Compact Representation of a Multi-dimensional Combustion Manifold Using Deep Neural Networks

Sushrut Bhalla[1 [0000−0002−4398−5052]], Matthew Yao[1 [0000−0001−6141−1477]], Jean-Pierre Hickey[1 [0000−0002−6944−3964]], and Mark Crowley[1 [0000−0003−3921−4762]]

University of Waterloo, Waterloo ON N2L 3G1, Canada
{sushrut.bhalla,matthew.yao,j6hickey,mcrowley}@uwaterloo.ca

Abstract. The computational challenges in turbulent combustion simulations stem from the physical complexities and multi-scale nature of the problem which make it intractable to compute scale-resolving simulations. For most engineering applications, the large scale separation between the flame (typically sub-millimeter scale) and the characteristic turbulent flow (typically centimeter or meter scale) allows us to evoke simplifying assumptions–such as done for the flamelet model–to pre-compute all the chemical reactions and map them to a low-order manifold. The resulting manifold is then tabulated and looked-up at run-time. As the physical complexity of combustion simulations increases (including radiation, soot formation, pressure variations etc.) the dimensionality of the resulting manifold grows which impedes an efficient tabulation and look-up. In this paper we present a novel approach to model the multi-dimensional combustion manifold. We approximate the combustion manifold using a neural network function approximator and use it to predict the temperature and composition of the reaction. We present a novel training procedure which is developed to generate a smooth output curve for temperature over the course of a reaction. We then evaluate our work against the current approach of tabulation with linear interpolation in combustion simulations. We also provide an ablation study of our training procedure in the context of over-fitting in our model. The combustion dataset used for the modeling of combustion of H2 and O2 in this work is released alongside this paper.

Keywords: Deep Learning · Combustion Manifold Modelling · Flamelet models.

1 Introduction

The field of turbulent combustion modeling is concerned with the prediction of complex chemical reactions–and the resulting heat release–coupled with an underlying turbulent flow field. Numerical combustion is a central design tool

within the fields of energy production, automotive engineering (internal combustion engines), and aerospace engineering (rocket engines, gas turbines). The computational challenges in turbulent combustion simulations stem from the physical complexities and multi-scale nature of the problem which make it intractable to compute scale-resolving simulations, in spite of having a set of equations governing the problem. For most engineering applications, the large scale separation between the flame (typically sub millimeter /microsecond scale) and the characteristic turbulent flow (typically centimeter or meter/minute or hour scale) allows us to evoke simplifying assumptions–such as done for the flamelet model– to pre-compute all the chemical reactions and map them to a low-order manifold; the resulting manifold can then be tabulated and looked-up at run-time. The main benefit of the flamelet model is that is allows a decoupling of the turbulent flow field (and the inherent mixing) and the chemical reactions. The Damköhler number, which represents a ratio of the time scale of the chemical reactions to the transport phenomena, is often used to bound the region of validity of the flamelet model. A simplified illustration of the flamelet model for non-premixed combustion is shown in Figure 1. With an increasing demand on the physical complexity of combustion simulations (through the inclusion of radiation, soot formation, pressure variations, wall-heat transfer etc.), the dimensionality of the resulting manifold increases and leads to the curse of dimensionality which impedes an efficient tabulation and look-up for engineering simulations. A central question in this field is how to efficiently model high dimensional manifolds necessary to capture the relevant physics of the problem and relating them to the chemical composition, pressures, and other factors computed during simulations. While the underlying governing equations for the physics are available, they are highly expensive computationally and not usable for the vast number of queries needed to run a combustion simulation of a realistic system. The ideal model would represent the well understood physical relationships in a fast, flexible form which could use varying dimensions as needed for a particular simulation or analysis task.

In the majority of the engineering-relevant, non-premixed combustion conditions, there is a large scale separation between the turbulent flow and the flame. This scale separation allows us to assume that a turbulent flame is comprised of a series of locally laminar flames, often called *flamelets*. This very convenient assumption is at the heart of flamelet modeling, one of the most common approaches in combustion modeling. For a given pressure, injection temperature, and fuel and oxidizer composition, all possible laminar flamelets can be uniquely characterized by the local *strain rate*, which is proportional to the velocity difference of the propellants in a counter-flow diffusion flame setup (see Figure 1 top right). As the laminar combustion only depends on the strain rate (for a given pressure, injector temperature and composition), all the flamelets can be pre-computed, tabulated and queried during run-time, which leads to a great computational advantage for the simulation. In the classical flamelet model, the combustion in any part of the computational domain can be uniquely defined by the local *mixture fraction*, Z (a conserved quantity which varies between 0 for

