Aerodynamic Data Predictions Based on Multi-task Learning

Liwei Hu
School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, Sichuan, 611731, China

Yu Xiang[*]
School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, Sichuan, 611731, China
jcxiang@uestc.edu.cn

Jun Zhang
School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, Sichuan, 611731, China

Zifang Shi
School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, Sichuan, 611731, China

Wenzheng Wang
China Aerodynamics Research and Development Center
Mianyang, Sichuna, 621000, China

October 20, 2020

ABSTRACT

The quality of datasets is one of the key factors that affect the accuracy of aerodynamic data models. For example, in the uniformly sampled Burgers' dataset, the insufficient high-speed data is overwhelmed by massive low-speed data. Predicting high-speed data is more difficult than predicting low-speed data, owing to that the number of high-speed data is limited, i.e. the quality of the Burgers' dataset is not satisfactory. To improve the quality of datasets, traditional methods usually employ the data resampling technology to produce enough data for the insufficient parts in the original datasets before modeling, which increases computational costs. Recently, the mixtures of experts have been used in natural language processing to deal with different parts of sentences, which provides a solution for eliminating the need for data resampling in aerodynamic data modeling. Motivated by this, we propose the multi-task learning (MTL), a datasets quality-adaptive learning scheme, which combines task allocation and aerodynamic characteristics learning together to disperse the pressure of the entire learning task. The task allocation divides a whole learning task into several independent subtasks, while the aerodynamic characteristics learning learns these subtasks simultaneously to achieve better precision. Two experiments with poor quality datasets are conducted to verify the data

---

[*]Corresponding Author, Professor

quality-adaptivity of the MTL to datasets. The results show than the MTL is more accurate than FCNs and GANs in poor quality datasets.

Keywords Muti-task learning · Aerodynamic data modeling · ClusterNet · K-means · Machine learning

1 Introduction

In the field of aerodynamic data predictions, the quality of datasets is one of the key factors that influence the accuracy of prediction models [1]. For example, in Burgers' equation, uniform sampling results in limited number of data with high-speed, see Fig.1. Limited high-speed data are submerged in lots of low-speed data, which leads to the inability for neural networks to predict high-speed aerodynamic data. This phenomenon, more common in unsteady flow fields [2], increases the difficulty of modeling and also reduce the accuracy of models.

Figure 1: The uniform sampling in Burgers' equation. x denotes the displacement and u denotes the velocity.

To improve the quality of datasets, data resampling technology is widely used in this field [3]. The aim of data resampling is to produce data conforming to the law of aerodynamic variations for the parts of flow fields with insufficient data to alleviate the defects caused by poor quality of datasets. Generally, the studies of data resampling can be divided into two categories: data recalculation and data generation.

Data recalculation is a traditional method that relies on computational fluid dynamics (CFD) solvers [4]. In aerodynamics, the CFD solvers usually simulate Navier-Stokes equations to express the variations of air flows [5]. However, even optimal CFD solvers, e.g. SU2 [6] or Fluent [7], take weeks to complete a high-dimensional aerodynamic calculation, which affects the instantaneity of predictions about fluid dynamics [8].

Data generation, usually based on neural networks, achieves great success both in the field of machine learning and aerodynamics [9]. This method adopts generative models to generate plentiful aerodynamic data. Among these generative models, the generative adversarial network (GAN) [10], the most promising deep learning model in recent years, is pervasively used in aerodynamic data generation [9]. The recursive CNNs-based GAN in unsteady flow predictions [11], the RNNs-based GAN in the analyses of the temporal continuity [12], the tempoGAN for high-resolution flow field image generations [13] and the RBF-GAN/RBFC-GAN for nonlinear sparse flow field data generation [14] are outstanding models.

Recently, in the field of natural language processing, the mixtures of experts have been applied to analyze complex sentences, which have achieved great success [15]. The idea of mixtures of experts is that different areas in a neural network are responsible for different parts of speech, i.e. a small area in a neural network is dedicated to processing a specific part, not the entire sentence. This mechanism provides a solution for eliminating the need for data resampling in aerodynamic data modeling, owing to that parts of data in flow fields can be processed by a dedicated area in neural networks-based aerodynamic models.

