https://doi.org/10.26599/FRICT.2026.9441239

Research Article

Generative deep learning for hydrodynamic lubrication prediction by a deconvolutional neural network

Yang Zhao[1]*, Ziniu Huang[2], Shuo Guo[2], Zhongxue Fu[2], Yanyan Lin[3]

[1]*Department of Construction, Environment and Engineering, Technological and Higher Education Institute of Hong Kong, 133 Shing Tai Road, Chai Wan, Hong Kong, China*

[2]*College of Mechatronic and Control Engineering, Shenzhen University, 3688 Nanhai Avenue, Shenzhen, Guangdong, China*

[3]*School of Automotive and Transportation Engineering, Shenzhen Polytechnic University, 7098 Liuxian Avenue, Shenzhen, Guangdong, China*

* Zhao, Yang: yzhao84-c@my.cityu.edu.hk

Received: September 15, 2025; Revised: January 20, 2026; Accepted: February 24, 2026

Abstract: Classical hydrodynamic lubrication simulation solves Reynolds equation to unveil the lubrication pressure and film thickness distributions, yet it overlooks the latent high-level representations buried beneath the lubrication data. While deep learning has illustrated that the mining of high-level representations helps to generate desired outputs from input prompts, the lubrication research, however, has not fully exploited generative deep learning in lubrication

prediction and generation. Here, we propose to adopt a deconvolutional neural network to learn the latent representations in hydrodynamic lubrication data and directly generate 2D lubrication scenario from the given working condition without solving any governing equation. Compared to classical method, our approach can output the distribution of lubrication pressure and film thickness in less than 0.1 s on a personal computer and be extended to more complicated scenarios including cavitation.

Key words: hydrodynamic lubrication, deconvolutional neural network, generative learning

1. Introduction

Sliding bearings play a significant role in supporting the efficient rotation of heavy-loaded shafts, e.g., the wind energy industry adopts sliding bearings to support the principal axis of the wind turbine generator [1]. Considering that these heavy-loaded rotating machines usually appear to be rare and energy-cost, the sliding bearing utilised in them needs customized design and optimization [2] to meet the machine's requirements, e.g., reduced friction and repair cost [3]. Thus, engineers need to first predict the bearing's lubrication performance under various working conditions and bearing geometries to select the most suitable bearing designs during the initial design process. Although modern numerical approach could easily obtain a sliding bearing's lubrication performance, running a single simulation of moderate accuracy still takes a few seconds [4]. Thus, when performing a series of simulations in bearing design, the entire simulation time scales up.

However, the outburst of deep learning provides an alternative strategy in lubrication prediction where, other than numerically solving the hydrodynamic lubrication's governing equation, researchers could train a deep learning model to mine the latent representations or features in lubrication data and, thus, unveil the complicated relation between the desired

friction [5], lubrication film [6], lubrication pressure [7] and the input working conditions. In these studies, the fully connected neural network, i.e., a multi-layer perceptron, is usually the most preferred one due to its simplicity in neural network design and training. For example, our previous work [4] illustrates that a simple two-layer neural network achieves accurate outlet film thickness prediction in hydrodynamic lubrication. Similarly, Bas et al [5] and Mahakur et al [8] exemplified the availability of shallow neural networks, i.e., neural networks with 2 or 3 layers, and other machine learning models, i.e., support vector machine and *K* nearest neighbour, in the prediction of friction in rolling and sliding contacts. Adopting the ready-to-use deep learning tools, e.g., the deep learning toolboxes in MATLAB [9], similar works that exploit shallow or deep neural networks to construct the implicit relation between the desired quantities, which are usually central or minimum film thickness, friction and temperature [10-11], and given working conditions have confirmed the feasibility and effectiveness of machine learning and deep learning in lubrication prediction under various scenarios. While these surrogate models are user-friendly to tribologists due to their simple structure that allows easy and fast training and deployment of the machine learning models, their outputs could only provide general information to the lubrication contact, which hinders extensive evaluation of the lubrication scenario such as the influences of cavitation where detailed knowledge on local pressure distribution becomes indispensable. To conduct detailed investigation in these scenarios, developing deeper and more complicated neural networks that generates local information on lubrication pressure is expected. While the design and implementation of the generative models is relatively more time-consuming than working with a simple surrogate model, the output of training generative models, however, is also more rewarding as it provides detailed information on the lubrication scenario, which allows more precise tuning and evaluation of the lubrication design. To unveil the lubrication characteristics over the lubrication domain, Cartwright et al [12] trained four deep neural networks, where each neural

network contains 8 layers, to independently predict the lubrication pressure, film thickness, contact deformation and temperature distributions in journal bearings from the given working conditions. Consequently, the trained neural networks successfully learn to extract latent features from the input and generate lubrication pressure and film distributions accordingly. Astonishingly, the neural network reduces the simulation time from 584 s to only 0.0567 s [12], which is comparable to performing a simple film thickness prediction by Hamrock-Dowson equation. Compared to conventional lubrication simulation that iteratively solves Reynolds equation, the prediction by neural network, however, only runs a simple forward process through the trained neural network, which is much faster than numerically solving the governing equations.

In addition to relying on a neural network's statistical ability on latent feature extraction and physical parameters generation, Almqvist [7] designed and trained a physics-informed neural network (PINN) [13] to conduct hydrodynamic lubrication analysis by integrating the governing equation and relevant boundaries conditions as PINN's loss. Thus, the training of PINN gradually minimizes the loss value to satisfy the governing equation and achieve reasonable hydrodynamic lubrication prediction. Different to typical neural networks that aim to find a nonlinear mapping from the inputs to the outputs by pure statistical approaches, PINN functions to guide the training process by setting the training target, i.e., loss of PINN, as a combination of Reynolds equation and the correspondent boundary conditions [7, 14]. Thus, the training process optimizes the neural network's parameters, e.g., biases and weights of each neuron, to approximate a hydrodynamic lubrication solution. However, our comparison [4] shows that PINN takes longer time than conventional method in running lubrication simulation under moderate accuracy. Thus, other than regarding PINN as another lubrication solver, the value of PINN lies in reverse problems of finding the most suitable working condition. For

example, Xi et al [15] expanded PINN into reverse problems to obtain the eccentricity ratio of journal bearings according to the pressure data of certain points.

Although researchers have successfully leveraged deep learning to broaden and deepen lubrication research, most of the tasks facilitated by deep learning adopt the surrogate model to give a general quantity of the lubricated contact, e.g., predicting the friction from a group of working conditions, few researches focus on obtaining the pressure distribution over the lubrication domain, which could facilitate more extensive understanding of the lubrication scenario including estimating the occurrence of cavitation and controlling pressure profile. Recently, the fast and amazing advances in generative deep learning, e.g., the renowned generative pre-trained transformer (GPT), has demonstrated that a neural network is able to generate complicated and reasonable images or sentences upon simple prompts. As illustrated in Fig. 1, the 2D film thickness or pressure distributions in hydrodynamic lubrication are similar to the pixels of an image as both are formed by matrixes/tensors. The convolutional operation in deep learning is to conduct matrix operations upon the elements inside the matrix/tensor. Thus, generating an image by a neural network is equivalent to constructing a tensor where the element of the tensor represents the RGB value of each pixel. Apparently, the tensor's element could also represent other physical quantities, such as the pressure or film thickness of lubricated contact. The only difference lies in the elements' magnitude where the RGB value is restricted as an integer within 0 to 255 while the pressure in hydrodynamic lubrication could be any positive value. However, dividing the elements by a great number, e.g., 255 for the image, transforms the elements into quantities within the range of 0 to 1, and diminishes the differences. Thus, if a neural network could generate an image (a tensor containing three matrixes), it would be suitable to generate lubrication pressure (a matrix), as long as the neural network is trained on the lubrication data with a customized loss. Inspired by this idea, we recently trained a fully connected network (FCN) that successfully predicts the

pressure and film thickness distributions from the input working condition parameters in 1D elastohydrodynamic lubrication (EHL) contact [16]. However, extending this method to 2D EHL prediction becomes difficult as FCN is usually incapable in mining latent representations from big data. To extract latent representations in 2D EHL, we further design a convolutional neural network (CNN) that is able to reconstruct high resolution EHL results from coarse data [17]. Compared to FCN, the convolutional kernels of CNN possess advanced ability in latent representation extraction and can capture rapid pressure variations inside the domain. Thus, similar to the work proposed by Liu et al. [18] that predicts stress distribution by convolutional deep learning, we deduce convolutional kernels should be effective in the generation of lubrication pressure in 2D lubricated contact.

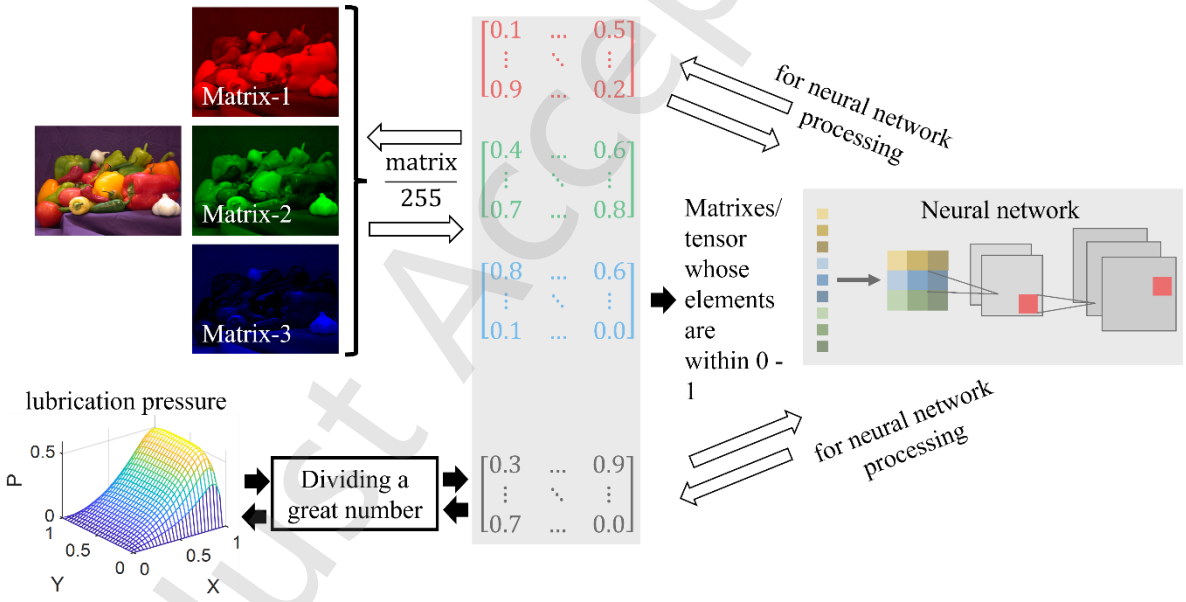

Fig. 1 The pressure distribution and an image are mathematically similar as they could be regarded as matrixes/tensors for neural network processing.

