Kinetics parameter optimization of hydrocarbon fuels via neural ordinary differential equations

Xingyu Su[a], Weiqi Ji[b], Jian An[c], Zhuyin Ren[a,c,*], Sili Deng[b], Chung K. Law[d,a]

[a] *Center for Combustion Energy, Tsinghua University, Beijing 100084, China*

[b] *Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

[c] *Institute for Aero Engine, Tsinghua University, Beijing, 100084, China*

[d] *Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ 08544, USA*

Submitted to **Combustion and Flame**

Article type: Full length

*Corresponding author:

Address: Mengminwei Science and Technology Building N317, Tsinghua University, Beijing 100084, China

E-mail: zhuyinren@tsinghua.edu.cn

Phone: +86-10-62797371

Abstract

Chemical kinetics mechanisms are essential for understanding, analyzing, and simulating complex combustion phenomena. In this study, a Neural Ordinary Differential Equation (Neural ODE) framework is employed to optimize the kinetics parameters of reaction mechanisms. Given experimental or high-cost simulated observations as training data, the proposed algorithm can optimally recover the hidden characteristics in the data. Different datasets of various sizes, types, and noise levels are systematically tested. A classic toy problem of stiff Robertson ODE is first used to demonstrate the learning capability, efficiency, and robustness of the Neural ODE approach. A 41-species, 232-reactions JP-10 skeletal mechanism and a 34-species, 121-reactions n-heptane skeletal mechanism are then optimized with species' temporal profiles and ignition delay times, respectively. Results show that the proposed algorithm can optimize stiff chemical models with sufficient accuracy, efficiency and robustness. It is noted that the trained mechanism not only fits the data perfectly but also retains its physical interpretability, which can be further integrated and validated in practical turbulent combustion simulations. In addition, as demonstrated with the stiff Robertson problem, it is promising to adopt Bayesian inference techniques with Neural ODE to estimate the kinetics parameter uncertainties from experimental data.

Key Words: Chemical kinetics; Parameter optimization; Adjoint sensitivity; Neural networks

1. Introduction

Well-developed chemical kinetics models, with satisfying accuracy and conciseness, are essential in the research and design of energy conversion devices and biochemical processes. Classical approaches to building these models include ab initio calculations and reaction templates developed with expert knowledge [1]. Further parameter estimation may use rate rules and other functional group methods. However, these parameter estimates are seldom accurate enough to predict the quantity of interests (QoIs), and a refinement process to optimize the kinetics models is often needed. Generally, the kinetics model structures for small hydrocarbon and oxygenated fuels are relatively complete, while the kinetics parameters still have large uncertainties [2]. Consequently, in order to meet the accuracy criterion, it is necessary to use experimental data or high-cost data from computational quantum chemistry to optimize the kinetics parameters. On the other hand, with more experimental data accumulated in the past decades, optimizing and updating existing mechanisms with newly acquired data also plays an important role in chemical kinetics research.

The kinetic model optimization process is generally treated as solving an inverse problem with the aid of sensitivity analysis [3, 4], uncertainty analysis [5, 6], and subsequent data regression [7-9]. For chemical kinetics, the procedure is generally divided into the forward and reverse uncertainty quantification (UQ) steps. Forward UQ will build a response surface mapping from the kinetics parameters to a specific QoI, such as ignition delay times (IDTs) and laminar flame speeds, with its form being unlimited. Thus, the output can be obtained at a relatively low cost. Commonly used mapping functions are polynomial chaos expansion (PCE) [5, 7], high dimensional model representation (HDMR) [8], and artificial neural network (ANN) [8]. Since the kinetics parameter space is always high-dimensional, techniques such as sensitivity analysis [10] and the active subspace method [6, 11] are often employed to perform a reduction in the parameter space. With the response surface acquired, Monte Carlo sampling can be straightforwardly conducted to quantify the uncertainty of the output QoI. Then the reverse UQ employs the response surface as a surrogate and optimizes the kinetics parameters, i.e., pre-exponential factors, temperature coefficients, and activation energies. With the response surface being constructed with sufficient samples, one can train the low-cost surrogate models with widely used optimization approaches, such as genetic algorithm (GA) [12], stochastic gradient descent (SGD) [13], and Bayesian regression [2, 9]. However, building the response surface is often time-consuming, while

insufficient samples could lead to non-physical behaviors in the surrogate models [2]. Besides, the response surface is usually used to map some representative global characteristics such as IDTs, and it is difficult to map the data with temporal and spatial evolution, such as the species profiles from shock tube experiments. To overcome these problems, efficient modeling of chemical kinetics and affordable gradient calculation holds the potential to update the parameters of chemical models directly.

During the past decade, advances in deep learning have offered opportunities for efficient high-fidelity combustion simulations. The most important advantages include the universal approximation ability, well-developed optimization techniques, and open-source ecosystems. Ihme et al. [14, 15] employed ANN to tabulate the filtered thermochemical quantities, including the filtered reaction rate of the progress variable, from the mean, the variance of mixture fraction and the mean of the progress variable. Owoyele et al. [16] stepped further with the mixture of expert (MoE) approach that splits the input space into different sections and fine-tunes specific neural networks for each section. Rassi et al. [17] proposed the physics-informed neural network (PINN) that allows physical constraint of governing equations during the neural network training process, which enables the distillation of the physical behavior in datasets to control the parameters of a given system. Following that, Ji et al. [18] employed PINN in combustion scenarios and proposed the stiff-PINN method to optimize kinetics parameters. There are also plenty of neural network-related studies focusing on building surrogate chemical models [19], constructing sub-grid chemical source terms [20], and discovering unknown reaction pathways [21]. Thus, it is a promising method to utilize neural networks to optimize kinetics mechanisms.

Recently, Chen et al. [22] proposed the neural ordinary differential equation (Neural ODE) approach in deep learning and showed its capability for learning computer vision tasks. It employs multi-layer perceptron (MLP) layers as the ODE layer and uses neural networks to approximate the given ODE systems or classification tasks by the gradient descent method, with the gradients calculated via the adjoint sensitivity method. The applications of Neural ODE have inspired extensive recent developments in the adjoint sensitivity algorithms [23, 24] and associated open-source software ecosystems [25, 26]. For example, Ma et al. [24] studied the performance of different implementations of the adjoint sensitivity method and suggested guidelines on the choice of adjoint sensitivity algorithms based on the size and stiffness of the ODE system. Ji et al.[21, 26-

28] proposed chemical reaction neural networks (CRNN) and augmented them with neural ODE to facilitate learning reaction pathways of the biomass pyrolysis process.

In the present work, we introduce the Neural ODE concept to optimize realistic chemical ODE systems to take advantage of these recent developments. Note that the neural ODEs are not black boxes or fully connected layers that are only used to fit experimental data, but are sparse connections that share the same pathways of chemical kinetics models. With automatic differentiation techniques enabled by differentiable programming, one can optimize the neural ODEs with training data efficiently. The proposed kinetics parameter optimization framework is integrated into a trackable data structure in the Julia language [25]. The gradient of the loss function against model parameters is obtained by the adjoint sensitivity method, which is accelerated by utilizing the Jacobian function from automatic differentiation. Specifically, the ODEs are written as neural networks and then integrated by stiff ODE solvers, and the kinetics parameters are then constrained by the loss function. Meanwhile, the neural network is still interpretable, which facilitates providing physical insights and integration of the kinetic model into large-scale turbulent combustion simulations.

This work is the first comprehensive study that introduces Neural ODE to optimize kinetics parameters of practical chemical models. The optimizations of hydrocarbon kinetics of various complexity with different data types, noise levels and mutable parameters are systematically tested. The subsequent contents are arranged as follows. In Section 2, the algorithm is detailed and the training progress is briefly introduced, with a toy problem Robertson ODE tested against different noise levels. In Section 3, a JP-10 skeletal mechanism is used to demonstrate the optimization ability of the proposed method and to show the accuracy against noise levels in the practical chemical system. Furthermore, an n-heptane skeletal mechanism is generated and optimized against its parent mechanism under the constraint of ignition delay times, with different mutable parameters. The relationship between the reverse UQ and the Bayesian Neural ODE is briefly discussed in Section 4. Conclusions are presented in Section 5.

2. Methodology

2.1 The neural network architecture

Consider a typical dynamics system described by an ordinary differential equation (ODE):

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{f}(\boldsymbol{u}, \boldsymbol{\theta}, t) \tag{1}$$

where $\boldsymbol{u}$ of size n is the state variable vector, $\boldsymbol{\theta}$ of size m the model parameters that control the dynamics and t the time. Given the initial state, this system can be integrated by an ODE solver

$$\boldsymbol{u}(t_1) = \boldsymbol{u}(t_0) + \int_{t_0}^{t_1} \boldsymbol{f}(\boldsymbol{u}, \boldsymbol{\theta}, t)dt \tag{2}$$

For simple non-stiff ODE systems such as the Lotka-Volterra model for predator-prey population dynamics [24], one can use explicit ODE solvers, e.g., Runge-Kuta methods. However, the chemical kinetics of hydrocarbon fuels generally involve a wide range of chemical timescales, and the resulting ODEs are highly stiff. The largest and smallest eigenvalues of the system's Jacobian matrix can differ by many orders of magnitude. In this study, the implicit second-order backward difference formula with trapezoidal rule (TRBDF2) is employed for integrating stiff chemistry assisted by the interface of the Arrhenius.jl package [29].

