makes predictions on a randomly-selected mini-batch of training examples, (b) determines the prediction quality using a loss function $\mathcal{L}$, and finally (c) updates the parameter weights in a process called *backpropagation*. This process is repeated until the loss converges to an acceptable minimum [10]. The trained model $\hat{f}$ can then be used to generate predictions $\hat{f} : x \longrightarrow \hat{y}$ for unseen input.

For these experiments, we are specifically trying to learn the autoregressive function $f : (T_t, p_t, Y_t) \longrightarrow (T_{t+\Delta t}, p_{t+\Delta t}, Y_{t+\Delta t})$ for some timestep Δt in a hydrogen oxidation reaction.

A. Network Architecture

Our network architecture (Figure 1) is currently composed of twelve jointly-trained sub-networks: one for temperature, pressure, and each of ten species. Each sub-network includes two densely-connected hidden layers with 30 neurons per layer (Figure 2).

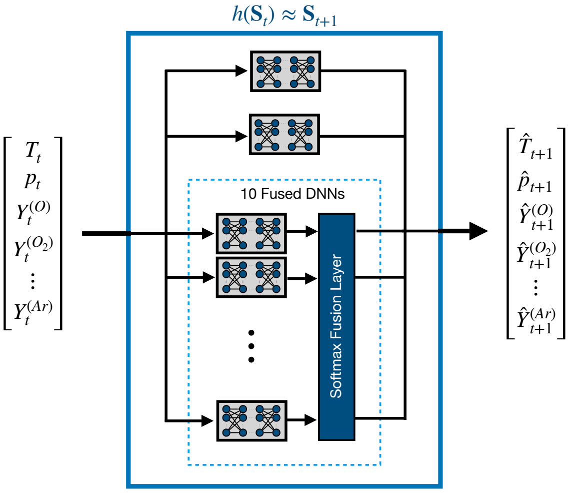


Figure 1 Network architecture, consisting of 12 jointly-trained artificial neural network (ANN). The species predictions are normalized to fractions using a softmax function, and then the weighted loss is calculated using all predictions.

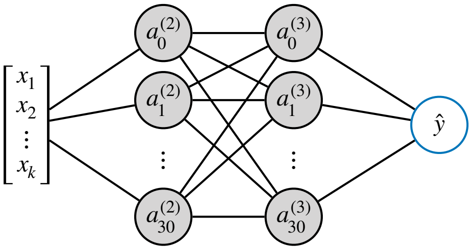


Figure 2 Each subnetwork is an densely-connected artificial neural network (ANN) consisting of two hidden layers, 30 neurons per layer.

In an accurate prediction, the mass fractions for each species must sum to one. To enforce this, we normalize the predicted values using softmax activation function (Eq. 1), which enforces a soft mass conservation constraint of $Y^{(k)} \in (0, 1)$ and $\sum_{j=1}^{k} Y^{(k)} = 1$ for k species predictions.

$$\sigma(\mathbf{z})_i = \frac{\exp z_i}{\sum_{j=1}^{k} \exp z_j} \tag{1}$$

B. Loss Function

Neural networks are trained by optimizing the loss function. This network loss function currently has three weighted components corresponding to the temperature, pressure, and species concentration losses, as seen below:

$$\mathcal{L} = \lambda_Y \mathcal{L}_Y + \lambda_T \mathcal{L}_T + \lambda_p \mathcal{L}_p + \lambda_R R(\theta) \tag{2}$$

$$\mathcal{L}_T = MSE_T = \frac{1}{n}\sum_{i=1}^{n}(T_i - \hat{T}_i)^2 \tag{3}$$

$$\mathcal{L}_p = MSE_p = \frac{1}{n}\sum_{i=1}^{n}(p_i - \hat{p}_i)^2 \tag{4}$$

$$\mathcal{L}_Y = MSLE_Y = \sum_{j=1}^{k} \frac{\lambda_Y^{(k)}}{n} \sum_{i=1}^{n} \left(\log(Y_i^{(j)}) - \log(\hat{Y}_i^{(j)})\right) \tag{5}$$

T, p, and Y represent the temperature, pressure, and species concentrations respectively, and $\hat{T}$, $\hat{p}$, and $\hat{Y}$ represent the respective network predictions. Loss weights are set according to the mean value of the associated outputs. While temperature ($\mathcal{L}_T$) and pressure ($\mathcal{L}_p$) loss functions are set to the mean squared error, the species concentration loss ($\mathcal{L}_Y$) uses mean squared logarithmic error due to the extremely low concentrations associated with free radicals.