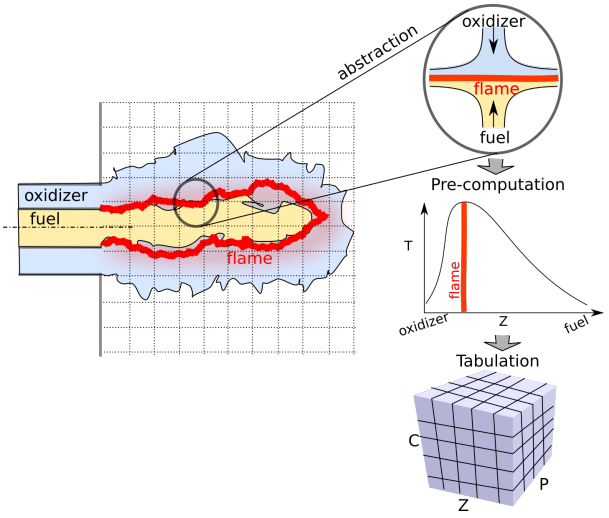

Fig. 1: A turbulent non-premixed flame is illustrated (left). The corrugated turbulent flame is assumed to be made up of locally laminar flames (as illustrated in the top right). The laminar flame (top right) can be computed, *a priori*, for all strain rates and tabulated for lookup at run-time. The tabulation is often done with respect to the known/transported variables from the simulation such as mixture fraction, Z, progress-variable C, variance of mixture fraction Z'', pressure, P (bottom right).

pure oxidizer and 1 for pure fuel), *strain rate*, χ (time scale ratio between the chemical reactions and the flow), and the *mixture fraction variance*, Z'' (which accounts for the turbulent mixing). More recent variations of the flamelet model have replaced the strain rate by a non-conservative *progress variable*, C. This latest model is called the *Flamelet-Progress Variable Approach (FPVA).* The reader interested in a more comprehensive overview of these combustion modeling paradigms is invited to consult [22].

In recent years, deep learning has been used to achieve state of the art results in image classification [5, 26], policy learning [18, 12] and natural language processing [25]. Deep learning has the ability to learn compact representations and can naturally handle interpolation of points which are not part of the training data. For the combustion modeling problem, our hypothesis is that deep learning can be used to learn a model of the higher dimension combustion manifold which can be used for simulation. In this paper we present an architecture to test this hypothesis and the techniques, including novel ones, we used to train a discriminative model for combustion. We show that we can reduce the combustion simulation running time and memory requirements compared to the current tabular methods. To examine the relative impact of each model component, we present an ablation study of our regularization methods and training techniques. We also propose an improved over-sampling procedure and a loss function which forces the model to focus on more difficult data points during training. Significant improvements in modeling for the combustion modeling problem could lead to a revolution in the ability to simulate complex combustion reactions and more

efficiently design better engines and power systems. This paper sets the groundwork for such a change by showing how to use deep learning to free modelers from the limits of tabular representations.

2 Related Work

Over the years the classical flamelet methodologies have been extended to tackle increasingly complex reactive flow problems. For example, the slow time-scale of NOx formation has motivated the use of unsteady flamelet [16] and the need to account for radiative heat loss effects due to the sensitivity to temperature of the NO formation has resulted in an FPV framework that includes an additional enthalpy term [7]. Similarly, the strong heat loss at the combustor walls has motivated a wall heat loss model [15, 24] which also includes an additional enthalpy term. Many combustion processes, such as liquid rocket combustion, are undertaken under variable pressure conditions which has been integrated into the flamelet framework [17]. Other extensions includes the consideration of multi-fuel systems [3]–which demand the use of two separate mixture fractions – or the inclusion of combustion chemistry that is highly sensitive to molecular diffusion (primarily in partially premixed settings) resulting in the introduction of multidimensional flamelet-generated manifolds (MFM) [20].

There is a growing demand to accurately represent the combustion process in more complex combustion scenarios such as those above. However, this requires a higher-dimensional manifold than the widely used FPVA method which commonly utilizes a three-dimensional tabulation of all the variables of interest, $\phi(Z, \widetilde{Z''^2}, c)$. Not only does the tabulated data occupy a larger portion of the available memory, the searching and retrieval of the pre-tabulated data becomes increasingly expensive in a higher-dimensional space. For example, assuming a standard flamelet table discretization of $(nZ, nZvar, nC) = (200, 100, 50)$ with say 15 tabulated 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. Recent consideration of a high-dimensional flamelet generated manifold for stratified-swirled flames with wall heat loss requires a 5D manifold which would be even larger [2]. This could only be achieved by using a very coarse discretization of the flamelet manifold (e.g. only 10 points are used to account for the variance of mixture fraction). One approach to address the large tabulative size has been proposed by using polynomial basis functions [27]; other approaches have looked at the use of Bézier patches [28]. These approaches provide adequate ways to reduce the tabulation size, retrieval time and improve accuracy, but face the inevitable curse of dimensionality. Other approaches seek to use principle component analysis to identify the optimal progress variable for the definition of a low-dimensional manifold [19]. Even if those approaches prove effective, inevitably many multi-physics systems will require a higher-dimensional space to adequately capture the relevant processes.

