Potential of Data-driven Approaches for Modeling Heat and Mass Convection Processes

by

Munku Kang

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved March 2022 by the
Graduate Supervisory Committee:

Beomjin Kwon, Chair
Patrick Phelan
Yi Ren
Konrad Rykaczewski
SungMin Sohn

ARIZONA STATE UNIVERSITY

May 2022

ABSTRACT

In convective heat transfer processes, heat transfer rate increases generally with a large fluid velocity, which leads to complex flow patterns. However, numerically analyzing the complex transport process and conjugated heat transfer requires extensive time and computing resources. Recently, data-driven approach has risen as an alternative method to solve physical problems in a computational efficient manner without necessitating the iterative computations of the governing physical equations. However, the research on data-driven approach for convective heat transfer is still in nascent stage. This study aims to introduce data-driven approaches for modeling heat and mass convection phenomena. As the first step, this research explores a deep learning approach for modeling the internal forced convection heat transfer problems. Conditional generative adversarial networks (cGAN) are trained to predict the solution based on a graphical input describing fluid channel geometries and initial flow conditions. A trained cGAN model rapidly approximates the flow temperature, Nusselt number (Nu) and friction factor (f) of a flow in a heated channel over Reynolds number (Re) ranging from 100 to 27750. The optimized cGAN model exhibited an accuracy up to 97.6% when predicting the local distributions of Nu and f.

Next, this research introduces a deep learning based surrogate model for three-dimensional (3D) transient mixed convention in a horizontal channel with a heated bottom surface. Conditional generative adversarial networks (cGAN) are trained to approximate the temperature maps at arbitrary channel locations and time steps. The model is developed for a mixed convection occurring at the Re of 100, Rayleigh number of 3.9×10^6, and

Richardson number of 88.8. The cGAN with the PatchGAN based classifier without the strided convolutions infers the temperature map with the best clarity and accuracy.

Finally, this study investigates how machine learning analyzes the mass transfer in 3D printed fluidic devices. Random forests algorithm is hired to classify the flow images taken from semi-transparent 3D printed tubes. Particularly, this work focuses on laminar-turbulent transition process occurring in a 3D wavy tube and a straight tube visualized by dye injection. The machine learning model automatically classifies experimentally obtained flow images with an accuracy > 0.95.

DEDICATION

To my family

for their unconditional love and encouragement

ACKNOWLEDGMENTS

First and foremost, I would like to acknowledge my advisor, Prof. Beomjin Kwon, for his generous support and guidance. I have greatly benefited from his incredible insight and intuition. I would also like to thank the members of my Supervisory Committee, Prof. Patrick Phelan, Prof. Yi Ren, Prof. Konrad Rykaczewski, and Prof. SungMin Sohn, for their time and providing constructive criticism, valuable suggestions, and comments. I would like to express my gratitude to my friends, who have always supported and encouraged me during my Ph.D.

I would like to acknowledge the funding sources that supported the research presented in this dissertation, including: (i) ASU startup fund; (ii) National Science Foundation (Grant No. 2053413).

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

1. INTRODUCTION

1.1.Traditional Methods of Modeling Heat and Mass Convection Processes

To model convective heat transfer mathematically, the conservation laws of mass, momentum, and energy are introduced. These laws can be expressed as formulas. First, the law of conservation of mass is expressed by Equation (1.1), where *V* is velocity.

$$\nabla \cdot \vec{V} = 0 \tag{1.1}$$

$$\rho \frac{\partial V_i}{\partial t} + \rho \vec{V} \cdot \nabla \vec{V_i} = -\nabla p + \mu \nabla^2 \vec{V_i} + \rho g \tag{1.2}$$

Second, the law of conservation of momentum is expressed by Equation (1.2), where V_i is the components of *V*, ρ is density, *p* is pressure, and *g* is acceleration due to gravity. This is Newton's second law, and the sum of external forces acting on a fluid particle equals the rate of change of linear momentum. The fluid velocity can be obtained by solving the conservation laws of mass and momentum together. Third, the law of conservation of energy is expressed as Equation (1.3), where α is thermal diffusivity, and *T* is temperature. This equation consists of the advection term, which explains a net rate where thermal energy leaves the control volume due to bulk fluid motion, and the other term is a net inflow of thermal energy through conduction. Here, viscous dissipation is ignored. Fluid temperature distribution can be obtained by solving the law of conservation of energy.

$$\nabla\left(\vec{V} \cdot T\right) - \alpha \nabla^2 T = 0 \tag{1.3}$$

The physical properties of the fluid are changed according to the temperature in the heating and cooling process. During this process, density is changed with temperature, and the difference in density between fluids causes the fluid move, which is buoyancy-driven flow. The change in buoyancy with temperature can be approximated using the Boussinesq

approximation. Many modeling methods use Boussinesq approximation. However, when the temperature difference is greater than 2℃ for water and 15℃ for air, the errors are increased by more than 1% (Ferziger, Perić, and Street 2020).