Motivated by this idea, we propose the multi-task learning (MTL) scheme which consists of two parts: task allocation and aerodynamic characteristics learning. The task allocation divides a complete aerodynamic task (dataset) into multiple subtasks (subsets) according to specific methods. In this paper, the data partition or K-means method [16] is employed to achieve this goal. Data partition, a vertical task allocation, divides a whole dataset into several subsets based on a certain dimension in the dataset. While K-means, a horizontal task allocation, clusters the samples in a dataset into multiple subsets. The aerodynamic characteristics learning learns the aerodynamic variations contained in each subsets at the same time, and outputs the corresponding predictions based on given inputs. In this paper, the ClusterNet is applied to implement the aerodynamic characteristics learning [17]. Two experiments with poor quality

of datasets are conducted to verify the feasibility of the MTL. Compared with fully connected networks (FCNs) and GANs, the MTL can accurately predict the velocity u, the coefficient of pressure C_P and the friction F_x in the entire flow field from datasets with poor quality. The results reflects than the MTL is a dataset quality-adaptive learning scheme. Besides, the MTL explains in detail how the ClusterNet automatically recognizes regions in datasets, which was not explained in the original paper [17].

To summarize, the contributions of our work are:

a) we propose the MTL in aerodynamic data modeling;

b) our work eliminates the data resampling, while dealing with poor quality aerodynamic datasets;

c) our work, for the first time, explains in detail how the ClusterNet automatically recognizes regions in datasets.

The structure of the remainder of this paper is as follows. Section II introduces the research status of K-means clustering algorithm and the ClusterNet in the field of aerodynamic response predictions. In Section III, the details of MTL are elaborated. In Section IV, two typical datasets from different scenes are applied to validate the effectiveness of the proposed scheme. The conclusion of our work is shown in Section V.

2 Previous Work

The MTL consists of two parts: task allocation and the aerodynamic characteristics learning. As the task allocation mainly uses the K-means method, and the aerodynamic characteristics learning mainly relies on the clusterNet model, in this section, we introduce the previous work of the K-means method and the ClusterNet model.

2.1 K-means Method

Ball presented the K-means method in 1967 and since then it has become the most popular of the clustering algorithms [16]. In aerodynamics, K-means method is widely used to data clustering and parameters estimation.

Organizing aerodynamic data into multiple sensible groups is a foundation of MTL. Usually, it is difficult to divide a whole aerodynamic dataset into several subsets even for experts, owing to the complex variations of air flows. K-means, as a fast and flexible unsupervised learning method, provides a viable solution [18]. [19] adopted k-means to obtain representative points on the upper and lower surface of a airfoil. [20] applied K-means to optimize data distribution to cover the input space as uniform as possible. [21] replaces the hierarchical clustering algorithm with a K-means that is suitable for large unstructured grids.

K-means algorithms are also used to estimate the hyper parameters. For example, the centers of a radial basis function can be determined by clustering the input data [22]. Besides, the parameters (weights and biases) of other models can also be initialized with k-menas [23, 24].

Different from the above studies, we employ K-means to clustering aerodynamic data into different independent subsets to provide learning labels for MTL.

2.2 ClusterNet Model

For general neural networks, each neuron will be activated both in training and testing process, which is difficult to adapt to poor quality learning samples. Although, large-scale neural networks seems to be a feasible approach [25, 26], the computational costs increase dramatically. The reason for this is that all neurons are activated for each sample, which is clearly different from the way the brain processes informations from real world [27].

The origin of ClusterNet can be traced back to mixtures of experts (i.e. dedicated neural networks) in natural language processing [28]. The idea of mixtures of experts is that different experts are responsible for analyzing different parts of speeches. The final results are obtained by linearly weighting results of all experts [29]. The experts can be implemented by different models, such as SVMs [30], gaussian processes [31], neural networks [15], etc. After 2019, the mechanism of partial activation of neurons is developed in the form of clusterNet in aerodynamics.

The clusterNet is a novel neural network in that different clusters automatically identify different regions in datasets. [17] proposed the ClusterNet in 2020, which is used to predict the velocity of air flows. Compared with general neural networks, ClusterNet reduced the errors of predicted velocity. [32] adopted a clusterNet-based physics-informed model to predict future trajectories of the swarm by approximating the nonlinear dynamics of the swarm model, which is more stable than other models in nonlinear ordinary differential equations systems.