In the following, a simple reaction system involving three species [A, B, C] is used to illustrate the network structure in neural ODEs

$$\begin{aligned} A &\overset{k_1}{\rightarrow} B \\ B + B &\overset{k_2}{\rightarrow} C + B \\ B + C &\overset{k_3}{\rightarrow} A + C \end{aligned} \tag{3}$$

Wherek_1, k_2, k_3 are the rate constants of each reaction. Without loss of generality, all the reactions can be written in the form of

$$v_{i,A}^f A + v_{i,B}^f B + v_{C,i}^f C \overset{k_i}{\rightarrow} v_{i,A}^r A + v_{i,B}^r B + v_{i,C}^r C. \tag{4}$$

The constants $v_{i,j}^f$, $v_{i,j}^r$ are forward and backward stoichiometric coefficients of species *j* in the *i*-th reaction. The reaction rate of the *i*-th elementary reaction is described by the power-law expression as

$$r_i = k_i [A]^{v_{i,A}^f} [B]^{v_{i,B}^f} [C]^{v_{i,C}^f}. \tag{5}$$

Viewing from the perspective of a neural network, this equation can be formed by a series of linear combinations and activation functions, i.e.,

$$r_i = \exp\left(\ln k_i + \nu_{i,A}^f \ln[A] + \nu_{i,B}^f \ln[B] + \nu_{i,C}^f \ln[C]\right), \tag{6}$$

in which the dependency of k_i on temperature is usually described by the three-parameters Arrhenius formula,

$$k_i = A_i T^{b_i} \exp\left(-\frac{Ea_i}{RT}\right). \tag{7}$$

where A_i is the pre-exponential factor, b_i is the temperature exponent, and Ea_i is the activation energy. And Eq. (7) can be rewritten as $\ln k_i = \ln A_i + b_i \ln T - \frac{Ea_i}{RT}$. Note that the reaction rate of species (say [A]) is a linear combination of the reaction rates of elementary reactions, i.e.,

$$\frac{d[A]}{dt} = \sum_i -\nu_{i,A}^f r_i + \nu_{i,A}^r r_i. \tag{8}$$

Therefore, the chemical reaction network can be formulated as a neural network that consists of multiple layers of activation functions and linear connections. As shown in Fig. 1, the ODEs of the simple reaction system are transformed to an ODE Layer in the Neural ODE's network. For the neural ODE framework, following ResNet [30], the integration process is achieved by adding the ODE Layer's input to its output, i.e., $\boldsymbol{u}_{i+1} = \boldsymbol{u}_i + \mathrm{ODELayer}(\boldsymbol{u}_i)$. Note that this neural ODE framework is different from the one proposed by Chen et al. [22, 31, 32]. In [22], the network also consists of ODE layers and is integrated by ODE solvers, but each ODE layer is still made of MLP layers constructed by fully connected neural networks, to approximate an unknown dynamical system. In this study, the ODE layer is however made of exactly the original chemical ODEs that are viewed, modeled, and optimized in the neural network perspective. Thus, the proposed framework is based on the knowledge that one has already known the controlling mechanism of the target system but with uncertain parameters to be fine-tuned. In other words, it can also be viewed as CRNN [21, 27] with all pathways fixed and all other unnecessary connections trimmed.

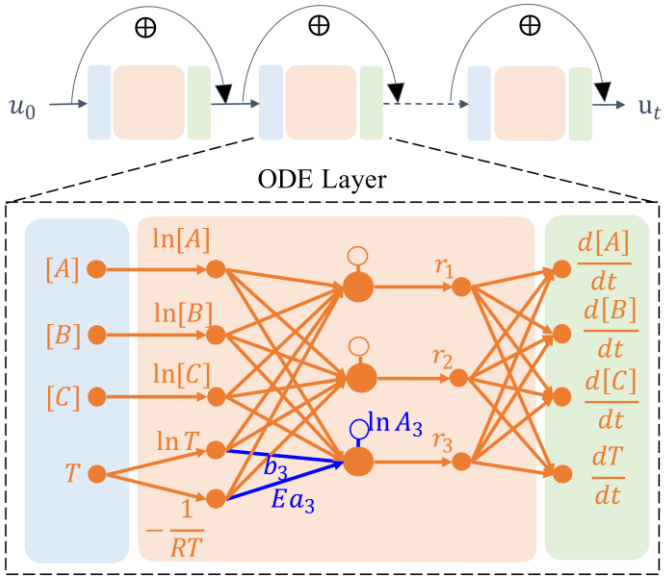

Fig. 1. Schematics of network architecture for neural ordinary differential equations.

Thus, the chemical reaction models are now transformed into neural networks and one can train the learnable parameters with datasets, as well as get the ODE system constrained on given observations. Under this implementation, one can optimize the kinetics parameters with SGD optimizers like Adam [33] once the gradient of the loss function is obtained.

2.2 Adjoint sensitivity method

There are generally two kinds of methods, i.e., forward sensitivity analysis and adjoint sensitivity analysis, to obtain the gradient of the loss function. For illustration, let us consider a loss function given by the integral up to the time point t_{end}:

$$\begin{aligned} &min\ L(\boldsymbol{u},\boldsymbol{\theta}) = \int_0^{t_{end}} g(\boldsymbol{u},\boldsymbol{\theta})\, dt \\ &s.t.\ \frac{d\boldsymbol{u}}{dt} = \boldsymbol{f}(\boldsymbol{u},\boldsymbol{\theta},t) \end{aligned} \tag{9}$$

To optimize this system, the gradient of loss function against the parameters, $\frac{dL}{d\boldsymbol{\theta}}$ is needed. And it can be obtained by

$$\frac{dL}{d\boldsymbol{\theta}} = \int_0^{t_{end}} \frac{dg}{d\boldsymbol{\theta}}\, dt = \int_0^{t_{end}} \left(\frac{\partial g}{\partial \boldsymbol{u}}\frac{d\boldsymbol{u}}{d\boldsymbol{\theta}} + \frac{\partial g}{\partial \boldsymbol{\theta}}\right) dt. \tag{10}$$

The terms $\frac{\partial g}{\partial u}$ of size $1 \times n$ and $\frac{\partial g}{\partial \theta}$ of size $1 \times m$ can be acquired by automatic differentiation (AD) techniques, so the only unknown is the first-order sensitivity coefficient matrix $\boldsymbol{w} \equiv \frac{d\boldsymbol{u}}{d\boldsymbol{\theta}}$ of size $n \times m$. Differentiating Eq. (1) against the parameters $\boldsymbol{\theta}$ leads to

$$\frac{d}{d\boldsymbol{\theta}}\left(\frac{d\boldsymbol{u}}{dt}\right) = \frac{d}{dt}\left(\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\theta}}\right) = \frac{d\boldsymbol{f}}{d\boldsymbol{\theta}} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\frac{d\boldsymbol{u}}{d\boldsymbol{\theta}} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}}, \tag{11}$$

where $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}$ of size $n \times n$ and $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}}$ of size $n \times m$ are the Jacobian matrices that can be acquired by AD techniques. Thus, one can get the governing equation for $\boldsymbol{w}$ as

$$\frac{d\boldsymbol{w}}{dt} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\boldsymbol{w} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}}. \tag{12}$$

In forward sensitivity analysis, one can integrate Eqs. (1) and (12) together and the final gradient of the loss function is acquired by Eq. (10).

However, the forward sensitivity analysis for a system with n state variables and m parameters requires solving a set of ODEs of size $n(m+1)$. Forward sensitivity analysis hence scales linearly in the number of parameters and in the number of state variables, which is computationally demanding for large n and m [34]. In the context of chemical kinetics optimizations, the state variables $\boldsymbol{u}$ consist of the temperature and mass fractions, while the parameters $\boldsymbol{\theta}$ are the Arrhenius parameters in kinetics mechanisms. The number of state variables n and the number of parameters m can be up to 10^3 and up to 10^4, respectively.

To reduce the computational cost and memory requirements, the adjoint sensitivity method [34-36] is instead employed by Chen et al. [22]. For the sake of better understanding, the derivations of the adjoint sensitivity method are described with the Lagrangian multiplier function, which is defined as

$$\mathcal{L}(\boldsymbol{\theta}) = \int_0^{t_{end}} g(\boldsymbol{u}, \boldsymbol{\theta}) dt + \int_0^{t_{end}} \boldsymbol{\lambda}^T \left(\frac{d\boldsymbol{u}}{dt} - \boldsymbol{f}(\boldsymbol{u}, \boldsymbol{\theta}, t)\right) dt, \tag{13}$$

where $\boldsymbol{\lambda}$ is of size $n \times 1$. One can optimize the original loss function $L(\boldsymbol{u}, \boldsymbol{\theta})$ along with the dynamics trajectory that satisfies $\frac{d\boldsymbol{u}}{dt} - \boldsymbol{f}(\boldsymbol{u}, \boldsymbol{\theta}, t) = \boldsymbol{0}$, where $\frac{d\mathcal{L}}{d\boldsymbol{\theta}} = \frac{dL}{d\boldsymbol{\theta}}$. Differentiating Eq. (13) leads to

$$\frac{dL}{d\boldsymbol{\theta}} = \int_0^{t_{end}} \left(\frac{\partial g}{\partial \boldsymbol{u}}\frac{d\boldsymbol{u}}{d\boldsymbol{\theta}} + \frac{\partial g}{\partial \boldsymbol{\theta}}\right) dt + \int_0^{t_{end}} \boldsymbol{\lambda}^T \left(\frac{d}{d\boldsymbol{\theta}}\left(\frac{d\boldsymbol{u}}{dt}\right) - \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\frac{d\boldsymbol{u}}{d\boldsymbol{\theta}} - \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}}\right) dt, \tag{14}$$

where the term involving in $\frac{d}{d\boldsymbol{\theta}}\left(\frac{d\boldsymbol{u}}{dt}\right)$ can be rewritten using integration by parts as

$$\int_0^{t_{end}} \boldsymbol{\lambda}^T \frac{d}{d\boldsymbol{\theta}}\left(\frac{d\boldsymbol{u}}{dt}\right) dt = \boldsymbol{\lambda}^T \frac{d\boldsymbol{u}}{d\boldsymbol{\theta}}\Big|_0^{t_{end}} - \int_0^{t_{end}} \frac{d\boldsymbol{\lambda}^T}{dt}\frac{d\boldsymbol{u}}{d\boldsymbol{\theta}} dt. \tag{15}$$

Substituting this into Eq. (14) and rearranging to group together terms involving the sensitivity matrix $\frac{d\boldsymbol{u}}{d\boldsymbol{\theta}}$ gives

$$\frac{dL}{d\boldsymbol{\theta}} = \int_0^{t_{end}} \left(\frac{\partial g}{\partial \boldsymbol{u}} - \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}} - \frac{d\boldsymbol{\lambda}^T}{dt}\right)\frac{d\boldsymbol{u}}{d\boldsymbol{\theta}} dt + \int_0^{t_{end}} \left(\frac{\partial g}{\partial \boldsymbol{\theta}} - \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}}\right) dt + \boldsymbol{\lambda}^T \frac{d\boldsymbol{u}}{d\boldsymbol{\theta}}\Big|_0^{t_{end}}. \tag{16}$$

The adjoint sensitivity vector $\boldsymbol{\lambda}^T$ of size $1 \times n$ can be used to eliminate the first term involving the sensitivity matrix, which requires

$$\frac{d\boldsymbol{\lambda}^T}{dt} = \frac{\partial g}{\partial \boldsymbol{u}} - \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}. \tag{17}$$