If there is an imbalance of concentration between species during the convection process, heat and mass transfer occur simultaneously. The mass transfer is proportional to a concentration gradient of substance. It can be expressed by Fick's law, as expressed in Equation (1.4), where D_{AB} is the diffusion coefficient of the species in the mixture and C_A is the concentration of species.

$$\dot{m}_{diff} = -D_{AB} A \frac{dC_A}{dx} \tag{1.4}$$

To model the transport equations, computational techniques have been introduced, such as finite difference, finite elements, and finite volume methods. First, the finite difference method (FDM) is the oldest and most direct approach to discretizing partial differential equations. However, it is not frequently used in commercial software due to difficulties in handling irregular geometry. Thus, now commercial software widely uses unstructured meshing because this technology can generate meshes up to complex 3D geometry. The finite element method (FEM) has been mainly hired in structural analysis. In FEM, the mesh quality is very important because it is directly proportional to the accuracy of the analysis result and greatly influences the calculation time. In specific cases, one-dimensional (1D) and two-dimensional (2D) are preferred over three-dimensional (3D) mesh. Although this method was initially developed for analyzing solid materials, it extended to thermofluids. The finite volume method (FVM) replaced the integral of the transport equation to the conservation form, and the spatial integral to the area integral using Gauss's divergence. This change made it possible to apply control volume analysis

to the problem. Engineers have widely employed FVM in heat transfer. These numerical modeling methods can solve heat transfer equations and fluid transport equations with high accuracy and be applied to arbitrary nonlinear physical problems and complex geometries, whereas these methods provide approximate solutions and require a large computational cost.

1.2. General Concepts of Machine Learning and Deep Learning

Machine learning (ML) has been employed in a wide range of areas such as medicine, health care, robotics, stock market where useful information (e.g., patterns) must be extracted out of large-sized data in a rapid fashion (Erickson et al. 2017; Bell 2015; Smart and Kaelbling 2002; Beam and Kohane 2018). Algorithms hired in ML are classified as supervised learning, unsupervised learning, and reinforcement learning. These algorithms contributed to offering considerable promise to classification and regression works in terms of accuracy. Figure 1.1 briefly illustrates the five stages of the machine learning process. The first stage of feature preparation preprocessing requires the preparation of the qualified dataset. Here the dimensionality and noise in datasets are removed by normalization, and then proper labels are attached to datasets. In the second stage, various ML models can be hired to choose a proper model to increase its accuracy and then, while the selected ML model is being trained, comparing predictions can be adjusted by updating weight and biases at each iteration. In the testing model, the trained ML model evaluates how well it predicts outputs. Following testing the ML model, the parameters are adjusted to improve its accuracy in the parameter tuning. At the final stage, the tuned ML model is applied to classify new or unseen similar data.

Figure 1. 1. Machine Learning

Deep learning is a class of machine learning algorithms that uses multiple stacked layers of processing units to learn high-level representation from nonlinear patterns of data (Lecun, Bengio, and Hinton 2015; Deng and Yu 2013) as shown in Fig. 1.2. Unlike the conventional machine learning techniques, deep learning algorithms allow us to readily discover features from high-dimensional data, e.g. images, which contributed to processing a large amount of data, including image recognition (Krizhevsky, Sutskever, and Hinton 2017), speech recognition (Mikolov et al. 2011), science (Brunton, Noack, and Koumoutsakos 2020; Gawehn, Hiss, and Schneider 2016; S. Kang and Cho 2019), business and government (Valter, Lindgren, and Prasad 2018; Tien Bui et al. 2020). Lately, more attention toward deep learning has been increasingly growing in the fields of fluid mechanics and heat transfer, where nonlinear patterns of data are frequently encountered due to complex physics. Thermo-fluidic problems usually consist of governing physical equations that require iterative calculations to attain accurate solutions with conventional numerical simulation techniques, e.g., a finite difference method, finite volume method, and finite element method. This iterative calculation for governing physical equations causes expensive computational cost and time.

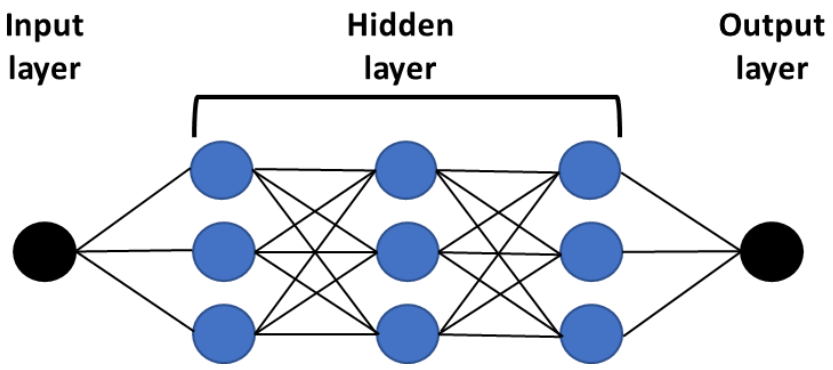

Figure 1. 2. Machine Learning Neural Network

1.3. The Emergence and Growth of Deep Learning for Engineering

Deep learning model is able to be classified into discriminative and generative model. The discriminative model finds the y that maximizes conditional probability $p(y|x)$ to determine the label y that classifies data x. That is, y is determined by learning $p(y|x)$ directly from the model. Conversely, the generative model indirectly learns the joint probability $P(x,y)$ to determine the label y of the data x or learns $p(x)$ if no label exists. In other words, if the discriminative model discovers data labels by learning the decision boundary, it properly classifies the difference in labels. The generative model learns the distribution of data to determine its type. A model then determines the probability of belonging to a distribution. This model has been developed from the basic generative model known as Naïve Bayes, which attempts to learn joint probability distribution $P(x,y)$ of the input x and the label y. It can be transformed from $P(y|x)$ using the Bayes rule. Although a discriminative model attempts to learn the conditional probability distribution $P(y|x)$, generative models can use the joint distribution $P(x,y)$ to generate likely (x,y) samples. The Naïve Bayes model has its merits as follows: it is simple and easy to

implement without requiring much training data, in addition, it can handle both continuous and discrete data fast and is less sensitive to irrelevant features. However, the problem is that it assumes that features are independent; and thus, it fails to handle images. Another generative model that has been widely adopted in ML is Gaussian mixture model (GMM) and Hidden Markov Model (HMM). However, due to its weakness in generating images, it has been mostly applied in classification work.