However, how the model automatically identify different regions in datasets and why the ClusterNet is better than others is not analyzed in detail. Therefore, we combine task allocation and the ClusterNet to form MTL, which effectively explains the mechanism of the ClusterNet.

3 Methodology

3.1 Overview

In this section, we introduce the MTL which consists of task allocation and aerodynamic characteristics learning, see Fig.2 (a). As for the task allocation, we use data partition or K-means method to divide a whole learning task into multiple independent subtasks according to specific rules. As for the ClusterNet, we use both a classification loss function and a regression loss function to learn these subtasks at the same time to achieve the purpose of partial activation.

Figure 2: (a) The scheme of MTL in aerodynamic data modeling. (b) The structure of a ClusterNet, in which the pink neurons denote the function network and the gray neurons denote the context network.

3.2 Task Allocation

Task allocation divides a given dataset into multiple independent subsets. Given a dataset $\mathbf{D} \subseteq \mathbf{R}^{n \times m}$, the goal of task allocation is to divide $\mathbf{D}$ into k independent subsets $\hat{\mathbf{D}} = \{\hat{\mathbf{D}}_z | \cap_{z=1}^{k} \hat{\mathbf{D}}_z = \emptyset\}$. In this paper, we provide two alternative methods: data partition and K-means method.

3.2.1 Data Partition

We analyze $\mathbf{D}$ from the perspective of column vector, i.e. $\mathbf{D} = \{\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_m\}$, where $\mathbf{d}_i$ denotes the ith dimension of $\mathbf{D}$. Assuming that for any $v \in \mathbf{d}_i$ satisfies $a \leq v < b$, then $\hat{v}$ satisfies:

$$
\begin{cases}
a + \sum_{j=0}^{z-1} \lambda_{ij} \leq \hat{v} < a + \sum_{j=0}^{z} \lambda_{ij}, z = 0, 1, 2, \cdots, k-1 \\
a + \sum_{j=0}^{z-1} \lambda_{ij} = b \\
\hat{v} \in \hat{\mathbf{d}}_{iz} \\
\hat{\mathbf{d}}_{iz} \subseteq \hat{\mathbf{D}}_z, z = 1, 2, \cdots, k
\end{cases}
\tag{1}
$$

where $\hat{\mathbf{D}}_z$ denotes the zth subset, $\hat{\mathbf{d}}_{iz}$ denotes the ith dimension of the zth subsets $\hat{\mathbf{D}}_z$, $\hat{v}$ denotes the elements of $\hat{\mathbf{d}}_{iz}$ and λ_{ij} denotes the change length of the ith dimension data in the jth subset.

3.2.2 K-means method

We analyze $\mathbf{D}$ from the perspective of row vectors, i.e. $\mathbf{D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}^\mathsf{T}$ where n denotes the number of data. The goal of K-means method is to cluster all the data in $\mathbf{D}$ into a set of k clusters (subsets), i.e. $\hat{\mathbf{D}}$. K-means method finds a clustering result to make the following formula reach its minimum value:

$$min(L) = \sum_{z=1}^{k} \sum_{x_r \in \hat{\mathbf{D}}_z} ||x_r - \mu_z||^2 \tag{2}$$

where x_r denotes the data in $\hat{D}_z$, μ_z denotes the center (mean) of $\hat{D}_z$. The algorithm obtains good clustering results by continuously adjusting the center μ_z.

K-means method is a greedy algorithm, which starts with an initial partition with K clusters. Usually, this method is considered to converge to a local minimum, while [33] showed that it can converge to the global minimum with a large probability. The details of K-means method can be found in [34].

3.3 ClusterNets

The ClusterNet is unique in that it can process multiple subtasks simultaneously, which is due to the special structure and the training method.

3.3.1 The Sturcture

As shown in Fig.2 (b), a ClusterNet consists of multiple clusters, each of which is responsible for a specific subtask, i.e. $q = k$, where q denotes the number of clusters. A cluster is composed of two parts: a function network and a context network. The function network learns the conditional probability $P(X|Y)$ to predict the corresponding value for given inputs, i.e. the function network is a regression model. While the context network learns the nonlinear relationship between the inputs and its category labels (from task allocation) to identify the subtask the cluster should handle, i.e. the context network is a classification model.

