

Predicting Unsteady Indoor Temperature Distributions by POD-DNN

Chenghao Wei^{1*}, Ryoza Ooka², Bingchao Zhang², and Qi Zhou²

¹Graduate School of Engineering, The University of Tokyo, Tokyo, Japan

²I.I.S., The University of Tokyo, Tokyo, Japan.

Abstract. In this study, to predict unsteady temperature distributions, POD-DNN was utilized, where DNN was trained to predicted coefficients of POMs. Two strategies, flatten POD-DNN and nested POD-DNN were compared. The flatten POD-DNN provided high accuracy if training data is sufficient, but otherwise very inaccurate. The nested POD-DNN roughly predicted the development of temperature fields even training data was small. The results showed their different sensitivities to the training data size.

1 Introduction

For efficient energy consumption of buildings, predicting the detailed indoor thermal environment, such as temperature and velocity distributions, is essential. Computational fluid dynamics (CFD) has been widely utilized to simulate the detailed building environment. But the time cost of CFD is expensive.

Some improved methods have been developed to reduce the computing time of CFD, such as FFD [1] and rCFD [2]. And recently, machine learning (ML) is introduced to speed up the prediction for its ability to build nonlinear relationships between boundary conditions and the indoor thermal environment rapidly [3]. The basic practice is training a deep neural network (DNN) by the results of CFD, and the well-trained DNN is hoped to predict temperature and velocity distributions of cases with other boundary conditions. But, as a data-driven method, DNN only fits data without understanding the flow structure of airflow development. Ottino et al. [4] introduced an ML-based reduced-order model (ROM) approach for indoor environment prediction. In that paper, proper orthogonal decomposition (POD) was used to extract an optimal set of orthonormal bases (modes) and calculate their combination coefficients from CFD results. A DNN was trained to predict combination coefficients of modes. Because these modes produced useful information of flow structures [5], by combining modes according to predicted coefficients, the building environment can be predicted under the guidance of some flow characteristics. The approach did well in steady-state indoor flow prediction. But the unsteady prediction application is remained.

In this study, the POD-DNN approach is utilized to predict unsteady state indoor temperature distributions with alternative boundary parameters. POD was used to extract characteristic modes. DNN is trained to output

combining coefficients of these modes to predict temperature distributions. For an unsteady problem with alternative parameters, by taking time and parameters into consideration, different strategies used to extract modes influenced the accuracy of predictions. In this study, two kinds of POD strategies, flatten POD and nested POD, were researched to investigate their influence on the prediction accuracy.

2 Methodology

2.1 Introduction of POD

POD analysis, also known as principal component analysis (PCA), is a method to extract the eigenvectors and eigenvalues of a matrix. And the matrix can be reproduced by combining the eigenvectors according to the eigenvalues, as shown in Eq. (1):

$$A(x, t) = \sum_1^r a_r(t) \varphi_r(x) \quad (1)$$

where φ_m is eigenvectors and a_m is coefficients. By the POD analysis, a reduced set of data [$\varphi_1(x), \dots, \varphi_r(x)$], namely POD modes (POMs), can be extracted from an original temperature field. These POMs provide useful information on temperature field structures. In practice, the coefficient relates to time and boundary condition parameters. If coefficients are precisely predicted from time and boundary parameters, complex temperature fields can be reconstructed quickly by combining them with only a few basic modes.