To overcome all these traditional generative models with limits in generating images, neural networks were adopted to generative models. As a class of reinforcement learning methods, deep neural networks were distinguished because of their substantial merit of automatically discovering features from raw data for pattern analysis or classification (Lecun, Bengio, and Hinton 2015; Deng and Yu 2013). These deep learning achievements made critical components large labeled datasets, the growth of computational power (Brenner, Eldredge, and Freund 2019; Brunton, Noack, and Koumoutsakos 2020), and the power of a multi-layer (deep) architecture- possible. In the past few years, GAN research and utilization among various generative models have been actively conducted. Therefore, I would like to take a detailed look at variational autoencoders (VAE) (Kingma and Welling 2013), convolutional neural networks (CNNs) (Dai, Lu, and Wu 2014) (used as generative models), and GANs (Goodfellow et al. 2014).

Although VAE has the same structure as an autoencoder (AE), VAE is a generative model. In latent space, AE expresses output as a set of simple numbers, whereas VAE expresses output as a distribution with a learned mean and standard deviation. The VAE's loss function comprises the total reconstruction errors between the original and loss function images and the Kullback-Leibler (KL) divergence. The KL divergence compares

the probability distribution of the latent space with the standard Gaussian distribution. Thus, VAE is a technique based on Bayesian theory, which learns distribution. The extracted distribution from the latent space is decoded, potentially generating a copy of the original image. Adjusting the vectors in latent space is attempted to generate objective images. VAE estimates one distribution mode by applying the maximum likelihood estimation, and a normal distribution close to the mean of the distributions is estimated to minimize the error. However, it is impossible to easily estimate the center of the images, and a safe middle zone is selected. This causes the eyes, nose, and mouth to be well expressed in portraits while the background is blurred.

CNNs introduce convolutional kernels to automatically extract features and expressions from high-dimensional data such as images. The number of dimensions to learn is reduced by reducing the number of parameters by having the same parameter value. Introducing this method has the advantage of extracting useful features while preventing overfitting. Also, it showed excellent performance in the image classification task. Because of these advantages of a CNN, it has been hired in structures similar to VAE. By adding an adversarial loss to the reconstruction loss of the CNN generative model, it is now possible to realistically reconstruct the background part, which was blurred in VAE. However, in the generative model, calculation loss and backpropagation learning are difficult, and there is no formal method to increase the similarity, so it did not receive much attention until the advent of GANs.

GAN is introduced as a generative model. GAN hires two architectures, a generator and a discriminator. Both architectures train competitively in the training process. This causes GAN to generate realistic images. In the discriminator, a high-quality image

produced from the generator is judged real or fake. Through this adversarial training, GAN can generate many more realistic images than any other generative model. However, it showed the problems of mode collapse and instability in the training process. To stabilize GAN training process, Wasserstein GANs (WGAN) (Arjovsky, Chintala, and Bottou 2017) that substitutes binary cross-entropy loss to Wasserstein loss and adds the Lipschitz Constraint using weight clipping to the discriminator was suggested. This enabled the gradient to converge more stable. However, sometimes, WGAN produced worse samples and failed to converge due to the weight clipping. To solve this, the Wasserstein GANs - Gradient Penalty (WGAN-GP) (Gulrajani et al. 2017) changed weight clipping into gradient-penalty in the discriminate loss function to provide more stable GAN training. Since then, many interesting studies using GAN with regard to image generation have appeared. After CNN was introduced to GAN, it provided a stable model, and it has been actively studied.

One of the newly suggested was the Image-to-image translation method using conditional GAN (cGAN) (Isola et al. 2016) that trains the network to predict images from paired images consisting of both ground truth images and input images. Also, by employing *L1* distance as a loss function, it provides less blurry images than *L2* distance. On the other hand, unlike cGAN, the CycleGAN (Zhu et al. 2017) did image translation, e.g., image style, patterns, seasons, and painting styles without requiring paired-images. In general, GAN has been widely hired for modifying images, e.g., by putting on glasses or changing hairstyle or skin color (Choi et al. 2017; Karras, Laine, and Aila 2018) and improving the resolutions of images by modifying the structures of networks based on GAN structure (Karras et al. 2017; 2019).