The output of a ClusterNet can be written as:

$$y_i = \sum_{j=1}^{q} f_{ij} \times c_{ij} \tag{3}$$

where f_{ij} and c_{ij} denote the output values of the function network and the context network in the jth cluster corresponding to the ith input, respectively.

3.3.2 The Training Method

In a ClusterNet, the function network and the context network are trained alternately to achieve both regression and classification simultaneously.

As for the regression, the loss function of function networks is:

$$L_f = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{4}$$

where N denotes the number of data in the training set, y_i denotes the predicted value, and $\hat{y}_i$ denotes the real value in the training set. Formula (4) dose not involve any information of context networks, therefore only the parameters of function networks in all clusters will be updated at this step.

As for the classification, the loss function of context networks is :

$$L_c = \frac{1}{N} \sum_{i=1}^{N} L_i = \frac{1}{N} \sum_{i=1}^{N} [-\sum_{j=1}^{q} c_{ij} log(p_{ij})] \tag{5}$$

where c_{ij} denotes the probability that the ith input is classified into the jth category by the jth context network, and p_{ij} denotes the true category the ith input belongs to. Formula (5) does not involve any information of function networks, hence only the parameters of context networks in all clusters will be updated at this step.

In conclusion, the training of function networks is only related to L_f, while the training of context networks is only related to L_c. The updating process of parameters in a ClusterNet is shown as:

$$\begin{cases} \mathbf{W}_{fj} = \mathbf{W}_{fj} + \eta \frac{\partial L_f}{\partial \mathbf{W}_{fj}} \\ \mathbf{b}_{fj} = \mathbf{b}_{fj} + \eta \frac{\partial L_f}{\partial \mathbf{b}_{fj}} \\ \mathbf{W}_{cj} = \mathbf{W}_{cj} + \eta \frac{\partial L_c}{\partial \mathbf{W}_{cj}} \\ \mathbf{b}_{cj} = \mathbf{b}_{cj} + \eta \frac{\partial L_c}{\partial \mathbf{b}_{cj}} \end{cases} \tag{6}$$

where $\mathbf{W}_{fj}$ and $\mathbf{b}_{fj}$ denotes the weight matrix and bias vector of the function network in the jth cluster, respectively, while $\mathbf{W}_{cj}$ and $\mathbf{b}_{cj}$ denotes the weight matrix and bias vector of the context network in the jth cluster, respectively.

4 Experimental Results

4.1 Dataset

Figure 3: Subgraph (a) denotes the visualization of Burgers' dataset and (b) denotes the visualization of cylindrical laminar dataset.

To validate the dataset quality-adaptivity of the MTL, two different datasets are employed, namely the Burgers' dataset and the cylindrical laminar dataset.

4.1.1 Burgers' Dataset

The Burgers' equation is an one-dimensional partial differential equation that expresses the movement of a shockwave across a tube:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = v \frac{\partial^2 u}{\partial x^2}$$

where u, the output, denotes the velocity of the shockwave, t, x, and v, the inputs, denotes the time, the displacement and the coefficient of viscosity, respectively. The variation range of every input is from 0.2 to 4.8, and the step is 0.2. Consequently, this dataset contains 13824 samples. Fig.3 (a) illustrates the visualization of whole Burgers' dataset. The red area in the lower left corner depicts the high-speed flow field data. We can see that the low-speed flow field data are plentiful, while the number of high-speed flow field data is limited, which can be used to verify the dataset quality-adaptivity of MTL.

4.1.2 Cylindrical Laminar Dataset

variables	x	y	Ma
significance	x-coordinate	y-coordinate	mach number
start	0.1	-0.5	0.1
end	1	0.5	0.24
step	0.005	0.0078	0.01

Table 1: The data format and variation range of input parameters in cylindrical laminar dataset.

6

variables	P	C_p	F_x	F_y
significance	pressure	coefficient of pressure	friction in x-coordinate	friction in y-coordinate

Table 2: The data format of output parameters in cylindrical laminar dataset.

The cylindrical laminar is a two-dimensional application of the Navier-Stokes equation that simulates the pressure change on the surface of a cylinder while a flow passes a cylinder. The Navier-Stokes equation is as follows:

$$\rho \left[\frac{\partial V}{\partial t} + (V.\nabla)V \right] = -\nabla P + \rho g + v\nabla^2 V$$

where P, V, t, ρ, v denote the pressure, velocity, time, density and coefficient of viscosity, respectively. In this dataset, we used SU2 to calculate 6000 sample points. The format and the variation range of the input parameters are shown in Tab.1. The format of the output parameters is shown in Tab.2.