2.2 POD-DNN

DNN is good at fitting non-linear relationships between input values and output values. In this study, DNN is introduced to predict coefficients from inputs of time or

* Corresponding author: c-wei@iis.u-tokyo.ac.jp

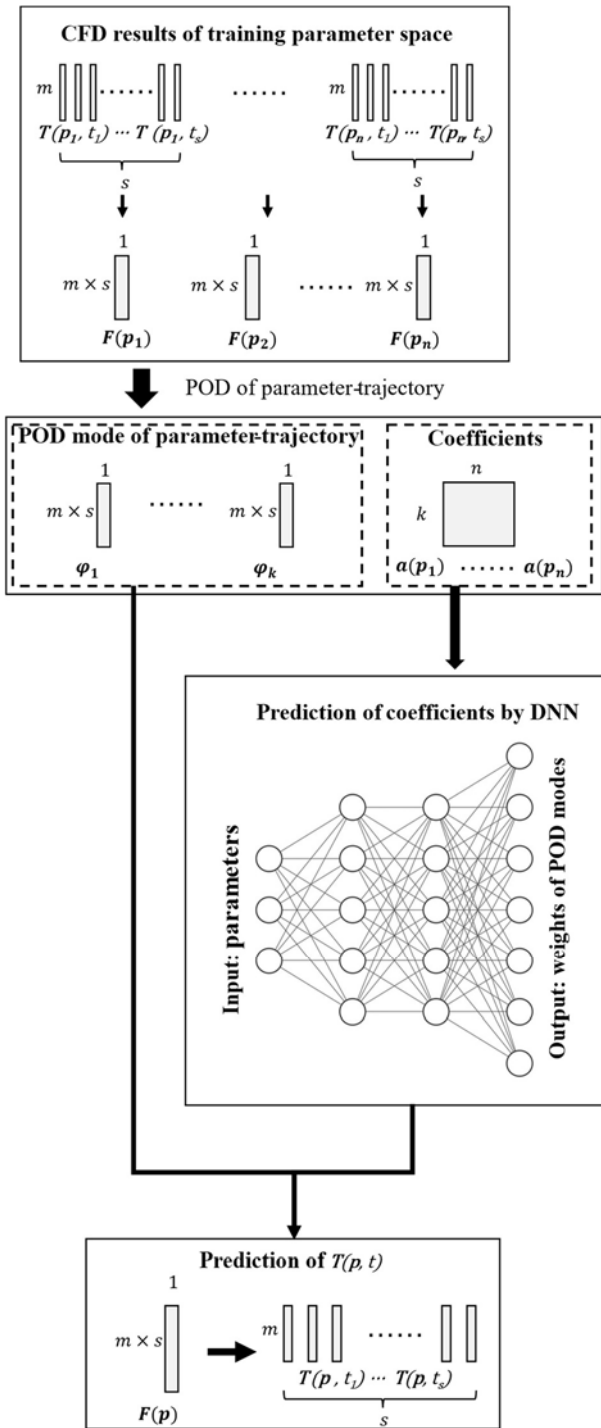


Fig. 1. Process of flatten POD-DNN.

boundary parameters. Unsteady temperature fields are predicted by combining POMs and coefficients predicted by DNN. The POD-DNN prediction process is mainly divided into four parts: (1) conducting unsteady CFD simulation of different boundary parameters to build a database, and dividing them into a training set, validation set, and testing set, (2) extracting POMs and coefficients from the training set, (3) training DNN by the coefficients of the training set and predicting coefficients of testing set, (4) combining POMs according to the predicted coefficients to reconstruct testing set.

For an unsteady problem with alternative boundary parameters, taking time and parameters into