There has been some previous work on using machine learning for combustion manifold modeling. In 2009, a thorough experimental comparison was carried out by Ihme et al. [8] using a simple *Multi-Layer Perceptron (MLP)* for learning a mapping function to replace the tabular lookup method and thus speed up combustion simulation calculations. They showed that a neural network approach could be more generalizable but they found it had much worse accuracy than the tabular approach. This led to the method not being adopted for combustion simulator evaluations by the community. The approach in [8] was limited to a simple range of MLP variations and focused on the optimization of the network structure relative to a particular metric. They also did not have the benefit of modern deep neural network training techniques or regularization methods and used the classical approach of sigmoid activation functions rather than rectified linear units. In this work we focus on finding an optimal deep neural network training strategy which can achieve results approximately close to the true data curves. We show that our loss function and over sampling methods can achieve better accuracy than [8]. We also demonstrate our approach provides an acceleration over tabular methods for realistic combustion simulator evaluations by leveraging a CUDA enabled GPU. We also implement trained models for the species composition, temperature, source term and heat release.

Recent work [11] has focused on predicting the subgrid scale wrinkling of a flame by using a convolutional neural network. The work focuses on training an autoencoder with a U-Net network structure, which uses the current Direct Numerical Simulation (DNS) snapshot to predict the next DNS snapshot. The training data is a collection of 2 DNS of different flames; a third DNS is used for testing. This is a very simplified model to account for combustion and doesn't make any flamelet-based assumptions. Using a temporally dependent data structure for training limits the work in [11] and it cannot be easily extended to instantaneous combustion evaluations. Their model also requires the flame to be extrapolated to a certain length for the CNN model to be effective, as the autoencoder model they use would fail at the boundary conditions of the flame. In contrast, our model is built with the understanding that there can be multiple flames being simulated at a given instance and they could be running at different resolutions. Resolution in combustion simulations represents the factor by which the flamelet has been discretized based on the precision and accuracy required by the researcher. Thus, our model is built using fully connected layers which map the pressure, mixture fraction and progress variable to the species composition, temperature, source term and heat release.

3 Modeling Methods for High Dimensional Data

In this section we present the techniques we used to train our deep neural network model to predict the composition of the species (H, O_2, O, OH, H_2, H_2O, HO_2, H_2O_2), temperature (T), source term (W) and heat release (HR) of any flamelet given the pressure (P), progress variable (C) and mixture fraction (Z). The training data consists of flamelets for pressure values in the set

$(1, 2, 5, 10, 20, 30, 35, 40, 50)bar$ and the associated progress variable (C) and mixture fraction (Z). We reserve flames at pressure values $(15, 25, 33, 42)bar$ for testing purposes. The validation data set is generated by sampling sections of the flames in the training data set. The data was generated using FlameMaster, a 1D solver for the solution of the laminar, diffusion flamelet equations. The flamelets were generated at varying strain rates, from equilibrium combustion to nearly the quenched solution at varying base pressure levels. For all flamelets, the inflow temperature of the fuel and oxidizer remain constant. At each condition, the flamelets are solved with 1001 grid points with local mesh adaptation.

3.1 Neural Network Design

Table 1: Neural Network design used for prediction. All models use fully connected layers with *Leaky ReLU* as the activation function. Only layers in **bold** are regularized.

Prediction Output	Hidden Layers
Temperature (T)	(**64,128,512,512,1024**, 1024)
Source Term (W)	(**64,128,512,512,512**, 512)
Heat Release (HR)	(**64,128,512,512,1024,2048**, 2048)

Four different neural networks are designed for predicting the Species, Heat Release, Temperature and Source Term. The species show a high correlation in combustion time series as the total mass in the systems stays constant. In our experiments for species we predict the fraction of total mass which belongs to each species. In the neural network shown in Fig. 2 the network shares a common model for the first 7 layers and a separate fully connected head of 64 units to predict individual species. This improves the prediction accuracy and also reduces the memory requirements for the model. Note that we also introduce *function generators* (FG) to augment the input data with the function set $(sin, cos, square, exp, log)$ applied over the inputs (Z, C, P). The FG output is concatenated to the input of network. We found that using FGs leads to faster training convergence by providing common transformations right in the training data.