In this study, the initial conditions do not depend on parameters $\boldsymbol{\theta}$, thus $\frac{d\boldsymbol{u}}{d\boldsymbol{\theta}}\Big|_{t=0} = \mathbf{0}$. And to eliminate the last term in Eq. (16), one can let $\boldsymbol{\lambda}^T|_{t=t_{end}} = \mathbf{0}$. Therefore, the gradient of the loss function is

$$\frac{dL}{d\boldsymbol{\theta}} = \int_0^{t_{end}} \left(\frac{\partial g}{\partial \boldsymbol{\theta}} - \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}}\right) dt. \tag{18}$$

To reduce time complexity, one can also backward integrate $\frac{\partial g}{\partial \boldsymbol{\theta}} - \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}}$ and combine it with the backward integration of $\boldsymbol{\lambda}^T$. Therefore, one can conduct the augmented dynamics with $\boldsymbol{\lambda}_{aug}^T \equiv [\boldsymbol{\lambda}^T, \boldsymbol{\mu}^T]$ of size $1 \times (n+m)$, $\boldsymbol{\mu}^T|_{t=t_{end}} \equiv \frac{\partial g}{\partial \boldsymbol{\theta}}\Big|_{t=t_{end}}$, $\boldsymbol{\lambda}^T|_{t=t_{end}} \equiv \mathbf{0}$ and with $\boldsymbol{u}_{aug} \equiv [\boldsymbol{u}, \boldsymbol{\theta}]$ of size $(n+m)$. The governing equation for $\boldsymbol{\lambda}_{aug}^T$ is

$$\frac{d\boldsymbol{\lambda}_{aug}^T}{dt} = \frac{\partial g}{\partial \boldsymbol{u}_{aug}} - \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}_{aug}}, \qquad \boldsymbol{\lambda}_{aug}^T\Big|_{t=t_{end}} = \left[\mathbf{0}, \frac{\partial g}{\partial \boldsymbol{\theta}}\Big|_{t=t_{end}}\right], \tag{19}$$

where $\frac{\partial g}{\partial \boldsymbol{u}_{aug}} \equiv \left[\left(\frac{\partial g}{\partial \boldsymbol{u}}\right), \left(\frac{\partial g}{\partial \boldsymbol{\theta}}\right)\right]$ is of size $1 \times (n+m)$ and $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}_{aug}} \equiv \left[\left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right), \left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}}\right)\right]$ is of size $n \times (n+m)$.

By backward integrating the augmented ODEs Eq. (19), one can obtain the full gradients of the given loss function with respect to the training datasets, which is $\frac{dL}{d\boldsymbol{\theta}} = -\boldsymbol{\mu}|_{t=0} + \left.\frac{\partial g}{\partial \boldsymbol{\theta}}\right|_{t=t_0}$. The detailed steps to compute the gradient of the loss function are shown in Algorithm 1. Thus, the adjoint sensitivity analysis solves a set of ODEs of size $(2n+m)$ and is more computationally effective than the $n(m+1)$ ones solved in the forward sensitivity analysis. Readers can refer to [22, 34, 35] for more details.

Algorithm 1 Gradient of an ODE problem using adjoint method

Input: dynamics parameters $\boldsymbol{\theta}$, initial state $\boldsymbol{u}_0$, time range $[0, t_{end}]$, functions $\boldsymbol{f}(\cdot)$ and $g(\cdot)$

1. Integrate $\frac{d\boldsymbol{u}}{dt} = \boldsymbol{f}(\boldsymbol{u}, \boldsymbol{\theta}, t)$ for $\boldsymbol{u}$ from $t = 0$ to t_{end} with initial condition $\boldsymbol{u}|_{t=0} = \boldsymbol{u}_0$.
2. Integrate $\frac{d\lambda_{aug}^T}{dt} = \frac{\partial g}{\partial \boldsymbol{u}_{aug}} - \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}_{aug}}$ for $\boldsymbol{\lambda}_{aug}^T \equiv [\boldsymbol{\lambda}^T, \boldsymbol{\mu}^T]$ from $t = t_{end}$ to 0 with initial condition $\boldsymbol{\lambda}_{aug}^T\big|_{t=t_{end}} = \left[\mathbf{0}, \left.\frac{\partial g}{\partial \boldsymbol{\theta}}\right|_{t=t_{end}}\right]$.
3. Compute the gradient $\frac{dL}{d\boldsymbol{\theta}} = -\boldsymbol{\mu}|_{t=0} + \left.\frac{\partial g}{\partial \boldsymbol{\theta}}\right|_{t=t_0}$.

Output: $\frac{dL}{d\boldsymbol{\theta}}$

Note: Jacobian functions, i.e., $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}, \frac{\partial g}{\partial \boldsymbol{u}}, \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}}, \frac{\partial g}{\partial \boldsymbol{\theta}}$, can be acquired via automatic differentiation.

For time-discrete measurements, the loss function is not continuous and one needs to re-initialize $\boldsymbol{\lambda}$ at each time point of the measurements. Taking the Lotka-Volterra ODEs, also known as the predator-prey equations,

$$\frac{d\boldsymbol{u}}{dt} = \frac{d}{dt}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \alpha u_1 - \beta u_1 u_2 \\ \delta u_1 u_2 - \gamma u_2 \end{bmatrix}, \tag{20}$$

as an example, where $u_1(t)$ and $u_2(t)$ denote the populations of prey and predator at time t, respectively. By default, the parameters are $= 1.5$, $\beta = 1$, $\delta = 3$, $\gamma = 1$ and initial conditions are $u_1 = 1$, $u_2 = 1$. If given measurements of six time points, one uses the loss function of

$$L(\boldsymbol{u}, \boldsymbol{\theta}) = \sum_{k=0}^{5} \left\| \frac{\boldsymbol{u}_{t_k} - \boldsymbol{u}_{t_k}^{obs}}{\boldsymbol{u}_{scale}} \right\|^2 + \alpha \boldsymbol{\theta}^T \boldsymbol{\theta}, \tag{21}$$

where $\boldsymbol{u}_{t_k}^{obs}$ is the observed data at time t_k, $\boldsymbol{u}_{scale}$ is a scaling vector to balance the contribution of u_1 and u_2, α is a regulation term to prohibit unnecessary parameter changes. During the backward integration of adjoint state variable $\boldsymbol{\lambda}(t)$, it should be re-initialized at each t_k by

$$\begin{aligned} \boldsymbol{\lambda}(t_k) &= \lim_{t \to t_k^+} \boldsymbol{\lambda}(t) + \nabla_{\boldsymbol{u}} \left(\left\| \frac{\boldsymbol{u}_{t_k} - \boldsymbol{u}_{t_k}^{obs}}{\boldsymbol{u}_{scale}} \right\|^2 \right) \\ &= \lim_{t \to t_k^+} \boldsymbol{\lambda}(t) + 2(\boldsymbol{u}_{t_k} - \boldsymbol{u}_{t_k}^{obs}) / \boldsymbol{u}_{scale}^2. \end{aligned} \tag{22}$$

As shown in the top panel of Fig. 2, the forward integration will get the state variables of the ODEs, and there are discrepancies between the state variables and the data represented by symbols. Backward integration of the augmented dynamics results in the adjoint state shown in the bottom panel of Fig. 2. When multiple time points are used to constrain a given system, the backward integration takes the errors at each time point into account, as demonstrated by the step change of adjoint state at corresponding time points. Readers can refer to [22, 24] for more details.

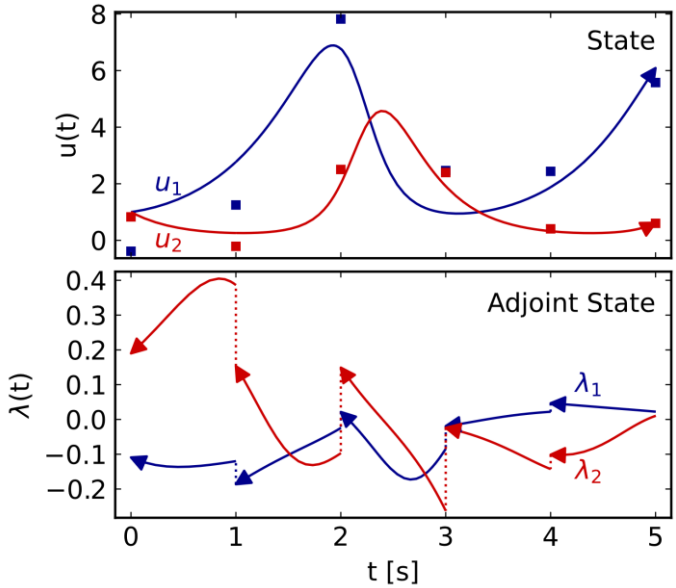

Fig. 2. Demonstration of forward and backward integration of the Lotka-Volterra ODEs and the augmented dynamics. The top panel shows the state variables $\boldsymbol{u}(t)$ obtained by forward integration, with symbols being the observed data. The bottom panel shows the adjoint state variables $\boldsymbol{\lambda}(t)$ obtained by backward integration of the augmented dynamics.

2.3 Kinetics parameter optimization

In general, traditional gradient evaluations for transient solutions involving stiff chemical kinetic models are extremely expensive, as the computation time of forward sensitivities usually scales with both the number of species and the number of parameters. Compared with the numerical Jacobian, the analytical Jacobian can be acquired with lower cost and higher accuracy by differentiable programming. Thus, once combined with the adjoint sensitivity method, the gradients of the loss function can be evaluated efficiently and hence can be used in stochastic gradient descent (SGD) methods to optimize kinetics parameters. The stiff ODE solvers enable the ability to handle stiff chemical models, while the SGD methods leverage the ability to extract information from noisy experimental data. Note that recently the newest released version of Cantera has also enabled the capability of analytical Jacobian. So, one can also implement the proposed method in Cantera, but all the other necessary gradient evaluations and adjoint sensitivities integration should be manually added.

By integrating the adjoint sensitivity, one can easily get the loss gradients and optimize the kinetics parameters of any chemical model under various types of datasets. As for biomedical processes or chemical pyrolysis processes, time-related species profiles are generally measured or estimated in experiments and models are trained to fit the experimental observations. In combustion scenarios, shock tube oxidation data plays a similar role, with much more available data being indirect characteristics of given models, such as ignition delay times (IDTs) or laminar flame speeds.