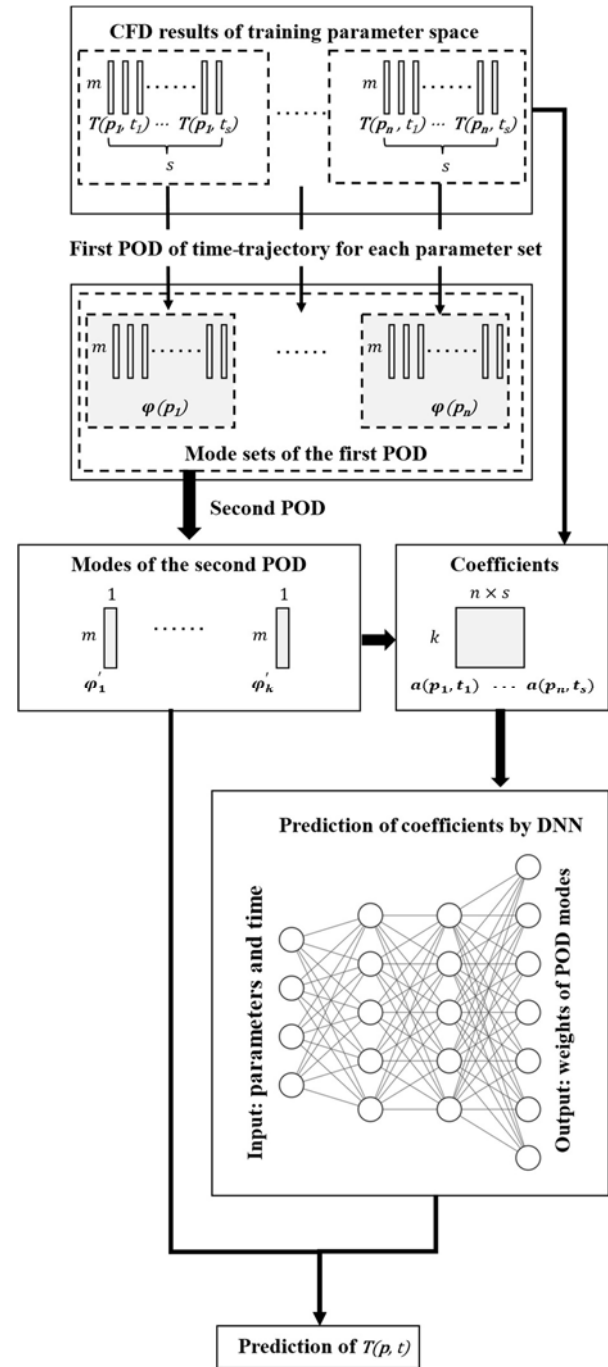


Fig. 2. Process of nested POD-DNN.

consideration makes it a multidimensional problem. It is important to find a proper way to extract POMs. In this study, two POD strategies combined with DNN are compared: flatten POD-DNN and nested POD-DNN.

2.2.1 Flatten POD-DNN

The process of flatten POD-DNN is shown in Fig. 1. Considering a CFD model with grid numbers of m , temperature field $T(\mathbf{p}_n, t_s)$ is simulated, where \mathbf{p}_n and t_s mean boundary parameter vectors and time, respectively. For each \mathbf{p}_n , there is a time series set of temperature field $\mathbb{T}(\mathbf{p}_n) = [T(\mathbf{p}_n, t_1) | \dots | T(\mathbf{p}_n, t_s)]$, which is a two-dimensional matrix shaped as $m \times s$. Then the matrix $\mathbb{T}(\mathbf{p}_n)$ is flattened to a vector $F(\mathbf{p}_n)$. POMs of parameter-trajectory $[\boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_k]$ and coefficients of training cases

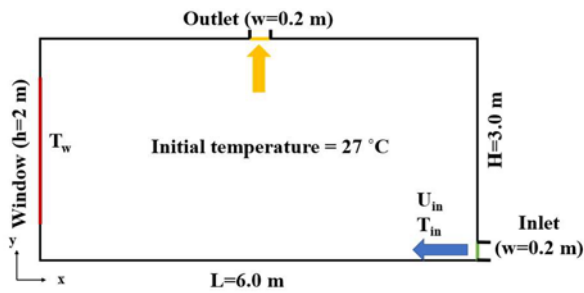


Fig. 3. Outline of the target space.

$[\mathbf{a}(\mathbf{p}_1), \dots, \mathbf{a}(\mathbf{p}_n)]$ are calculated from POMs and $[\mathbf{F}(\mathbf{p}_1), \dots, \mathbf{F}(\mathbf{p}_n)]$. Trained by coefficients of training cases, DNN is used to predict coefficients of testing cases. $\mathbf{F}(\mathbf{p})$ of the testing case is predicted by combining predicted coefficients and the POMs. Finally, the $\mathbf{F}(\mathbf{p})$ is sliced through time-trajectory.