The visualization of cylindrical laminar dataset is shown in Fig.3(b) that reflects the same feature as Fig.3(a): the number of data with large C_p is limited.

4.2 Experiment Results

We compare several approaches including FCN, GAN and MTL with different structures. The learning rate is 0.0001, with the batch size 128, and the number of iterations 2000 within all models. All these approaches compared in this paper are implemented based on tensorflow framework [35]. All programs run on four Tesla K80 GPUs. The whole dataset is divided into three subsets: a training set, a validation set and a test set with a ratio of 8:1:1. The errors of these models are evaluated by mean squared error (MSE) and mean absolute error (MAE):

$$\begin{cases} MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \\ MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \end{cases}$$

where y_i denotes the predicted value, $\hat{y}_i$ denotes the real value for the same inputs.

4.2.1 Burgers' Experiment

Method	structure	MSE	MAE
FCN_1	3*32	1.64×10^{-4}	7.71×10^{-3}
FCN_2	3*64	1.85×10^{-4}	8.82×10^{-3}
cGAN_1	G(62,1*64,4)D(4,1*64,1)	2.63×10^{-4}	1.29×10^{-2}
cGAN_2	G(62,3*64,4)D(4,3*64,1)	diverges	diverges
MTL_k-means	4;3*64;1*5	1.79×10^{-4}	8.11×10^{-3}
MTL_v	4;3*64;1*5	1.78×10^{-4}	8.91×10^{-3}
MTL_x	4;3*64;1*5	9.86×10^{-5}	6.70×10^{-3}

Table 3: The Burgers' experiment results. In the column related to structure, "3*32" denotes that the FCN has 3 hidden layers, each of which has 32 neurons. "G(62,1*64,4)" indicates that generator of the GAN has 3 layers, the number of neurons in the input layer, the hidden layer and the output layer is 62, 64 and 4, respectively. "D(4,1*64,1)" indicates that the discriminator of the GAN has 3 layers, 4, 64 and 1 denotes the number of neurons in the input layer, the hidden layer and the output layer, respectively. "4;3*64; 1*5" means that the ClusterNet consists of 4 clusters. The functional network has 3 hidden layers, each of which has 64 nodes. The context network has 1 hidden layer, each of which has 5 neurons. MTL_k-means denotes the multi-task learning that uses k-means for subtask division. MTL_v denotes the multi-task learning whose subtasks are divided by v. Similarly, MTL_x denotes that the subtasks are divided by x.

The results of Burgers' experiment are shown in Tab.3. We can learn that MTL_x is the optimal approach in term of MSE and MAE. Besides, the rest two MTLs are similar to FCNs, and the cGANs are the worst among them all.

The visualization results of velocity u predicted by the above approaches are shown in Fig.4. Obviously, the FCN can not predicted u precisely where $u > 3.5$. What's worse, the cGAN could not achieve accurate prediction of u in the entire flow field. Therefore, we can get a conclusion that the velocity u predicted by MTLs are much more accurate than FCNs and cGANs, i.e. the MTL is a dataset quality-adaptive learning scheme.

Figure 4: The u predicted by FCN_1 (a), cGAN_1 (b), MTL_k-means (c), MTL_v (d) and MTL_x (e).

Fig.5 also reflects the same results. The red part in subgraph (b) is lighter than others, which means than the value of predicted u is smaller than that of other approaches. Besides, the contour line in subgraph (b) and (c) are not smooth, which means that the predicted u of the FCN and the cGAN are very different from the real one. It is worth noting that subgraph (d), (e) and (f), i.e. the three MTLs, are much more close to subgraph (a).

Figure 5: The u predicted by CFD (a), FCN_1 (b), cGAN_1 (c), MTL_k-means (d), MTL_v (e) and MTL_x (f).

Figure 6: The results of task allocation in Burgers' experiment. Subgraph (a) denotes the results of k-means, (b) denotes the results of data partition by v and (c) denotes the results of data partition by x.

