Neural flow-augmented global reaction for the reduction of complex chemical kinetics

By C. Laurent

1. Motivation and objectives

The integration of complex chemical kinetics into reactive flow simulations enhances accuracy and facilitates the capture of intricate chemico-physical phenomena (Felden *et al.* 2018), yet it also poses considerable challenges (Lu & Law 2009). These challenges stem primarily from the high computational costs associated with the large number of reactions and species involved. Additionally, the inherent stiffness of chemical kinetics, often characterized by significant separations in spatial and temporal scales among species, further complicates the process. Traditional methods typically rely on a combination of physics-based heuristics and mathematical optimization to simplify kinetics by reducing the number of species and reactions (Pepiot-Desjardins & Pitsch 2008; Jaravel *et al.* 2019; Wu *et al.* 2020). An extreme example of this simplification is the use of global reaction schemes, where the entire chemical network is condensed into a single reaction between reactants and their products. Although such global schemes are computationally less demanding, they often lack the accuracy found in more detailed mechanisms. Recent progress in machine learning provides new avenues to address these challenges. Typically, these advancements involve the use of autoencoder (AE) models to compress the dimensionality of the kinetic system into a few latent variables. This is coupled with various methods for learning the temporal dynamics of the latent equations, such as physics-informed neural networks (Raissi *et al.* 2019; Ji *et al.* 2021), neural ordinary differential equations (ODEs) (Chen *et al.* 2018; Kim *et al.* 2021; Dikeman *et al.* 2022) or deep operator networks (DeepONets) (Lu *et al.* 2021; Goswami *et al.* 2023). Previous authors have emphasized the necessity of addressing the stiffness in the temporal evolution of chemical kinetics and have developed methods intended to tackle this issue.

In this work, a methodology is introduced that blends principles from both chemistry and machine learning. Instead of the conventional use of an autoencoder for reducing the dimensions of the kinetic system, only the species from a global reaction are retained. However, the unique reaction of a one-step mechanism is replaced with a deep neural network (DNN). This approach aims at combining the low cost of one-step mechanisms with the quantitative precision offered by data-driven methods. More specifically, an encoder is designed to encode the chemical state into a latent space where the temporal dynamics have a reduced stiffness. Temporal solutions of the kinetic system are obtained within the latent space; the learning of this latent solution operator is performed by a neural-flow model (Biloš *et al.* 2021), akin to neural ODEs. It differs by directly learning the continuous-time solution operator rather than the differential equation itself, aligning more closely with a DeepONet approach. This strategy avoids the challenges associated with training a neural ODE for stiff problems, a common issue in previous attempts. To further mitigate stiffness in the latent space, a gradient-smoothing regularization term is introduced, leading to a multitask optimization objective. The method is applied to time series generated from methane-oxygen ignition simulations in a constant-volume

reactor, covering a wide range of temperatures, pressures and equivalence ratios using a 30-species detailed mechanism (Smith *et al.* 2016). We then assess the surrogate's reliability for predicting ignition sequences and statistics for a new set of unseen reactor simulations.

The remainder of this paper is organized as follows: Section 2 describes the DNN model, the regularization term, the training procedure, and the data generation process. Section 3 presents validation of the trained model and comparison with classical mechanisms used in combustion kinetics, followed by an analysis of the training process and the effect of the introduced regularization. Finally, Section 4 presents conclusions and potential future research directions.

2. Method

2.1. *Latent neural-flow model*

In a typical setting, disregarding spatial phenomena for simplicity, chemical kinetics involving N_S species are governed by a dynamical system of dimension $N_S + K$

$$\frac{d\mathbf{y}}{dt} = \mathbf{F}(\mathbf{y}) \tag{2.1}$$

Here, $\mathbf{y}$ represents a vector that includes the chemical composition of the mixture, characterized by the species mass fractions $(Y_k)_{k=1,\dots,N_S}$ or mole fractions $(X_k)_{k=1,\dots,N_S}$, along with K thermodynamic variables, which vary depending on the system considered (such as temperature T, energy U, etc.). The computational cost associated with the integration of chemical kinetics is multifaceted: (1) a high number of species N_S results in an extensive set of ODEs to solve; (2) numerous reactions, each governed by nonlinear Arrhenius laws, lead to computationally demanding calculations for $\mathbf{F}(\mathbf{y})$; and (3) the stiffness of the ODEs necessitates the employment of specialized stiff-ODE solvers (Wanner & Hairer 1996). To address points (1) and (2), we focus on retaining only a global one-reaction mechanism, characterized by a single reaction with a limited number of reactants and products. This method is commonly used in large-eddy simulations of real engines (Poinsot & Veynante 2005). We therefore only retain the corresponding species in $\mathbf{y}$, such that it reduces to a vector $\mathbf{x}$ of smaller dimension. Point (3) is tackled using a neural-flow DNN (Biloš *et al.* 2021) combined with an AE, as described in Figure 1.