During training, we also use equally-weighted L1 and L2 regularization (Eq. 6) at each node to prevent overfitting and encourage a more generalizable function. This is especially important for time series integration: if the network is not resilient to small errors in input, errors will propagate and the time history will diverge. Both L1 and L2 regularization encourage small parameter weights, but L1 regularization has the additional quality of shrinking unimportant parameters to zero. This makes it useful for feature selection and determining feature saliency, which we may investigate in more detail in future work.

$$R_{L1,L2} = \sum_{j=1}^{p} \left(\left|\theta_j\right| + \theta_j^2\right) \tag{6}$$

C. Datasets

We use the Cantera [29] library to generate a synthetic well-mixed hydrogen oxygenation dataset using the Westbrook H_2-air reaction mechanism [30]. This is a small mechanism consisting of four elements, ten species, and 34 reactions. The dataset size and parameter space varied for each experiment; more details can be found in the corresponding experimental sections.

The network is trained on tuples of initial and final conditions: temperature T (measured in Kelvin), pressure p (measured in standard atmospheres), and species concentration $\mathbf{Y} = (Y_0, ..., Y_k)$ (measured as a percentage). The initial conditions, or *input data*, are represented by **x**, and the known final conditions, or *labels*, are represented by **y**.

ANNs train more efficiently and effectively when data is normalized and scaled to small range, so we use log normalization for our input state, as in [13]. As with the species loss above (Eq. 5), log normalization was chosen because this yields better results for free radicals, whose species concentrations skew towards zero [13]. The normalization constants are calculated from the training set only and applied to the data during both training and inference.

$$\begin{aligned} \mu_{\log} &= \frac{1}{N_{\text{train}}} \sum_{j=0}^{N_{\text{train}}} \log(x_j) \\ \sigma_{\log} &= \sqrt{\text{var}(\log(x_{\text{train}}))} \\ z_i &= \frac{\log(x_i) - \mu_{\log}}{\sigma_{\log}} \end{aligned} \tag{7}$$

In order to make the model robust to small input errors, the generated dataset was augmented with zero-centered Gaussian noise, which was scaled according to input magnitude.

D. Implementation and Training

The neural network was implemented using the TensorFlow machine learning library [31]. In order to improve prediction performance of free radicals, which have extremely small concentrations compared to the major species, the species sub-networks were trained independently for 100 epochs. The full network was trained for 1,000 epochs in minibatches of 400 examples. We use the Adam optimizer [32] with optimization hyperparameters set to learning rate $\alpha = 0.00003$ and momentum terms $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Examples were reshuffled and batched before each epoch to facilitate training and avoid local minima.

The dataset was randomly split into training, testing, and cross-validation sets with 80%, 10%, and 10% of the data respectively.

E. Integration into CFD Codes

There have been a number of previous attempts to integrate machine learning models into reacting flow CFD code [21–25], including remote function calls from legacy Fortran codes to modern machine learning libraries, re-implementing the full fluid solver in TensorFlow, and embedding the network directly into the codes. Our approach to integration was to develop a lightweight C/C++ wrapper library for our model. We used the experimental TensorFlow Lite for Microcontrollers [31] library to export our trained model to a byte array and then perform lightweight inference without requiring the full TensorFlow ecosystem. The primary benefit to this approach is flexibility: the implementation can be changed, extended, or optimized without affecting the code integration, the library can be integrated into new codes without requiring a complicated extraction, and the code runs on multiple types of hardware.

III. Experiments

We conducted three experiments (a) to verify the basic functionality of our model in a limited parameter range, (b) to evaluate our model's performance in very stiff regimes, and (c) finally to demonstrate our model real-world potential for reacting flow simulations.

A. Variable Temperature

In this first experiment, we demonstrate our model's ability to approximate chemical kinetics by training and evaluating it for a fixed timestep of $\Delta t = 50$ ns. To accomplish this, we limit training and evaluation to regimes in which all temperature, pressure, and species graphs have a similar stiffness.

Parameter	*Range*	*Steps*
Time (Δt)	0–5×10^{-8} s	1,000
Temperature (T)	1,300–2,500 K	100
Pressure (p)	1 atm	1
Equivalence Ratio (ϕ)	1.0	1

Table 1 Initial conditions for a limited-range hydrogen combustion in a homogenous reactor. The initial gas mixture was fixed at O_2:1.0 and N_2:3.76. All steps are evenly spaced in the given ranges.

