



ChemTab: A Physics Guided Chemistry Modeling Framework

Amol Salunkhe^(✉), Dwyer Deighan, Paul E. DesJardin, and Varun Chandola

University at Buffalo, Buffalo, NY 14260, USA
{aas22,dwyerdei,ped3,chandola}@buffalo.edu

Abstract. Modeling of turbulent combustion system requires modeling the underlying chemistry and the turbulent flow. Solving both systems simultaneously is computationally prohibitive. Instead, given the difference in scales at which the two sub-systems evolve, the two sub-systems are typically (re)solved separately. Popular approaches such as the *Flamelet Generated Manifolds* (FGM) use a two-step strategy where the governing reaction kinetics are pre-computed and mapped to a low-dimensional manifold, characterized by a few reaction progress variables (model reduction) and the manifold is then “looked-up” during the run-time to estimate the high-dimensional system state by the flow system. While existing works have focused on these two steps independently, we show that joint learning of the progress variables and the look-up model, can yield more accurate results. We propose *ChemTab* an architecture that learns jointly and demonstrate its superiority.

Keywords: Physics guided neural networks · DNN

1 Introduction

Modeling of turbulent flow combustion is central in the development of new combustion technologies in aviation, automotive and power generation [6]. Turbulent flow combustion combines two nonlinear and multi-scale phenomena: *turbulent flow* and *chemical reactions*. This coupling of the kinetic chemical reaction equations with the set of Navier-Stokes flow equations results in a problem that is too complex to be solved, at full resolution, by the current computational means. Even for a simple fuel such as methane, the combustion chemistry mechanism involves 53 species and 325 chemical reactions [19], and the numbers increase with increasing fuel complexity. Solving the details of such mechanisms during the flow simulation can consume up to 75% of the solution time [4].

In most cases, the large scale separation between the combustion chemistry/flame (typically sub millimeter/microsecond scale) and the characteristic turbulent flow (typically centimeter or meter/minute or hour scale) allows simplifying assumptions to be made that enable increased computational efficiency by (re)solving chemistry and flow separately [16]. In this paper, we focus on approximate methods that deal with handling the chemistry, and in particular,

the methods based on laminar flames [15]. Here the 1-D or single-species flame reactions are solved *a priori* and stored. During the flow simulation, these reactions are looked-up to estimate the high-dimensional thermochemical state of the system, as shown in Fig. 1.

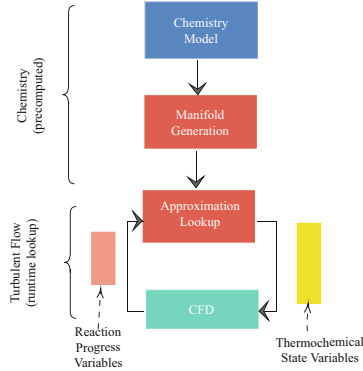


Fig. 1. (Re)solving systems separately

Most models developed for increased computational efficiency rely on the existence of a theoretical low-dimensional thermochemical state-space manifold to which the combustion chemistry can be mapped [11]. The central question then is, *how to efficiently model low-dimensional thermochemical manifolds that capture the relevant physics of the problem; and parametrize and approximate these manifolds which can then be accessed during turbulent flow simulations?*

While existing approaches (collectively referred to as *state-space parametrization* [16,17]) have been successful, they have primarily solved the two sub-problems – *progress variable generation* to characterize the manifold, and *manifold approximation* to perform the lookup during run-time, independently. This can result in sub-optimal solutions because the progress variables, learnt using methods such as *Principal Component Analysis* (PCA) [2,20], are not necessarily optimized to perform the run-time lookup. Similarly, while the traditional lookup approaches that use tabulation, or the recently proposed neural network based data-driven alternatives [1], facilitate efficient look-ups, the construction of the underlying data-structure or machine learning based model is not informed by the learning of the progress variables.

Our main hypothesis is that by simultaneously learning the progress variables and the manifold approximation (lookup model), we can achieve higher accuracy in terms of the estimation of the thermochemical state at run-time. But how does one combine the progress variable learning, an inherently linear mapping task, with a highly non-linear lookup model, while ensuring that the components influence each other during the learning phase? To that end, we propose a framework called *ChemTab*, in which the learning of these two components is

formulated as a joint optimization task. An implementation of *ChemTab*, using a novel deep learning architecture, is proposed. The joint optimization includes a set of mathematical constraints that ensure that the progress variable learning is approximately similar to a PCA-type linear reduction, and, at the same time, can also predict the thermochemical state using a non-linear predictive component.