The model comprises three components: an encoder $\mathcal{E}$, tasked with encoding a state $\mathbf{x}_t$ at a given time t into a latent variable $\mathbf{z}_t$. This process is particularly utilized for projecting an initial condition $\mathbf{x}_0$ to $\mathbf{z}_0$. The AE does not serve for dimensionality reduction but as a regularizer, projecting the kinetics variables into a state where the governing equations are less stiff. The second component is a neural-flow network $\mathcal{P}$, operating in the latent space, which learns the solution operator $\mathcal{S}$ of Eq. (2.1) in this regularized latent space. This operator maps any initial condition $\mathbf{z}_0$ to the solution of the ODE $\mathbf{z}(t;\mathbf{z}_0)$ at any given time t. Finally, latent trajectories $\mathbf{z}(t;\mathbf{z}_0)$ are mapped back into the physical space through a decoder $\mathcal{D}$. Similar architectures have been popularized in the context of DeepONets (Lu *et al.* 2021; Goswami *et al.* 2023), where the AE is typically employed for dimension reduction rather than for regularizing stiff dynamics.

The network is trained to minimize the following multitask objective

$$L = L_{\mathcal{EPD}} + L_{\mathcal{EP}} + L_{\mathcal{ED}} + L_{reg} \ , \tag{2.2}$$

where $L_{\mathcal{EPD}}$ quantifies the final prediction errors in the physical space. Denoting N as

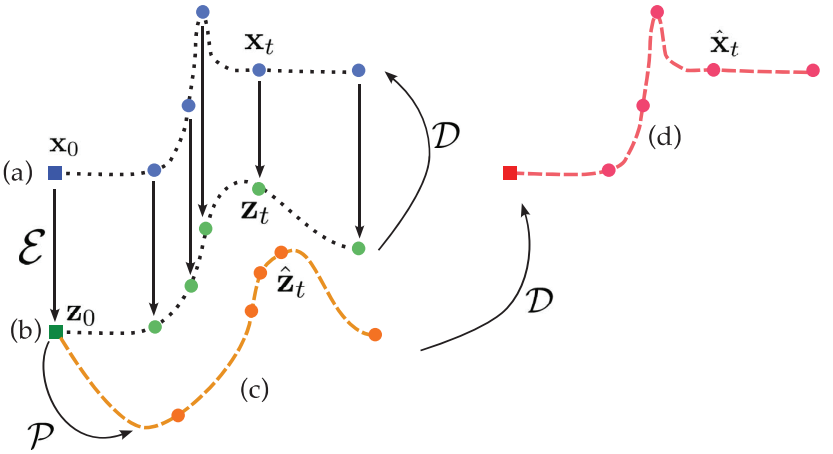

FIGURE 1. Overview of the neural-flow AE process. The encoder $\mathcal{E}$ independently encodes each point $\mathbf{x}_t$ in the physical space (a) into a latent representation $\mathbf{z}_t$ (b). Subsequently, the neural-flow predictor $\mathcal{P}$ determines the entire time solution $\hat{\mathbf{z}}(t, \mathbf{z}_0)$ from a given initial condition $\mathbf{z}_0$ (c). This is facilitated by the stiffness reduction of the dynamics in the latent space. The decoder $\mathcal{D}$ then reconstructs the predicted trajectory $\hat{\mathbf{x}}(t, \mathbf{x}_0)$ back in the physical space (d). Additionally, for ensuring consistency, $\mathcal{D}$ is also trained to invert the operation of $\mathcal{E}$.