1. Setup

This experiment used a dataset consisting of 99,900 temporal pairs. In this experiment, the pressure and equivalence ratio were held steady while the initial temperature varied. The initial conditions for these sections are described in Table 1.

2. Results

Our model was able to predict temperature, pressure, and species concentration with a high degree of accuracy when compared to numerical results from our hydrogen combustion mechanism. Our model was able to predict the temperature within 0.6% accuracy (±0.3 K), with an average species mass fraction error of 4.86×10^{-5}. Converted mole fraction predictions for the major and minor reacting species can be seen in Figure 3. Qualitatively, the model is able to generate highly accurate predictions. While direct prediction is a useful metric, long-term stability is important to reacting flow simulations. Many real-world problems require a time series integration over multiple timesteps before a "fresh" chemical state is provided. To simulate this, we performed a time series integration in which the only the initial conditions are given as input. The function then feeds subsequent species mass fraction predictions into the model, generating a full time history of the reaction.

In order to stabilize the time series integration, several constraints were imposed on the function. First, raw mass fraction predictions were clipped to [0,1] and then normalized to sum to 1 using Eq. 1. Second, predicted values were adjusted to obey conservation properties before being fed back into the model. Finally, mass fractions near zero were damped to prevent small numerical errors from propagating, as in [22].

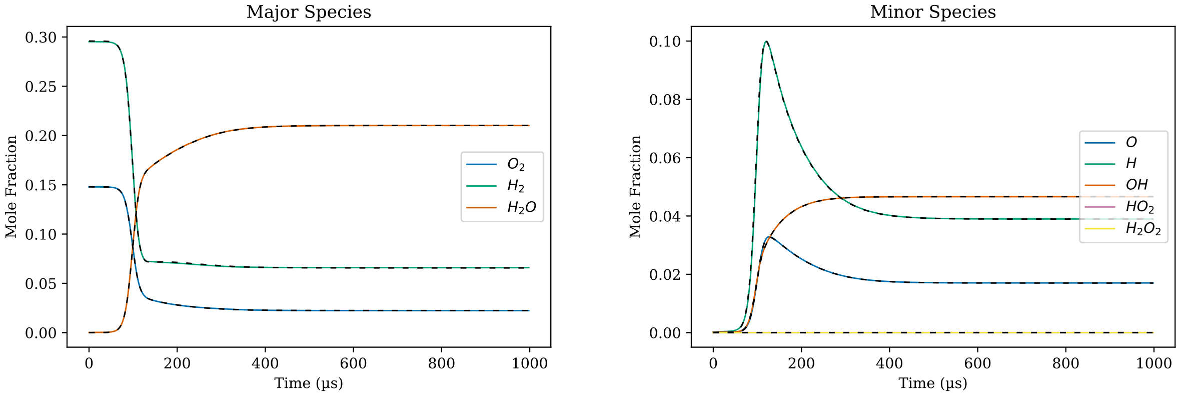


Figure 3 Predicted vs. actual species mole fractions with initial conditions T_0 = 1900 K, p_0 = 1 atm, ϕ = 1.0, and t_0 = 0. The graphs show the raw prediction (colored dots) with the numerical results (black dashed line) overlaid as a function of time.

We evaluate our model quantitatively two metrics: the root mean squared error (or RMSE, shown in Eq. 8) gives the absolute magnitude of error for each species, especially penalizing large outliers. On the other hand, the root mean squared logarithmic error (or RMSLE, shown in Eq. 9) penalizes order-of-magnitude differences, which better reflects the error for free radicals and species at near-zero states.

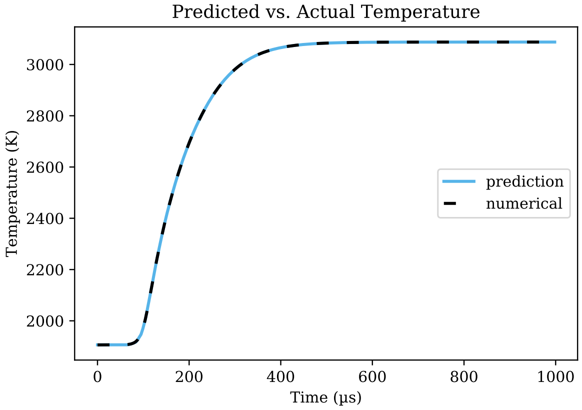


Figure 4 Predicted vs. actual temperatures (right) with initial conditions $T_0 = 1900$ K, $p_0 = 1$ atm, $\phi = 1.0$, and $t_0 = 0$.