The network design for T and W is similar to the design used for the prediction of species, see Table. 1. However, even though T and W are highly correlated, their numerical scales are vastly different, thus, we use separate networks for them. We use *Leaky Rectified Linear Units (ReLu)* as the activation function for all layers in our neural network models unless otherwise specified [4]. All biases are initialized to 0.0 and weights are initialized using *He initialization* [6] with sampling from a normal distribution. All inputs to the network are standardized using Z-score standardization. We follow the same procedure for test set, where we use the mean and variance of the training data set to normalize

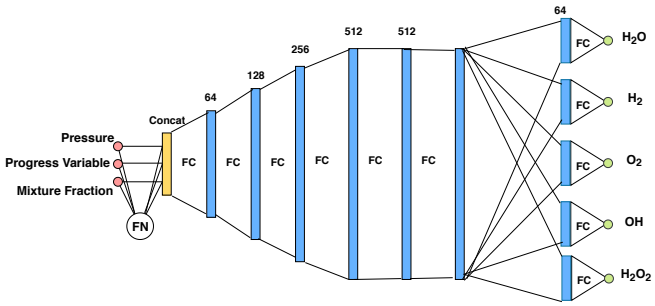

Fig. 2: Deep Neural Network structure used for training of Species. *FC* indicates fully connected layers. *FN* is the function generator.

the inputs. We use the *Adam optimizer* [10] with a learning rate of 0.001 and set the optimizer hyper-parameters to $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e-08$.

3.2 Over Sampling Hard Examples

For the remainder of the paper we focus on temperature which is the most essential prediction for combustion simulation. The temperature curve for a single combustion (H_2, O_2) flamelet looks like a parabola as shown in Fig. 3. The temperature curve shown, usually has relatively large gradients in the beginning and end (pure oxidiser or pure fuel). The semi-dome shape at the top of the temperature curve accounts for more than 50% of the input domain and shows the highest variability from flame to flame. When training deep neural networks with uniform sampling, they are easily able to learn the edges of the temperature curve as it shows low variability, however, the model performs poorly in the domain of high variability (the semi-dome). To improve this we employ oversampling of the "*hard examples*" defined as examples having error larger than median error of the batch:

$$HE(d)_{d\sim D} = \begin{cases} 1 & Error_{epoch_{t-1}}(d) \geq median(\overline{Error_{epoch_{t-1}}(batch_{smpl})}) \\ 0 & otherwise \end{cases} \quad (1)$$

New batches are created by sampling with replacement 75% hard / 25% easy examples from the sampling batch $batch_{smpl}$. We arrive at a value of 75% of hard examples through cross validation. A sampling batch refer to a uniformly sampled batch from the training dataset which is at least $2\times$ larger than the training batch size. Using $batch_{smpl}$ for our over-sampling reduces training time and memory requirements when compared to the rank based method employed by online batch selection methods [14].

3.3 Importance Weights Error and Gradient Clipping

Recent techniques for object detection and classification have shown that weighting the loss function separately for hard and easy examples leads to better model

predictions [13]. Training of deep neural networks or recurrent networks shows improvements in training stability by performing gradient clipping [21]. We incorporate these techniques in training our model for regression prediction. The gradient clipping clips large gradients in the backpropagation phase of neural network training and thus reduces any damage to our trained model caused by anomalies in our data. Through cross validation, we arrive at a value of 5.0 for gradient clipping. We choose to weight the cost function using a constant value of $\alpha = 0.4$ (in Eq. 2) for all easy data-points ($d \sim D$). The data points with loss in the lower 25 percentile are considered easy. This approach leads to good results in the semi-dome area of the temperature curve because it weighs down the contribution of a large number of easy examples with small gradient updates and allows the training to focus on the data points (d) with higher cost. The approach can also be used as a replacement to over sampling of hard examples. The new loss function is shown in Eq. 2.

$$L_{imp}(d)_{d\sim D} = \begin{cases} \alpha \times Cost(T(d), \hat{T}(d)) & loss(d) \leq rank_{25\%}(loss(batch_{smpl})) \\ Cost(T(d), \hat{T}(d)) & otherwise \end{cases} \tag{2}$$

3.4 Regularization

Over-fitting leads to large oscillation on the training data set which would lead to poor performance in numerical simulators used for modeling combustion. To reduce the chance of oscillation we combine $L1$ and $L2$ regularizers with an *L1L2* regularized loss function given by the following formula:

$$Loss = Cost(batch) + \lambda_{l1} \times \sum_i (|NN_{w_i}|) + \lambda_{l2} \times \sum_i (NN_{w_i})^2 \tag{3}$$

Through cross-validation over the accuracy metric and qualitative assessments of the smoothness of the predicted temperature curve, we arrive at the values of $\lambda_{l1} = 0.00015$ and $\lambda_{l2} = 0.000125$. The *L1L2* cost adds a regularization term to the total loss to be jointly minimized by the optimizer. This forces the network to predict with smaller values of weights which produces smoother predictions of the temperature curve.