the number of trajectories $\mathbf{x}_{<T} = (\mathbf{x}_t)_{t=0,\dots,T-1}$ in our data set, and T as their length, $L_{\mathcal{EPD}}$ is expressed as

$$L_{\mathcal{EPD}} = \frac{1}{N}\frac{1}{T}\sum_{n=1}^{N}\sum_{t=0}^{T-1} ||\mathbf{x}_t - \hat{\mathbf{x}}_t(\mathbf{x}_0)||^2 \ . \tag{2.3}$$

$L_{\mathcal{EP}}$ ensures the consistency of temporal prediction and encoding in the latent space

$$L_{\mathcal{EP}} = \frac{1}{N}\frac{1}{T}\sum_{n=1}^{N}\sum_{t=0}^{T-1} ||\mathbf{z}_t - \hat{\mathbf{z}}_t(\mathbf{z}_0)||^2 \ , \tag{2.4}$$

and $L_{\mathcal{ED}}$ ensures that the decoder learns the inverse of the encoder

$$L_{\mathcal{ED}} = \frac{1}{N}\frac{1}{T}\sum_{n=1}^{N}\sum_{t=0}^{T-1} ||\mathbf{x}_t - \mathcal{D}(\mathbf{z}_t)||^2 \ . \tag{2.5}$$

Finally, L_{reg} is a regularization term that encourages the encoder to learn a latent representation where the temporal dynamics are less stiff

$$L_{reg} = \frac{1}{N}\sum_{n=1}^{N} ||\mathbf{z}_{<T} - \overline{\mathbf{z}_{<T}}||^2 \ , \tag{2.6}$$

where $\overline{\mathbf{z}_{<T}}$ designates a convolutional, temporal, low-pass filter applied to the encoded trajectory $\mathbf{z}_{<T}$. This regularization encourages $\mathcal{E}$ to learn smooth trajectories in the latent space, which are easier to approximate for the predictor $\mathcal{P}$. The objective L of Eq. (2.2) is trained with the Adam optimizer (Kingma & Ba 2014); note that L is simply the unweighted sum of the four losses presented in Eqs.(2.3)-(2.6). The encoder and decoder are simple feedforward networks with three layers of 128 hidden units each; the output layer of $\mathcal{E}$ includes a batch normalization to encourage the encoder to learn a standardized latent representation. The predictor $\mathcal{P}$ is a residual-flow network, which is

formulated as

$$\begin{aligned}&\mathcal{P}(t,\mathbf{z}_0)=\mathbf{z}_0+\mathbf{A}(\mathbf{z}_0)\cdot\mathbf{B}(t)-\mathbf{A}(\mathbf{z}_0)\cdot\mathbf{B}(0),\text{with}\\&\mathbf{A}(\mathbf{z}_0)=[\mathbf{A}_1(\mathbf{z}_0),...,\mathbf{A}_K(\mathbf{z}_0)]\ ,\ \mathbf{B}(t)=[\mathbf{B}_1(t),...,\mathbf{B}_K(t)]\,.\end{aligned}\tag{2.7}$$

Both the initial condition-dependent amplitudes $\mathbf{A}(\mathbf{z}_0)$ and time-dependent basis functions $\mathbf{B}(t)$ are parameterized by two compact feedforward networks, each comprising two layers with 32 units. Furthermore, $K = 64$ basis functions are utilized in Eq. (2.7). It is important to note that the neural-flow predictor $\mathcal{P}$ does not learn the right-hand side of an differential equation like a neural ODE. Instead, it directly models the solution operator $\mathcal{S}(t;\mathbf{z}_0)$ of the differential equation in the latent space (i.e., the flow). Consequently, it does not depend on any ODE solver, either during training or prediction, and thus can provide predictions at any given time t for a specified initial condition $\mathbf{z}_0$ with a single pass through the neural network. This approach circumvents the difficulty of training neural ODE models for stiff equations, as such a method relies on a differentiable stiff ODE solver. Finally, the structure adopted for $\mathcal{P}$ aligns closely with that of a DeepONet (Lu *et al.* 2021; Goswami *et al.* 2023), where $\mathbf{A}(\mathbf{z}_0)$ is analogous to the branch network and $\mathbf{B}(t)$ resembles the trunk network.