Species Mass Fraction, Y_i	*RMSE (Eq. 8)*	*RMSLE (Eq. 9)*
O	5.89×10^{-5}	5.82×10^{-5}
O_2	1.06×10^{-4}	9.67×10^{-5}
H	2.25×10^{-5}	2.24×10^{-5}
H_2	4.24×10^{-5}	4.19×10^{-5}
OH	5.38×10^{-5}	5.30×10^{-5}
HO_2	5.49×10^{-6}	5.48×10^{-6}
H_2O	9.60×10^{-5}	8.70×10^{-5}
H_2O_2	3.84×10^{-6}	3.83×10^{-6}
Overall	4.86×10^{-5}	4.61×10^{-5}

Table 2 Average mass fraction error for the fixed timestep model on the test dataset; this includes samples within the training distribution that the model did not see during training. We provide both absolute (RMSE) and relative (RMSLE) error metrics to capture error in both the major species and free radicals.

$$\text{RMSE} = \sqrt{\frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \left(Y_i - \hat{Y}_i\right)^2} \tag{8}$$

$$\text{RMSLE} = \sqrt{\frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \left(\log(Y_i) - \log(\hat{Y}_i)\right)^2} \tag{9}$$

Quantitative analysis shows that the model yields highly accurate predictions within its training distribution, yielding extremely low (on the order of 1×10^{-5}) error for both metrics. These results are displayed in Table 2. While the prediction accuracy is high across the board, the low concentrations of free radicals pose the greatest risk. While this does not appear to strongly affect prediction quality, even a small error can be several orders of magnitude larger than a free radical mass fraction.

B. Expanding the Parameter Range

Our second case extends the model to stiffer regimes that more accurately represent real-world reacting flow problems. These challenging cases are important because they represent the cases where stiff ODE solvers struggle: due to the small timesteps required, load balancing is challenging, and the computational requirements can be high. Furthermore, the full parameter range is quite diverse: there are regions of equilibrium, regions of non-stiff chemistry,

Parameter	*Range*	*Steps*
Time (Δt)	0–5×10^{-8} s	1,000
Temperature (T)	800–2,500 K	170
Pressure (p)	0.9–1.1 atm	3
Equivalence Ratio (ϕ)	0.1–1.5	14

Table 3 Initial conditions for expanded hydrogen combustion in a homogenous reactor. The initial gas mixture was fixed at O_2:1.0 and N_2:3.76. All steps are evenly spaced in the given ranges.

Species Mass Fraction, Y_i	*RMSE (Eq. 8)*	*RMSLE (Eq. 9)*
O	8.38×10^{-5}	8.26×10^{-5}
O_2	1.77×10^{-4}	1.59×10^{-4}
H	2.38×10^{-5}	2.37×10^{-5}
H_2	6.60×10^{-5}	6.47×10^{-5}
OH	8.89×10^{-5}	8.75×10^{-5}
HO_2	7.69×10^{-6}	7.66×10^{-6}
H_2O	2.02×10^{-4}	1.89×10^{-4}
H_2O_2	8.13×10^{-6}	8.10×10^{-6}
Overall	2.19×10^{-4}	7.78×10^{-5}

Table 4 Average mass fraction error for the larger parameter range. While these errors are still small, they are an increase from the more limited parameter range. They also include more outliers and extreme values, which could pose a problem for real-world simulations.

and regions of very stiff chemistry; it is challenging for one model to perform well in all cases. In this case, we extend the temperature range as well as varying the pressure and equivalence ratio (Table 3).

To verify our model accuracy, we conduct an error analysis comparing our reaction state predictions with the numerical "ground truth" solution (Table 4). We found that while the absolute error is acceptably low, there are large absolute errors in some regions (particularly affecting major species predictions) which affects the stability of the model in time series predictions.

1. Setup

The expanded dataset consisted of 7,128,360 temporal pairs from the initial conditions described in Table 3, which cover most canonical cases in hydrogen combustion.

2. Results

The results in this expanded parameter range are encouraging due to their low magnitude (Table 4). The RMSE is about an order-of-magnitude larger in this larger parameter range than in the more limited range; this would imply that there are more outliers and errors in the major species than in the limited range which may prove troublesome in real-life simulations. Less dramatically, RMSLE for this expanded parameter range is about twice has large as the RMSLE error in the limited parameter range.