1.4. Research Objectives

This research attempts to develop time-efficient and accurate modeling method using machine learning for heat and mass convection processes. Although existing traditional methods of modeling have shown excellent accuracy, there is still a possibility to advance them when it comes to time, and cost aspects caused by iterative calculation. The data-driven approaches using machine learning are more advantageous in calculation time and reduced efforts for updating the designs without numerical calculations. Most importantly, it leads data-driven approaches to a time-efficient process by omitting the iterative calculation and updating design process. The objectives of this work are as follows:

1. A data-driven model for 2D convection heat transfer in steady-state is developed based on a generative model that can rapidly infer the temperature fields and convection properties of channel heat sinks from given boundary conditions, geometries, and flow conditions.
2. Test and investigate the potential of the deep conditional generative adversarial networks that can predict for the 3D transient laminar mixed convection phenomena.
3. Analysis and prediction methodologies of various flow regimes based on machine learning are developed to provide comprehensive information for accurate prediction of heat transfer characteristics and pressure drop in geometrically complex flow devices.

2. DEEP GENERATIVE MODELING FOR 2D STEADY STATE HEAT CONVECTION PROCESS

2.1. Introduction

Deep learning is a class of machine learning methods that automatically discovers features from raw data for pattern analysis or classification (Lecun, Bengio, and Hinton 2015; Deng and Yu 2013). Unlike the conventional machine learning techniques, deep learning algorithms allow us to readily discover features from high-dimensional data, e.g. images, and have been impacting various areas dealing with a large amount of data such as image recognition (Krizhevsky, Sutskever, and Hinton 2017), speech recognition (Mikolov et al. 2011), science (Brunton, Noack, and Koumoutsakos 2020; Gawehn, Hiss, and Schneider 2016; S. Kang and Cho 2019), business and government (Valter, Lindgren, and Prasad 2018; Tien Bui et al. 2020). Recently, the interest in deep learning has been growing in the fields of fluid mechanics and heat transfer where nonlinear patterns of data are frequently encountered due to complex physics. Efforts are underway to develop deep learning techniques that can infer the patterns of thermofluidic processes from provided conditional information, e.g., system geometry, boundary and initial conditions. Previous studies show that the deep learning techniques are able to predict the patterns of flows (Raissi, Yazdani, and Karniadakis 2020; Lee and You 2019; Farimani, Gomes, and Pande 2017; McClure and Carey 2021) or temperature distributions (Farimani, Gomes, and Pande 2017; Sharma et al. 2018; Edalatifar et al. 2020; Cai et al. 2021) of thermofluidic processes if physical conditions are prescribed. When predicting the solutions of physics problems, the deep learning techniques approximate the output without iteratively calculating the governing physical equations, thus they demand lower computational costs than the

conventional numerical simulation techniques, e.g., finite difference method, finite volume method, and finite element method. Although the conventional numerical approaches can offer accurate solutions to the intricate problems, e.g., transient, two-dimensional, three-dimensional, or conjugate problems, the computational costs are often tremendous particularly when requiring high-resolution, large-scale, or long-period solutions. Thus, researchers have been investigating the deep learning techniques as an alternative modeling approach for thermofluidic processes.

Several recent publications explored how to cost-effectively infer the thermofluidic processes using deep learning techniques (Raissi, Yazdani, and Karniadakis 2020; Yang et al. 2019; Sharma et al. 2018; Edalatifar et al. 2020; Farimani, Gomes, and Pande 2017; Lee and You 2019). Some studies employed conditional generative adversarial networks (cGAN) (Farimani, Gomes, and Pande 2017), fully convolutional encoder-decoder network (Sharma et al. 2018), or autoencoder (Edalatifar et al. 2020) to generate the solutions for steady-state two-dimensional (2D) heat conduction problems. When the temperature along the boundary of the model domain was given in a 2D image format, the deep learning model inferred the corresponding temperature distribution within the domain similar to the conventional numerical techniques without solving the heat diffusion equation. Another demonstration used the cGAN model to predict the cooling effectiveness distributions of an effusion cooling technique while varying the design of a porous plate (Yang et al. 2019). The deep learning techniques based on generative adversarial networks (GAN) (Lee and You 2019), cGAN (Farimani, Gomes, and Pande 2017), or convolutional neural networks (CNN) (Lee and You 2019) also succeeded in approximating the solutions for fluid mechanics problems. When the input data were three-channel images describing

the 2D velocity vectors and pressure fields along the 2D domain boundary, the cGAN model was able to predict the corresponding flow and pressure fields within the domain (Farimani, Gomes, and Pande 2017). For the unsteady flow over a cylinder, GAN and CNN models predicted the flow fields around the cylinder during a short period of time, if the flow fields during the past moment were provided (Lee and You 2019). More recently, a physics-informed deep neural network was developed that was capable of inferring the flow and pressure fields from several snapshots of solute concentration fields (Raissi, Yazdani, and Karniadakis 2020). While the early works demonstrate the potential of deep learning for heuristic modeling, still many questions remain regarding how to leverage the potential of deep learning to solve complex thermofluidic problems. Some questions are related to the sizes of training data required for different problems, settings in algorithms to avoid underfitting and overfitting issues, and methods of generating sufficient amount of training data.

This chapter presents a heuristic model for forced convection heat transfer problems based on cGAN that rapidly infers the convection properties and temperature fields from boundary conditions. The cGAN model learns forced convection heat transfer through provided numerical simulation solutions, and predicts the solutions for unseen boundary conditions. To improve the accuracy of the cGAN model, the influence of important factors is investigated such as the size of training set, training epoch, and hyperparameters, e.g., the trade-off parameter λ, and learning rate.