2.2. *Data generation*

The latent neural-flow, as described above, is applied to the ignition of methane-oxygen mixtures. The FFCM detailed mechanism consisting of 30 species and 278 reactions is employed (Smith *et al.* 2016). Training data are generated through simulations of ignition in a constant-volume reactor, governed by the following equation

$$\begin{aligned}\frac{dT}{dt}&=-\frac{V}{mc_v}\sum_{s=1}^{N_S}u_sW_s\dot{\omega}_s\\\frac{dY_s}{dt}&=\frac{V}{m}W_s\dot{\omega}_s\ ,\ s=1,\ldots,N_S,\end{aligned}\tag{2.8}$$

where m and V represent the total mass and volume of the reactor, c_v is the mass heat capacity at constant volume, u_s is the species internal energy, W_s is the species molar weight, and $\dot{\omega}_s$ denote the species net production rate. Approximately 10^4 ignition trajectories, each containing $T = 1000$ time steps, are generated by solving Eq. (2.8) using the Cantera library (Goodwin *et al.* 2023). Note that the time steps are not uniformly sampled, but follow the ODE solver time stepping strategy, with higher density of points in region of sharp gradients. Initial temperature T_0 and pressure P_0 are sampled following normal distributions, while the initial equivalence ratio ϕ_0 is sampled from a uniform distribution $T_0 \sim \mathcal{N}(1600\text{K}, 300\text{K})$, $P_0 \sim \mathcal{N}(1\text{atm}, 0.5\text{atm})$, and $\phi_0 \sim \mathcal{U}(0.5, 2)$. Out of the 10^4 trajectories, 80% are retained for training and the rest are held out for validation. A typical ignition trajectory is displayed in Figure2.

In the state vector $\mathbf{x}$ of the surrogate model, only five variables are retained: the temperature and the four species involved in the global reaction – the reactants CH_4 and O_2 and the products CO_2 and H_2O. The dimension of the latent variable $\mathbf{z}$ is fixed at 8.

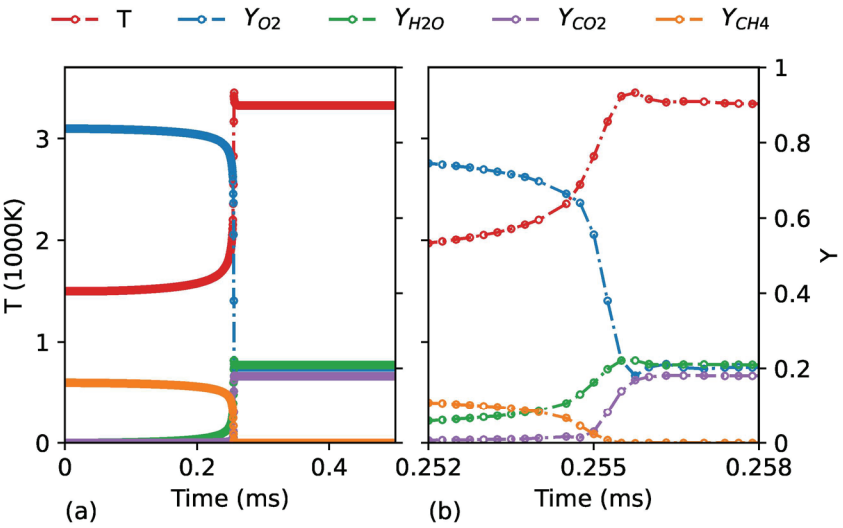

FIGURE 2. (a) An entire ignition trajectory from the training data set. (b) Close-up view around the area of ignition. Symbols represent the time instants sampled for training. This shows significant separation of scale, with a long ignition delay time of 0.2 ms, followed by fast reactions with a timescale of approximately 0.001 ms.

3. Results

3.1. *Model validation*

After training, the model's predictive ability and generalization are evaluated using a new data set, consisting of 1000 trajectories sampled from simulations of constant-volume reactors. These new trajectories are generated using a similar process, but with variances in T_0, P_0 and ϕ_0 increased by 20%. Figure 3(a) contrasts the trajectory obtained with the detailed FFCM mechanism against that learned by the neural-flow operator. The temperature and species profiles appear similar, accurately predicting the ignition delay time δ_i and the adiabatic flame temperature T_{ad}. However, the inferred ignition sequences are slightly smoother, and most importantly, they fail to capture the sharp temperature overshoot. This sharp overshoot is characteristic of methane-oxygen Combust. and leads to significant endothermic recombinations, a phenomenon that can only be captured by detailed kinetic mechanisms. For reference, Figure 3(b) presents the same simulation with the BFER mechanism (Franzelli *et al.* 2012), a commonly used global mechanism for methane combustion. This mechanism underpredicts the ignition delay time by two orders of magnitude and predicts an adiabatic flame temperature of up to 7000K. These unrealistic results are due to the complexity of ignition, a phenomenon governed by both slow and fast reactions, a process that one-step chemistry cannot accurately model.