We tested other standard regularization techniques including Batch Normalization (BN) [9], Dropout [23] and Layer Normalization (LN) [1] for the regression tasks. BN normalizes the output of a hidden unit based on the mean and variance of training data, this performs poorly when the testing data set is vastly different from the training data. Dropout on the other hand, does not perform well during training. Dropout works by dropping a connection between hidden units with a probability $p = 0.5$ (typical for classification tasks) and optimizing the sub-network. In regression prediction, the scaling factor p used in dropout leads to training predictions lower than the true data and the reverse is true during the testing phase (due to weight scaling). The network with dropout is not able to train well on toy regression problems (for the function $f(x) = x^2$). LN

normalizes the input features of the hidden layers on a per data point basis and thus doesn't exhibit the problems from batch normalization. LN seems to be a better fit than batch normalization and dropout for regression tasks, however as we show in the experiments section, the performance with $L1L2$ regularization is better.

3.5 Ensemble Model

Many machine learning tasks have shown improved results by using an ensemble of models. In regression, ensembles of models would be highly beneficial as it would reduce the amount of regularization required per model and the accuracy could be improved with careful selection of a set of prediction models. We design an averaging ensemble of trained models. We train five deep neural network models for prediction of the temperature curve. The ensemble model also helps in improving the accuracy of the model. We compute the final prediction by averaging the four best (in terms of deviation from the mean prediction) individual models. This allows us to ignore results from models which didn't perform well on certain data-points and pick the best of all trained models.

4 Results and Discussions

4.1 Quantitative Analysis

Table 2 shows the accuracy and the loss for each prediction variable. A prediction is considered accurate if the model prediction ($\hat{T}(d)$) is within a standard error range E_T, as shown in the formula below for the temperature model:

$$accuracy_{d\sim D} = \begin{cases} 1 & \hat{T}(d) \in \{T(d) - E_T, T(d) + E_T\} \\ 0 & otherwise \end{cases} \tag{4}$$

The value of E_O (where $O \in \{T, W, HR, Species\}$) is computed based on the resolution of the combustion simulator and an expert's opinion. The value of $0.005 \times range$ is used based on the discretization error in typical combustion simulations arising from tabulation method. We detail the values of E_O for each output label in the Table 2. Accuracy is used for model comparison since using the resulting MSE could be misleading. That's because the model could be performing well on easy examples (head and tail of T curves) and poorly on the hard examples (mostly in the semi-dome part of the curve). This would result in a lower loss value but poor accuracy for simulator evaluation purposes.

We focus on the predictions of T and W for our work as their performance is vital for our model to be useful for the combustion community. The mean error results in Table. 2 are computed using the formula $ME = \frac{1}{N}\sum_1^N(|y - \hat{y}|)$. For temperature, we see a mean error of 82.44 which is the deviation from the tabulated data on test data set. We show using our simulation tests that this error is sustainable and can be used in combustion simulators. The accuracy of

Table 2: Quantitative analysis of predictions of species composition, temperature (T), source term (W) and Heat release (HR) using the regression model in Fig. 2

	Range $(O_{MAX} - O_{MIN})$	Standard Error Range (E_O)	Mean Error	Training Accuracy	Validation Accuracy
H	0.0219	0.0001	0.00431	29.23	21.44
O_2	1.0	0.005	0.3419	20.17	12.56
O	0.0665	0.00033	0.00892	55.79	37.43
OH	0.1279	0.00064	0.0345	56.91	43.34
H_2	1.0	0.005	0.2606	38.40	25.41
H_2O	0.8865	0.0044	0.6056	26.19	14.59
HO_2	0.0142	$7.1e-05$	0.00137	60.36	41.42
H_2O_2	0.0091	$4.5e-05$	0.00259	55.92	36.05
$HR(J/m^3)$	$93.4e+81$	$6.4e+78$	$1.05e+76$	81.23	81.18
$T(K)$	3295	34.47	82.44	61.97	54.60
W	30	0.149	3.023	71.55	58.144

54.60% represents the number of data points which are predicted within a range of $34.47K$ range of the tabulated data. We see similar accuracy for source term (W) with a mean error value of 3.023. The R^2 statistic for our model predictions of T and achieve an average value of 0.94 per flamelet predictions. The model is able to capture a large part of the combustion manifold and can be used for evaluation with combustion simulators.

4.2 Ablation study and qualitative analysis

In this section we present an ablation study of our training methodology by varying or removing components and comparing the quantitative results. We also support this with qualitative results where we pick a flamelet from the test data set and analyze the predicted temperature curve generated by our model. The curve used for the qualitative study in this section is for $P = 25bar$. As mentioned, the closest pressure values in the training data are at $20bar$ and $30bar$. Thus the ablation results also provide us an insight into the interpolation skills of the deep neural network model. The flamelet used for comparison was selected at random and represents an above average performance case of our deep neural network model.

Table 3: Quantitative results for regularization of NN on combustion manifold.