2.2. Convection Problem

A simple internal forced convection problem occurring in a 2D straight channel is considered where the channel width (w) is 66.6mm and length (l) is 153mm as shown in Fig. 2.1a. At an inlet, water at 20°C enters with a uniform velocity distribution while the wall temperature is constantly 60°C. At the inlet, the Reynolds number (Re) varies from 100 to 27750.

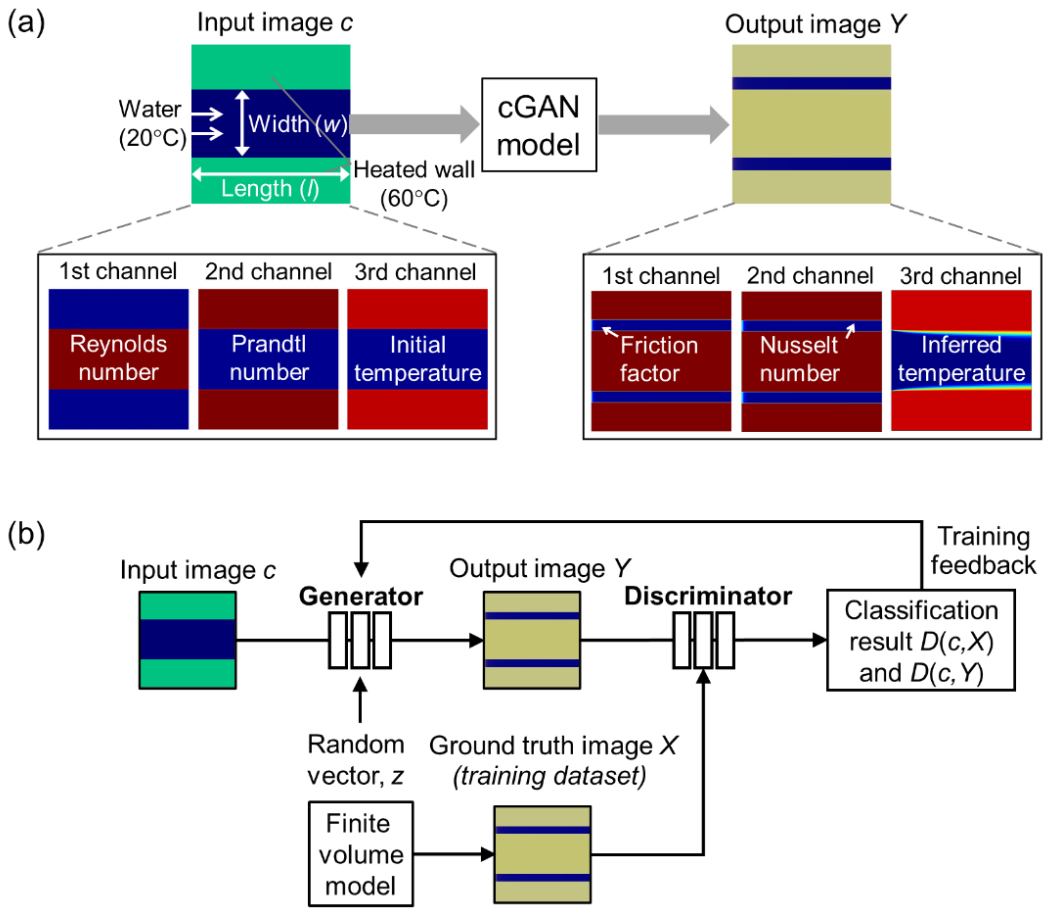

Figure 2. 1. (a) Structures of Input and Output Images. (b) Conditional Generative Adversarial Networks Architecture

2.3. Deep Learning Methodology

2.3.1. Conditional Generative Adversarial Networks

A modern deep learning model is trained, conditional generative adversarial networks (cGAN), to generate the property fields of interest from graphical inputs. Due to the ability to generate data that own similar characteristics to the input data, the cGAN has been one of the popular deep learning methods. Figure 2.1a illustrates the structures of input and output images that are employed for our cGAN convection model. Conditional inputs are two-dimensional (2D), 256 × 256-resolution, three-channel images representing the boundary and initial conditions of the convection problem. The pixel values in the first channel represent Re. In wall region (colored in blue), Re is an arbitrary constant, i.e., 0, while Re is calculated by the flow properties in fluid region (colored in dark red). The pixel values in the second channel correspond to Prandtl number (Pr) distribution. In wall region (colored in dark red), Pr is an arbitrary constant 50000 while in fluid region (colored in blue) Pr is that of liquid water, 7. The pixel values in the third channel are the temperature distribution at an initial moment. The trained cGAN model approximates an output image by statistically learning a possible output data group for input images. The outputs are 2D, 256 × 256-resolution, three-channel images. The pixel values along the fluid domain boundary (colored in blue) are friction factor (f) in the first channel and are Nusselt number (Nu) in the second channel. The pixel values of the third channel represent the temperature distribution at steady state. Figure 2.1b shows a cGAN architecture. The cGAN algorithm uses two neural networks: a generator neural network (G) and a discriminator neural network (D). The generator creates an output image (Y) when a random noise vector (z) and a conditional input (c) are provided.