The accuracy of the model is further assessed by computing ignition statistics on the validation data set. Figure 4 presents statistics of the relative errors committed by the DNN model in calculating the ignition time delay δ_i and the adiabatic flame temperature T_{ad}. Errors in the adiabatic flame temperature are marginal, consistently staying below 5%. Predicting the ignition delay is more challenging, but the error remains within a reasonable margin and well within the experimental uncertainty associated with the measurements of this quantity. Further analysis reveals that cases with significant ignition delay errors correlate with extended ignition delays, caused either by low temperatures

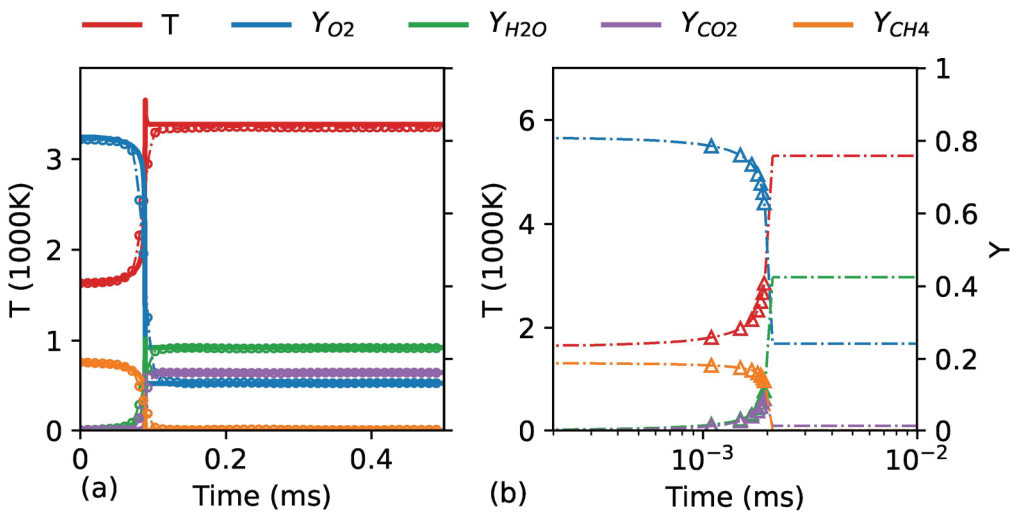

FIGURE 3. (a) Comparison between the FFCM mechanism and the neural-flow operator for time series data of temperature and retained species, obtained from simulating a constant-volume reactor. Thick lines indicate the ground truth, and dashed lines with circles indicate the surrogate predictions. (b) Simulation using identical initial conditions with the one-step BFER mechanism; dashed lines with triangles represent the BFER solution. Note that the FFCM predictions are not displayed here, as they significantly differ from those of BFER. Observe also the difference in scale for the temperature and the logarithmic scale for time.

or initial compositions at the edge of the flammability limit. These samples are underrepresented in the training data set; thus it is not surprising that the model has difficulties in predicting them accurately.

An additional advantage of the neural-flow model is its capability to learn the entire solution operator of the latent dynamical system. This provides access to the chemical composition and temperature at any given moment t, and it does so with constant time complexity through a single pass via the encoder, predictor and decoder. In terms of computational cost, the aforementioned comparisons were conducted on an Apple M1 Pro Chip, with physical simulations executed using Cantera and the DNN operations performed using Pytorch. The average runtime of the DNN to generate the temporally resolved solution for a single reactor is 0.26 ms, compared to 0.41 ms for the BFER mechanism and 8.6 ms for the detailed FFCM mechanism. Thus, the neural-flow operator is significantly more efficient than these alternatives.