The deep learning implementation of ChemTab is shown to reduce the error by 73%, when compared to an existing tabulation based framework, in predicting one of the key thermochemical term, *source energy*, when applied to flames data for a Methane–Air fuel–oxidizer combination generated using the GRI–Mech 3.0 simulator. Moreover, the proposed architecture of ChemTab is shown to outperform a recently proposed state-of-art decoupled PCA+neural network based solution by 24%.

2 Related Work

In this section we provide a brief overview of existing in low-dimensional thermochemical manifold modeling, focusing more on data-driven methods. We note that there have been works that use physics-driven machine learning models for solving other physics problems [10,23], however, these methods generally focus on simpler physics and are not necessarily applicable in the domain of turbulent combustion.

Common approaches to low-dimensional thermochemical manifold modeling are combustion chemistry mechanism reduction and thermochemical state-space parametrization [18,20]. Chemistry mechanism reduction approach cannot be generalized and in the recent past state-space parametrization approach has been the most dominant method comprising of two phases *progress variable generation* and *manifold approximation*. For progress variable generation, existing methods have either used domain models or numerical methods.

Domain models like steady Laminar Flamelet Method (SLFM) [15], Flamelet-Generated Manifold (FGM) [21,22], Flamelet Progress Variable approach (FPVA) [7,17] and Flamelet-Prolongation of ILDM model (FPI) [5] theorize that a multi-dimensional flame can be considered as an ensemble of multiple one-dimensional locally laminar flames (flamelets). These flamelets are parametrized by a combination of conserved and reactive scalars [3,17,21,22]. A lot of research in this area builds on the principles laid out in [9] for progress variables regularization however the fundamental problem of generating adequate number of progress variables that capture the underlying physics is still open.

Numerical methods, like PCA, have shown significant promise for parametrization of the thermochemical state. PCA provides a method of generating reaction progress variables using the flamelet solutions, the state-space variables are still nonlinear functions of the reaction progress variables, and a nonlinear regression is learned to approximate the state-space manifold [2,12,13,20]. This purely numerical parametrization lack interpretability and may also not

be generalizable enough due to variation capture maximization that may overlearn the numerical errors in the data. Linear Autoencoders have also been suggested [14] however this definition lacks incorporation of a principled approach to progress variable generation and thus may not be generalizable.

While domain based model have traditionally relied on tabular lookup, these are not scalable. Tabulated data occupies a larger portion of the available memory on every node where the flow simulation is computing. Also the searching and retrieval of this pre-tabulated data becomes increasingly expensive in a higher-dimensional space. For example, assuming a standard 3 progress variable discretization (200, 100, 50) with say 15 tabulated thermochemical state variables, we obtain a pre-computed combustion table of 120 Mb. The addition of a variable such as *enthalpy* with a very coarse discretization of 20 points, brings the size of the table to 2.4 Gb. To address the tabulation problem researchers like [1, 24] build on the work of [8] to investigate the use of a neural networks for manifold approximation which replaces the Tabulation. The mapping between the progress variables (reduced dimensionality) and thermochemical state variables obtained using the flamelets solutions is learnt using a neural network. However, due to the highly non-linear, knotted and discontinuous nature of the lower dimensional manifolds formed by the progress variables generated *a priori* the accuracy gained by a neural network is not satisfactory.

3 ChemTab: Joint Learning Progress Variables and Manifold Approximation

To reduce the computational effort in coupled simulations, state-space parametrization approaches follow a two-phase strategy. First, parametrize and tabulate *a priori* the scalar evolution of a reactive turbulent environment by few progress variables that govern the scalar evolution in a laminar flame. Second, use a tabular lookup at run-time to determine the high-dimensional chemical state required by the CFD solver. For instance, the FGM approach replaces all species and temperature by a *mixture-fraction* and a single *reaction progress variable* or reaction progress parameter. In this study, we focus on state-space parametrization using *Unsteady Flamelet Generated Manifolds* or Unsteady FGMs [3]. We modify this approach in three ways: the progress variable generation is different, the manifold is not tabulated and lastly, the progress variables and manifold approximation are done jointly.