The deconvolutional neural network [19] is one of the classical models in generative tasks where the deconvolutional layers work as a decoder that gradually constructs a complete image from the input latent features. Conventionally, the latent feature could be extracted from the inputs by convolutional layer, as exemplified by U-Net [20-21]. Here, the left side of the U-Net utilises convolutional layers to extract the latent features of the input. Then, the right side

of the U-Net reconstructs the output from the latent features by deconvolutional layers. Thus, a key point in U-Net is to obtain the representative features of the input image. However, in lubrication problems, the given working condition, e.g., load, entrainment velocity, lubricant's viscosity, etc., arbitrarily determines the entire lubrication scenario, which naturally provides an explicit representation of the lubrication condition (the lubrication pressure/film thickness matrix could be obtained from the working conditions). Thus, instead of adopting a convolutional neural network to extract the latent features of hydrodynamic lubrication (the left side of U-Net), we could manually designate the working condition parameters as the representation of the lubrication scenario and, correspondently, generate lubrication pressure by a deconvolutional neural network (the right side of U-Net).

Therefore, we design and train a deconvolutional neural network to directly output the hydrodynamic lubrication pressure from the given working conditions where the deconvolutional kernels establish the mapping principle from the working condition to pressure distributions via neural network training. Compared to other works, our method expands the application of deep learning from predicting global parameters, e.g., friction coefficient, to detailed lubrication parameters distributions, e.g., pressure distribution, over the lubrication domain while maintaining fast lubrication prediction.

2. Hydrodynamic lubrication dataset: preparation and pre-processing

Here, we consider the hydrodynamic lubrication generated inside a fixed-incline-pad slider bearing, as illustrated in Fig. 2. Assuming the lower surface moves with the velocity of U and the upper surface is stationary, the moving lower surface brings the lubricant into the wedged space between the lower surface and static upper surface, which generates hydrodynamic lubrication that could be described by Reynolds equation.

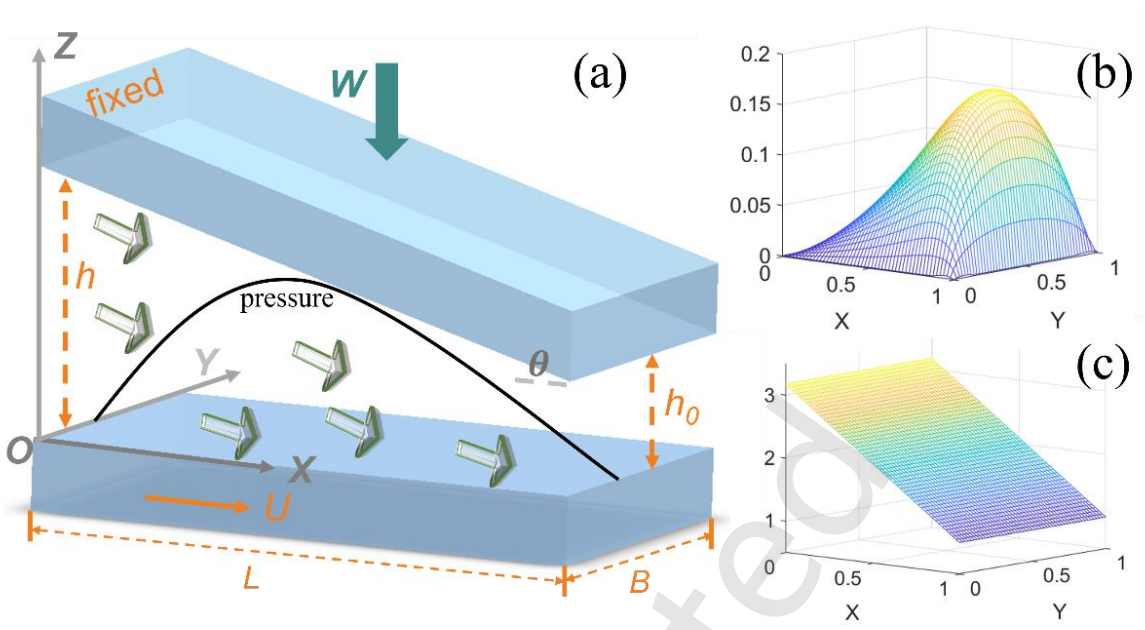

Fig. 2 Illustrations of (a): hydrodynamic lubrication contact of a slider bearing; (b): typical pressure distribution under hydrodynamic lubrication; (c): typical lubrication film distribution under hydrodynamic lubrication.

We introduce the dimensionless treatment to the original governing equations with the scaling of:

$$X = x/L, Y = y/B, H = h/h_0, P = ph_0^2/(U\eta L) \tag{1}$$

where L and B are the dimensions of the lubrication area, h is the lubrication film thickness, h_0 is outlet film thickness and η is the viscosity of the lubricant. Thus, the dimensionless Reynolds equation is:

$$\frac{\partial}{\partial X}\left(H^3\frac{\partial P}{\partial X}\right) + \left(\frac{\boldsymbol{L}}{\boldsymbol{B}}\right)^{\mathbf{2}}\frac{\partial}{\partial Y}\left(H^3\frac{\partial P}{\partial Y}\right) - 6\frac{\partial H}{\partial X} = 0 \tag{2}$$

where the film thickness is:

$$H(X) = \boldsymbol{\theta L}\frac{1-X}{h_0} + 1 \tag{3}$$

In addition, the pressure generated inside the lubrication area supports the applied load, W:

$$\int_{\Omega} P\mathrm{d}X\mathrm{d}Y = \frac{\boldsymbol{W}}{\boldsymbol{U\eta L^2 B}} h_0^2 \tag{4}$$

In Eqs. 2-4, the bold parameters are known constants. Solving Eqs. 2-4 obtains the hydrodynamic lubrication film thickness (*H*) and pressure (*P*) distributions, which would be used to evaluate the bearings performance. Fig. 2(b-c) displays the typical pressure and film thickness profiles in a slider bearing contact where the pressure distribution has no analytical expression while the film thickness distribution could be determined by the outlet film thickness, h_0, and geometrical parameters of the slider, i.e., θ and L, as evidenced in Eq. 3. Here, further inspection of governing equations of hydrodynamic lubrications reveals that their solution could be arbitrarily determined by the constant coefficients of L/B in Eq. 2, θL in Eq. 3, and $W/(U\eta L^2 B)$ in Eq. 4. When these three coefficients are determined, the only unknowns are the pressure and film thickness, which are to be obtained. Therefore, defining the constant coefficients of Eqs. 2-4 as representative parameters (RP) of $RP1 = L^2/B^2$,$RP2 = \theta L$, $RP3 = W/(U\eta L^2 B)$, we could arbitrarily determine the lubrication film and pressure distributions by the representative parameters. Conventionally, researchers need to adopt finite difference or finite element methods to solve Eqs. 2-4 simultaneously with an iterative process. Although running a single hydrodynamic lubrication simulation only takes a few seconds, running series of simulations to determine a suitable bearing performance under a wide range of working conditions easily sums to great computing costs in terms of both simulation time and simulation resources.

However, extracting the pressure from the evenly distributed mesh, the pressure forms a matrix that has similar form with the pixel distribution of an image, as illustrated in Fig. 1. For image generation, the generative deep learning model generates a tensor containing three matrixes corresponding to the RGB channels of the image. Thus, a matrix that records the pressure distribution, which is the desired output in lubrication prediction, could also be

generated using similar deep learning models. In generative deep learning for image generation, the input image will first be normalized by dividing 255 to make all elements of the tensor ranges within 0 to 1, which could be completed by the "*transforms*" function in *PyTorch*'s *torchvision* package. Following such operation, we normalize the pressure matrix by dividing a sufficiently great value, which could be a value that is slightly greater than the maximum pressure of the training dataset. After these treatments, the normalized image has the shape of $3 \times N_w \times N_L$ and the normalized pressure matrix has the shape of $N_w \times N_L$, where N_w and N_L are number of pixels/nodes along the sides of the image/matrix. To make the pressure matrix's format be aligned with the normalized image, we adopt the *reshape* function to add one dimension to the pressure matrix, transforming the pressure matrix's shape into $1 \times N_w \times N_L$. Here, declaring the number of channels in *PyTorch*'s *ConvTranspose2d* function (1 channel for the pressure matrix), we could utilise the deconvolutional function to achieve deconvolutional operations.

We implement and train the generative deep learning model to generate a matrix that stores the pressure distribution of lubricated contact. The input of the model are the working conditions, which are the three representative parameters of Esq. 2-4. To achieve this goal, we first prepare relevant hydrodynamic lubrication dataset for the training and test of the generative neural network. Here, we use COMSOL to obtain the pressure distribution and outlet film thickness of hydrodynamic lubrication where $RP1$ ($(L/B)^2$) varies from 0.01 to 100, $RP2$ (θL) varies from 2×10^{-7} to 2×10^{-5}, and $RP3$ ($W/(U\eta L^2 B)$) varies from 1×10^{7} to 5×10^{15}. To achieve a linear distribution inside these dimensionless parameters, we first apply logarithmic treatment to the representative parameters to make log(RP1) ranges within -2 to 2, log(RP2) ranges within -6.7 to -4.7 and log(RP3) ranges within 7 to 15.7. Then, we select 40 RPs from each RP's range, i.e., log(RP1) = -2, -1.8974, -1.7947, …, 2, log(RP2) = -6.7, -6.4077, -6.1154, …, 4.7, log(RP3) = 7, 7.2231, 7.4462, …, 15.7, to create 64000 potential working

conditions. If the input working condition results in a converged result, COMSOL outputs the hydrodynamic lubrication pressure distributions and film thickness under each working condition. Here, the output pressure is a 28×28 matrix for each working condition. We save the pressure distribution of each working condition as a separate file in the pressure folder to let the deep learning program easily reads the pressure data. In correspondence to the pressure files, we also save each working condition as a separate file in the working condition folder to construct the mapping from working condition to the pressure distribution. Resultantly, we successfully obtain 39353 converged results out of 64000 working conditions where the outlet film thickness ranges within 0.1 to 90 μm, indicating that our dataset covers various hydrodynamic lubrication conditions from very thin to thick lubrication films.