3.2. *Regularization and conflicting gradients*

The training of the latent neural-flow operator, particularly the effect of the encoder regularization L_{reg} defined in Eq. 2.6, is now analyzed. Figure 5(b) displays two of the encoded variables $\mathbf{z}_1$ and $\mathbf{z}_2$ for a given trajectory $\mathbf{x}_{<T}$, along with the corresponding predicted latent solutions $\hat{\mathbf{z}}_{1,<T}$ and $\hat{\mathbf{z}}_{2,<T}$. The impact of L_{reg} in the latent space is evident: the latent trajectories are smoother compared to the time series displayed in Figure 3, which facilitates their learning by the predictor $\mathcal{P}$. Despite the significant smoothing of pre-ignition and post-ignition phases, sharp gradients are still present. These sharp gradients correspond to the temperature overshoot visible in Figure 3(a), which the DNN does not capture accurately. This effect of sharp gradients is also observable in the fluctuation

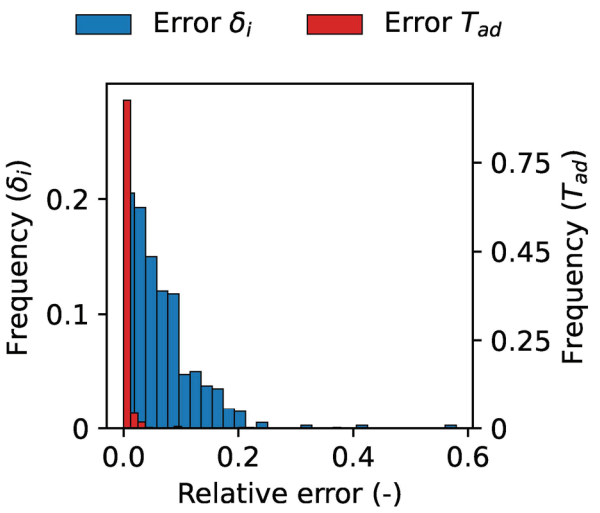

FIGURE 4. Distribution of relative error in adiabatic flame temperature T_{ad} (red) and ignition delay δ_i (blue) between the DNN and the detailed mechanism, as computed for the validation data set.

of losses shown in Figure 5(a), where the losses exhibit considerable noise. Additionally, the encoding-decoding loss $L_{\mathcal{ED}}$ appears to decrease faster than the other components, while the encoding-predicting-decoding loss $L_{\mathcal{EPD}}$, crucial for our applications, does not show a decreasing trend over time.

These observations raise questions about the encoder's ability to sufficiently regularize the latent space. To investigate this, the encoder is trained as a stand-alone model with the sole objective of minimizing the regularization term L_{reg}. Figure 6 shows similar results for this experiment. Figure 6(a) indicates a more rapid decrease of L_{reg} than when the encoder is trained jointly with the decoder. Figure 6(b) demonstrates that the encoder is fully capable of transforming sharp trajectories in the physical space into smooth trajectories in the latent space. This observation, coupled with the swift decrease of $L_{\mathcal{ED}}$ during the joint training as shown in Figure 5(a), suggests an adverse effect of the decoder during joint training. This effect, particularly common in multitask settings, is referred to as conflicting gradients (Yu *et al.* 2020; Liu *et al.* 2021, 2023). In essence, improving the encoder via $L_{\mathcal{ED}}$ can only occur at the expense of the regularization term L_{reg}, leading to either one dominating the other or no improvement at all. These observations call for the adoption of advanced multitask gradient manipulation techniques, such as FAMO (Liu *et al.* 2023), which essentially consists of dynamically weighting the individual terms in the objective of Eq. (2.2), such that all losses improve at a comparable rate.

4. Conclusions

This study presents the development of a DNN surrogate for combustion chemical kinetics of methane-oxygene. It is based on a global one-step mechanism augmented with an AE, aimed at reducing the stiffness of chemical kinetic evaluation in a latent space. This is further combined with a neural-flow model that learns the entire temporal solution of the latent ODE. The introduction of a regularization term, acting as a low-pass filter of the encoded trajectories, is crucial for regularizing the latent space. This model