3.1 Background: Unsteady FGM

FGM is a widely used tabulated chemistry method and can deal with a range of complicated conditions. FGM model shares the same theoretical basis with flamelet approaches [15], in which a multi-dimensional flame can be considered as an ensemble of multiple one-dimensional flames. Generally FGM model used for combustion modeling follows three steps as shown below:

1. Calculation of the representative 1-D flamelets.

2. Transformation of 1–D flamelets solutions to progress variables space.
3. Retrieval of thermo–chemical variables from the FGM tables according to FGM control variables from CFD simulations.

Table 1. Definitions for terms used in Sect. 3.1

	Description		Description
Z_{mix}	Mixture fraction	T	Temperature of the mixture
C_{pv}	Progress variable	\widehat{S}	Reactive scalars source terms
Y	Species mass fraction	ψ	Non–linear function of \widehat{Y} and Z_{mix}
\dot{S}	Species source terms	k	No. of species used to generate progress variables
ρ	Density of the mixture	p	Number of progress variables
\widehat{Y}	Reactive scalars	n	Number of data points
Le	Lewis number	\dot{S}_i	Source term of the i^{th} species
μ	Viscosities	$h_{f,i}^0$	Heat of formation of the i^{th} species
\mathcal{D}_i	Diffusivity of i^{th} species	h	Total enthalpy
κ	Thermal conductivity	s	Total no. of species in mechanism
Pr	Prandtl number	ϕ	Non–linear function of Y
Sc	Schmidt number	ζ	Non–linear function of \widehat{Y}

Governing Equations. Conservation equations for mass, species, momentum and energy for the 1–D, fully compressible, and viscous flames, are given by:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_x)}{\partial x} = 0 \quad (1)$$

$$\frac{\partial (\rho Y_i)}{\partial t} + \frac{\partial \rho u_x Y_i}{\partial x} = \frac{\partial}{\partial x} \left(\rho \mathcal{D}_i \frac{\partial Y_i}{\partial x} \right) + \dot{S}_i \quad (2)$$

$$\frac{\partial (\rho u_x)}{\partial t} + \frac{\partial (\rho u_x^2)}{\partial x} = -\frac{\partial p}{\partial x} + \frac{\partial}{\partial x} \left(\mu \frac{\partial u_x}{\partial x} \right) \quad (3)$$

$$\begin{aligned} \frac{\partial (\rho e_t)}{\partial t} + \frac{\partial}{\partial x} (\rho u_x H_t) &= \frac{\partial}{\partial x} \left(u_x \mu \frac{\partial u_x}{\partial x} \right) + \mu \frac{c_p}{Pr} \left(1 - \frac{1}{Le} \right) \frac{dT}{dx} \\ &+ \frac{1}{Sc} \frac{dh}{dx} - \sum \dot{S}_i h_{f,i}^o \end{aligned} \quad (4)$$

where the different terms are defined in Table 1.

We simplify the above equations making some well known assumptions. In 1D cartesian coordinates, the steady state solution to (1)–(4) is obtained only when the total mass flux is zero, i.e., velocity field is zero ($u_x = 0$) and so the four equations reduce to:

$$\frac{\partial}{\partial x} \left(\rho \mathcal{D}_i \frac{\partial Y_i}{\partial x} \right) + \dot{S}_i = 0 \quad (5)$$

$$\frac{\partial}{\partial x} \left(\kappa \frac{\partial T}{\partial x} + \sum \rho \mathcal{D}_i \frac{\partial Y_i}{\partial x} h_i \right) - \sum \dot{S}_i h_{f,i}^o = 0. \quad (6)$$

In (6), the final term in the energy equation is represented by the total sum of the product of all the source species and their respective heat of formation and is collectively called the source energy. Source energy is one of the crucial parameters in the combustion simulation and accurate chemistry description is required to define it. Prediction error of this term is used as the basis of comparison of our method against the other state of the art methods.