The discriminator learns to distinguish the ground truth images from the generator outputs. The ground truth images contain numerically calculated Nu, *f*, and the temperature. The discriminator receives batches of both ground truth images (*X*) and generated images (*Y*), and classifies the images into real and fake classes. During the training, the discriminator learns to maximize the probability that it correctly classifies the images, while the generator tries to minimize the probability by generating realistic samples. Thus, the objective of a cGAN is formulated as

$$L_{cGAN}(G, D) = E_{c,X}[\log D(c, X)] + E_{c,Y}[\log\{1 - D(c, Y)\}] \tag{2.1}$$

where $D(c, X \text{ or } Y)$ is the probability that the discriminator classifies X or Y as ground truth for a given c. $E_{c,X \text{ or } Y}$ is the expected value over the entire group of X or Y. The generator attempts to minimize L_{cGAN} while an adversarial discriminator tries to maximize L_{cGAN}. The cGAN algorithm also considers a traditional loss function such as $L1$ distance that estimates the errors in Y against X.

$$L_{L1}(G) = E_{c,X,Y}[\|Y - X\|] \tag{2.2}$$

Thus, the final objective becomes

$$G^* = \arg\ min_G max_D L_{cGAN}(G, D) + \lambda L_{L1}(G) \tag{2.3}$$

where the hyperparameter λ is a weight for $L_{L1}(G)$.

To train the neural networks, a finite volume model (FVM) was used to prepare a dataset. A commercial software, ANSYS 19.0 was used to developed the FVM that simulated the steady-state flow and heat transfer in the internal forced convection problem. A dataset consisting of *N* pairs of conditional inputs and outputs were prepared by changing Re of flow. As Re was linearly varied from 100 to 27750, the data for transition and

turbulent flow regimes were five times more than the data for laminar flow regime. Then, the dataset was split into training and testing sets with a ratio of 9:1.

2.4. Optimization

To optimize the networks, the effects of dataset size N, training epoch and the hyperparameter λ were studied. Generally, deep neural networks require N to be greater than 1000. However, if generating such a large dataset is not practical, it is necessary to examine an appropriate N that ensures a sufficient accuracy level. Figure 2.2 compares the temperature maps inferred by the cGAN and a ground truth image calculated by the FVM for Re = 300. During training, the input and output pair for Re = 300 was not provided to cGAN, indicating that Fig. 2.2 shows the test result with an unseen input. When N = 60, the cGAN did not correctly approximate the wall temperature, exhibiting dark spots near the channel exit and thin thermal boundary layer. However, as N increased to 180, the cGAN accurately produced the thermal boundary layers as well as temperature in other regions. To quantitatively examine the effect of N, the root mean square error (RMSE) was evaluated, i.e., the standard deviation of the prediction error of the cGAN model against the ground truth, and maximum absolute error (MAX), i.e., the maximum absolute temperature difference between the ground truth and cGAN output. The RMSE and MAX were calculated in the thermal boundary layer region, i.e., fluid domain spanning 33 vertical pixels both from the top and bottom walls, since the difference between the ground truth and cGAN output was the most profound in this region due to a large temperature gradient. The RMSE and MAX were 3.71°C and 22.3°C, respectively, with N = 60, but RMSE and MAX drastically reduced to 0.31°C and 2.74°C with N = 180.

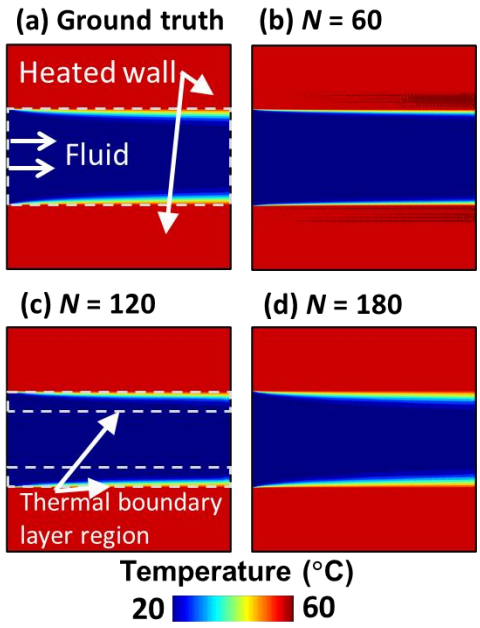

Figure 2. 2. Temperatures Maps Predicted by (a) FVM and (b-d) cGAN. The Dataset Size *N* Was Selected as (b) 60, (c) 120, and (d) 180.

Figures 2.3a and 2.3b show the local Nu generated by the cGAN along with the ground truth for Re = 300 and 23500. During training, the input and output pairs for Re = 300 and 23500 were not provided to cGAN, indicating that Fig. 2.3 presents the test result with unseen inputs. When N = 60 and Re = 300, the cGAN predicted Nu distribution with an accuracy of 32.4%. Here, the accuracy is defined as 100% – percentage error that is averaged over all channel locations, i.e., accuracy = $\sum(1-|\mathrm{Nu_X}-\mathrm{Nu_Y}|/\mathrm{Nu_X})/n$ where $\mathrm{Nu_X}$ is true Nu obtained from the FVM, $\mathrm{Nu_Y}$ is Nu inferred by the cGAN model and n is the total number of nodes along the channel. As N increased to 180, the accuracy improved to 95.9% for Re = 300. In turbulent regime (Re > 10000), the cGAN exhibited higher accuracies probably because there were five times more training data as compared to the laminar flow regime. When N = 60 and Re = 23500, the accuracy was 97.2%. When N =

180 and Re = 23500, the accuracy was 97.6%. Figures 2.3c and 2.3d present *f* predicted by the cGAN when Re = 300 and 23500. Similar to Nu prediction, the cGAN approximated more accurately for the turbulent regime than the laminar regime.