While the low errors are promising, the stability of this model will likely need to be improved before it is useful in production applications. This can be done several ways, including: model ensembles (as in , each of which is responsible for a narrower parameter range; guided optimization to enforce physical constraints, and

C. Model Analysis and Explainability

As discussed in the introduction, despite promising early results in modeling stiff chemical kinetics with ANNs, their usage is not widespread. One hurdle to ANN integration into production codes is the lack of interpretability: it can be difficult to reason about why a model is making a given prediction, so it can be difficult to trust its output. This is especially true in problems such as time series prediction, in which the model must make accurate predictions given noisy data near its training boundaries.

There are a number of ways to build confidence in a model. Performance validation over the region of interest is straightforward, but fully validating a complex model can be time-consuming and prohibitively expensive. Uncertainty quantification, a popular technique in scientific computing, is gaining traction in the machine learning community. In this paper, we look at two topics in explainability: model robustness within its training domain and feature saliency.

This model was trained using equally-space values across the parameter space, but some regions of the landscape are likely more difficult to model than others. This can be seen in Figure 5: we have plotted the average prediction error across the temperature input range. As we would expect, the prediction accuracy suffers near the edge of the training distribution; this is especially true at low temperatures. Future work could include supplementing training with higher concentration of examples in these challenging regions.

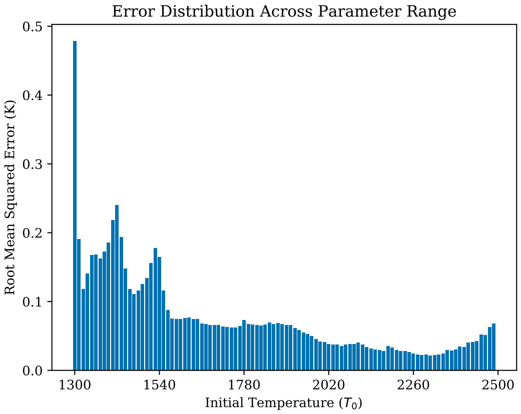


Figure 5 The distribution of temperature prediction error across the range of input temperatures in the limited hydrogen case. Unsurprisingly, error increases near the edge of the training distribution.

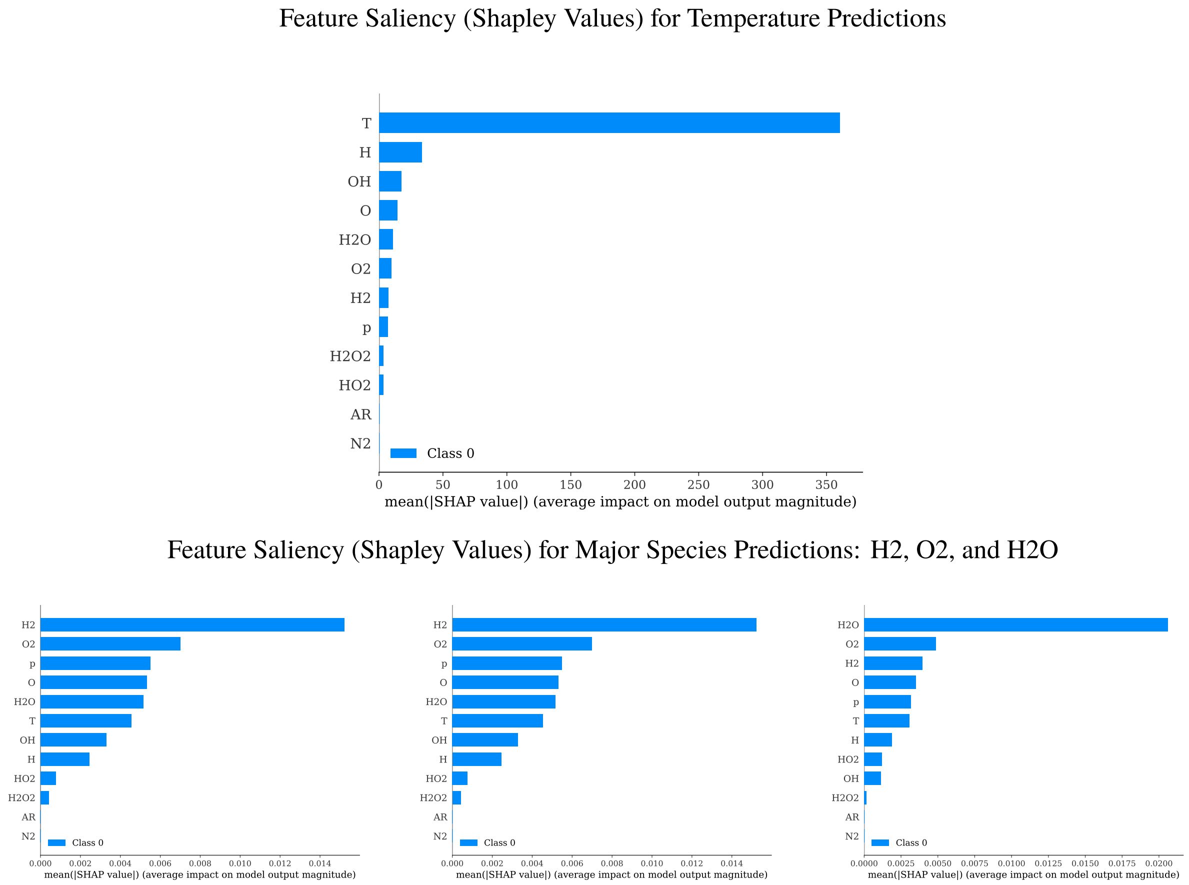
Because our model is emulating a well-understood chemical mechanism, we can build confidence in its predictions through feature saliency analysis: analyzing how important different features are for individual prediction or global quantity. We study the forcing functions for different outputs using Shapely Additive Explanations (SHAP) [33], a surrogate analysis technique by game theory to determine feature saliency in a robust, interpretable way. SHAP is a class of additive feature attribution methods that use a locally linear surrogate model to quantifiy each input's effect on a given prediction [33]. Here, we calculate the SHAP values across our testing dataset to determine relative feature importance temperature and the major species (Figure 6).