Flamelet Solutions. The data is generated by solving 1-D Steady State Flamelets differential equations in (6) using a finite volume PDE solver. The species Y and thermochemical state variables \dot{S} are generated using the solver.

$$Y = \begin{bmatrix} Y_{11} & \dots & Y_{1s} \\ \dots & \dots & \dots \\ Y_{n1} & \dots & Y_{ns} \end{bmatrix}, \quad \dot{S} = \begin{bmatrix} S_{11} & \dots & S_{1s} \\ \dots & \dots & \dots \\ S_{n1} & \dots & S_{ns} \end{bmatrix}, \quad Z_{mix} = \begin{bmatrix} Z_{mix_1} \\ \dots \\ Z_{mix_n} \end{bmatrix} \quad (7)$$

3.2 ChemTab

In ChemTab, the unsteady FGM approach is replaced with the following three steps:

1. Calculation of the representative 1D flamelets (data generation)
2. Using the data generated jointly generate Progress Variables (encoder) and Manifold Approximation (regressor) using ChemTab
3. Retrieval of thermo-chemical variables from the ChemTab-regressor according to progress variables from CFD simulations.

Formulation. The generated data described in (7) is then used by ChemTab. Conceptually the following equations summarize the relationships:

$$\dot{S} = \phi(Y) \quad (8)$$

$$S_{energy} = - \sum_i^s h_{f,i}^0 * \dot{S}_i \quad (9)$$

The two sub-problems of state-space parametrization are formulated as a joint optimization problems as follows:

$$\min \left(\sum_{i=1}^k \sum_{j=1}^n \|\dot{S}_{ij} - \zeta_i(\widehat{Y}_j)\|_t + \sum_{j=1}^n \|S_{energy} - \psi(\widehat{Y}, Z_{mix})\|_t \right) \quad (10)$$

$$\text{s.t. } t \in R \quad (11)$$

$$\widehat{Y} = Y \times W \quad (12)$$

$n \times p$ $n \times s$ $s \times p$

$$p \ll s \quad (13)$$

$$\|W\| = 1 \quad (14)$$

$$W^T \times W = I \quad (15)$$

$$(\widehat{Y} \oplus Z_{mix})^T \times (\widehat{Y} \oplus Z_{mix}) = I \quad (16)$$

$$\dot{S} \approx \widehat{S} = \zeta(\widehat{Y}, Z_{mix}) \quad (17)$$

$$S_{energy} \approx \widehat{S_{energy}} = \psi(\widehat{Y}, Z_{mix}) \quad (18)$$

The formulation described in Eq. 10 learns the optimal reactive scalars C_{pv} s (described by the embedding $Y \times W$) that along with Z_{mix} form the progress variables. This is a linear dimensionality reduction problem such that the new basis retains the inherent physics in higher dimensions described by the non-linear relation between Y and \dot{S} . To facilitate the development of transport equations using the progress variables it is necessary that the embedding of the variables in the low-dimensional space be linear. The constraints on the linear embedding are inspired by the work of [9] and the key ideas from PCA.

Implementation. The joint optimization problem is solved using a Deep Neural Architecture. ChemTab jointly optimizes two neural networks for the tasks of reaction progress variable generation (*encoder*) and manifold approximation (*regressor*). The encoder network focuses on linear dimensionality reduction and creates a linear embedding for the input. The regressor network focuses on learning the manifold approximation: a regression function whose input is the linear embedding and the output are the desired thermo-chemical state variables

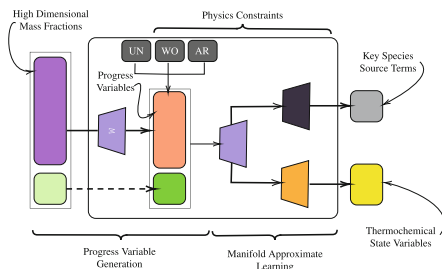


Fig. 2. ChemTab architecture

Table 2. Symbols used in Sect. 3.2

	Description		Description
f_θ	Prediction function	in	$s \times n$
y	Input/output matrix	s	Total no. of species in mechanism
W	Weight matrix	out	No. of thermo-chemical variables
b	Bias matrix	d_{in}	Input dimensions s
\mathcal{S}	Thermochemical state variables	L	No. of layers
σ	scalar/activation function	d_{out}	Output dimensions $s + 1$
o	Entry-wise operation	m	Number of neurons
n	Number of data points		

(Fig. 2 and Table 2).