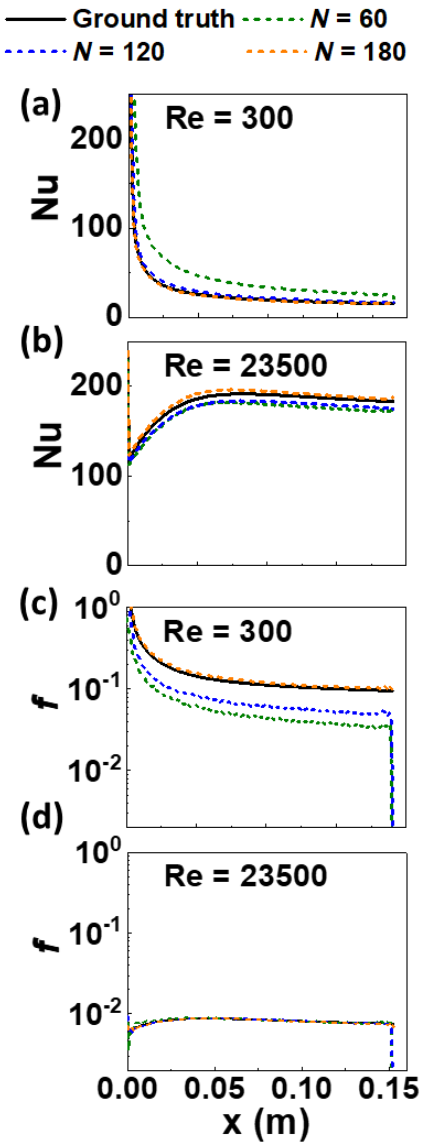

Figure 2. 3. Local Nusselt Number and Friction Factor Predicted by cGAN when the Dataset Size *N* Was Varied from 60 to 180: (a) Nu at Re = 300, (b) Nu at Re = 23500, (c) *f* at Re = 300 and (d) *f* at Re = 23500.

The epoch of neural networks training is raised until the loss functions and errors become sufficiently small. With an epoch of 500, MAX was 3.8 – 13.9°C and training duration was 1.5 hours. However, with an epoch of 2000, MAX reduced to 3.3 – 6.3°C and the training duration increased to 9.6 hours. To balance the error and training duration, the epoch as 1000 was selected in the subsequent cGAN trainings, resulting in a training duration of 3.3 hours.

The hyperparameter λ balances the mismatch in the orders of magnitudes of L_{cGAN} and L_{L1}. Figures 2.4a and 2.4b show L_{cGAN} and L_{L1} as a function of epoch for $\lambda = 10^5$. In our trainings, L_{cGAN} was on the order of 10 while L_{L1} was on the order of 10^{-2}. Thus, when λ was varied from 10^4 to 5×10^5, the total generator loss G^* was adjusted to the order of 10^2 – 10^3 (Fig. 2.4c).

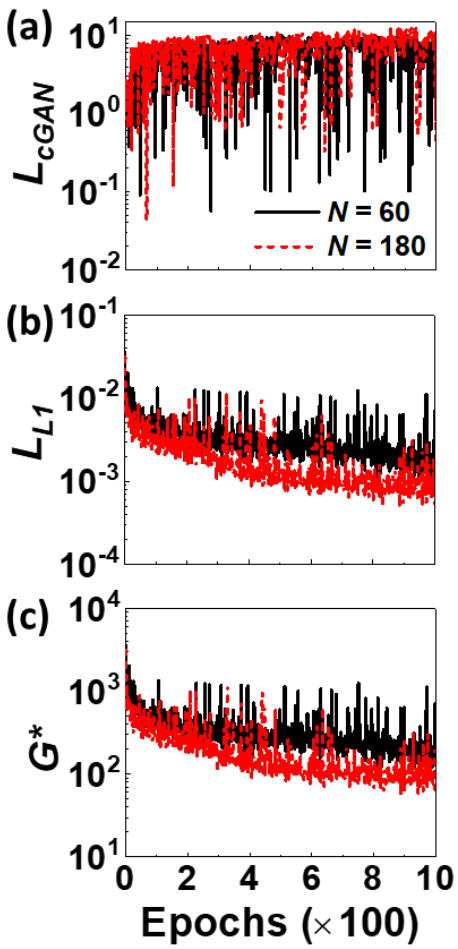

Figure 2. 4. Losses as a Function of Epoch: (a) Generator Loss, (b) *L1* Distance Loss, and (c) Total Generator Loss.

To understand the influence of dataset size *N* and hyperparameter *λ,* MAX and accuracies of Nu and *f* were evaluated as a function of both parameters. In Tables 2.1 and 2.2, the cGAN model accuracies are shown for laminar flow data (Re = 300, Table 1) and turbulent flow data (Re = 23500, Table 2.2). Overall, the cGAN model exhibited greater accuracy for the turbulent flow than the laminar flow. In general, a large *N* improves the model accuracy. However, a simple relation was not observed between *λ, N* and accuracies.