Temperature is primarily affected by the prior timestep's temperature with some slight dependence on free radicals associated with enthalpy change, as we would expect. In contrast, the major species are much more evenly affected by the species concentrations: all of them are heavily influenced by the other major species; are moderately influenced by pressure, temperature, and O concentration; are barely impacted by the smaller free radials (H_2O_2 and HO_2), and are not impacted at all by the nonreactive species N_2 and Ar.

IV. Conclusion

Stiff chemistry integration is a major bottleneck in many science and engineering design problems, and this challenge is increasing as researchers seek more and more complex mechanisms. Current state-of-the-art chemistry ODE solvers are resource-intensive for large mechanisms, causing many desirable simulations to be very expensive or even impossible. Data-driven models could make these simulations possible, allowing researchers to develop new fuels, engines, and vehicles that were previously out of reach. Due to its flexibility, representational power, and prediction efficiency, deep learning is a particularly promising approach for building models that work over a large parameter space, including very stiff regimes.

We have developed a model for approximating chemical kinetics in an efficient, scalable way. We validate our model in a H_2 oxidation reaction, testing its accuracy and efficiency in stiff regimes. We then analyze the robustness and



Feature Saliency (Shapley Values) for Temperature Predictions

Feature Saliency (Shapley Values) for Major Species Predictions: H2, O2, and H2O

Figure 6 Feature saliency analysis for temperature predictions in the hydrogen combustion model.

representation of our model, helping us build confidence in its predictions. Deep learning has the potential to overcome many challenges in CFD, but there is much work to do before these techniques are ready for full production use. Future work could include stabilizing the model for time series integration through model and optimization constraints, modeling larger mechanisms such as hydrocarbon combustion reactions, incorporating uncertainty quantification to improve confidence in model predictions, exploring transfer learning and other weak supervision methods for modeling larger mechanisms, and exploring ways to optimize the trained model for CFD code.

Acknowledgements

This work is sponsored by the Office of Naval Research through the Unlocking Combustion Mechanisms Project under the NRL 6.1 Materials Science Task Area.

References

[1] Knio, O., Najm, H., and Wyckoff, P., "A Semi-Implicit Numerical Scheme for Reacting Flow: II. Stiff, Operator-Split Formulation," *Journal of Computational Physics*, Vol. 154, No. 2, 1999, pp. 428–467. doi:10.1006/jcph.1999.6322.

[2] Lanser, D., and Verwer, J., "Analysis of Operator Splitting for Advection-Diffusion-Reaction Problems from Air Pollution Modelling," *Journal of Computational and Applied Mathematics*, Vol. 111, No. 1-2, 1999, pp. 201–216. doi:10.1016/S0377-0427(99)00143-0.

[3] Oran, E. S., and Boris, J. P., *Numerical Simulation of Reactive Flow*, 2$^{\text{nd}}$ ed., Vol. 12, Cambridge University Press, New York, NY, 2000. doi:10.1017/CBO9780511574474.