Considering that a standard neural network accepts the inputs whose range in within -1 to 1, we normalize the representative parameters as:

$$\overline{\log(RPi)} = \frac{\log(RPi) - \min\left(\log(RPi)\right)}{\max\left(\log(RPi)\right) - \min\left(\log(RPi)\right)}, i = 1,2,3 \tag{5}$$

to let all representative parameters distribute within 0 to 1 to facilitate neural network training. In addition, we randomly split 70% of the entire data into training dataset (Fig. 3(a)) and designate the remaining 30% data as test dataset (Fig. 3(b).

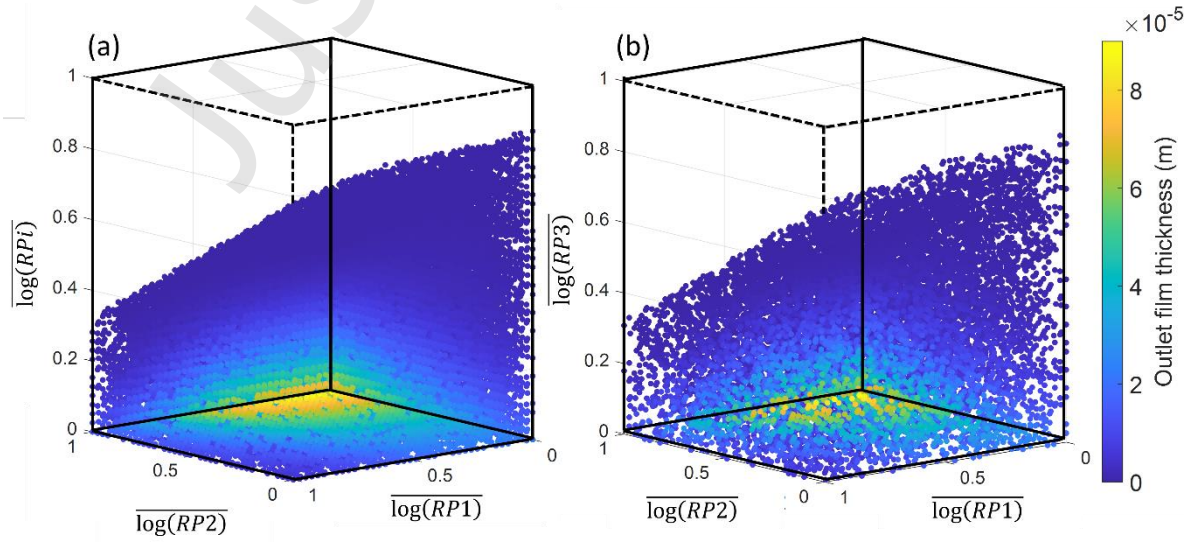

Fig. 3 Outlet film thickness distribution of the dataset. (a) Training dataset of 27548 samples; (b) test dataset of 11805 samples.

3. Deconvolutional neural network for lubrication pressure generation

To directly obtain the hydrodynamic lubrication pressure, we propose to train a deconvolutional neural network (also called transposed convolutional neural network) [19] that gradually up-scale and reshape the representative parameters (the vector of $[L/B, \theta L, W/(U\eta L^2 B)]$, where the elements are the constants in Eqs. 2-4) into the pressure distribution (a $1 \times 28 \times 28$ tensor), as illustrated in Fig. 4. To enable 2D convolutional operations, we first utilise a fully connected layer to expand the input into a 1×9 vector to allow the following reshaping of the latent feature. Alternatively, we could expand the inputs into 16, 25 or even more latent feature, but using more latent features indicates more computing cost. Here, all the representative parameters contribute to the value of each latent feature as they are linked by the fully connected neural network where the weights and biases of the neural network are learnable parameters to be optimized during neural network training. Then, we reshape the latent feature into a $1 \times 3 \times 3$ tensor to perform the deconvolutional operations. The deconvolutional operation involves 5 deconvolutional block layers where each layer contains several deconvolutional kernels, rectified linear unit (ReLU) activation functions and the batch normalization (BN) processing (Fig. 4(a)). In these blocks, the deconvolutional operations gradually increase the tensor size from $1 \times 3 \times 3$ to $128 \times 28 \times 28$ where the original 1 feature upscales to 128 latent features and the size of each feature are expanded by the deconvolutional operations. Then, we aggregate these 128 representations into the desired pressure distribution by the last deconvolutional layer. Different to the other convolutional layers, the last layer utilises a sigmoid activation function to perform nonlinear transformation and the last layer does not contain a batch normalization layer. Fig. 4(b) illustrates the deconvolutional operation where the deconvolutional kernel augments a single element into a

matrix and dispatch the matrix to the correspondent position for final latent feature construction. Here, the deconvolutional kernels' elements are learnable parameters that will be optimized during neural network training. After the deconvolutional layers, the complete deconvolutional neural network in Fig. 4(c) could directly output the hydrodynamic lubrication pressure distribution from the given representative parameters of a specific working condition.

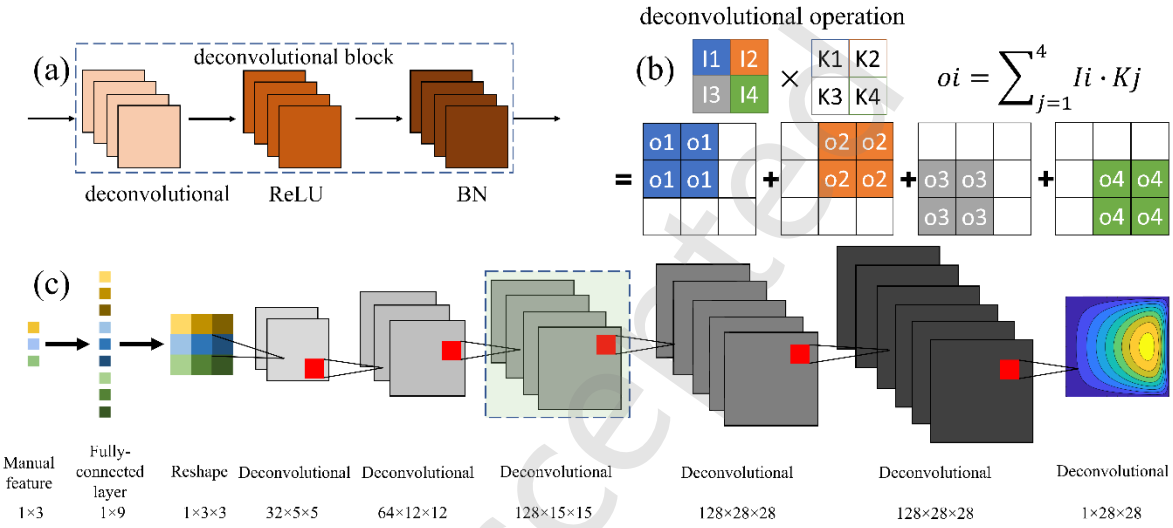

Fig. 4 Illustration of the deconvolutional neural network for hydrodynamic lubrication pressure generation. (a): the deconvolutional block for the 1st to 5th layers; (b): the deconvolutional operation; (c): deconvolutional neural network structure.

To train the deconvolutional neural network, we divide the training dataset into 108 batches where each batch contains 256 randomly selected hydrodynamic lubrication samples. Each sample contains the working conditions (inputs of the network) and pressure distributions (output of the neural network). During one epoch of neural network training, the neural network first loads a batch of data to output the pressure distribution corresponding to the input. Then, the neural network calculates the mean square error (MSE) between neural network output and the reference value within one batch:

$$\mathrm{MSE} = \frac{1}{N_k}\sum_{k=1}^{N_K}\sum_{i=1}^{28}\sum_{j=1}^{28}\left(o_{i,j,k} - d_{i,j,k}\right)^2 \tag{6}$$

where N_k is number of samples of the current batch, $o_{i,j,k}$ is the pressure of the node (i, j) generated by neural network and $d_{i,j,k}$ is the pressure provided by the training dataset. Based on the MSE value, the neural network utilises an optimizer (Adam in our model) to update the neural network's learnable parameters, e.g., the elements of convolutional kernels. With the updated neural network parameters, the neural network loads the next batch of data and repeats the process to optimize the parameters again. Such a process continues until all batches have been fed to the neural network for the training of one complete epoch. Then, the next epoch repeats the process until the trainer has reached the maximum epoch. Here, we utilise the deep learning framework, *PyTorch*, to implement neural network design, dataset definition, and neural network training. In specific, we inherent *PyTorch*'s *Dataset* and *DataLoader* to define how to load and pre-process the original data and divide them into different batches, inherent *torch.nn.module* class to construct the neural network and write a neural network training function to calculate the loss and update neural network's parameters according to the loss. To facilitate neural network training, we initialize the neural network by setting the weights as a normalized distribution with mean value of 0 and standard deviation of 0.001 and letting all biases as 0. In addition, we set the maximum epoch as 2000 and the learning rate as 0.001. Quantitatively, the entire training takes about 10 h on a single NVIDIA 4090D GPU. Although the training process is quite time-consuming, the inference process, i.e., outputting hydrodynamic lubrication pressure from the given working condition, costs less than 0.1 s, which is much faster than a conventional lubrication solver.

Fig. 5 plots the pressure distribution obtained by the trained neural network (NN, Fig. 5(a-c)), finite element method (FEM, Fig. 5(d-f)), and their differences (Fig. 5(g-i)) on the training dataset. Clearly, the deconvolutional neural network outputs consistent pressure generation with FEM calculations under various working conditions: for long slider contact in Fig. 5(a, d), the neural network is able to depict the rapid pressure drop near the outlet; for the contact where

the slider's length is comparable to its width (Fig. 5(b, e)), the neural network shows similar result with FEM; for wide slider contact in Fig. 5(c, f), the neural network also successfully displays steep pressure reduction at the side leakage boundary. The pressure difference in Fig. 5(g-i) is all within 5% of the maximum pressure, which quantifies the high accuracy of the generative deep learning model.