2.2.2 Nested POD-DNN

This method is introduced by Wang et al. [6]. Fig. 2 shows the process. In this method, POMs are extracted by applying POD two times. The first time, POMs $\boldsymbol{\varphi}(\mathbf{p}_n)$ are extracted from $\mathbb{T}(\mathbf{p}_n)$ for each boundary parameter \mathbf{p}_n . The second time, final POMs $[\boldsymbol{\varphi}'_1, \dots, \boldsymbol{\varphi}'_k]$ are extracted from $[\boldsymbol{\varphi}(\mathbf{p}_1) | \dots | \boldsymbol{\varphi}(\mathbf{p}_n)]$. Then, coefficients relating to time and parameters are calculated from $\mathbf{T}(\mathbf{p}_n, t_s)$ and the POMs. The set of coefficients are used to train a DNN. The trained DNN is applied to predicted coefficients of testing cases by inputting parameters and time. Finally, the POMs are combined according to predicted coefficients to predict temperature fields of testing cases.

2.3 A two-dimensional indoor problem

Temperature distributions of a two-dimensional indoor space, shaped as 6.0 m (x) \times 3.0 m (y), were investigated. Fig. 3 shows the outline of the target space. One inlet boundary and one outlet boundary were set near the floor and at the centre of the ceiling, respectively. And one window was across from the inlet boundary. The lengths of the inlet boundary, outlet boundary, and window were 0.2, 0.2, and 2 m, respectively. The

Table 1. Settings of CFD simulation

Items	Conditions
Employed code	Star CCM+ V14.06.013
Dimension	6.0 m (X) \times 3.0 m (Y)
Object	Air (Incompressible)
Analysis Method	Transient Analysis (0 s~300 s, Interval: 0.1 s)
Turbulent model	Realizable k- ϵ model
Mesh size	0.05 m \times 0.05 m
Inlet boundary	Fixed temperature T_{in} Fixed velocity U_{in} Turbulent intensity: 4%
Outlet boundary	Free outflow
Wall	Non-slip, wall-function, adiabatic
Window	Fixed surface temperature T_w
Boundary parameter space	U_{in} : 0.25, 0.5, 0.75 and 1.0 m/s T_{in} : 16, 18, 20, and 22 $^{\circ}$ C T_w : 35, 40 and 45 $^{\circ}$ C

window was set to a fixed temperature to exchange heat by convection. Other envelopes were assumed to be adiabatic.

The database was built by CFD simulation. Settings of the CFD simulation are shown in Table 1. STAR CCM+ v14.06.013 was employed for the CFD simulation in this study. The initial values of the temperature and velocity fields were set to 27 $^{\circ}$ C and 0 m/s, respectively. The alternative boundary parameters space consists of three kinds of boundary conditions: inlet air velocity U_{in} , inlet air temperature T_{in} , window surface temperature T_w . Values of them are provided in Table 1. Cases are named by parameter set: (U_{in}, T_{in}, T_w) , for example, (1.0, 18, 35) means the case whose U_{in} , T_{in} , T_w is 1.0 m/s, 18 $^{\circ}$ C, and 35 $^{\circ}$ C. The time of each simulation was in a range of 0 to 300 s, where the time step was set to 0.1 s. But only results of selected time points were used to obtain the dataset (the first second, every 5 seconds from 5 to 100 s, every 10 seconds from 110 to 200 s, and every 20 seconds from 220 to 300 s).