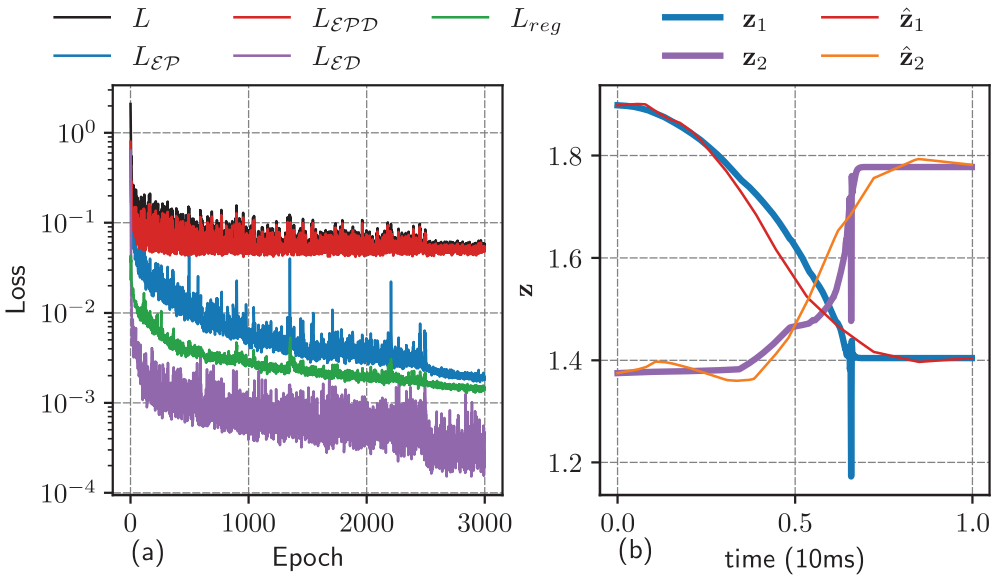

FIGURE 5. (a) Evolution of various validation losses during training. (b) Two latent variables, $\mathbf{z}_1$ and $\mathbf{z}_2$ (represented by thick blue and purple lines, respectively), for the same trajectory, accompanied by their corresponding latent temporal predictions (represented by thin orange and red lines).

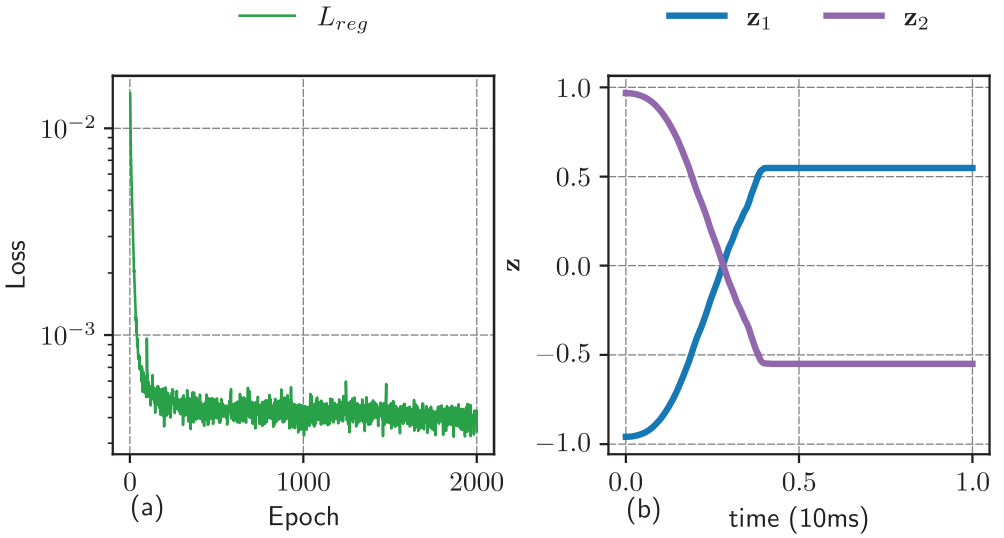

FIGURE 6. (a) Evolution of the regularization L_{reg} during the training of a stand-alone encoder. (b) Two latent variables, $\mathbf{z}_1$ and $\mathbf{z}_2$ for a common trajectory, obtained after this training.

was trained and evaluated on simulations from constant-volume reactors. The trained model demonstrated accuracy comparable to the full 30-species detailed mechanism, but with significantly lower computational costs. Its fidelity greatly surpasses that of typical one-step mechanisms such as BFER. Certain features of ignition, such as effects due to temperature overshoots and endothermic recombinations, remain unresolved by the model. This challenge is attributed to the complexities of jointly training a regularizing encoder with a decoder in a multitask setting, which is prone to gradient conflicts. Future work will focus on exploring the out-of-distribution generalizability of this model, including its applicability to other types of reactors and to multidimensional reactive fluid simulations. Advanced gradient manipulation techniques to balance the training between the various losses and mitigate the effects of conflicting gradients will also be explored.