$$\begin{aligned}
 f_{\theta}(y) &= W^{[L-1]} \sigma \circ (W^{[L-2]} \sigma \circ (\dots (W^{[1]} \sigma \circ \\
 &\quad (W^{[0]} y + b^{[0]}) + b^{[1]}) \dots) + b \\
 &\quad \text{where, } W^{[l]} \in R^{m_{l+1} \times m_l} \\
 b^{[l]} &= R^{m_{l+1}} \quad ; \quad m_0 = d_{in} = d \quad ; \quad m_L = d_{out}
 \end{aligned} \tag{19}$$

As described by (19) a Deep Neural Network can be conceptualized as a series of operations. The input of the network is the data for each of the species for each flame at each axial coordinate.

$$\begin{aligned}
 f_{\theta}^{[0]}(y) &= y \\
 f_{\theta}^{[1]}(y) &= (W^{[0]} f_{\theta}^{[0]}(y)) \\
 f_{\theta}^{[2]}(y) &= (f_{\theta}^{[2]}(y) \oplus Z_{mix}) \\
 f_{\theta}^{[l]}(y) &= \sigma \circ (W^{[l-1]} f_{\theta}^{[l-1]}(y) + b^{[l-1]}) \quad \forall \quad l \quad \text{s.t.} \quad 3 \leq l \leq L-1 \\
 f_{\theta}(y) &= f_{\theta}^{[L]}(y) = \sigma \circ (W^{[L-1]} f_{\theta}^{[L-1]}(y) + b^{[L-1]})
 \end{aligned} \tag{20}$$

As described by (20) the network is a layer-wise composition. The input of the network is reduced at the first layer linearly: this creates the linear embedding/reacting scalars (C_{pvs}). The next layer concatenates the conserved scalar Z_{mix} with the reacting scalars. These progress variables are then fed to the next layer. The subsequent layers together make up the regressor that learns a non-linear function between the progress variables and the thermo-chemical state variables.

$$\begin{aligned}
 \arg \min_{\theta} \quad & |f_{\theta}(y) - \mathcal{S}| \\
 \text{s.t.} \quad & W^{[0]T} W^{[0]} = I \\
 & \|W^{[0]}\| = 1 \\
 & f_{\theta}^{[2]}(y)^T f_{\theta}^{[2]}(y) = I
 \end{aligned} \tag{21}$$

As described by (21), ChemTab minimizes the Mean Absolute Error in predicting the thermo-chemical state variables (Source Energy in the current work) while ensuring that the linear embedding conforms to the following constraints:

1. Embedding Weights w learnt are unit norm (UN)
2. Embedding Weights w learned for the species mass fractions Y_i are uncorrelated/orthogonal (WO)
3. The reaction progress variables are uncorrelated/orthogonal (AR)

The constraints in (21) will be also added to the objective in addition to the predictions of key source terms, corresponding to a few important species, which serve as the physics constraints.

Extensions. The current framework and implementation can be very easily extended to include the prediction of additional thermochemical state variables and the projection of the embedding to get back the high dimensional mass fractions. These can be implemented as two other neural networks and their respective prediction errors can be added to the objective function.

4 Experimentation and Results

In this section we explain the specifics of the data set used, the training strategy, impact of the number of C_{pv} , comparison with the existing framework and relevant machine learning methods and the performance of the best model in the context of the multiple objectives.

4.1 Dataset

The training data was generated by solving 1-D Steady State Flamelets differential equations using a finite volume PDE solver. GRI-Mech 3.0 is one of the widely used Methane mechanism to model the reaction kinetics. This mechanism consists of 53 chemical species and 325 reactions.

The Flamelet solver discretizes the domain into 200 grid points (200 observations on the axial coordinate) in between the fuel and the air boundary and 100 flame are solved to steady-state. To train the model 20,000 data points (100 flames and 200 grid points) for a single pressure setting are used. Some of the generated data that represent extinguished flames were discarded, which led to exclusion of approximately 3,500 data points.

We experiment the model training and evaluation using two strategies:

1. *50% Flamelets* – Train using data from 50% of flamelets selected randomly and test using data from the remaining 50% of the flamelets, and,
2. *50% Data points* – Train using 50% data points selected randomly, and test on remaining 50% data.

4.2 Evaluation

We use the Mean Absolute Error of the Source Energy across the entire dataset as the metric to compare the performance as described in Eq. (21).