	No Regularizer	*L1L2* Regularizer	Layer Normalization	Batch Normalization
Training Accuracy	87.59	59.01	44.53	61.68
Testing Accuracy	31.44	39.83	23.63	12.28

Ablation study of regularization techniques. Fig. 3(a) shows the qualitative results for training the neural network with *L1L2* regularization and without any regularization. The *L1L2* regularization forces the weights of the neural network to be closer to 0. This affects the temperature curve generated by our regularized model as seen in Fig. 3(a). The curve shows very little oscillations which is necessary condition for our model to be used in real combustion simulators. Due to the restriction on the weights of the neural network, our model suffers in terms of performance as we get a temperature curve distant from the true data curve. Table. 3 summarizes the performance of using different regularization techniques. The value of $\lambda_{L1}, \lambda_{L2}$ is set to $0.00015, 0.000125$ respectively. We see that *L1L2* regularization performs the best on test data set. We thus use *L1L2* for all the experiments. BN is also able to train well on the training data set, but performs poorly when the distribution of the data is changed during testing. BN was also not able to scale to larger batch sizes and thus suffered in performance when compared to other methods. LN alleviates the dependence on the training data and normalizes each layer's input features. This results in better performance, and the results of LN are more predictable than batch normalization.

Ablation study of over sampling. Next we examine the incremental effect of using over sampling of hard examples from the data set based on the mean square error for each data-point in the previous training epoch. Fig. 3(b) shows the comparison of the predicted temperature curves for using over-sampling compared to basic uniform sampling. Both training methods use *L1L2* regularization with the same hyper-parameters as last subsection. The results clearly show that over sampling is able to better approximate the temperature curve. The over sampling of hard examples allows the network to improve its predictions on semi-dome structure near the top of the temperature curve. The network is able to approximate the beginning and end of the temperature curve as seen with a naive neural network model in Fig. 3(b). The plot shows the temperature (same as previous subsection) curve for a flame at pressure value of $25bar$ over the mixture fraction. The mixture fraction denotes the progress of the $3-D$ combustion flame. Table. 4 shows the quantitative improvements in the prediction accuracy of the temperature curves for over-sampling with the same neural network structure and training hyper-parameters used for uniform sampling case. The results show a direct improvement in the qualitative curve structure and the accuracy metric. The fraction of hard examples per batch was set to 0.75 through cross validation.

Ablation study of ensemble model. Next we present an ablation study of the ensemble model used in our work. The Fig. 3(c) shows the comparison of predicted temperature curves for the ensemble method against the best single model temperature curve achieved using over sampling. While regularization reduces the oscillation in the neural network predictions, the over-sampling technique forces the network to shift its weights to focus on the hard examples. This

Table 4: Quantitative results for over sampling of NN on combustion manifold.

	Uniform Sampling	Over Sampling	Ensemble Model
Training Accuracy	59.01	63.58	61.97
Testing Accuracy	39.83	48.87	47.73

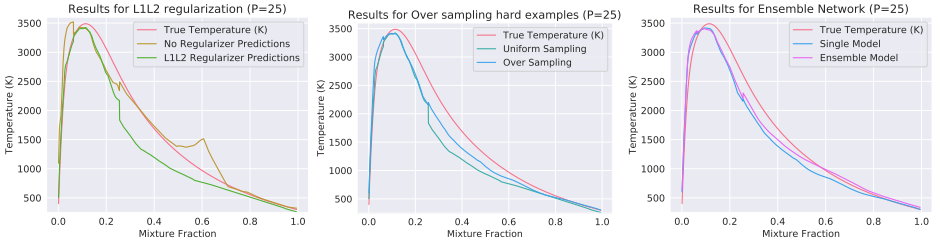

Fig. 3: Ablation study of (a) Regularization techniques, (b) Over Sampling for Neural Network Training and (c) Ensemble models.

reduces the regularization effect on the network's predictions. Thus, to further reduce the oscillations in the predicted model, we use an ensemble of neural networks. The neural networks differ based on the number of hidden layers, size of hidden layers, weight initialization technique, optimization technique (RMSProp or ADAM optimizer). These variations force each model to converge to a different saddle point in the optimization landscape. The ensemble method achieves better results than the individual model in terms of accuracy and smoothness of the predicted temperature curve. The Fig. 3(c) shows the results for an averaging ensemble network comprised of 4 separately trained neural networks. The Table. 4 shows the quantitative comparison of using an ensemble method when compared to the best single deep neural network model.

Ablation study of importance loss weighting. In this subsection, we present an ablation study for importance weighting the loss of easy and hard data-points. The reduced loss of easy examples, reduces the gradients and thus affects a smaller change in the weights of the neural network. Hard examples, on the other hand, can be highly weighted which leads to large gradients in the neural network and thus the model over-fits to the hard examples. We use cross-validation to arrive at a value of constant error weights of 0.4 for easy examples and 1.0 for hard examples. Table. 5 shows the quantitative improvements in the training and test accuracy of the same neural network model with and without importance loss weighting. The importance loss weighting is an important aspect of our training procedure as it can lead to large accuracy improvements in some cases.