Acknowledgments

This investigation was funded by the Advanced Simulation and Computing program of the U.S. Department of Energy's National Nuclear Security Administration via the PSAAP-III Center at Stanford, Grant No. DE-NA0003968. The author is grateful to Prof. Gianluca Iaccarino and Prof. Alireza Doostan for useful comments and discussions.

REFERENCES

Biloš, M., Sommer, J., Rangapuram, S. S., Januschowski, T. & Günnemann, S. 2021 Neural flows: Efficient alternative to neural odes. *Adv. Neur. In.* **34**, 21325–21337.

Chen, R. T., Rubanova, Y., Bettencourt, J. & Duvenaud, D. K. 2018 Neural ordinary differential equations. *Adv. Neur. In.* **31**.

Dikeman, H. E., Zhang, H. & Yang, S. 2022 Stiffness-reduced neural ode models for data-driven reduced-order modeling of combustion chemical kinetics. In *AIAA* Paper 0226.

Felden, A., Esclapez, L., Riber, E., Cuenot, B. & Wang, H. 2018 Including real fuel chemistry in LES of turbulent spray combustion. *Combust. and Flame* **193**, 397–416.

Franzelli, B., Riber, E., Gicquel, L. Y. & Poinsot, T. 2012 Large eddy simulation of combustion instabilities in a lean partially premixed swirled flame. *Combust. and flame* **159**, 621–637.

Goodwin, D. G., Moffat, H. K., Schoegl, I., Speth, R. L. & Weber, B. W. 2023 Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes.

Goswami, S., Jagtap, A. D., Babaee, H., Susi, B. T. & Karniadakis, G. E. 2023 Learning stiff chemical kinetics using extended deep neural operators. *arXiv:2302.12645* .

Jaravel, T., Wu, H. & Ihme, M. 2019 Error-controlled kinetics reduction based on non-linear optimization and sensitivity analysis. *Combust. and Flame* **200**, 192–206.

Ji, W., Qiu, W., Shi, Z., Pan, S. & Deng, S. 2021 Stiff-PINN: Physics-informed neural network for stiff chemical kinetics. *J. of Phys. Chem. A* **125**, 8098–8106.

Kim, S., Ji, W., Deng, S., Ma, Y. & Rackauckas, C. 2021 Stiff neural ordinary differential equations. *Chaos* **31**, 093122.

Kingma, D. P. & Ba, J. 2014 Adam: A method for stochastic optimization. *arXiv:1412.6980* .

Liu, B., Feng, Y., Stone, P. & Liu, Q. 2023 FAMO: Fast adaptive multitask optimization. *arXiv:2306.03792* .

Liu, B., Liu, X., Jin, X., Stone, P. & Liu, Q. 2021 Conflict-averse gradient descent for multi-task learning. *Adv. Neur. In.* **34**, 18878–18890.

Lu, L., Jin, P., Pang, G., Zhang, Z. & Karniadakis, G. E. 2021 Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat. Mach. Intell.* **3**, 218–229.

Lu, T. & Law, C. K. 2009 Toward accommodating realistic fuel chemistry in large-scale computations. *Prog. Energ. Combust.* **35**, 192–215.

Pepiot-Desjardins, P. & Pitsch, H. 2008 An efficient error-propagation-based reduction method for large chemical kinetic mechanisms. *Combust. and Flame* **154**, 67–81.

Poinsot, T. & Veynante, D. 2005 *Theoretical and Numerical Combustion*. RT Edwards, Inc.

Raissi, M., Perdikaris, P. & Karniadakis, G. E. 2019 Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707.

Smith, G. P., Tao, Y. & Wang, H. 2016 Foundational fuel chemistry model version 1.0 (ffcm-1). epub, accessed July 26, 2018.

Wanner, G. & Hairer, E. 1996 *Solving Ordinary Differential Equations II (Vol. 375)*. Ney York: Springer Berlin Heidelberg.

Wu, Y., Liu, Y. & Lu, T. 2020 A linearized error propagation method for skeletal mechanism reduction. *Combust. and Flame* **211**, 303–311.

Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K. & Finn, C. 2020 Gradient surgery for multi-task learning. *Adv. Neur. In.* **33**, 5824–5836.