The CFD database was separated to training set, validation set, and testing set by boundary parameters. Two training scenarios were compared for each POD strategies. In one scenario, case (0.25, 16, 35), (1.0, 16, 35), (0.25, 22, 35), (1.0, 22, 35), (0.25, 16, 45), (1.0, 16, 45), (0.25, 22, 45), (1.0, 22, 45) and (0.75, 18, 40) were selected as training cases (9 cases in 48 cases), (0.25, 20, 45), (0.5, 18, 40), (0.75, 16, 35) and (1.0, 20, 40) were validation cases, and others were testing cases. In the other scenario, (0.25, 16, 40), (0.25, 20, 45), (0.25, 22, 45), (0.5, 16, 40), (0.5, 22, 45), (0.75, 18, 35), (0.75, 20, 35) and (1.0, 18, 35) were selected as testing cases, (0.25, 18, 45), (0.5, 18, 40), (0.75, 22, 40) and (1.0, 20, 35) were validation cases, and others were training cases (36 cases in 48 cases). In addition, the purpose of this study is to reproduce CFD database via DNN, the validity of the CFD database is not paid attention to.

Regarding DNN prediction, the process is mentioned in Section 2.2. Parameters U_{in} , T_{in} , T_w were input to the flatten POD-DNN, on the other side, time t_s and parameters U_{in} , T_{in} , T_w were input. Coefficients of POMs were output. Both input and out data were normalized by transferring them to a range of 0 to 1. A dense DNN with 34 hidden layers was used, where there were 64 nodes in each hidden layer. Activation function, optimization algorithm, and loss function were set to ELU, Adam, and mean squared error, respectively.

3 Results

Contour figures of a testing case (0.5, 16, 40) were shown in Fig.4 to check prediction accuracy. In the results of CFD, a vortex in low temperature gradually became larger and moved away from the inlet until it filled the entire space. Regarding results of the flatten POD-DNN, when the training set was large, the prediction results were almost the same as the CFD results. However, when the training set was small, there were obvious multiple vortices, different from the results of CFD, showing low accuracy. For the nested POD-DNN, when the training set was large, compared with the flatten POD-DNN, there was more noise, and