Table 5: Comparison of impact on accuracy for differing levels of α in Eq. 2 for hard and easy data. Uses the Mean Square Error cost function.

	H	O_2	O	OH	H_2	H_2O	HO_2	H_2O_2	$T(K)$	W
$\alpha = 1.0$	29.23	18.05	55.79	45.58	38.40	60.56	60.36	55.92	61.97	
$\alpha = 0.4$	46.57	43.6	62.3	51.6	38.51	85.76	62.14	67.41	63.64	

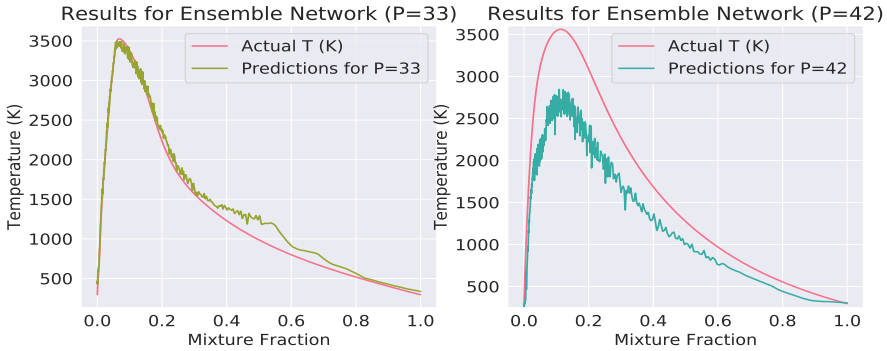

Fig. 4: Study of the Consistency of the Training Data.

4.3 Study of the Consistency of the Training Data.

The training data used for our experiments uses pressure values in the set $(1, 2, 5, 10, 20, 30, 35, 40, 50)bar$. The validation data is generated by reserving 20% of the training data for validation purposes. This approach ensures that the training data and the validation data share the data distribution. The testing data set is comprised of pressure values the neural network has never witnessed before and thus our model must precisely interpolate between the training data-points to generate the accurate temperature curves for the testing data-set. The testing data-set is comprised of pressure values in the set $(15, 25, 33, 42)$. We use our best trained single neural network model to predict the interpolation results for all pressure values in the test set. The model achieve an accuracy of 72.21% and mean error of $68.47K$ for pressure of $33bar$. The model achieve an accuracy of 38.83% and mean error of $661.35K$ for pressure of $42bar$. Fig. 4 shows the qualitative comparison for the interpolation results of the neural network model at pressure value of $(33, 42)bar$. Through experiments on pressure values of $15, 25bar$, we see similar performance. Based on the experimental evaluations, we should be training our model at intervals of $5bar$ to achieve better accuracy at interpolated pressure values.

4.4 Model Analysis using Combustion Simulator

The efficacy of DNN-based flamelet combustion modelling rests on two important parameters for turbulent combustion simulations. First, the modelling of the high-dimensional manifold should result in a smaller memory footprint compared to traditional tabular approaches. For a three-dimensional manifold, a

Table 6: Memory Requirements and Inference Time Analysis

	Tabulation Method	Deep Neural Networks
Parallel Inference Time (in ms)	1.2×10^5 ms	13.92
Serial Inference Time (in s)	10.997 ms	55.27
Memory Requirements	184.64 MB	24.158 MB

compact representation takes up about 8 times less memory compared to an adequately resolved table. With increasing dimensionality of the combustion manifold (inclusion of wall heating, NO_x computations etc.) the impact on the DNN model is modest whereas we typically expect a two decade increase in the table size per additional dimension. Second, the query time should be quick for $N-$dimensional interpolations for simulator evaluations as the size of data and dimensionality increases. Table. 6 shows the query time for $50,000$ data points performed in serial and batch fashions. The simulator evaluations can be easily parallelized to take advantage of the batched inference time.

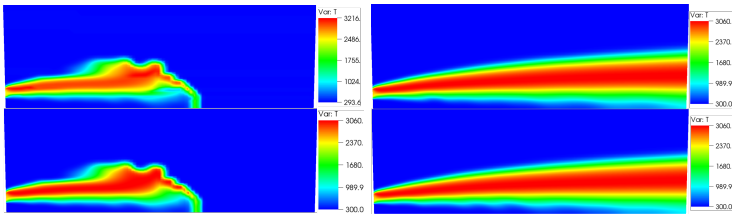

Fig. 5: Comparison of simulation using DNN data (top) and tabulated data (bottom).

As a proof of concept test, a simulation of a piloted hydrogen/oxygen diffusion flame was conducted using OpenFOAM. At a single timestep, the calculated temperature field was replaced using a predicted temperature field obtained via the deep neural net predictions. The solution was then advanced using the newly predicted temperature field. The results are shown in Figure 5. Although the DNN predicted slightly higher temperature values, both solutions reach the same steady state.