To demonstrate the proposed framework's capability and robustness of optimizing kinetics parameters under different scenarios, a classical model problem, the stiff Robertson ODE [37], is first tested. The Robertson ODE shares the same dynamics with Eq. (3) but does not involve temperature-dependent or pressure-dependent rate constants. The generally used rate constants $\boldsymbol{k}^{true} = [0.04, 3 \times 10^7, 1 \times 10^4]$ are adopted to generate the ground truth data with the initial value $\boldsymbol{y}_0 = [1, 0, 0]$. All the simulations and optimizations for the Robertson ODE are under the default tolerances, i.e., $atol = 10^{-6}$ for the absolute tolerance and $rtol = 10^{-3}$ for the relative tolerance. The data samples are drawn from 50 uniform time instants in the log time scale of range $[10^{-4}, 10^8]$ s.

Randomly perturbs the kinetics parameters by sampling from $\log\left(\frac{\boldsymbol{k}^{init}}{\boldsymbol{k}^{true}}\right) \sim U[-1,1]$, which is equivalent to an uncertainty factor [5] of $UF_i = \frac{k_i^{max}}{k_i^{true}} = \frac{k_i^{true}}{k_i^{min}} \approx 2.7$ for all parameters, and leads to $\boldsymbol{k}^{init} = [0.023, 1.88 \times 10^8, 1.96 \times 10^3]$. With 10% noise added to the state values of the datasets, one can then train the Robertson ODE with loss function constraining on the state variable $\boldsymbol{y}$. Based on the fact that chemical species concentration span across several orders of magnitude, the loss function utilizes a normalization via $\boldsymbol{y}_{scale}$, which is a vector in the same shape of $\boldsymbol{y}_0$ and each value describes the scale of each state variable, $L_{\boldsymbol{y}}(\boldsymbol{\theta}) = MSE\left(\frac{\boldsymbol{y}^{model} - \boldsymbol{y}^{obs}}{\boldsymbol{y}_{scale}}\right)$ where $\boldsymbol{y}^{model}$ is the prediction of neural networks, $\boldsymbol{y}^{obs}$ is the observed noisy data, $\boldsymbol{y}_{scale}$ is obtained by subtracting the min from the max of each state variable, and MSE represents the mean squared error function. The loss of parameters is defined as $L_{\boldsymbol{k}}(\boldsymbol{\theta}) = MSE\left(\frac{\boldsymbol{k}^{model} - \boldsymbol{k}^{true}}{\boldsymbol{k}^{true}}\right)$, where $\boldsymbol{k}^{model}$ is the current parameters of neural networks.

As shown in Fig. 3, the predictions with parameters $\boldsymbol{k}^{init}$ differ significantly from the ground truth data. For each epoch of the training progress, the gradients of the loss function are computed and fed to the Adam [33] optimizer with an initial learning rate of $lr = 0.1$ and default weight decay parameters of 0.9 for the 1st-order moment and 0.999 for the 2nd-order moment. After 200 epochs of training, the optimized ODE well predicts the data behavior and the optimized parameters $\boldsymbol{k}^{opt} = [0.041, 3.00 \times 10^7, 1.02 \times 10^4]$ is very close to the ground truth ones $\boldsymbol{k}^{true}$.

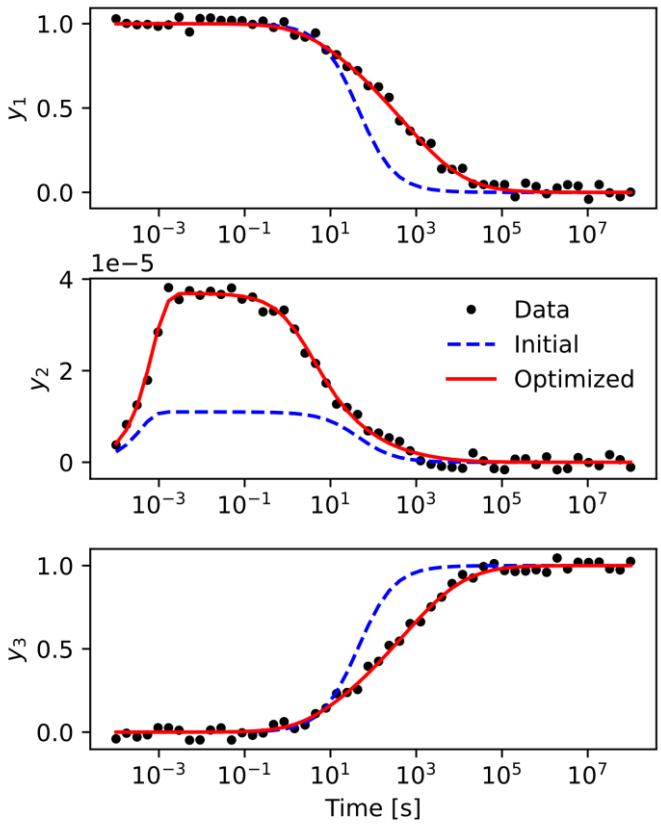

Fig. 3. Training results of Robertson ODE with 10% noise level.

To further examine the robustness of the Neural ODE method, another 100 training experiments are performed under noise levels between 0.1% to 20%. Each experiment uses random initial parameters generated with $UF \approx 2.7$ and a fixed 300 epochs training with the same setting is employed. The evolution of losses is shown in the left panel of Fig. 4, which guarantees each training process has converged to its optimum, To quantify the training results, normalized errors of $\boldsymbol{y}$ and $\boldsymbol{k}$ are defined as $y_{err} = \sqrt{\frac{1}{n \cdot d}\sum\left(\frac{\boldsymbol{y}^{opt}-\boldsymbol{y}^{obs}}{\boldsymbol{y}_{scale}}\right)^2}$ and $k_{err} = \sqrt{\frac{1}{m}\sum\left(\frac{\boldsymbol{k}^{opt}-\boldsymbol{k}^{true}}{\boldsymbol{k}^{true}}\right)^2}$, where $n = 3$ is the dimension of state variables, $d = 50$ is the number of data samples and $m = 3$ is the number of parameters. The results of y_{err} and k_{err} are shown in the right panel of Fig. 4. Intuitively, converged training leads to y_{err} being comparable to noise level so y_{err} collapses around the line $y = x$. The k_{err} results show that the learned kinetics parameters have a similar or smaller scale of error compared to the noise level, which suggests good learning ability and robustness of the Neural ODE method.

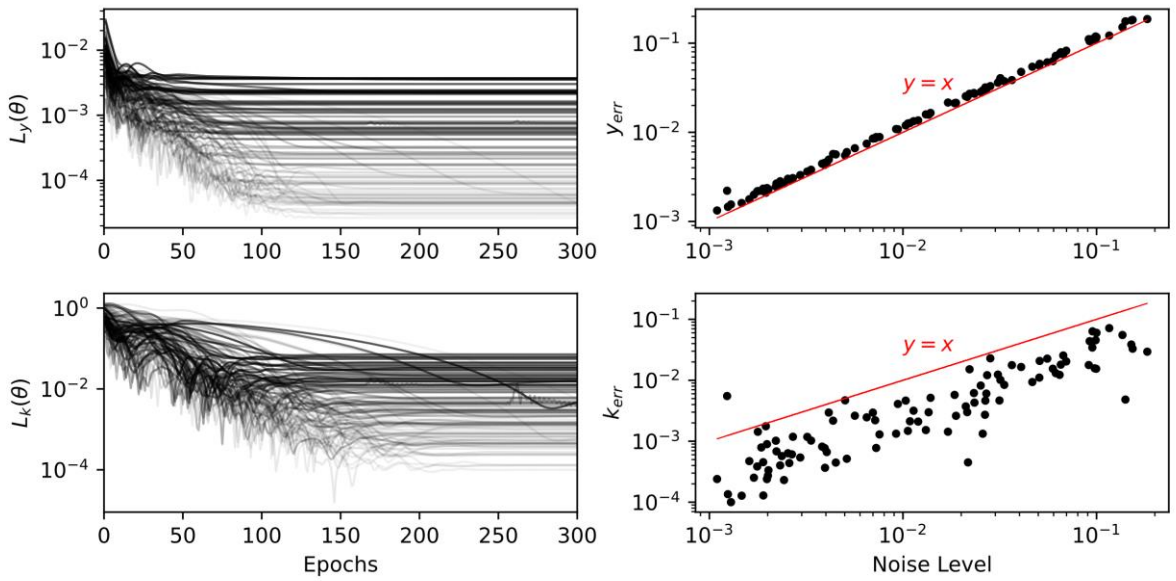

Fig. 4. The evolution of losses and final errors of Robertson ODE's training results against different noise levels.

Note that the tests against noise are performed with random initial parameters from fixed UF. With large uncertainty factors, it will be harder to converge to the optimum. One can take a series of UFs, and conduct the performance tests for each UF to validate the ability of convergency with different parameter uncertainties. However, it will relate to not only the studied dynamics system but also the hyper-parameters of the optimizer, which is beyond the scope of this paper.

3. Results

In this section, we demonstrate the capability of the Neural ODE approach in optimizing the kinetics parameters of a JP-10 skeletal mechanism with known species profiles and optimizing an over-reduced *n*-heptane mechanism with known ignition delay times from the detailed one.

3.1 Case 1: JP-10 pyrolysis

Jet propellant 10, or JP-10, is one of the leading high volumetric energy density fuel candidates for propulsion devices. Tao et al. [38] developed a 41-species, 232-reaction mechanism for JP-10 surrogate $C_{10}H_{16}$ by constraining this hybrid chemistry model with high temperature pyrolysis experimental data. Here by artificially perturbing the kinetics parameters, e.g., rate constants $\boldsymbol{k}$, the test aims to demonstrate whether Neural ODE can recover the original ones with the dataset of species profiles generated from the Tao mechanism. The datasets consist of 20

training sets with an initial temperature range of 1000-1200 K, an initial mass fraction of $C_{10}H_{16}$ in the range of 0.02-0.2; and another 5 validation sets with an initial temperature of 1200-1400 K and an initial $Y_{C_{10}H_{16}}$ in 0.2-0.3. The detailed initial conditions are shown in Table A1 in the Appendix. The true parameters $\boldsymbol{k}^{true}$ are the original Arrhenius parameters proposed by Tao et al. [38]. By defining $\boldsymbol{\theta} \equiv \log\left(\frac{\boldsymbol{k}}{\boldsymbol{k}^{true}}\right)$, the true kinetics parameters are hence represented by $\boldsymbol{\theta}^{true} = \boldsymbol{0}$. The datasets are obtained by solving the chemical ODEs with $\boldsymbol{k}^{true}$ and sampling 10 points uniformly in the range of 10^{-6} s – 10^{-1} s in the log scale.