4.3 Implementation and Settings

We implemented ChemTab using Tensorflow 2.3.0, Keras and Adam optimizer. Models were trained on a server with Nvidia Quadro RTX 5000 GPU and cuDNN 8.0 and CUDA 11.0. We performed a coarse grid search on the hyperparameters (dropouts, learning rate, early stopping, batch size) & standard model architecture (number of layers, number of nodes in the layers, activation functions). After the initial model architecture and hyper-parameter search, all subsequent models in the subsequent studies were trained for 500 epochs. Results are reported as average over 10 runs (Table 3).

Table 3. Model parameters

Parameter	Value	Parameter	Value
Learning rate	0.001	Number of layers	11
Momentum	0.5	Layer shapes	32 64 128 256 512
Dropout	5%	Activation Functions	ReLU
Early stopping	Yes	Number of epochs	500 (short run) 20000 (long run)
Batch size	32	Network weight initialization	Uniform distribution

Table 4. ChemTab architectural variants

Abbreviation	Description
UN	Unit norm constraint on weights of the linear embedding
WO	Orthogonality constraint on weights of the linear embedding
AR	Orthogonality constraint on linear embedding concatenated with Z_{mix}
UN + WO	Unit norm constraint and orthogonality constraint on weights of the linear embedding
UN + AR	Unit norm constraint on the weights and orthogonality constraint on linear embedding concatenated with Z_{mix}
WO + AR	Orthogonality constraint on the weights and linear embedding concatenated with Z_{mix}
All	Unit norm and orthogonality constraint on the weights and linear embedding concatenated with Z_{mix}

4.4 Compared Methods

We compare the 7 variants of ChemTab with the relevant constraints on the Linear Embedding and the Progress Variables with a series of state-of-the-art baselines for Source Energy prediction Sect. 2.

4.5 Results

Current Framework Comparison. The current framework uses FGM based progress variables and Conformal Mapping based Tabulation and Lagrange Polynomial Interpolation based lookup. The tabulation was generated by using the entire data-set. The best MAE that the framework generated on the data-set was 2.243 E+09. The best ChemTab model trained on 50% of the data showed a 73% reduction in error. This reduction although high comes from the limitation of the current framework to include more than 2 progress variables and the realization of that through conformal mapping. We present a more principled comparison with the state-of-the-art methods in the next section.

Other Baseline Comparisons. We include *DNN-PVG(NL)-DNN* as reference although it cannot be used due to non-linear embedding. Similarly we

Table 5. Current state of the art methods and ChemTab variants

Method abbreviation	Progress variable generation
FGM-CPVG-DNN	FGM Constrained
PCA-PVG-DNN	PCA
DNN-PVG(NL)-DNN	Non-linear encoder
DNN-PVG(UL)-DNN	Unconstrained linear encoder
CT-PVG(ALL)-DNN	Physics constrained linear encoder Table 4
CT-PVG(UN)-DNN	Physics constrained (UN) linear encoder Table 4
CT-PVG(WO)-DNN	Physics constrained (WO) linear encoder Table 4
CT-PVG(AR)-DNN	Physics constrained (AR) linear encoder Table 4
CT-PVG(UN+WO)-DNN	Physics constrained (UN+WO) linear encoder Table 4
CT-PVG(UN+AR)-DNN	Physics constrained (UN+AR) linear encoder Table 4
CT-PVG(WO+AR)-DNN	Physics constrained (WO+AR) linear encoder Table 4

did not consider Gaussian Processes as there are several challenges with operationalization of Gaussian Process in our context and so we focus more on benchmarking against the relevant DNN based approaches (Table 5).

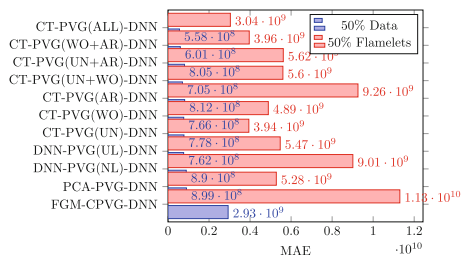
**Fig. 3.** MAE for source energy: data set split strategy

Figure 3 shows the results of an ablation study for both types of sampling strategies. When the trained using the sampled points, all models consistently do better than when trained using sampled flamelets. Essentially the flame is considered as an ensemble of multiple one-dimensional flamelets, each of which captures some of the highly nonlinear state-space and hence almost all models struggle in this training regime. ChemTab models still perform better and our assertions are that our constraints help in the generalization process. Our dataset is limited and so we limit ourselves to use only 50% of the data for training.