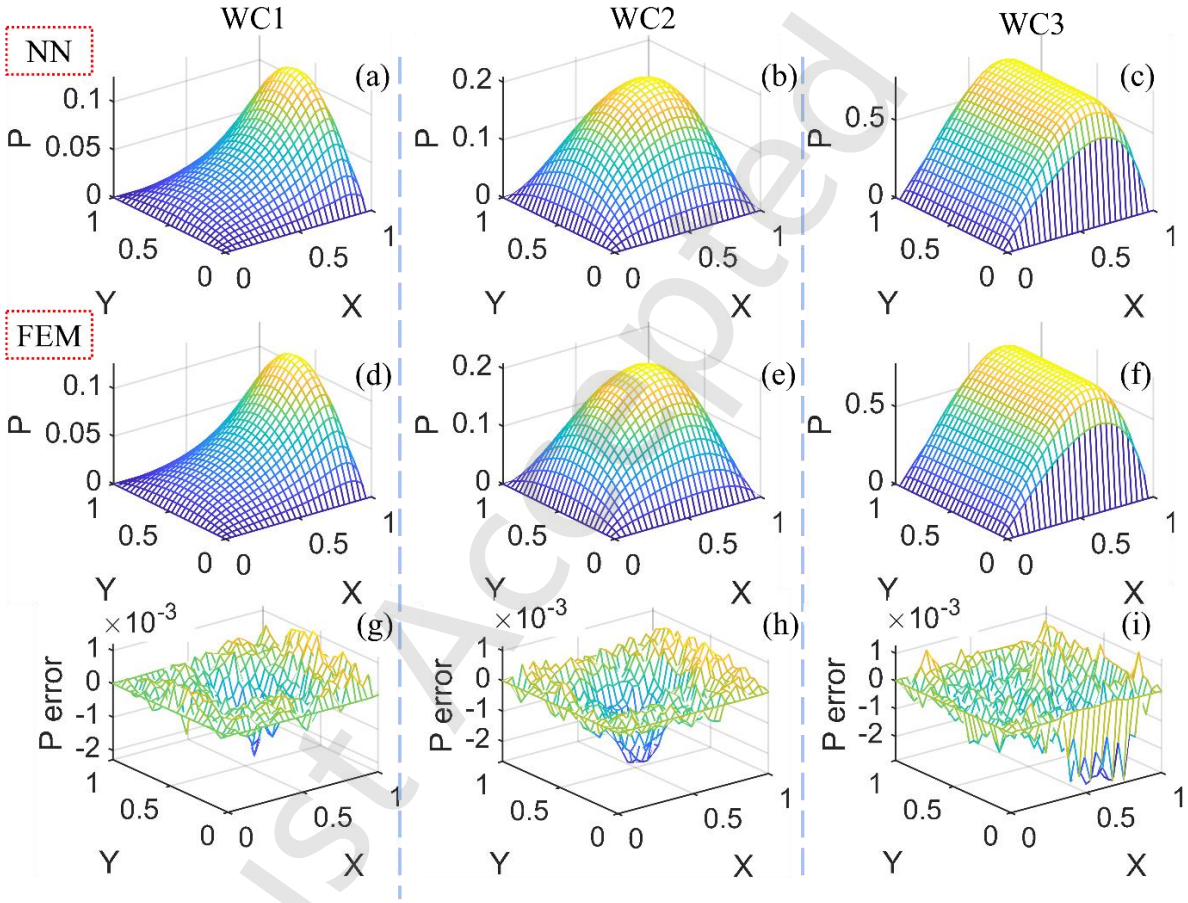

Fig. 5 Comparison of the hydrodynamic lubrication pressure between neural network generation (a-c) and FEM calculation (d-f) on the training dataset. (g-i): error between neural network prediction and FEM results.

(WC1: $L^2/B^2 = 11.9378$, $\theta L = 3.829 \times 10^{-6}$, $W/(U\eta L^2 B) = 1.7002 \times 10^9$;

WC2: $L^2/B^2 = 1.8047$, $\theta L = 7.3305 \times 10^{-7}$, $W/(U\eta L^2 B) = 4.749 \times 10^9$;

WC3: $L^2/B^2 = 0.01$, $\theta L = 8.751 \times 10^{-6}$, $W/(U\eta L^2 B) = 1.0173 \times 10^9$).

Fig. 6 further compares central pressure distribution along the entrainment direction (considering that the 28×28 pressure matrix does not contain data at $Y = 0.5$, we extract pressure distribution at $Y = 0.4815$) and at the boundaries. While Fig. 5(a) confirms that the pressure generated by neural network corresponds well with FEM results, Fig. 5(b-d) shows oscillated pressure behaviour around the zero-pressure boundary condition where the pressure deviates the zero-pressure condition at the exit boundary ($X = 1$) and satisfies the boundary condition best at the inlet boundary ($X = 0$). Quantitatively, the pressure deviation at the boundary is less than 1% of the maximum pressure magnitude. Thus, the non-zero pressure error at the boundaries does not result in significant influence on the whole results. Statistically, the MSE value over the entire training dataset ranges from 1×10^{-4} to 3×10^{-4} in ten independent trainings, which endorses the effectiveness and stability of the current neural network.

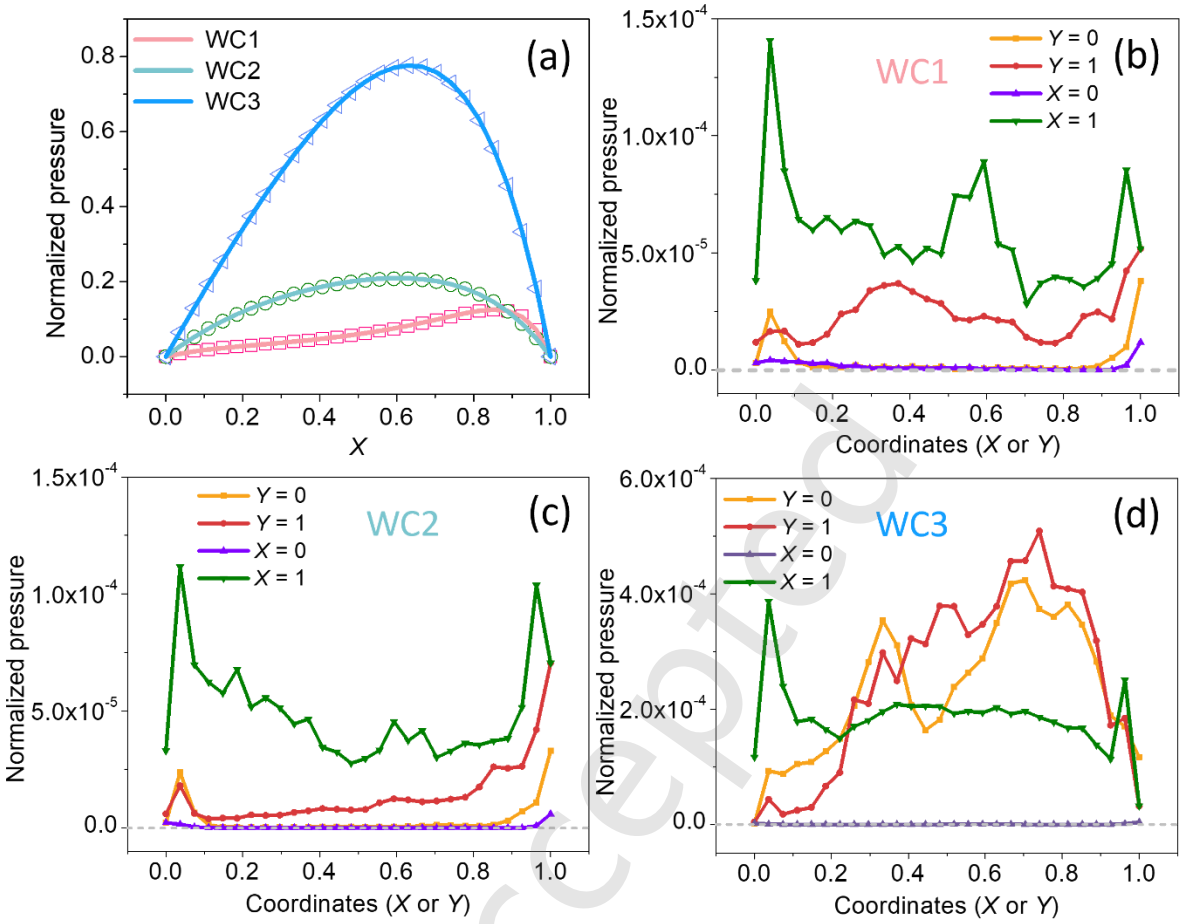

Fig. 6 Pressure distribution along entrainment direction and at boundaries. (a): Pressure distribution along $Y = 0.4815$ (solid line: FEM calculation, scatters: neural network generation); (b-d): pressure distribution at the boundaries under WC1, WC2 and WC3.

To further exhibit the neural network's performance in hydrodynamic lubrication pressure prediction, we compare the neural network's output with FEM results on the test dataset, i.e., data that never appears during neural network training, in Fig. 7. Similar to Fig. 5, the neural network also achieves consistent results with FEM calculations under working conditions of long, wide or quasi-square sliders that possesses different pressure distribution characteristics, which confirms that the trained neural network captures the latent high-dimensional features of pressure distribution by the deconvolutional kernels via neural network training and, thus, is able to predict hydrodynamic lubrication pressure from the given working conditions.