The loss function for training has a MSE term and a regularization term as

$$L(\boldsymbol{y}, \boldsymbol{\theta}) = MSE\left(\frac{\boldsymbol{y}^{model} - \boldsymbol{y}^{obs}}{\boldsymbol{y}_{scale}}\right) + \alpha \boldsymbol{\theta}^T \boldsymbol{\theta} \tag{23}$$

where α is a small number, e.g., 1×10^{-4} to avoid unnecessary parameter changes. During the training process, the 20 training sets are randomly permutated and employed to compute gradients in each epoch. To accelerate the training process, larger ODE tolerances and fewer timesteps are used at the beginning of training and gradually adjusted to finer tolerances and timesteps.

The initial parameters are given randomly in the range of $[-1, 1]$, which means they can be around 2.7 times smaller or larger than the original rate constants k^{true} and $UF \approx 2.7$. The simulations are conducted with zero-dimensional, adiabatic, constant pressure reactors. The training process is divided into three steps to reduce the computational cost: (i) train 10 epochs with 50% data of each initial condition, with tolerances $atol = 10^{-9}$, $rtol = 10^{-6}$ and learning rate $lr = 0.05$; (ii) train 40 epochs with 100% data, $atol = 10^{-9}$, $rtol = 10^{-6}$ and $lr = 0.01$; (iii) train 50 epochs with 100% data, $atol = 10^{-12}$, $rtol = 10^{-9}$ and $lr = 0.01$. Note that these hyper-parameters are manually decided and are not optimal. Further investigation of hyper-parameters might need to be conducted for better performance on large kinetics mechanisms.

After 100 epochs of training, the parameters are well optimized to capture the evolution characteristics of data. As shown in Fig. 5, the initial parameters $\boldsymbol{\theta}_{init}$ result in significant errors in the pyrolysis process. After training, all the species correlate perfectly with the ground truth data. Note that the validation sets are not exposed to the training program, which means the algorithm learned the inner kinetics from training sets and it applies to other thermochemical conditions.

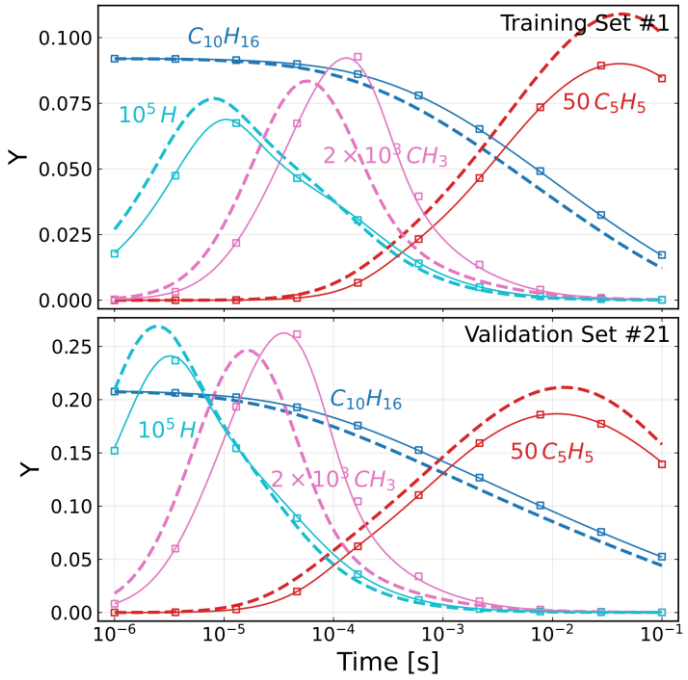

Fig. 5. Training results of JP-10 pyrolysis with temperature and all profiles as training data. Squares: Data, dash lines: Initial, solid lines: Optimized.

One can further check if the parameters are trained to the true kinetics parameters used in datasets. As shown in Fig. 6, the ground truth $\boldsymbol{\theta}^{true} = \mathbf{0}$ for every elementary reaction, and the initial parameters $\boldsymbol{\theta}^{init}$ randomly distribute in the range $[-1, 1]$. In contrast, for the sensitive reactions, most optimized parameters $\boldsymbol{\theta}^{opt}$ collapse to the x-axis, which means they are optimized toward the ground truth ones. This further confirms that the optimization algorithm extracts the inner kinetics behavior from datasets and gets the parameters optimized to the real ones effectively.

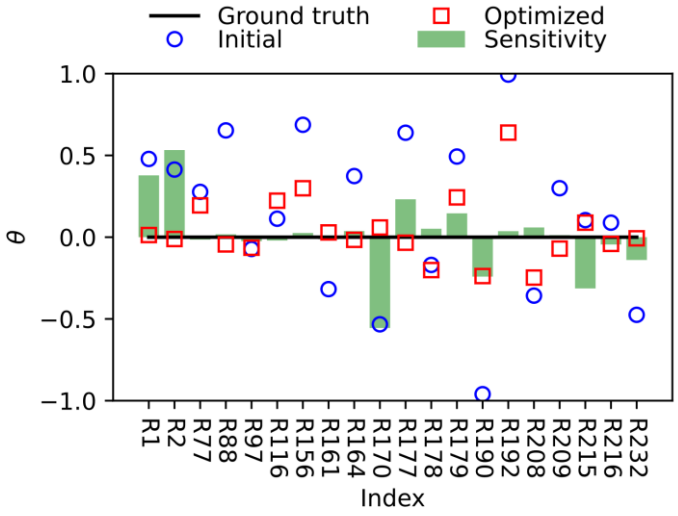

Fig. 6. The initial and optimized kinetics parameters with the ground truth being all zero. Only the most sensitive 20 reactions under initial condition #1 in Table A1 are plotted.

Figure 7 further demonstrates the capability of the optimization algorithm by showing that it can well reproduce the pyrolysis process of all species even if only the profiles of a small subset of species are available for training. Note that the information of species C_5H_5, CH_3 and H are not available, but their evolutions are well reproduced after the kinetics parameters are optimized. The demonstration reveals that it is possible to learn the kinetics parameters from partially observed species profiles. For instance, one can employ the approach on synthesized data to identify the most influential measurements on the kinetics parameter optimization and guide the design of experiments.

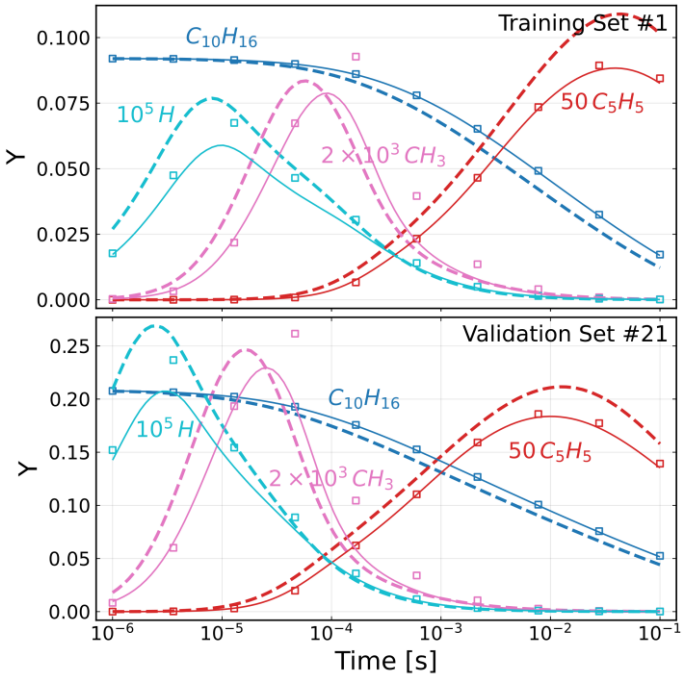

Fig. 7. Training results of JP-10 pyrolysis with only the profiles of CH_4, C_2H_4 as training data. Squares: Data, dash lines: Initial, solid lines: Optimized.

The accuracy of measurements will definitely influence the accuracy of model prediction. Therefore, various noise levels are tested and compared in the JP-10 cases, with either all species profiles or only the profiles of CH_4, C_2H_4 as the training data. For example, with 5% noise added to all species profiles, after the training 100 epochs mentioned before, one can get a set of optimized parameters. With that optimized parameters, the predictions under all initial conditions

listed in Table A1 are computed and compared to the ground truth ones without noise, and then the average error under each initial condition can be obtained and statistically counted for its PDF among 25 initial conditions.

As shown in Fig. 8, for two kinds of training data, the errors of the final predictions get gradually increased with increasing noise levels. When all data profiles are employed, the errors under the same noise levels are generally smaller than that trained with CH_4 and C_2H_4. It is worth noting that with only the data profiles of CH_4 and C_2H_4 being used, noises smaller than 5% lead to only limited performance of final errors. However, for the one trained with all data profiles, there are still improvements with the noise level getting down to 1%. It suggests that when optimizing this chemical system with limited data, adding data abundance may perform better than pursuing ultimate data accuracy.

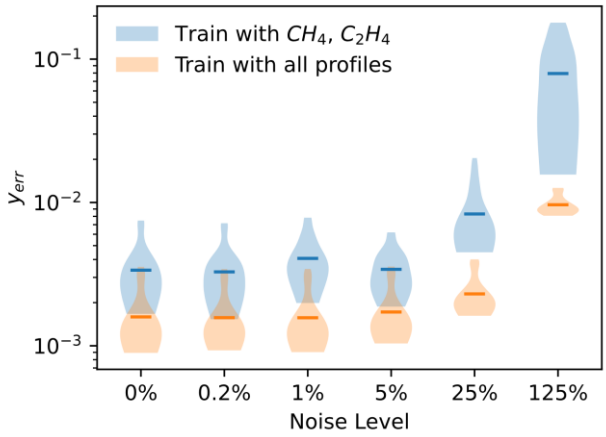

Fig. 8. Final errors of the JP-10 pyrolysis case with different training data under different noise levels. The width of each violin-shaped plot shows the probability of the corresponding y_{err} among the 25 datasets shown in Table A1 and the bars are the corresponding means of y_{err}.