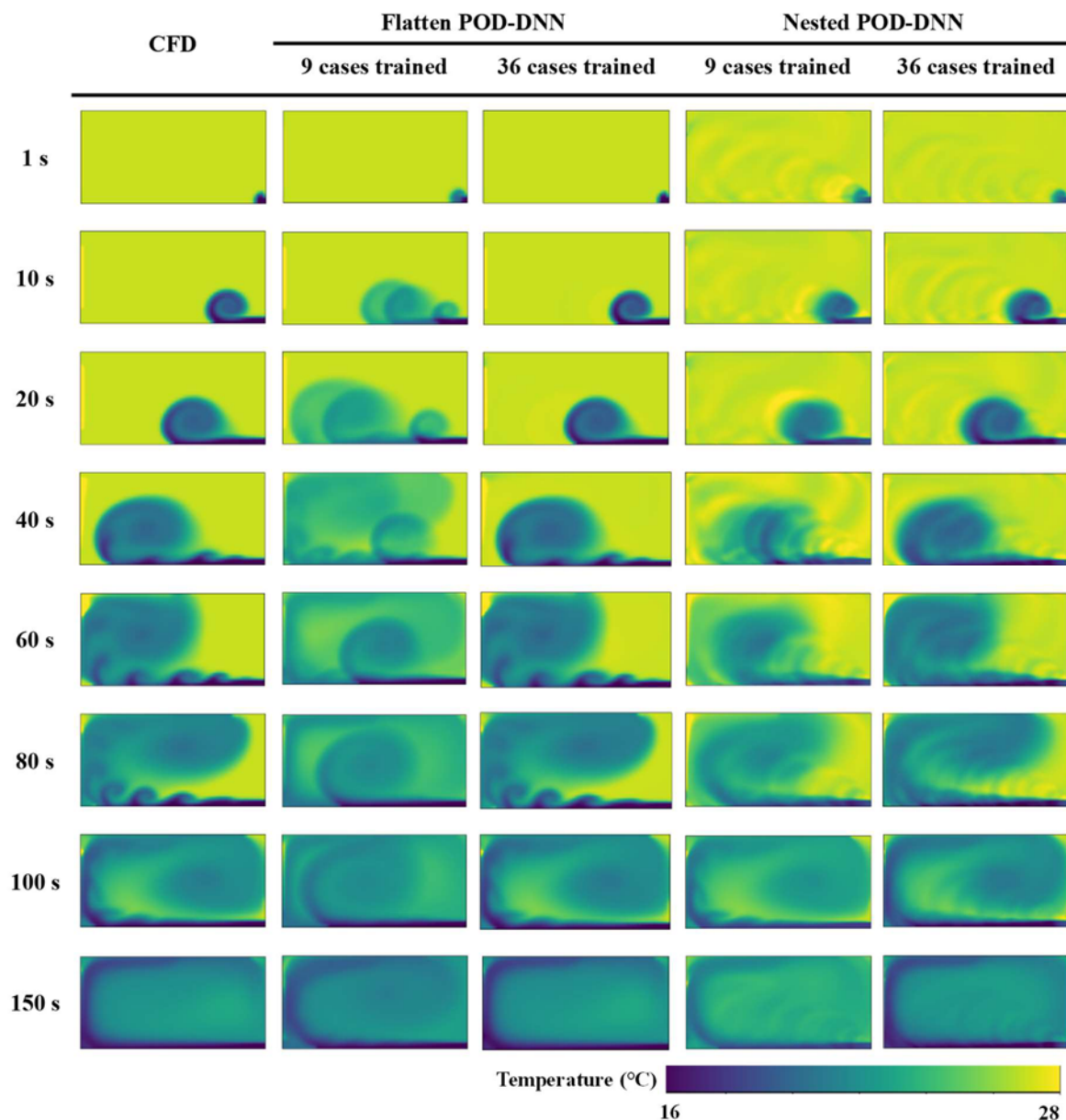


Fig. 4. Temperature distributions of a testing case, $U_{in}=0.5$ m/s, $T_{in}=16$ °C, $T_w=40$ °C.

small vortices near the ground were not reproduced clearly. But generally, the development of the dominant vortex was well reproduced. When the training set was small, not like the results of the flatten POD-DNN, although accuracy dropped somehow, the development of the dominant vortex was reproduced. As a summary, if training data was sufficient, the flatten POD-DNN predicted accurately. But this method is sensitive to the training data size, less training data leading to obvious larger errors. Although there are some noises and errors, the nested POD-DNN can roughly predict the development of airflow regardless of the training data size, showing its less sensitivity to the training data size.

4 Conclusion

In this study, to predict unsteady temperature distributions, POD-DNN was utilized, where DNN was trained to predicted coefficients of POMs. Two kinds of POD strategies were compared. The results showed different their sensitivities to the training data size. The flatten POD-DNN provided high accuracy if training data is sufficient, but otherwise very

inaccurate. The nested POD-DNN roughly predicted the development of temperature fields even training data was small. Reasons for the different sensitivities should be investigated in future works.

Acknowledgment: This work was supported by JST SPRING, Grant Number JPMJSP2108.

References

1. W. Zuo, Q. Chen, Indoor Air, **19**, 33–44 (2009).
2. Y. Du, B. Blocken, S. Pirker, Build Environ, **170**, 106604 (2020).
3. Q. Zhou Q, R. Ooka, Build Simul, **13**, 1409–1423 (2020).
4. G. Ottino, A. Lario, 15th ROOMVENT Conference, **NV1** (2021).
5. C. Howard, S. Gupta, A. Abbas, T. Langrish, D. Fletcher, Chem Eng Res Des, **123**, 333-346 (2017)
6. Q. Wang, J. Hesthaven, D. Ray, J. Comput. Phys., **384**, 289-307 (2019)