As we increase the number of C_{pv} the computational time of the flow simulation goes up, so we want to use the least number of C_{pv} while still capturing the essential physics. Figure 4 shows the MAE decreases with increase in the number of C_{pv} and then starts to increase again. As we add more C_{pv} the embedding has too many degrees of freedom and hence may start diverging.

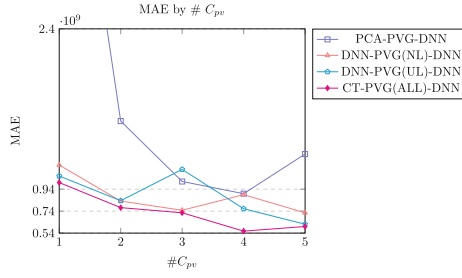


Fig. 4. MAE for source energy: Cpv ablation

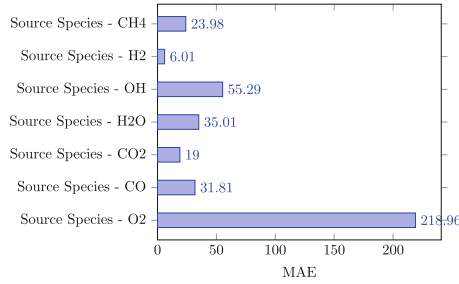


Fig. 5. MAE for key source terms – best model

Table 6. Constraints – best model

UN				WO			
w_1	w_2	w_3	w_4	w_1	w_2	w_3	w_4
1.004	1.005	1.001	0.998	1.004	-0.003	-0.002	0.005
				-0.003	1.004	-0.003	0.002
				-0.002	-0.003	1.001	0.001
				-0.005	0.002	0.001	0.99

AR					
	Z_{mix}	Cpv_1	Cpv_2	Cpv_3	Cpv_4
Z_{mix}	0.004	0.00	0.00	0.00	0.00
Cpv_1	0.00	0.008	-0.001	0.001	0.00
Cpv_2	0.00	-0.001	0.008	0.00	0.00
Cpv_3	0.00	0.001	0.00	0.007	-0.001
Cpv_4	0.00	0.00	0.00	-0.001	0.067

Best Model Performance. Table 6 shows the conformity of the constraints of Eq. 10. The first tabulation shows the (14) constraint conformity. The second tabulation shows the (15) constraint conformity and the third (16) constraint conformity. The (16) is also adequately satisfied as the constraint conformity is measured through covariance (Fig. 5).

Best Model Long Run Performance. We trained best model architecture on a 50% Data Points strategy for a long run of 20000 epochs and generated a MAE of $1.80\text{E}+08$.

5 Conclusion

We propose ChemTab, a novel framework for jointly learning the progress variables and the manifold approximation. ChemTab follows the principle of physics guided neural networks [10], however no solutions exist that can directly benefit the combustion community. ChemTab outperforms the state-of-the-art state-space parametrization in combustion. Crucially, ChemTab generated reaction progress variables can be interpreted by examining the weight matrix, W , and thus, allow for physical insights into the systems being modeled. Incorporation of ChemTab into a flow simulation will be explored as part of future work.

Acknowledgments. Funded by the United States Department of Energy’s (DoE) National Nuclear Security Administration (NNSA) under the Predictive Science Academic Alliance Program III (PSAAP III) at the University at Buffalo, under contract number DE-NA0003961.

References

1. Bhalla, S., Yao, M., Hickey, J.-P., Crowley, M.: Compact representation of a multi-dimensional combustion manifold using deep neural networks. In: Brefeld, U., Fromont, E., Hotho, A., Knobbe, A., Maathuis, M., Robardet, C. (eds.) ECML PKDD 2019. LNCS (LNAI), vol. 11908, pp. 602–617. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-46133-1_36
2. Biglari, A., Sutherland, J.C.: An a-posteriori evaluation of principal component analysis-based models for turbulent combustion simulations. *Combust. Flame* **162**(10), 4025–4035 (2015)
3. Bojko, B.T., DesJardin, P.E.: Formulation and assessment of flamelet-generated manifolds for reacting interfaces. *Combust. Flame* **173**, 296–306 (2016)
4. El-Asrag, H.A.: A comparison between two different flamelet reduced order manifolds for non-premixed turbulent flames. In: *Laminar & Turbulent Flames* (2013)
5. Fiorina, B., Gicquel, O., Carpentier, S., Darabiha, N.: Validation of the FPI chemistry reduction method for diluted nonadiabatic premixed flames. *Combust. Sci. Technol.* **176**(5–6), 785–797 (2004)
6. Giusti, A., Mastorakos, E.: Turbulent combustion modelling and experiments: recent trends and developments. *Flow Turbul. Combust.* **103**(4), 847–869 (2019)