3.2 Case 2: n-heptane autoignition

Most skeletal chemical models are obtained by removing unimportant species and the associated reactions from the master model, leaving the kinetics parameters of the remaining pathways unchanged after the reduction. There has been increasing interest in optimizing the kinetics parameters in overly reduced reaction models to compensate for the error introduced by over-reduction [12, 28]. Thus, one can obtain a smaller model with higher fidelity than those acquired via traditional reduction methods.

Here a widely used 41-species, 168-reaction n-heptane mechanism developed by Nordin et al. [39] is employed to demonstrate the ability of neural ODE for optimizing reduced models. The species C_3H_5, C_3H_4, C_2H_6, CH_4O_2, CH_3O_2, and C_2H_2 are removed via an iterative reduction process involving direct relation graph (DRG) [40], sensitivity analysis (SA) [41] and directioni relation graph with error propagation (DRGEP) [42]. The overly reduced model is denoted as SK34 with 34 species and 121 reactions. It is noted that one can also optimize an existing empirical semi-global reaction model against a detailed model without consulting skeletal reduction.

Ignition delay times (IDTs) are used as the single target characteristics to optimize the skeletal mechanism SK34. Note that the parameter studies [43, 44] suggested that there is strong correlation between the temperature exponent b, the activation energy E_a and the pre-exponential factor A; thus, adding b, E_a to be mutable parameters will not help much on the final accuracy. Therefore, usually only A is optimized in many related works. In this study, the optimizations are conducted either with only A being optimized or with all the Arrhenius parameters being mutable to validate the efficiency and effectiveness of the proposed method. The mechanism optimized with only A and the one optimized with A, b, E_a are hereby denoted as SK34OPa and SK34OPb, respectively.

The training sets τ^{obs} are IDTs under 500 thermochemical conditions sampled in the range of 700-1600 K for initial temperature T_0, 1-50 atm for pressure P, and 0.5-2.0 for the equivalence ratio ϕ. Splitting the datasets with 80% as the training set and 20% as the validation set, one can optimize the problem with the training set and check the validity and reliability with the validation set. The loss function adopted is also an MSE term accompanied with a regularization term:

$$L(\boldsymbol{\theta}) = \left(\log\frac{\tau}{\tau^{obs}}\right)^2 + \alpha\boldsymbol{\theta}^T\boldsymbol{\theta}, \tag{24}$$

where $\alpha = 1 \times 10^{-4}$. The gradient of $L(\boldsymbol{\theta})$ is

$$\frac{dL}{d\boldsymbol{\theta}} = \frac{\partial L}{\partial \tau}\frac{d\tau}{d\boldsymbol{\theta}} + \frac{\partial L}{\partial \boldsymbol{\theta}} = 2\left(\log\frac{\tau}{\tau^{obs}}\right)\frac{d\tau}{dT}\frac{dT}{d\boldsymbol{\theta}} + 2\alpha\boldsymbol{\theta}, \tag{25}$$

where $\frac{dT}{d\boldsymbol{\theta}}$ are obtained via the adjoint sensitivity method and $\frac{d\tau}{dT}$ are obtained after the integration of the ODE solver (by numerical gradient evaluation from temperature profile).

The ignition delay times are simulated by zero-dimensional, adiabatic, const pressure reactors. All the simulations share the same tolerances $atol = 10^{-12}$ and $rtol = 10^{-9}$. In each

epoch, a batch of 40 samples are randomly chosen from the training set and the gradients are fed to the Adam [33] optimizer with an initial learning rate of 2×10^{-3} and default weight decay parameters of 0.9 for the 1st-order moment and 0.999 for the 2nd-order moment.

After 100 epochs training, as shown in Fig. 9, the skeletal mechanism is optimized and the predictions of the optimized ones on IDTs match well with the original Nordin mechanism. By averaging over all the tested thermochemical conditions, i.e., T_0=[650, 700, 750, 800, 850, 900, 950, 1000, 1100, 1200, 1300, 1400, 1600] K, $P = [10, 30, 60]$ atm and $\phi = [0.5, 1.5, 2.0]$, one can compute the mean relative error of IDT as $\tau^{err} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\tau_i - \tau_i^{obs}}{\tau_i^{obs}}\right|$, where N is the number of tested thermochemical conditions and $|\cdot|$ is the absolute function. The mean relative errors for SK34OPa and SK34OPb are 3.5% and 2.2%, respectively, which implies that the optimization with A, b, E_a leads to a little higher accuracy than the one with only A.

As for the computational efficiency, the entire optimization process with 363 mutable parameters is about 6 times expensive than the one with 121 pre-exponential factors, and it takes about 10 CPU hours on a normal PC. For reference, the optimization of an overly reduced Jet A kinetic model with 30 mutable parameters among 254 reactions using evolutionary algorithms has to rely on high-performance clusters and costs around hundreds of CPU hours [12]. The MUM-PCE method [7] can provide a comparable time cost with around 30 mutable parameters, but if all parameters are taken into account, it will be unaffordable to build the response surface.

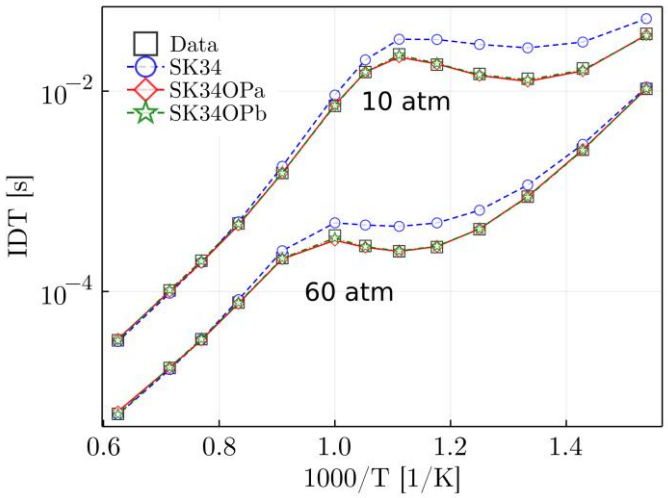

Fig. 9. Prediction on IDTs of Nordin (Data) with its skeletal (SK34) and the optimized mechanisms with A (SK34OPa) and with A, b, E_a(SK34OPb).

Figure 10 further validates the evolution of species and temperature during the autoignition process. As shown in Fig. 10, SK34OPa and SK34OPb both perform better than SK34, while SK34OPb performs even better for the high-temperature pathway. Generally, the optimized skeletal mechanisms, especially SK34OPb, can not only well predict IDTs and temperature evolution, but also preserve the physical interpretability of the original mechanism, which leads to correct predictions of minor species evolution. Similar observations can be made for other species.

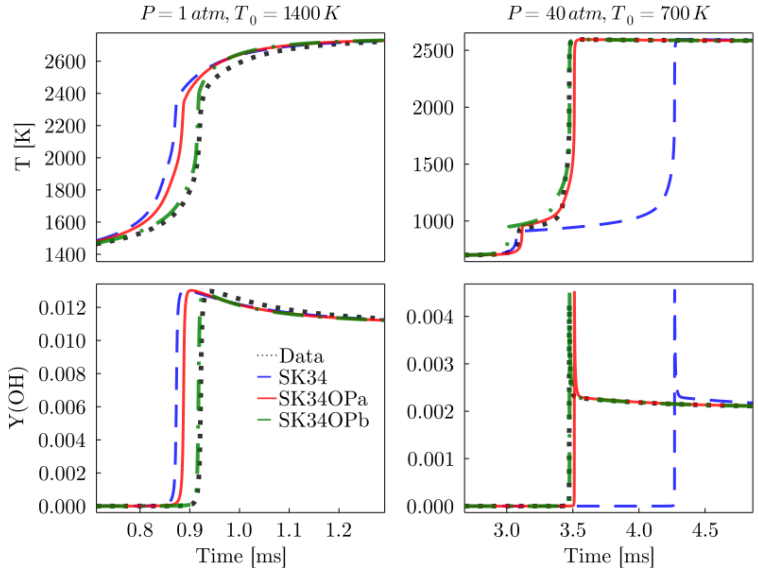

Fig. 10. Profiles of temperature and species OH during autoignition, where the left panels show high-temperature pathway and the right panels for the low-temperature pathway. The lines are for the Nordin (Data) with its skeletal (SK34) and the optimized mechanisms with A (SK34OPa) and with A, b, E_a (SK34OPb).

4. Discussions

Kinetics parameter optimizations with experimental data and theoretical calculations as constraints are generally widely used in the development of combustion chemical models [2, 5]. In addition to point estimation of model parameters, the reverse UQ strategies have been advocated to constrain the uncertainty space of the input parameters, i.e., to reconstruct the probability distribution of input parameters via given constraints. Li and Frenklach [45, 46] developed the bound-to-bound data collaboration (B2BDC) method with source code released. Sheen [7, 10] and Wang [5] proposed the method of uncertainty minimization by a polynomial chaos expansion (MUM-PCE). Cai and Pitsch [47, 48] adopted the MUM-PCE and optimized large alkane

mechanisms. Tao and Wang [38, 43] investigated the uncertainty and sensitivity of all the Arrhenius parameters of the FFCM-1 chemistry model.

As mentioned above, by using the reverse UQ methods, the uncertainties of model predictions are estimated and minimized against the given dataset. The inherent nature behind is Bayesian regression that treats the model parameters as random variables and estimates the posterior distributions from prior distributions and the ground truth distribution of the QoI [49]. The reverse UQ methods utilize the response surfaces to map the input parameters to the QoI, which facilitates the sampling process of the Markov chain Monte Carlo (MCMC) approach. Wang et al. [9] introduced artificial neural networks (ANN) as the response surface and combined it with the Bayesian analysis to estimate the posterior of kinetics parameters.

In neural network scenarios, the reverse UQ is often termed as variational inference (VI) and has been studied by many researchers [50, 51]. The universal approximation ability and automatic differential platform empower the neural networks to be a promising method to estimate posterior from limited datasets. Recently, Dandekar and Rackauckas [52] proposed the Bayesian Neural ODE with stochastic gradient Hamiltonian Monte Carlo (SGHMC) sampler and illustrated the ability to forecast the dynamics against ground truth data. Besides, Xu et al. [53] utilized the Bayesian neural networks with stochastic differential equation (SDE) and trained it with the evidence lower bound (ELBO) loss function.