5 Conclusion

In this paper we presented a novel training procedure for approximating high dimensional combustion manifolds for use in combustion simulations using deep neural networks. We propose a novel loss function for regression tasks with examples of varying degree of difficulty. We also propose a fast over sampling methodology based on the cost of each data point. The proposed model achieves sufficient accuracy when compared with tabulated data and runs fast enough to integrate into high dimensional multi-physics simulators of combustion. Our

model allows for cheap computation of very complex physics, compared to the traditional tabulation methods which can not scale to high dimensions. We plan to extend this work to focus on dimensionality reduction to understand core aspects of the combustion manifold. Our prediction model for HR does not perform adequately as the output range of HR is large (10^{81}). The drawering technique discussed in [29] can be used to reduce the complexity of learning the range of HR.

References

1. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
2. Donini, A., M. Bastiaans, R.J., van Oijen, J.A., H. de Goey, L.P.: A 5-D Implementation of FGM for the Large Eddy Simulation of a Stratified Swirled Flame with Heat Loss in a Gas Turbine Combustor, vol. 98. Flow, Turbulence and Combustion (2017)
3. Felsch, C., Gauding, M., Hasse, C., Vogel, S., Peters, N.: An extended flamelet model for multiple injections in di Diesel engines. Proceedings of the Combustion Institute **32 II**, 2775–2783 (2009)
4. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. pp. 315–323 (2011)
5. Graham, B.: Fractional max-pooling. arXiv preprint arXiv:1412.6071 (2014)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
7. Ihme, M., Pitsch, H.: Modeling of radiation and nitric oxide formation in turbulent nonpremixed flames using a flamelet/progress variable formulation. Physics of Fluids **20**(5) (2008)
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. Proceedings of the Combustion Institute **32 I**(1), 1527–1535 (2009),
9. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
11. Lapeyre, C., Misdariis, A., Cazard, N., Veynante, D., Poinsot, T.: Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates. arXiv preprint arXiv:1810.03691 (2018)
12. Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971 (2015)
13. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
14. Loshchilov, I., Hutter, F.: Online batch selection for faster training of neural networks. arXiv preprint arXiv:1511.06343 (2015)

15. Ma, P.C., Wu, H., Ihme, M., Hickey, J.P.: Nonadiabatic Flamelet Formulation for Predicting Wall Heat Transfer in Rocket Engines. AIAA Journal **56**(6), 1–14 (2018),
16. Mauss, F., Keller, D., Peters, N.: A lagrangian simulation of flamelet extinction and re-ignition in turbulent jet diffusion flames. Symposium (International) on Combustion **23**(1), 693–698 (1991)
17. Mittal, V., Pitsch, H.: A flamelet model for premixed combustion under variable pressure conditions. Proceedings of the Combustion Institute **34**(2), 2995–3003 (2013),
18. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529 (2015)
19. Najafi-Yazdi, A., Cuenot, B., Mongeau, L.: Systematic definition of progress variables and Intrinsically Low-Dimensional, Flamelet Generated Manifolds for chemistry tabulation. Combustion and Flame **159**(3), 1197–1204 (2012),
20. Nguyen, P.D., Vervisch, L., Subramanian, V., Domingo, P.: Multidimensional flamelet-generated manifolds for partially premixed combustion. Combustion and Flame **157**(1), 43–61 (2010). https://doi.org/10.1016/j.combustflame.2009.07.008,
21. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: International conference on machine learning. pp. 1310–1318 (2013)
22. Poinsot, T., Veynante, D.: Theoretical and numerical combustion. RT Edwards, Inc. (2005)
23. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research **15**(1), 1929–1958 (2014)
24. Trisjono, P., Kleinheinz, K., Pitsch, H., Kang, S.: Large eddy simulation of stratified and sheared flames of a premixed turbulent stratified flame burner using a flamelet model with heat loss, vol. 92 (2014)
25. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems. pp. 5998–6008 (2017)
26. Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., Fergus, R.: Regularization of neural networks using dropconnect. In: International conference on machine learning. pp. 1058–1066 (2013)
27. Weise, S., Popp, S., Messig, D., Hasse, C.: A Computationally Efficient Implementation of Tabulated Combustion Chemistry based on Polynomials and Automatic Source Code Generation. Flow, Turbulence and Combustion **100**(1), 119–146 (2018)
28. Yao, M., Mortada, M., Devaud, C., Hickey, J.P.: Locally-Adaptive Tabulation of Low-Dimensional Manifolds using Bezier Patch Reconstruction. Spring Technical Meeting of the Combustion Institute Canadian Section (2017)
29. Żołna, K.: Improving the performance of neural networks in regression tasks using drawering. In: 2017 International Joint Conference on Neural Networks (IJCNN). pp. 2533–2538. IEEE (2017)