7. Ihme, M., Cha, C., Pitsch, H.: Prediction of local extinction and re-ignition effects in non-premixed turbulent combustion using a flamelet/progress variable approach. *Proc. Combust. Inst.* **30**, 793–800 (2005)
8. Ihme, M., Schmitt, C., Pitsch, H.: Optimal artificial neural networks and tabulation methods for chemistry representation in les of a bluff-body swirl-stabilized flame. *Proc. Combust. Inst.* **32**, 1527–1535 (2009)
9. Ihme, M., Shunn, L., Zhang, J.: Regularization of reaction progress variable for application to flamelet-based combustion models. *J. Comput. Phys.* **231**(23), 7715–7721 (2012)
10. Karpatne, A., et al.: Theory-guided data science: a new paradigm for scientific discovery from data. *IEEE Trans. Knowl. Data Eng.* **29**(10), 2318–2331 (2017)
11. Maas, U., Pope, S.B.: Implementation of simplified chemical kinetics based on intrinsic low-dimensional manifolds. In: Symposium (International) on Combustion, vol. 24, pp. 103–112. Elsevier (1992)
12. Malik, M.R., Isaac, B.J., Coussement, A., Smith, P.J., Parente, A.: Principal component analysis coupled with nonlinear regression for chemistry reduction. *Combust. Flame* **187**, 30–41 (2018)
13. Malik, M.R., Obando Vega, P., Coussement, A., Parente, A.: Combustion modeling using principal component analysis: a posteriori validation on Sandia flames D, E and F. *Proc. Combust. Inst.* **38**, 2635–2643 (2020)
14. Perry, B.A., de Frahan, M.T.H., Yellapantula, S.: Evaluation of co-optimized machine-learned manifolds for modeling premixed combustion (2021). <https://ui.adsabs.harvard.edu/abs/2021APS.DFDF09003P/abstract>
15. Peters, N.: Laminar diffusion flamelet models in non-premixed turbulent combustion. *Prog. Energy Combust. Sci.* **10**(3), 319–339 (1984)
16. Peters, N.: *Turbulent combustion*. IOP Publishing (2001)
17. Pierce, C.D., Moin, P.: Progress-variable approach for large-eddy simulation of non-premixed turbulent combustion. *J. Fluid Mech.* **504**, 73–97 (2004)
18. Rastigejev, Y., Brenner, M.P., Jacob, D.J.: Spatial reduction algorithm for atmospheric chemical transport models. *Proc. Natl. Acad. Sci.* **104**(35), 13875–13880 (2007)
19. Smith, G.P., et al.: GRI-mech 3.0 is an optimized mechanism designed to model natural gas combustion, including no formation and reburn chemistry
20. Sutherland, J.C., Parente, A.: Combustion modeling using principal component analysis. *Proc. Combust. Inst.* **32**(1), 1563–1570 (2009)
21. Van Oijen, J., De Goey, L.: Modelling of premixed laminar flames using flamelet-generated manifolds. *Combust. Sci. Technol.* **161**(1), 113–137 (2000)
22. van Oijen, J., Lammers, F., de Goey, L.: Modeling of complex premixed burner systems by using flamelet-generated manifolds. *Combust. Flame* **127**(3), 2124–2134 (2001)
23. Willard, J., Jia, X., Xu, S., Steinbach, M., Kumar, V.: Integrating scientific knowledge with machine learning for engineering and environmental systems (2021)
24. Zhang, Y., Xu, S., Zhong, S., Bai, X.S., Wang, H., Yao, M.: Large eddy simulation of spray combustion using flamelet generated manifolds combined with artificial neural networks. *Energy AI* **2**, 100021 (2020)