The basic idea of Bayesian inference comes from the Bayes' theorem

$$p(\boldsymbol{\theta}|\boldsymbol{\mathcal{D}}) = \frac{p(\boldsymbol{\theta})p(\boldsymbol{\mathcal{D}}|\boldsymbol{\theta})}{p(\boldsymbol{\mathcal{D}})}, \tag{26}$$

where $p(\boldsymbol{\theta}|\boldsymbol{\mathcal{D}})$ is the posterior probability of the parameters $\boldsymbol{\theta}$ with given data $\boldsymbol{\mathcal{D}}$; $p(\boldsymbol{\theta})$ is the prior distribution of parameters $\boldsymbol{\theta}$, representing the initial beliefs about the parameters before observing the data; $p(\boldsymbol{\mathcal{D}}|\boldsymbol{\theta})$ is the likelihood that describes the probability of $\boldsymbol{\mathcal{D}}$ with given parameters $\boldsymbol{\theta}$; $p(\boldsymbol{\mathcal{D}})$ the marginal likelihood or evidence, which is the normalizing constant that ensures the posterior is a proper probability distribution. Since $p(\boldsymbol{\mathcal{D}})$ is constant, $p(\boldsymbol{\theta}|\boldsymbol{\mathcal{D}}) \propto p(\boldsymbol{\theta})p(\boldsymbol{\mathcal{D}}|\boldsymbol{\theta})$, therefore log posterior

$$\log p(\boldsymbol{\theta}|\boldsymbol{\mathcal{D}}) = \log p(\boldsymbol{\mathcal{D}}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}), \tag{27}$$

is often used as the optimization target and once the optimization is complete, one needs to normalize $p(\boldsymbol{\theta}|\boldsymbol{\mathcal{D}})$. The log likelihood $\log p(\boldsymbol{\mathcal{D}}|\boldsymbol{\theta})$ is usually defined as the negative sum of the

residuals between the observed data and the model predictions, weighted by the observational noise:

$$\log p(\boldsymbol{\mathcal{D}}|\boldsymbol{\theta}) = -\frac{1}{2}\sum\left(\frac{\boldsymbol{\mu}^{obs} - \boldsymbol{u}^{pred}}{\boldsymbol{\sigma}^{obs}}\right)^2, \tag{28}$$

where $\boldsymbol{u}^{pred}$ is the model prediction with given $\boldsymbol{\theta}$, $\boldsymbol{\mu}^{obs}$ is the observed mean and $\boldsymbol{\sigma}^{obs}$ is the observed noise or standard derivation. The gradient of $\log p(\boldsymbol{\theta}|\boldsymbol{\mathcal{D}})$ can be calculated by differentiating the log likelihood $\log p(\boldsymbol{\mathcal{D}}|\boldsymbol{\theta})$ and log prior $\log p(\boldsymbol{\theta})$ via automatic differentiation.

With the gradient of the likelihood obtained, MCMC sampler can then use it to efficiently explore the posterior distribution and generate samples from the posterior. The No-U-Turn Sampler (NUTS) was introduced by Hoffman et al. [54] as an efficient and automatic alternative to traditional MCMC methods and has been shown to be highly effective in a wide range of applications, including complex models with multi-model distributions and high-dimensional parameters.

With the NUTS algorithm implemented in Turing.jl [55], the Robertson ODE described in Section 2.3 is employed to demonstrate the variational inference ability of the Bayesian Neural ODE. Firstly, the arbitrary independent PDFs of parameters $\boldsymbol{\theta} \equiv \frac{\boldsymbol{k}}{\boldsymbol{k}^{true}}$ are taken as the ground truth to generate the training data. The prior parameter distributions are specified to be uniform distributions, as shown in the right panel of Fig. 11. Then as constrained by the data with the corresponding uncertainties, the posterior profiles and parameter distributions represent the learned dynamics and the learned uncertainties. As shown in Fig. 11, the posterior profiles lie between 2σ range of training data, with the marginal parameter distributions overlapping well with the ground truth ones. Note that the Bayesian inference of Neural ODE shown here is only for demonstration and warrants further analysis of its performance and robustness, under more practical conditions.

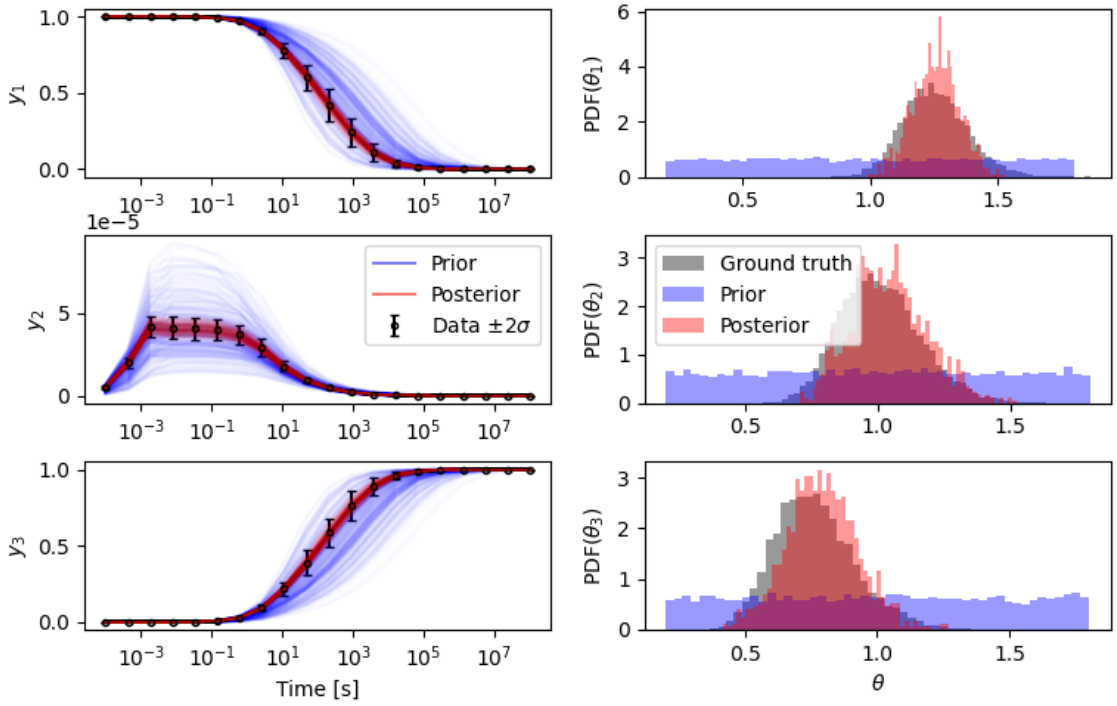

Fig. 11. Profiles and parameter distributions of Robertson ODE in the Bayesian inference process of Neural ODE.

Regarding the mechanisms of hydrocarbon fuels, using Bayesian Neural ODE to infer kinetics parameter distributions would be a promising way to conduct efficiently reverse UQ from limited datasets and is worthy of further investigation.

5. Conclusion

In this work, the Neural ODE architecture with the adjoint sensitivity method is proposed for kinetics parameter optimization with efficient and accurate gradient evaluation of chemical ODE models. And it is the first time demonstrated with systematically tests for the optimization of hydrocarbon fuels' mechanisms of various complexity, with different noise levels and mutable parameters.

The numerical experiments of optimizing the Robertson problem show that the proposed method can optimize kinetics parameters effectively and robustly, even with a substantial level of noise. The case study of JP-10 pyrolysis demonstrates the ability of the proposed method to learn the true kinetics parameters instead of being trapped into a local minimum, in practice chemical models. Though few data are exposed to the optimization process, the kinetics parameters are

adequately optimized and their predictions well fit the training sets and validation sets. Further tests against different noise levels suggest that when optimizing the chemical system with limited data, adding data abundance may perform better than pursuing ultimate data accuracy. Additionally, the parameter optimization of the *n*-heptane skeletal mechanism shows that reduced models can be trained to perform as well as the detailed ones, while not only matching the datasets constraint but also preserving the physical interpretability. And the optimization with all the Arrhenius parameters only brings little improvement than the optimization with only pre-exponential factors A.

Despite the capability to optimize various types of data, including temporal profiles and typical combustion characteristics, one can also adopt Bayesian inference techniques with Neural ODE to estimate the parameter uncertainties from experimental uncertainties. The proposed method can also be employed to optimize global reaction mechanisms, with available experimental data in species profiles or global characteristics such as laminar flame speeds. Assisted with high-efficiency neural network platforms and specific acceleration hardware units, e.g., TPU, one can optimize even larger kinetics models with thousands of reactions.

Acknowledgements

The work was supported by the National Natural Science Foundation of China 52025062.

Appendix

The initial conditions for the JP10 pyrolysis example are shown in Table A1, where $Y_{0,C_{10}H_{16}}$ is the initial mass fraction of JP10, and T_0 is the initial temperature. The first 20 conditions are used as training set, with T_0 in the range of 1000-1200K and $Y_{0,C_{10}H_{16}}$ in the range of 0.02-0.2; and the other 5 conditions are used as validation set, with T_0 in the range of 1200-1400K and $Y_{0,C_{10}H_{16}}$ in the range of 0.2-0.3.

Table A1. Initial conditions for the JP10 pyrolysis example.

		$\boldsymbol{Y_{0,C_{10}H_{16}}}$	$\boldsymbol{T_0}$ (K)
Training Set	#1	0.0920	1192
$\boldsymbol{T_0} \in [\mathbf{1000}, \mathbf{1200}]$ K	#2	0.1064	1104
$\boldsymbol{Y_{0,C_{10}H_{16}}} \in [\mathbf{0.02}, \mathbf{0.2}]$	#3	0.1640	1080
	#4	0.1928	1048
	#5	0.1496	1184

	#6	0.0272	1088
	#7	0.1208	1096
	#8	0.0560	1136
	#9	0.1784	1176
	#10	0.0848	1168
	#11	0.0992	1072
	#12	0.1352	1040
	#13	0.1856	1064
	#14	0.0632	1032
	#15	0.1424	1152
	#16	0.1568	1024
	#17	0.0776	1128
	#18	0.1280	1008
	#19	0.1136	1016
	#20	0.0416	1112
Validation Set	#21	0.2080	1256
$T_0 \in [1200,1400]$ K	#22	0.3000	1320
$Y_{0,C_{10}H_{16}} \in [0.2,0.3]$	#23	0.2160	1360
	#24	0.2840	1400
	#25	0.2280	1344

References

[1] S.J. Klippenstein, C. Cavallotti, Ab initio kinetics for pyrolysis and combustion systems, Computer Aided Chemical Engineering, Elsevier, 2019, pp. 115-167.