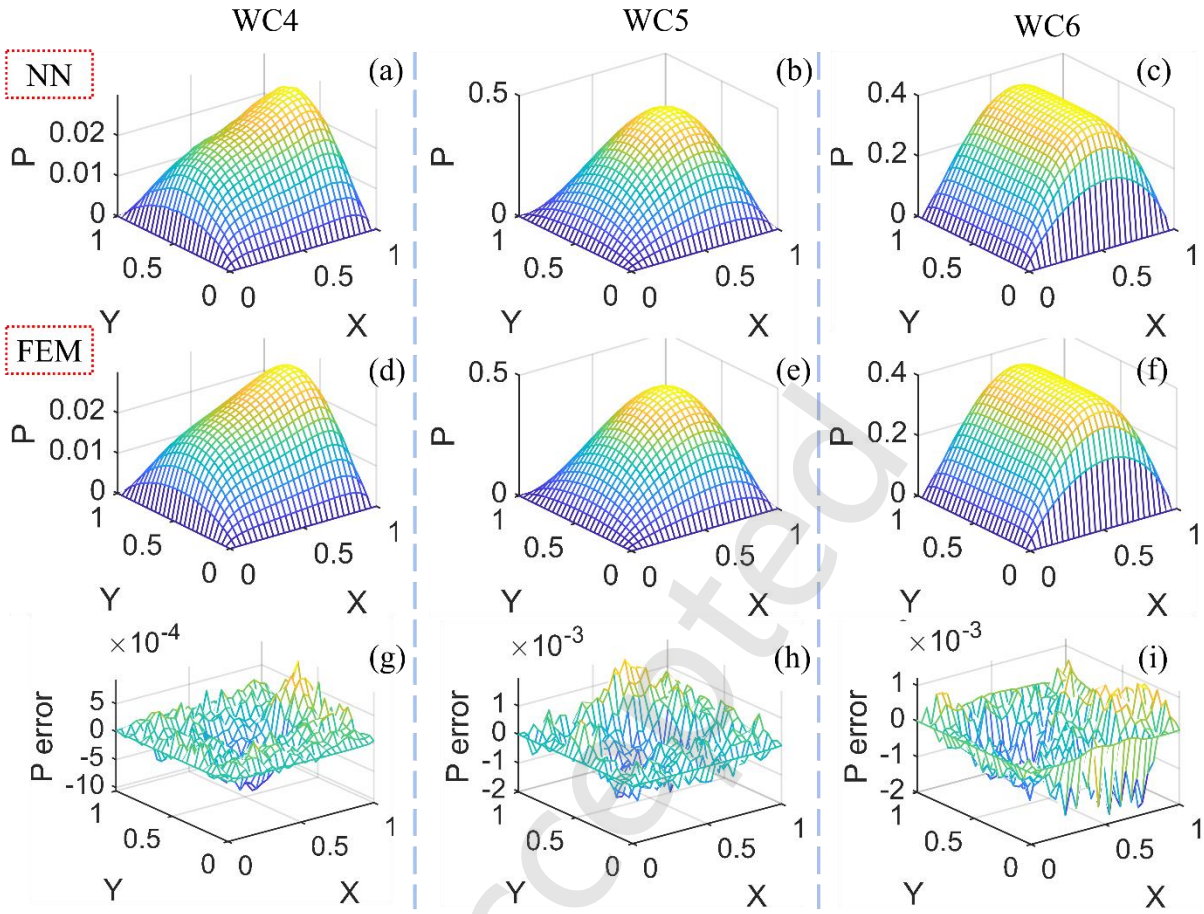

Fig. 7 Comparison of the hydrodynamic lubrication pressure between neural network generation (a-c) and FEM calculation (d-f) on the test dataset. (g-i): error between neural network prediction and FEM results.

(WC4: $L^2/B^2 = 19.1448, \theta L = 1.3229 \times 10^{-6}, W/(U\eta L^2 B) = 2.1792 \times 10^8$;

WC5: $L^2/B^2 = 0.8886, \theta L = 9.2832 \times 10^{-7}, W/(U\eta L^2 B) = 2.217 \times 10^{10}$;

WC6: $L^2/B^2 = 0.0203, \theta L = 1.4034 \times 10^{-5}, W/(U\eta L^2 B) = 1.6713 \times 10^7$).

Fig. 8 plots the pressure distribution along the entrainment direction and at the slider boundaries. In Fig. 8(a), the pressure distribution generated by neural network correlates well with FEM results, confirming the effectiveness and feasibility of running lubrication simulation by deep learning. Similar to Fig. 6(b-d), the generated pressure also slightly deviates from the zero-pressure boundary condition. Considering that the last layer's sigmoid activation function, $f(x) = 1/(1 + e^{-x})$, could only approach but never obtain 0 value, the pressure, which is

obtained after sigmoid function processing, deviates from the standard zero-pressure boundary condition. Thus, although the deconvolutional layers in the neural network's last block has obtained negative latent characteristics with relatively great magnitude, the transformation from the deconvolutional output to neural network output via the sigmoid function could not strictly obey the zero-pressure boundary condition.

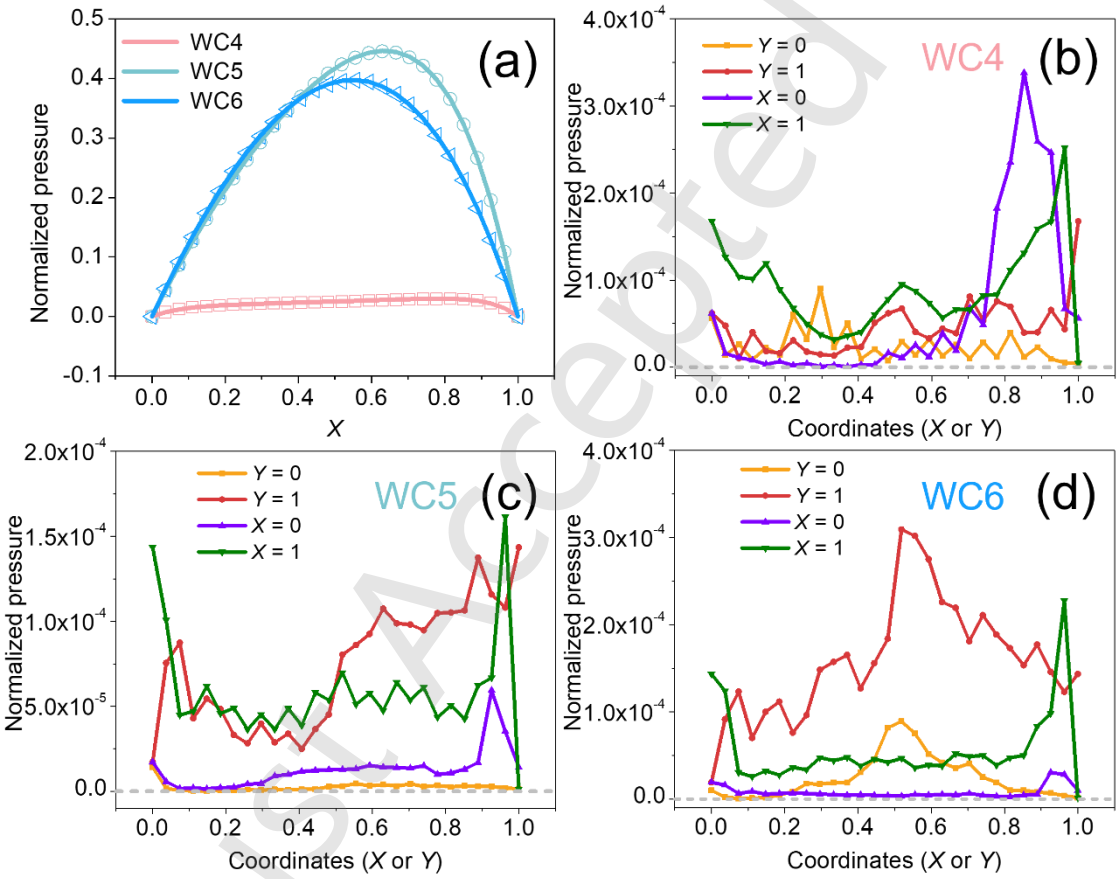

Fig. 8 Pressure distribution along entrainment direction and at boundaries. (a): Pressure distribution along $Y = 0.4815$ (solid line: FEM calculation, scatters: neural network prediction); (b-d): pressure distribution at the boundaries with WC4, WC5 and WC6.

To illustrate how the last layer's activation function works to transforms the latent feature to the desired pressure distribution, Fig. 9 compares the pressure distribution after the last deconvolutional layer and the neural network's output under WC5. Fig. 9(a-b) shows that, while the latent feature at the inlet and bottom boundaries has reached -15, the latent feature in

the other region ranges within -5 to 0. Resultantly, the activation function in Fig. 9(c) transforms the inlet and bottom latent feature into boundary pressure that approaches zeros, as illustrated by the $X = 0$ and $Y = 0$ lines (purple and yellow lines) in Fig. 8(c). For the latent features at the outlet and upper boundaries, the increased latent feature magnitude corresponds to increased pressure deviation of the zero-pressure condition than the other boundaries, as illustrated by the $X = 1$ and $Y = 1$ lines (green and red lines) in Fig. 8(c). For the non-boundary latent features that ranged within -5 to 0, the activation function transforms them into the normalized hydrodynamic pressure in Fig. 9(d).

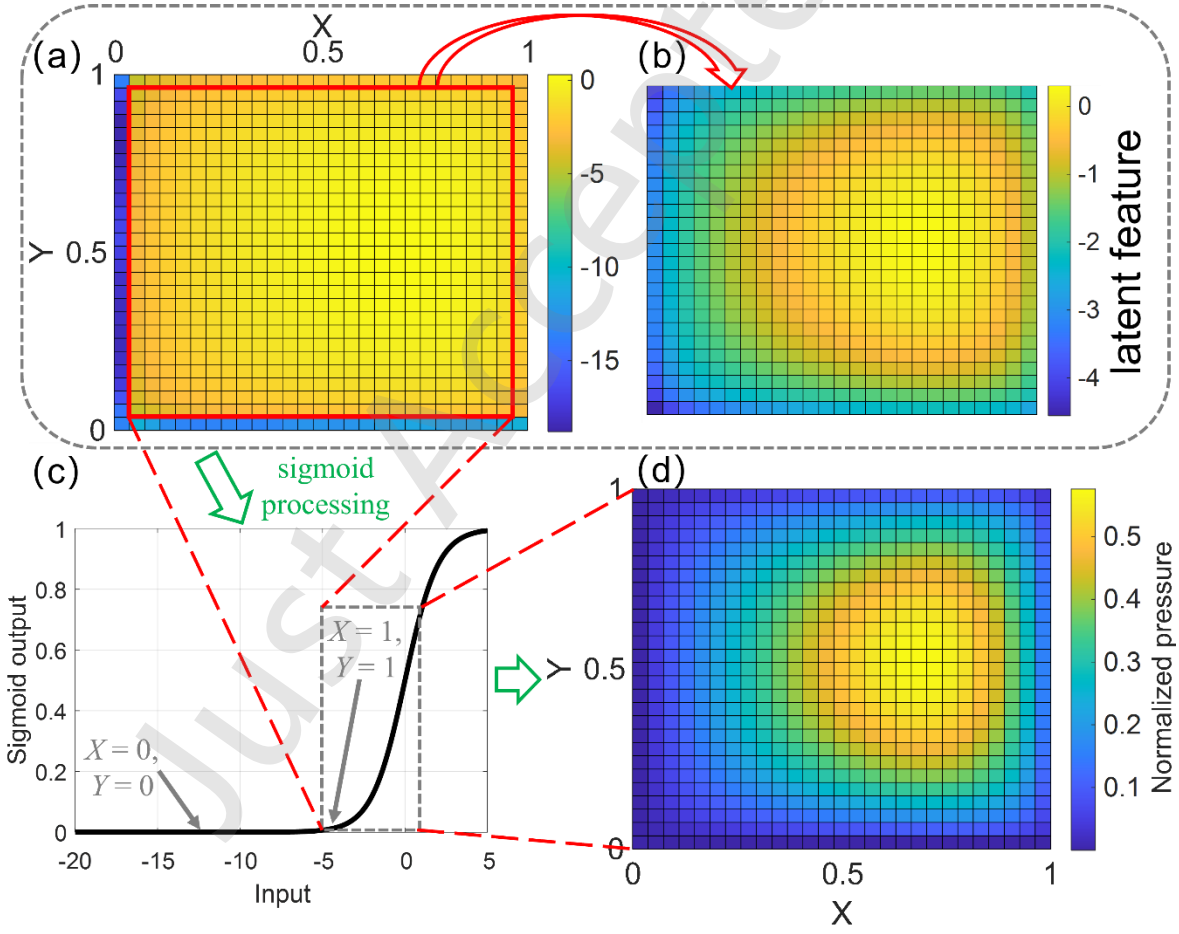

Fig. 9 The transformation from the 6th layer to the final output under WC5. (a): output of the 6th layer; (b): pressure distribution inside the red square; (c): the non-linear mapping of sigmoid activation function; (d): final output of the deconvolutional neural network.