[4] Cuoci, A., Frassoldati, A., Faravelli, T., and Ranzi, E., “Numerical Modeling of Laminar Flames with Detailed Kinetics Based on the Operator-Splitting Method,” *Energy & Fuels*, Vol. 27, No. 12, 2013, pp. 7730–7753. doi:10.1021/ef4016334.

[5] Lu, T., and Law, C. K., “Toward Accommodating Realistic Fuel Chemistry in Large-Scale Computations,” *Progress in Energy and Combustion Science*, Vol. 35, No. 2, 2009, pp. 192–215. doi:https://doi.org/10.1016/j.pecs.2008.10.002.

[6] Tao, Y., Xu, R., Wang, K., Shao, J., Johnson, S. E., Movaghar, A., Han, X., Park, J.-W., Lu, T., Brezinsky, K., Egolfopoulos, F. N., Davidson, D. F., Hanson, R. K., Bowman, C. T., and Wang, H., “A Physics-Based Approach to Modeling Real-Fuel Combustion Chemistry - III. Reaction Kinetic Model of JP10,” *Combustion and Flame*, Vol. 198, 2018, pp. 466–476. doi:10.1016/j.combustflame.2018.08.022.

[7] Wang, H., Xu, R., Wang, K., Bowman, C. T., Hanson, R. K., Davidson, D. F., Brezinsky, K., and Egolfopoulos, F. N., “A Physics-Based Approach to Modeling Real-Fuel Combustion Chemistry - I. Evidence from Experiments, and Thermodynamic, Chemical Kinetic and Statistical Considerations,” *Combustion and Flame*, Vol. 193, 2018, pp. 502 – 519. doi:10.1016/j.combustflame.2018.03.019.

[8] Wang, K., Xu, R., Parise, T., Shao, J., Movaghar, A., Lee, D. J., Park, J.-W., Gao, Y., Lu, T., Egolfopoulos, F. N., Davidson, D. F., Hanson, R. K., Bowman, C. T., and Wang, H., “A Physics-Based Approach to Modeling Real-Fuel Combustion Chemistry - IV. HyChem Modeling of Combustion Kinetics of a Bio-Derived Jet Fuel and Its Blends with a Conventional Jet A,” *Combustion and Flame*, Vol. 198, 2018, pp. 477–489. doi:10.1016/j.combustflame.2018.07.012.

[9] Xu, R., Wang, K., Banerjee, S., Shao, J., Parise, T., Zhu, Y., Wang, S., Movaghar, A., Lee, D. J., Zhao, R., Han, X., Gao, Y., Lu, T., Brezinsky, K., Egolfopoulos, F. N., Davidson, D. F., Hanson, R. K., Bowman, C. T., and Wang, H., “A Physics-Based Approach to Modeling Real-Fuel Combustion Chemistry - II. Reaction Kinetic Models of Jet and Rocket Fuels,” *Combustion and Flame*, Vol. 193, 2018, pp. 520–537. doi:10.1016/j.combustflame.2018.03.021.

[10] Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning*, MIT Press, 2016.

[11] Ling, J., Kurzawski, A., and Templeton, J., “Reynolds Averaged Turbulence Modelling Using Deep Neural Networks with Embedded Invariance,” *Journal of Fluid Mechanics*, Vol. 807, 2016, pp. 155–166. doi:10.1017/jfm.2016.615.

[12] Bartók, A. P., Kermode, J., Bernstein, N., and Csányi, G., “Machine Learning a General-Purpose Interatomic Potential for Silicon,” *Phys. Rev. X*, Vol. 8, No. 4, 2018, p. 041048. doi:10.1103/PhysRevX.8.041048.

[13] Peng, W. Y., and Pinkowski, N. H., “Efficient and Accurate Time-Integration of Combustion Chemical Kinetics Using Artificial Neural Networks,” 2017.

[14] Blasco, J. A., Fueyo, N., Dopazo, C., and Ballester, J., “Modelling the Temporal Evolution of a Reduced Combustion Chemical System With an Artificial Neural Network,” 1998, p. 15.

[15] Blasco, J., Fueyo, N., Larroya, J., Dopazo, C., and Chen, Y.-J., “A Single-Step Time-Integrator of a Methane–Air Chemical System Using Artificial Neural Networks,” *Computers & Chemical Engineering*, Vol. 23, No. 9, 1999, pp. 1127–1133. doi:10.1016/S0098-1354(99)00278-1.