Fig.6 shows the visualization results of task allocation. Fig.7 depicts the variations of loss functions of three MTLs. Apparently, the loss variations of function networks are similar, as well as the context networks. Besides, the loss value of context networks is close to 0, which implies that all three MTLs learn aerodynamic characteristics strictly according to the subtasks showed in the Fig.6.

8

Figure 7: The loss variations of function network and context network of MTL_K-means, MTL_v and MTL_x in Burgers' experiment. The 'function_kmeans' and 'context_kmeans' denote the loss function of the function networks and the context networks from MTL_K-means, respectively. The rest of signs are similar.

| (a) | (b) | (c) | (d) | (e) |

Figure 8: The C_p predicted by FCN_1 (a), GAN_1 (b), MTL_k-means (c), MTL_x (d) and MTL_Ma (e).

4.2.2 Cylindrical Laminar Experiment

In cylindrical laminar experiment, we test the same approaches, whose test errors are shown in Tab.4. We can learn that all MTLs are better than others in terms of MSE and MAE. Besides, the basis of task allocation do not affects the test errors apparently.

Method	structure	MSE	MAE
FCN_1	3*32	1.31×10^{-4}	8.44×10^{-3}
FCN_2	3*64	2.45×10^{-4}	8.81×10^{-3}
GAN_1	G(62,1*64,4)D(4,1*64,1)	6.35×10^{-4}	1.91×10^{-2}
GAN_2	G(62,3*64,4)D(4,3*64,1)	diverges	diverges
MTL_K-means	4;3*64;1*5	8.51×10^{-5}	6.80×10^{-3}
MTL_x	4;3*64;1*5	8.01×10^{-5}	6.28×10^{-3}
MTL_Ma	4;3*64;1*5	7.65×10^{-5}	6.32×10^{-3}

Table 4: The test errors of cylindrical laminar experiment.

The visualization results of C_p and F_x predicted by the above approaches are shown in Fig.8 and Fig.9. Like Burgers' experiment, the FCN still can not predicted C_p and F_x precisely where their values are relatively large. In addition, the cGAN could not achieve accurate prediction of C_p and F_x in the entire flow field. On the contrary, all the three MTLs can achieve accurate prediction of both C_p and F_x in the entire flow field.

The visualization results of Cp predicted by the above approaches are shown as in Fig.10. Intuitively, all these models can predict the Cp around the surface of a cylindrical. However differences can be seen clearly after zooming in those subgraphs. We enlarge the left half of the cylinder (i.e. the flow field with a large C_p) in Fig.12(a). We can see that the red part comes from subgraph (b) in Fig.10 is lighter than others, which is consistent with the conclusion derived from Fig.8

Fig.11 depicts the visualization results of task allocation. Fig.12(c) shows the variations of loss functions of three MTLs. Also, the loss variations of context networks are similar and close to 0. However, the fluctuations of function networks are different. As we can see, the fluctuations of function loss of the MTL which divides subtasks by Ma is more drastic than others.

9

Figure 9: The F_x predicted by FCN_1 (a), GAN_1 (b), MTL_k-means (c), MTL_x (d) and MTL_Ma (e).

Figure 10: The C_p predicted by CFD (a), FCN_1 (b), cGAN_1 (c), MTL_k-means (d), MTL_v (e) and MTL_x (f).

Figure 11: The results of task allocation in cylindrical laminar experiment. Subgraph (a) denotes the results of k-means, (b) denotes the results of data partition by x and (c) denotes the results of data partition by Ma.

Figure 12: Subgraph (a) denotes the enlargement of experimental visualization results in Fig.10. Subgraph (b) denotes the loss variations of function network and context network of MTL_K-means, MTL_x and MTL_Ma in cylindrical laminar experiment.

10

Sample Id	v	real u	predicted u	f_1	c_1	f_2	c_2	f_3	c_3	f_4	c_4
1	1.0	0.69	0.68	0.09	**0.98**	0.14	0.48	0.00	0.00	0.00	0.00
2	1.8	2.79	2.08	0.62	0.00	0.96	**0.48**	0.89	0.01	0.81	0.00
3	3.0	3.01	2.11	0.06	0.00	0.88	0.00	0.97	**0.49**	0.80	0.00
4	3.8	4.04	3.77	0.00	0.00	0.87	0.00	0.94	0.49	0.40	**0.99**

Table 5: The output of each cluster of MTL_v in Burgers' experiment. In the column related to "real u" and "predicted u", the data are not normalized, however the data in the rest of columns are normalized. Bold numbers means that the corresponding cluster is activated.