In Fig. 10, further inspection of the latent features under working conditions of WC4 and WC6 exhibits similar characteristics with Fig. 9 where the latent feature at the inlet and bottom boundaries is usually less than -5 and the latent feature at the other boundaries ranges within -5 to 0. Thus, the pressure at the inlet and bottom boundaries approaches zero-pressure boundary condition better than that at the outlet and upper boundaries after applying the sigmoid activation function. However, considering that the maximum deviation is still within 1% of the maximum pressure, the non-zero pressure fluctuation at the boundaries only results in insignificant influence on the results.

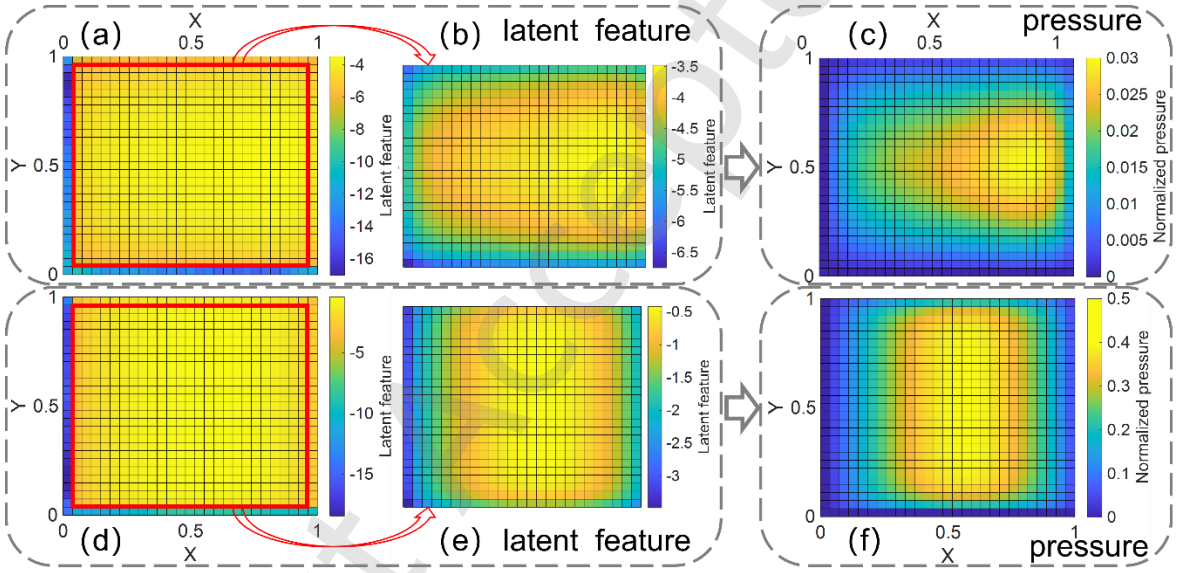

Fig. 10 illustration of the transformation from the last latent feature to neural network output. (a-c): transformation for WC4; (d-f): transformation for WC6.

4. Discussion on generative deep learning for lubrication modelling

4.1 The latent representation learned during training

In the previous section, we have shown that deep learning could directly generate lubrication pressure from the given working conditions. The general idea behind is that a neural network could learn to extract the latent features of the hydrodynamic lubrication during neural network training. However, what specific features did the neural network learn remains vague. To look

deep into the latent features, Fig. 11 visualizes some latent features obtained after the 5th convolutional operations. Fig. 11 shows that different kernels specialize in constructing different kinds of latent features. In Fig. 11(a-b), while one specific kernel forms continuously distributed features across (Fig. 11(a)) or along (Fig. 11(b)) the entrainment direction, the same kernel also leads to discretely distributed features at the other direction. For Fig. 11(c-d), the kernels obtain features resembling a checkboard, i.e., the obtained features are isolated by zero-value features. Fig. 11(e-f) show kernels specializing in learning features at boundaries where there are more kernels contributing to learning inlet and lower boundary features than learning outlet and upper boundary features. A complete visualization of the latent features after each layer could be referred to the supplemental information. Although these kernels work to construct different features, none of these kernels can obtain a consecutive and smooth latent feature distribution inside a local region but shows the checkerboard-like features, which indicates that the learned features vary abruptly along the neighbouring nodes. Thus, the neural network needs different convolutional kernels to learn different latent features and aggregates them to fill in the zero-value region and smoothen the feature distribution to finally generate the pressure map. Further inspection of the latent features generated at other layers also shows similar characteristics of (1): different kernels learn various latent features after training; (2): a single kernel usually learns discretely distributed latent features resembling a checkerboard. In addition, we also find not all kernels work well to learn latent features, as there are several kernels in layers 1-5 that outputs zero (as visualized in the supplemental information).

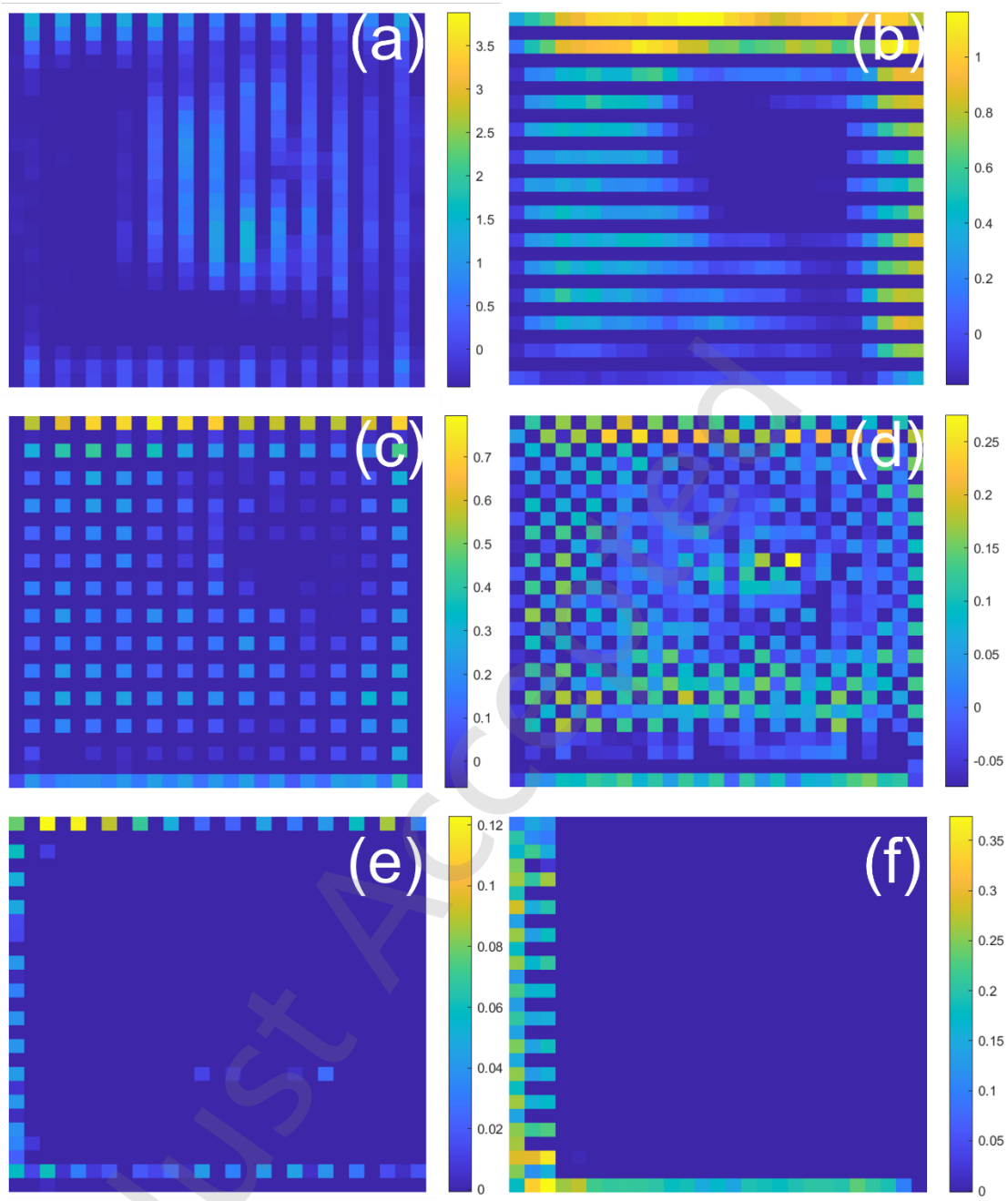

Fig. 11 Some latent features generated in the 5th layer under the working condition of $L^2/B^2 = 0.7017, \theta L = 1.2471 \times 10^{-5}, W/(U\eta L^2 B) = 3.6421 \times 10^8$. (a-b): Latent feature generated along and across the entrainment direction. (c-d): latent features inside the domain resembling a checkerboard; (e-f): latent features generated at the boundaries.