[16] Chen, J.-Y., Blasco, J., Fueyo, N., and Dopazo, C., “An Economical Strategy for Storage of Chemical Kinetics: Fitting in Situ Adaptive Tabulation with Artificial Neural Networks,” *Proceedings of the Combustion Institute*, Vol. 28, No. 1, 2000, pp. 115–121. doi:10.1016/S0082-0784(00)80202-7.

[17] Christo, F., Masri, A., and Nebot, E., “Artificial Neural Network Implementation of Chemistry with Pdf Simulation of H2/CO2 Flames,” *Combustion and Flame*, Vol. 106, No. 4, 1996, pp. 406–427. doi:10.1016/0010-2180(95)00250-2.

[18] Christo, F., Masri, A., Nebot, E., and Pope, S., “An Integrated PDF/Neural Network Approach for Simulating Turbulent Reacting Systems,” Symposium (International) on Combustion, 1996.

[19] Shenvi, N., Geremia, J., and Rabitz, H., “Efficient Chemical Kinetic Modeling through Neural Network Maps,” *The Journal of Chemical Physics*, Vol. 120, No. 21, 2004, pp. 9942–9951. doi:10.1063/1.1718305.

[20] Ranade, R., Alqahtani, S., Farooq, A., and Echekki, T., “An ANN Based Hybrid Chemistry Framework for Complex Fuels,” *Fuel*, Vol. 241, 2019, pp. 625–636. doi:10.1016/j.fuel.2018.12.082.

[21] Keller, C. A., and Evans, M. J., “Application of Random Forest Regression to the Calculation of Gas-Phase Chemistry within the GEOS-Chem Chemistry Model V10,” *Geosci. Model Dev.*, 2019, p. 18.

[22] Buchheit, K., Owoyele, O., Jordan, T., and Essendelft, D. T. V., “STEV: A Stabilized Explicit Variable-Load Solver with Machine Learning Acceleration for the Rapid Solution of Stiff Chemical Kinetics,” 2019, p. 25.

[23] Sen, B. A., and Menon, S., “Linear Eddy Mixing Based Tabulation and Artificial Neural Networks for Large Eddy Simulations of Turbulent Flames,” *Combustion and Flame*, Vol. 157, No. 1, 2010, pp. 62–74. doi:10.1016/j.combustflame.2009.06.005.

[24] Sen, B. A., Hawkes, E. R., and Menon, S., “Large Eddy Simulation of Extinction and Reignition with Artificial Neural Networks Based Chemical Kinetics,” *Combustion and Flame*, Vol. 157, No. 3, 2010, pp. 566–578. doi:10.1016/j.combustflame.2009.11.006.

[25] Cerri, G., Michelassi, V., Monacchia, S., and Pica, S., “Kinetic Combustion Neural Modelling Integrated into Computational Fluid Dynamics,” *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, Vol. 217, No. 2, 2003, pp. 185–192. doi:10.1243/09576500360611218.

[26] Hornik, K., Stinchcombe, M., and White, H., “Multilayer Feedforward Networks Are Universal Approximators,” *Neural Networks*, Vol. 2, No. 5, 1989, pp. 359–366. doi:10.1016/0893-6080(89)90020-8.

[27] Krizhevsky, A., Sutskever, I., and Hinton, G. E., “Imagenet Classification with Deep Convolutional Neural Networks,” *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[28] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J., “Distributed Representations of Words and Phrases and Their Compositionality,” *Advances in Neural Information Processing Systems*, 2013, pp. 3111–3119.

[29] Goodwin, D. G., Speth, R. L., Moffat, H. K., and Weber, B. W., “Cantera: An Object-Oriented Software Toolkit for Chemical Kinetics, Thermodynamics, and Transport Processes,” 2018. doi:10.5281/zenodo.1174508, version 2.4.0.

[30] Ó Conaire, M., Curran, H. J., Simmie, J. M., Pitz, W. J., and Westbrook, C. K., “A Comprehensive Modeling Study of Hydrogen Oxidation: A Comprehensive Modeling Study of Hydrogen Oxidation,” *International Journal of Chemical Kinetics*, Vol. 36, No. 11, 2004, pp. 603–622. doi:10.1002/kin.20036.

[31] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Yangqing Jia, Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X., “TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems,” 2015.

[32] Kingma, D. P., and Ba, J., “Adam: A Method for Stochastic Optimization,” *International Conference on Learning Representations*, 2015.

[33] Lundberg, S. M., and Lee, S.-I., “A unified approach to interpreting model predictions,” *Advances in Neural Information Processing Systems*, 2017, pp. 4765–4774.