4.3 Analysis

We take Burgers' experiment as an example to analyze the data quality-adaptivity of MTLs. Tab.5 shows four samples in the test set. We can see that sample 1 activates the first cluster in the MTL_v, sample 2 activates the second cluster, and so on. It is obvious that samples with different v are processed by different clusters in the MTL_v. Therefore, the MLT_v distributes all learning pressure on every cluster, so that a cluster could focus on a small learning task and the learning process among clusters does not affect each other. This mechanism is not affected by the quality of the dataset, i.e. it is an dataset quality-adaptive learning scheme, which perfectly explains why the MTLs are more accurate than FCNs and GANs in entire flow field.

5 Conclusions

In this paper, the MTL in aerodynamic data modeling is proposed. We first pointed out the impact of the quality of datasets on the accuracy of aerodynamic data models. Then, we propose the MTL, a dataset quality-adaptive learning scheme, which consists of tasks allocation and aerodynamic characteristics learning. Finally, the dataset quality-adaptivity of the MTL is verified through Burgers' and cylinder laminar experiments. Compared with existing models, MTL possess the ability of adapting to poor quality datasets.

The detailed conclusions are as follows:

a) the MTL is a dataset quality-adaptive learning scheme;

b) the MTL is more accuracy than FCNs and GANs;

c) the advantages described in a) and b) come from the partial activation mechanism of the MTL.

References

[1] Yang Yang, Kehua Ye, Chun Li, Constantine Michailides, and Wanfu Zhang. Dynamic behavior of wind turbines influenced by aerodynamic damping and earthquake intensity. *Wind Energy*, 21(5):303–319, 2018.

[2] Reik Thormann and Sebastian Timme. Efficient aerodynamic derivative calculation in three-dimensional transonic flow. *The Aeronautical Journal*, 121(1244):1464–1478, 2017.

[3] Hady Benyamen, Aaron Mckinnis, and Shawn Keshmiri. Effects of propwash on horizontal tail aerodynamics of pusher uass. In *2020 IEEE Aerospace Conference*, pages 1–9. IEEE, 2020.

[4] Zhaobin Li, Benjamin Bouscasse, Guillaume Ducrozet, Lionel Gentaz, David Le Touzé, and Pierre Ferrant. Spectral wave explicit navier-stokes equations for wave-structure interactions using two-phase computational fluid dynamics solvers. *arXiv preprint arXiv:2005.12716*, 2020.

[5] An Zhang, Mingjing Jiang, and Colin Thornton. A coupled cfd-dem method with moving mesh for simulating undrained triaxial tests on granular soils. *Granular Matter*, 22(1):1–13, 2020.

[6] S Vitale, M Pini, and P Colonna. Multistage turbomachinery design using the discrete adjoint method within the open-source software su2. *Journal of Propulsion and Power*, 36(3):465–478, 2020.

[7] Mohd Zarif Md Shah, Bambang Basuno, and Aslam Abdullah. Comparative study on several type of turbulence model available in ansy-fluent software for onera m6 wing aerodynamic analysis. *Journal of Advanced Mechanical Engineering Applications*, 1(1):9–19, 2020.

[8] Jiao He, Xin Jin, Shuangyi Xie, Le Cao, Yaming Wang, Yifan Lin, and Ning Wang. Cfd modeling of varying complexity for aerodynamic analysis of h-vertical axis wind turbines. *Renewable Energy*, 145:2658–2670, 2020.

[9] Liwei Hu, Jun Zhang, Yu Xiang, and Wenyong Wang. Neural networks-based aerodynamic data modeling: A comprehensive review. *IEEE Access*, 8:90805–90823, 2020.

[10] Jie Gui, Zhenan Sun, Yonggang Wen, Dacheng Tao, and Jieping Ye. A review on generative adversarial networks: Algorithms, theory, and applications. *arXiv preprint arXiv:2001.06937*, 2020.

[11] Sangseung Lee and Donghyun You. Data-driven prediction of unsteady flow over a circular cylinder using deep learning. *Journal of Fluid Mechanics*, 879:217–254, 2019.