4.2 The model's performance on extrapolated prediction

The previous section proves that the trained deconvolutional neural network can generate hydrodynamic lubrication pressure within the working condition range. To further test the neural network's performance, we compare the neural network's results with FEM in Fig. 12 where the 1st representative parameter is out of the training dataset's range. Fig. 12 illustrates that under extreme conditions where the slider is very long or wide, the neural network is still able to output lubrication pressure correctly. Such results further prove that, rather than simply "remembers" the pressure distribution during the training process, the neural network has learned to extract the latent features and utilise these learned features to generate pressure distribution from the input parameters. However, compared to the results in Figs. 5-8, the maximum relative error is about 10% of the maximum pressure, which is greater than that in Figs. 4-7. Thus, our current work needs further development, e.g., adopting more advanced models or expand the dataset's size, to further increase the generative deep learning model's accuracy on data that are out of the training dataset's range.

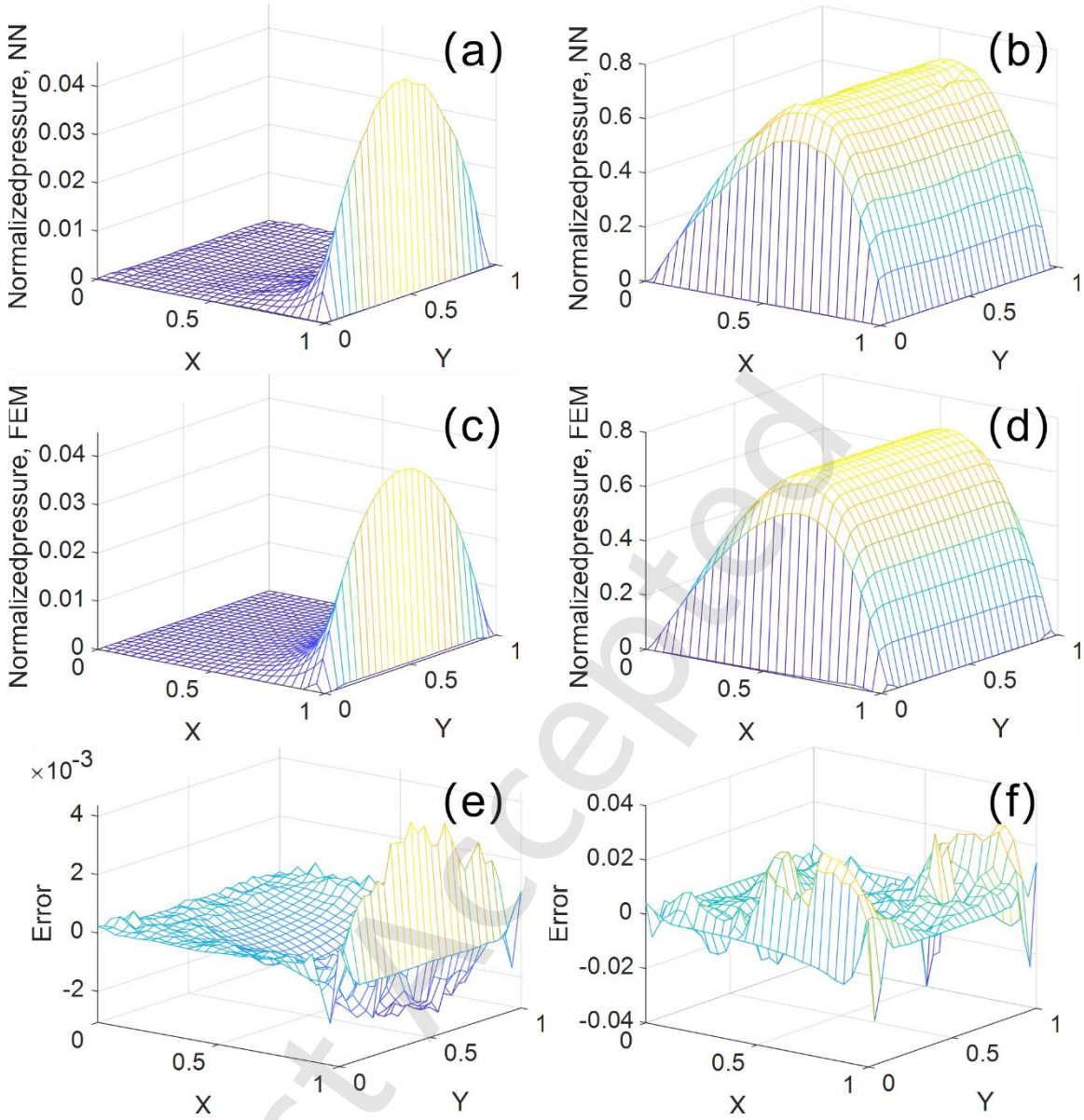

Fig. 12 Comparison of pressure between NN output and FEM calculation. (a, b): normalized pressure from NN; (c, d): normalized pressure from FEM; (e, f): pressure error between NN and FEM;

((a, c, e: long slider): $L^2/B^2 = 150$, $\theta L = 1 \times 10^{-6}$, $W/(U\eta L^2 B) = 5 \times 10^{10}$;

(b, d, f: wide slider): $L^2/B^2 = 0.005$, $\theta L = 1 \times 10^{-6}$, $W/(U\eta L^2 B) = 5 \times 10^{10}$.)

4.3 Fully connected neural network for lubrication film thickness prediction

The previous section has confirmed the feasibility of deep learning in lubrication pressure generation. Besides lubrication pressure, the lubrication film is another key indicator of the

lubrication performance. Compared to hydrodynamic pressure whose distribution parameters remains unknown *a prior*, Eq. 2 asserts a linear film thickness distribution over the entire lubrication domain where the only unknown variable is the outlet film thickness. Thus, rather than training another deep neural network to generate 2D lubrication distribution, we only train a surrogate model to predict the outlet film thickness and deducing the entire lubrication film accordingly. In Fig. 13(a), we construct and train a two-layer neural network to predict the outlet film thickness from the given working conditions. Here, the first layer expands the representative parameters into 100 latent features by the sigmoid activation function, and the second layer aggregates these latent features to output the outlet film thickness. Fig. 13(b) shows that the trained neural network achieves accurate outlet film thickness distribution on both the training and test dataset. Quantitatively, the neural network obtains the MSE value of 3.63×10^{-7} and 5.87×10^{-7} on the training and test dataset, respectively.

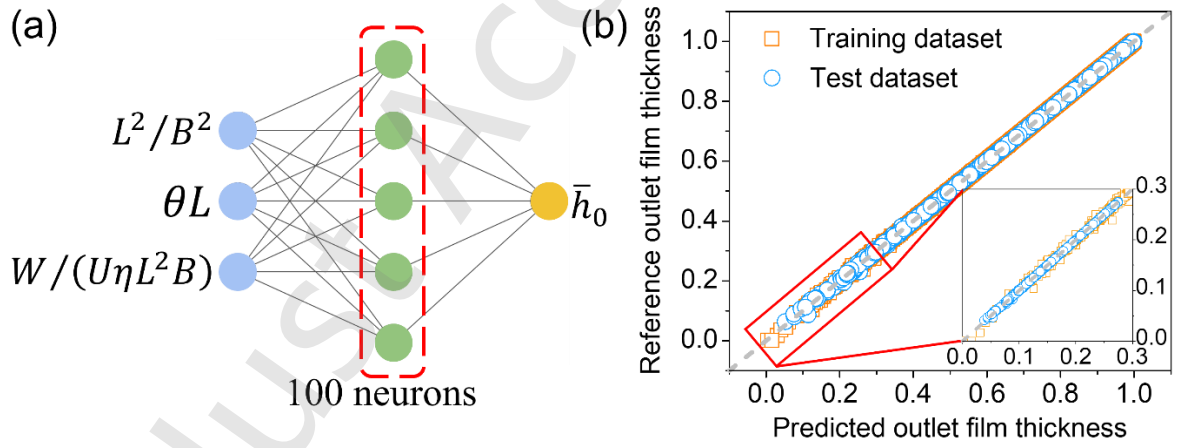

Fig.13 Performance of a single neural network for outlet film thickness prediction. (a) neural network structure, (b): neural network performance.

Different to the generative models, here, the surrogate model achieves lubrication film prediction task more rapidly without sacrificing predicting accuracy. In hydrodynamic lubrication, the lubrication film could be analytically described by several key parameters. Thus, a surrogate model that only outputs a global parameter, i.e., the outlet film thickness, is

preferred due to its simplicity. For the lubrication pressure distribution, a surrogate model becomes incapable as the pressure distribution lacks an analytical expression, which requires a generative model to obtain the lubrication pressure. Thus, the preference over surrogate models and generative models should be made upon the specific expectation: when global parameters, e.g., minimum film thickness and maximum pressure, are needed in evaluating the bearing's performance, a surrogate model is a more suitable candidate due to its simplicity in neural network training; when local parameter, e.g., the pressure distribution, is needed to further optimize the bearing design, a generative model is preferred as it provides local information on the lubrication scenario.

4.4 Extension to journal bearing lubrication with cavitation

The preceding contents have demonstrated generative models' ability in obtaining the pressure distribution over the entire lubrication domain. However, predicting the pressure/film distribution of hydrodynamic lubrication is usually an easy task, which could be completed by surrogate models and conventional methods. To illustrate generative deep learning's versatility in handling different kinds of lubrication, we extend the generative model into journal bearing lubrication with cavitation. To prepare the dataset for training, we utilise the "*journal bearing with cavitation*" model provided in COMSOL's application libraries to generate pressure distribution in journal bearings where the bearing's lubrication film is:

$$h(\theta) = c\big(1 + 0.6\cos(\theta)\big) \tag{7}$$

where θ is angle along circumference and c is bearing clearance. We consider the cavitation phenomenon by characterizing the lubricant's density as:

$$\rho(p) = \rho_c e^{\beta p} \tag{8}$$

where $\rho_c = 1000\ \mathrm{kg}/m^3$ is the default lubricant density at cavitation pressure and the bulk modulus $\beta = 10^{-7}\ \mathrm{Pa}^{-1}$. Here, the simulation model is used for illustrative purposes, we keep most of the parameters' value as the default value in COMSOL. With the current model, we

obtain pressure distributions under 63999 various working conditions where the lubricant's viscosity (η) varies from 0.005 to 0.03 Pas, the clearance (c) varies from 10 to 100 μm and the rotation speed (w) varies from 500 to 2000 RPM. Dividing 70% of the data into training dataset and the remaining as test dataset, we retrain the deconvolutional neural network to predict the pressure distribution in journal bearings with cavitation. Fig. 14 illustrates the pressure distribution in journal bearings where the model results (Fig. 14(a-b)) correspond well with the FEM results (Fig. 14(c-d)) on the training and test datasets (please be noted that the angle is adjusted to centre the pressure). Fig. 14(e-f) displays the error between the neural network outputs and the FEM results, where the maximum error between neural network output and FEM results is about 10% of the maximum pressure. Here, while the generative model could reproduce the abrupt pressure reduction at the outlet area, Fig. 14(e) shows that the maximum prediction error occurs at the outlet area too, indicating that further improvement of the neural network is expected to predict the pressure distribution more precisely. When the pressure decrement at the outlet area becomes less abrupt, the maximum prediction error in Fig. 14(f) reduces to 5% of the maximum pressure. Therefore, further improvement of the neural network structure is expected to diminish the prediction error at the outlet area. Moreover, considering that the parameters such as ambient density of the lubricant and the applied load also influence the lubrication condition, a more general and robust generative model would be desired in the future work. In addition, introducing more advanced neural network structure, e.g., ResNet, to let the neural network improve its accuracy at the outlet area and adopting a conditional variational autoencoder (conditional VAE) to involve more working condition parameters, would be a potential solution for applying deep learning in lubrication prediction.