[2] B. Yang, Towards predictive combustion kinetic models: Progress in model analysis and informative experiments, Proc. Combust. Inst. 38 (2021) 199-222.

[3] W. Ji, Z. Ren, C.K. Law, Evolution of sensitivity directions during autoignition, Proc. Combust. Inst. 37 (2019) 807-815.

[4] V. Gururajan, F.N. Egolfopoulos, Direct sensitivity analysis for ignition delay times, Combust. Flame 209 (2019) 478-480.

[5] H. Wang, D.A. Sheen, Combustion kinetic model uncertainty quantification, propagation and minimization, Prog. Energy Combust. Sci. 47 (2015) 1-31.

[6] X. Su, W. Ji, Z. Ren, Uncertainty analysis in mechanism reduction via active subspace and transition state analyses, Combust. Flame 227 (2021) 135-146.

[7] D.A. Sheen, H. Wang, The method of uncertainty quantification and minimization using polynomial chaos expansions, Combust. Flame 158 (2011) 2358-2374.
[8] H. Rabitz, Ö.F. Aliş, J. Shorter, K. Shim, Efficient input—output model representations, Comput. Phys. Commun. 117 (1999) 11 - 20.
[9] J. Wang, Z. Zhou, K. Lin, C.K. Law, B. Yang, Facilitating Bayesian analysis of combustion kinetic models with artificial neural network, Combust. Flame 213 (2020) 87-97.
[10] D.A. Sheen, X. You, H. Wang, T. Løvås, Spectral uncertainty quantification, propagation and optimization of a detailed kinetic model for ethylene combustion, Proc. Combust. Inst. 32 (2009) 535-542.
[11] W. Ji, Z. Ren, Y. Marzouk, C.K. Law, Quantifying kinetic uncertainty in turbulent combustion simulations using active subspaces, Proc. Combust. Inst. 37 (2019) 2175-2182.
[12] J.I. Ryu, K. Kim, K. Min, R. Scarcelli, S. Som, K.S. Kim, J.E. Temme, C.-B.M. Kweon, T. Lee, Data-driven chemical kinetic reaction mechanism for F-24 jet fuel ignition, Fuel 290 (2021) 119508.
[13] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT press, 2016.
[14] M. Ihme, C. Schmitt, H. Pitsch, Optimal artificial neural networks and tabulation methods for chemistry representation in LES of a bluff-body swirl-stabilized flame, Proc. Combust. Inst. 32 (2009) 1527-1535.
[15] M. Ihme, W.T. Chung, A.A. Mishra, Combustion machine learning: Principles, progress and prospects, Prog. Energy Combust. Sci. 91 (2022) 101010.
[16] O. Owoyele, P. Kundu, P. Pal, Efficient bifurcation and tabulation of multi-dimensional combustion manifolds using deep mixture of experts: An a priori study, Proc. Combust. Inst. 38 (2021) 5889-5896.
[17] M. Raissi, A. Yazdani, G.E. Karniadakis, Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations, Science 367 (2020) 1026-1030.
[18] W. Ji, W. Qiu, Z. Shi, S. Pan, S. Deng, Stiff-pinn: Physics-informed neural network for stiff chemical kinetics, J. Phys. Chem. A 125 (2021) 8098-8106.
[19] J. An, G. He, K. Luo, F. Qin, B. Liu, Artificial neural network based chemical mechanisms for computationally efficient modeling of hydrogen/carbon monoxide/kerosene combustion, Int. J. Hydrogen Energy 45 (2020) 29594-29605.

[20] M.T.H. de Frahan, S. Yellapantula, R. King, M.S. Day, R.W. Grout, Deep learning for presumed probability density function models, Combust. Flame 208 (2019) 436-450.
[21] W. Ji, S. Deng, Autonomous discovery of unknown reaction pathways from data by chemical reaction neural network, J. Phys. Chem. A 125 (2021) 1082-1092.
[22] R.T. Chen, Y. Rubanova, J. Bettencourt, D.K. Duvenaud, Neural ordinary differential equations, Advances in neural information processing systems 31 (2018).
[23] M. Lemke, L. Cai, J. Reiss, H. Pitsch, J. Sesterhenn, Adjoint-based sensitivity analysis of quantities of interest of complex combustion models, Combust. Theory Model. 23 (2019) 180-196.
[24] Y. Ma, V. Dixit, M.J. Innes, X. Guo, C. Rackauckas. A comparison of automatic differentiation and continuous sensitivity analysis for derivatives of differential equation solutions. 2021 IEEE High Performance Extreme Computing Conference (HPEC); 2021: IEEE. p. 1-9.
[25] C. Rackauckas, Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. Skinner, A. Ramadhan, A. Edelman, Universal differential equations for scientific machine learning, arXiv preprint arXiv:2001.04385, (2020).
[26] W. Ji, X. Su, B. Pang, Y. Li, Z. Ren, S. Deng, SGD-based optimization in modeling combustion kinetics: Case studies in tuning mechanistic and hybrid kinetic models, Fuel 324 (2022) 124560.
[27] W. Ji, F. Richter, M.J. Gollner, S. Deng, Autonomous kinetic modeling of biomass pyrolysis using chemical reaction neural networks, Combust. Flame 240 (2022) 111992.
[28] W. Ji, J. Zanders, J.-W. Park, S. Deng. Data-Driven Approaches to Learn HyChem Models. Internal Combustion Engine Division Fall Technical Conference; 2021: American Society of Mechanical Engineers. p. V001T006A011.
[29] W. Ji, P. Zhao, T. He, X. He, A. Farooq, C.K. Law, On the controlling mechanism of the upper turnover states in the NTC regime, Combust. Flame 164 (2016) 294-302.
[30] K. He, X. Zhang, S. Ren, J. Sun. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 770-778.
[31] O. Owoyele, P. Pal, ChemNODE: A neural ordinary differential equations framework for efficient chemical kinetic solvers, Energy and AI 7 (2022) 100118.
[32] S. Kim, W. Ji, S. Deng, Y. Ma, C. Rackauckas, Stiff neural ordinary differential equations, Chaos: An Interdisciplinary Journal of Nonlinear Science 31 (2021) 093122.

[33] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980, (2014).
[34] P. Stapor, F. Fröhlich, J. Hasenauer, Optimization and profile calculation of ODE models using second order adjoint sensitivity analysis, Bioinformatics 34 (2018) i151-i159.
[35] B. Sengupta, K.J. Friston, W.D. Penny, Efficient gradient computation for dynamical models, Neuroimage 98 (2014) 521-527.
[36] Y. Cao, S. Li, L. Petzold, R. Serban, Adjoint sensitivity analysis for differential-algebraic equations: The adjoint DAE system and its numerical solution, SIAM J. Sci. Comput. 24 (2003) 1076-1089.
[37] H. Robertson, Numerical analysis, an introduction, chapitre The solution of a set of reaction rate equations, Academic Press, 1966.
[38] Y. Tao, R. Xu, K. Wang, J. Shao, S.E. Johnson, A. Movaghar, X. Han, J.-W. Park, T. Lu, K. Brezinsky, A Physics-based approach to modeling real-fuel combustion chemistry–III. Reaction kinetic model of JP10, Combust. Flame 198 (2018) 466-476.
[39] N. Nordin, Numerical simulations of non-steady spray combustion using a detailed chemistry approach, Licentiate of Engineering thesis, Department of Thermo and Fluid Dynamics, Chalmers University of Technology, Goteborg, Sweden, (1998).
[40] T. Lu, C.K. Law, A directed relation graph method for mechanism reduction, Proc. Combust. Inst. 30 (2005) 1333-1341.
[41] T. Turányi, Applications of sensitivity analysis to combustion chemistry, Reliability Engineering & System Safety 57 (1997) 41-48.
[42] P. Pepiot-Desjardins, H. Pitsch, An efficient error-propagation-based reduction method for large chemical kinetic mechanisms, Combust. Flame 154 (2008) 67-81.
[43] Y. Tao, H. Wang, Joint probability distribution of Arrhenius parameters in reaction model optimization and uncertainty minimization, Proc. Combust. Inst. 37 (2019) 817-824.
[44] T. Nagy, T. Turányi, Uncertainty of Arrhenius parameters, Int. J. Chem. Kinet. 43 (2011) 359-378.
[45] R. Feeley, M. Frenklach, M. Onsum, T. Russi, A. Arkin, A. Packard, Model discrimination using data collaboration, J. Phys. Chem. A 110 (2006) 6803-6813.

[46] W. Li, A. Hegde, J. Oreluk, A. Packard, M. Frenklach, Representing model discrepancy in bound-to-bound data collaboration, SIAM/ASA Journal on Uncertainty Quantification 9 (2021) 231-259.
[47] L. Cai, H. Pitsch, Optimized chemical mechanism for combustion of gasoline surrogate fuels, Combust. Flame 162 (2015) 1623-1637.
[48] L. Cai, H. Pitsch, S.Y. Mohamed, V. Raman, J. Bugler, H. Curran, S.M. Sarathy, Optimized reaction mechanism rate rules for ignition of normal alkanes, Combust. Flame 173 (2016) 468-482.
[49] K. Braman, T.A. Oliver, V. Raman, Bayesian analysis of syngas chemistry models, Combust. Theory Model. 17 (2013) 858-887.
[50] M.D. Hoffman, D.M. Blei, C. Wang, J. Paisley, Stochastic variational inference, J. Mach. Learn. Res., (2013).
[51] X. Li, T.-K.L. Wong, R.T. Chen, D. Duvenaud. Scalable gradients for stochastic differential equations. International Conference on Artificial Intelligence and Statistics; 2020: PMLR. p. 3870-3882.
[52] R. Dandekar, K. Chung, V. Dixit, M. Tarek, A. Garcia-Valadez, K.V. Vemula, C. Rackauckas, Bayesian neural ordinary differential equations, arXiv preprint arXiv:2012.07244, (2020).
[53] W. Xu, R.T. Chen, X. Li, D. Duvenaud. Infinitely deep bayesian neural networks with stochastic differential equations. International Conference on Artificial Intelligence and Statistics; 2022: PMLR. p. 721-738.
[54] M.D. Hoffman, A. Gelman, The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo, J. Mach. Learn. Res. 15 (2014) 1593-1623.
[55] H. Ge, K. Xu, Z. Ghahramani. Turing: a language for flexible probabilistic inference. International conference on artificial intelligence and statistics; 2018: PMLR. p. 1682-1690.