[12] Junhyuk Kim and Changhoon Lee. Deep unsupervised learning of turbulence for inflow generation at various reynolds numbers. *Journal of Computational Physics*, 406:109216, 2020.

[13] You Xie, Erik Franz, Mengyu Chu, and Nils Thuerey. tempogan: A temporally coherent, volumetric gan for super-resolution fluid flow. *ACM Transactions on Graphics (TOG)*, 37(4):1–15, 2018.

[14] Liwei Hu, Wenyong Wang, Yu Xiang, and Jun Zhang. Flow field reconstructions with gans based on radial basis functions. *arXiv preprint arXiv:2009.02285*, 2020.

[15] Lianbo Zhang, Shaoli Huang, Wei Liu, and Dacheng Tao. Learning a mixture of granularity-specific experts for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8331–8340, 2019.

[16] Geoffrey H Ball and David J Hall. A clustering technique for summarizing multivariate data. *Behavioral science*, 12(2):153–155, 1967.

[17] Cristina White, Daniela Ushizima, and Charbel Farhat. Fast neural network predictions from constrained aerodynamics datasets. In *AIAA Scitech 2020 Forum*, page 0364, 2020.

[18] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[19] Kotaro Taguchi, Koh Fukunishi, Shogo Takazawa, Yasuto Sunada, Taro Imamura, Kenichi Rinoie, and Tomohiro Yokozeki. Experimental study about the deformation and aerodynamic characteristics of the passive morphing airfoil. *TRANSACTIONS OF THE JAPAN SOCIETY FOR AERONAUTICAL AND SPACE SCIENCES*, 63(1):18–23, 2020.

[20] Konrad Bamberger, Julian Belz, Thomas Carolus, and Oliver Nelles. Aerodynamic optimization of centrifugal fans using cfd-trained meta-models. 2016.

[21] Matt Edmunds, B Evans, I Masters, and RS Laramee. Enhanced flow visualisation of complex aerodynamic phenomena using automatic stream surface seeding with application to the bloodhound ssc land speed record vehicle. *The Aeronautical Journal*, 120(1226):547–571, 2016.

[22] Jitu Sanwale and Dhan Jeet Singh. Aerodynamic parameters estimation using radial basis function neural partial differentiation method. *Defence Science Journal*, 68(3):241, 2018.

[23] Dhan Jeet Singh, Nischal K Verma, AK Ghosh, Jitu Sanwale, and Appasaheb Malagaudanavar. Aerodynamic parameter estimation using two-stage radial basis function neural network. In *2017 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC)*, pages 461–467. IEEE, 2017.

[24] Jichao Li, Sicheng He, and Joaquim RRA Martins. Data-driven constraint approach to ensure low-speed performance in transonic aerodynamic shape optimization. *Aerospace Science and Technology*, 92:536–550, 2019.

[25] Ryan Hamerly, Liane Bernstein, Alexander Sludds, Marin Soljačić, and Dirk Englund. Large-scale optical neural networks based on photoelectric multiplication. *Physical Review X*, 9(2):021032, 2019.

[26] Mattie Tops, Markus Quirin, Maarten AS Boksem, and Sander L Koole. Large-scale neural networks and the lateralization of motivation and emotion. *International Journal of Psychophysiology*, 119:41–49, 2017.

[27] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[28] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[29] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[30] Afsane Rajaei, Hamidreza Shayegh, and Nasrollah Moghaddam Charkari. Human detection in semi-dense scenes using hog descriptor and mixture of svms. In *ICCKE 2013*, pages 229–234. IEEE, 2013.

[31] Marc Deisenroth and Jun Wei Ng. Distributed gaussian processes. In *International Conference on Machine Learning*, pages 1481–1490, 2015.

[32] Jiahao Zhang, Yeshas Thadimari, Venkat Varun Velpula, and Jayanth Bhargav. Learning interactions and dynamics of swarms. 2020.

[33] Marina Meilă. The uniqueness of a good optimum for k-means. In *Proceedings of the 23rd international conference on Machine learning*, pages 625–632, 2006.

[34] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[35] Bo Pang, Erik Nijkamp, and Ying Nian Wu. Deep learning with tensorflow: A review. *Journal of Educational and Behavioral Statistics*, 45(2):227–248, 2020.