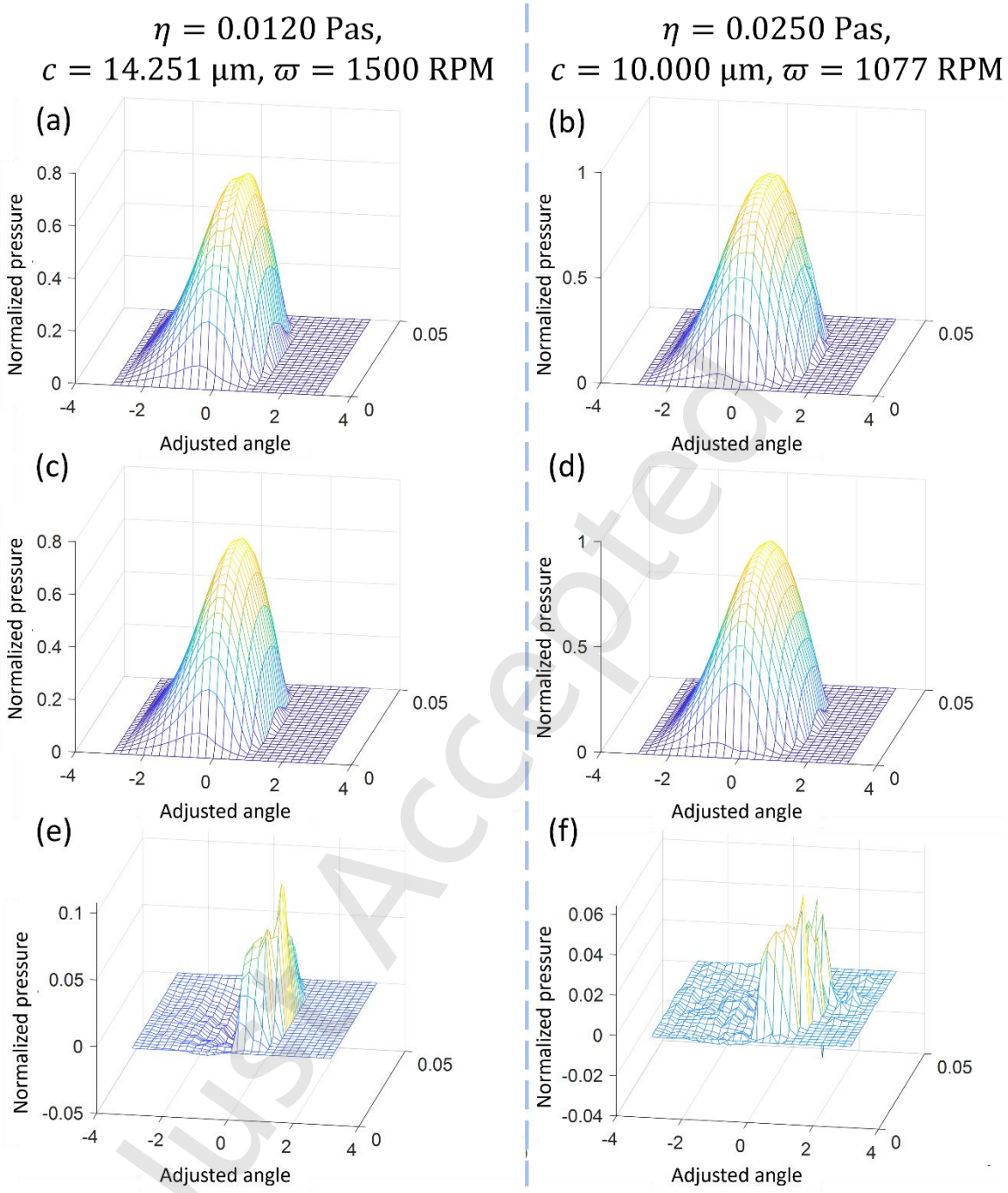

Fig. 14 Comparison of lubrication pressure in journal bearings with cavitation in the training (1st column) and test datasets (2nd column), ((a-b): lubrication pressure distribution obtained from the deconvolutional neural network, (c-d): lubrication pressure distribution calculated by FEM; (e-f): pressure error between the neural network prediction and FEM results).

6. Conclusions

The generation of images and prediction of lubrication pressure distribution shares similar concepts as both involves the operations upon matrixes/tensors. Thus, we propose to predict lubrication pressure distribution by a deconvolutional neural network. Training with a dataset that covers 27000 different working conditions, the deconvolutional kernels in the neural network has learned to extract latent features from the input representative parameters and generate the pressure distribution accordingly. Resultantly, the trained neural network is able to predict the hydrodynamic lubrication pressure within 0.1 s under various working conditions, which would accelerate the bearing design process that involves numerous lubrication predictions. Moreover, the neural network also works well when the working condition is out of the training data's range and is extended to journal bearing lubrication with cavitation.

Acknowledgement

This paper is supported by the Guangdong Basic and Applied Basic Research Foundation (No. 2023A1515240032) and Scientific Research Startup Fund for Shenzhen High-Caliber Personnel of SZPU (No. 6023330008K).

Author contributions

YZ designed and programmed the deconvolutional network codes, collected data, drafted the manuscript and acquired research funding. ZH & SG contributed to dataset generation and programming. ZF contributed to the finetune of the neural network. YL revised the neural network structure and acquired research funding.

References

[1] Wang L, Cai J, Ding X, Wang Z, Wang X. (2024). Multidisciplinary design and optimization of journal bearing for high-power wind turbine speed increaser. *Tribol Int* **197**: 109765 (2024)

[2] Rolink A, Jacobs G, Schröder T, Keller D, Jakobs T, Bosse D, Lang J, Knoll G. Methodology for the systematic design of conical plain bearings for use as main bearings in wind turbines. *Forsch. Ingenieurwes* **85**(2): 629-637 (2021)

[3] Euler J, Jacobs G, Loriemi A, Jakobs T, Rolink A, Röder J. Scaling challenges for conical plain bearings as wind turbine main bearings. *Wind* **3**(4): 485-495 (2023)

[4] Zhao Y, Wong P P. A hybrid data-driven approach for the analysis of hydrodynamic lubrication. *Proc I Mech Eng J - J Eng Tribol* **238**(3): 320-331 (2024)

[5] Hasan B A Ş, Karabacak Y E. Triboinformatic modeling of the friction force and friction coefficient in a cam-follower contact using machine learning algorithms. *Tribol Int* **181**: 108336 (2023)

[6] Marian M, Mursak J, Bartz M, Profito F J, Rosenkranz A, Wartzack S. Predicting EHL film thickness parameters by machine learning approaches. *Friction* **11**(6): 992-1013 (2023)

[7] Almqvist A. Fundamentals of physics-informed neural networks applied to solve the Reynolds boundary value problem. *Lubricants* **9**(8): 82 (2021)

[8] Mahakur V K, Bhowmik S, Patowari P K. Tribo-informatics evaluation of dry sliding friction of silanized jute filler reinforced epoxy composites using machine learning techniques. *Tribol Int* **183**: 108388 (2023)

[9] Kim P. MATLAB deep learning with machine learning, neural networks and artificial intelligence. A Press Berkeley, CA (2017)

[10] Ji H, Wang J, Zhang W, Zhao Z, Li Y. Prediction of maximum temperature of fluid-lubricated bearing based on machine learning algorithm. *Int Commun Heat Mass* **149**: 107109 (2023)

[11] Singh A, Wolf M, Jacobs G, König F. Machine learning based surrogate modelling for the prediction of maximum contact temperature in EHL line contacts. *Tribol Int* **179**: 108166 (2023)

[12] Cartwright S, Rothwell B C, Figueredo G, Medina H, Eastwick C, Layton J, Ambrose S. A machine learning-driven approach to predicting thermo-elasto-hydrodynamic lubrication in journal bearings. *Tribol Int* **196**: 109670 (2024)

[13] Karniadakis G E, Kevrekidis I G, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. *Nat Rev Phys* **3**(6): 422-440 (2021)

[14] Zhao Y, Guo L, Wong, P P L. Application of physics-informed neural network in the analysis of hydrodynamic lubrication. *Friction* **11**(7): 1253-1264 (2023)

[15] Xi Y, Sun R. Inverse problems in hydrodynamics lubrication: parameter identification in the Reynold equation by using physics-informed neural networks. *Proc I Mech Eng J - J Eng Tribol* **239**(4): 423-435 (2025)

[16] Yu G, Zhao Y, Fu Z, Chen Z. Application of back propagation neural network in the analysis of isothermal elastohydrodynamic lubrication. *Tribol Int* **198**: 109883 (2024)

[17] Zhao Y, Guo S, Fu Z, Deng Z. Reconstructing high-resolution EHL results from coarse data by a deep convolutional neural network. *Tribol Int* **210**: 110787 (2025)

[18] Liu X, Al-Qadi I L. Three-dimensional tire-pavement contact stresses prediction by deep learning approach. *Int J Pavement Eng* **23**(14): 4991-5002 (2022)

[19] Zeiler M D, Krishnan D, Taylor G W, Fergus R. Deconvolutional networks. In 2010 IEEE computer society conference on computer vision and pattern recognition: 2528-2535 (2010)

[20] Elasri M, Elharrouss O, Al-Maadeed S, Tairi H. Image generation: a review. *Neural Process Lett* **54**(5): 4609-4646 (2022)

[21] Zhang A, Lipton Z C, Li M, Smola A J. Dive into deep learning. Cambridge University Press (2023)