It seems that λ should be tuned for a specific N. Considering MAX and accuracies of Nu and f at the same time, $N = 180$ and $\lambda = 10^5$ were selected to minimize MAX and maximize the accuracies in both flow regions.

Table 2. 1. The Maximum Absolute Error (MAX, °C) of Temperature Maps and Accuracies of Nu and f at Re = 300 as a Function of N and λ.

		λ			
N		10^4	5×10^4	10^5	5×10^5
60	MAX	24.2	22.55	22.34	17.8
	Nu	0.67	0.4	0.32	0.93
	f	0.15	0.6	0.4	0.55
120	MAX	12.14	11.04	10.73	7.67
	Nu	0.87	0.71	0.9	0.74
	f	0.81	0.6	0.6	0.74
180	MAX	13.71	7.1	2.74	10.99
	Nu	0.92	0.84	0.96	0.79
	f	0.66	0.62	0.93	0.63

Table 2. 2. The Maximum Absolute Error (MAX, °C) of Temperature Maps and Accuracies of Nu and f at Re = 23500 as a Function of N and λ.

		λ			
N		10^4	5×10^4	10^5	5×10^5
60	MAX	3.24	2.08	2.45	1.5
	Nu	0.99	0.99	0.97	0.99
	f	0.95	0.95	0.94	0.96
120	MAX	1.58	1.75	0.81	1.44
	Nu	0.99	0.99	0.96	0.96
	f	0.88	0.98	0.97	0.84
180	MAX	16.65	1.14	1.4	1.27
	Nu	0.99	0.99	0.98	0.93
	f	0.95	0.97	0.99	0.91

2.4.1. Test and Cross-validation

The cGAN model trained with optimally selected parameters is tested and validated with the dataset unseen during training. Figures 2.5a and 2.5b compare the temperature maps for a developing laminar flow at Re = 300 that are obtained by the FVM (denoted as ground truth) and the cGAN. Despite a large temperature variation across the thermal boundary layer region, the RMSE and MAX of the cGAN prediction are merely 0.36°C and 2.74°C, respectively. Figures 2.5c and 2.5d depict a ground truth image and cGAN prediction for a transition flow at Re = 10875. Due to the flow mixing in the transition flow, the temperature is much more uniform than in the laminar flow with significantly reduced RMSE (0.17°C) and MAX (0.79°C). Figures 2.5e and 2.5f illustrate the temperature maps for a turbulent flow at Re = 23500. For the turbulent flow, RMSE is 0.19°C and MAX is 1.4°C.

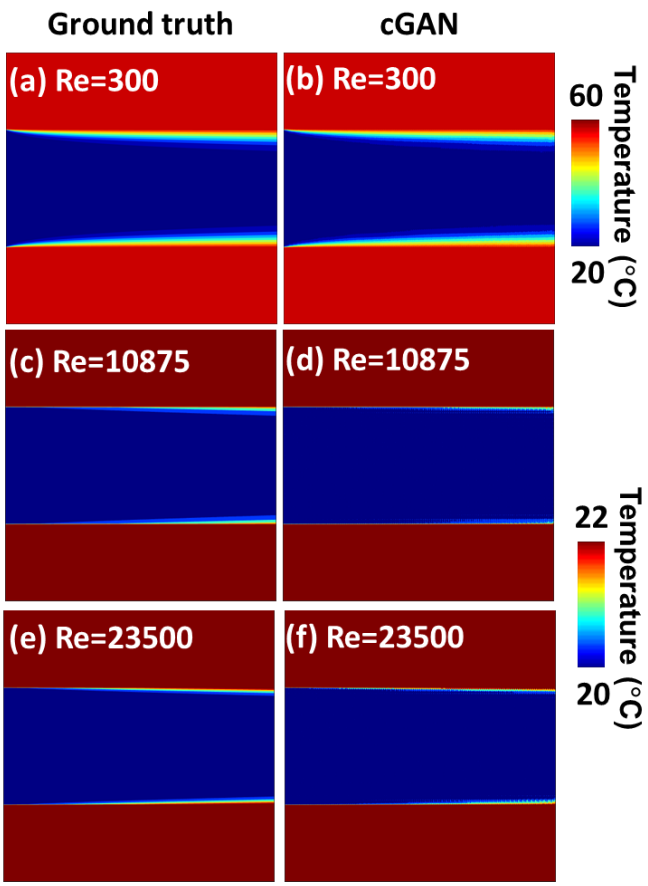

Figure 2. 5. Comparison of the Temperature Maps Obtained by cGAN and FVM (Ground truth): (a, b) Re = 300 (MAX = 2.74°C, RMSE = 0.36°C), (c, d) Re = 10875 (MAX = 0.79°C, RMSE = 0.17°C), and (e, f) Re = 23500 (MAX = 1.4°C, RMSE = 0.19°C).

The optimally trained cGAN model accurately predicts the local distribution of convection properties. Figure 2.6a shows the predicted Nu at three Re. Note that the cGAN model directly infers the distribution of Nu from provided input without requiring additional calculations with the predicted temperature distribution. Although the Nu distribution dramatically and nonlinearly changes with Re and the location in channel, the predicted Nu is highly accurate. Figure 2.6b shows the predicted *f*. Across the flow regimes

and along the channel, *f* changes more than an order of magnitude, but the cGAN was able to infer such trends.

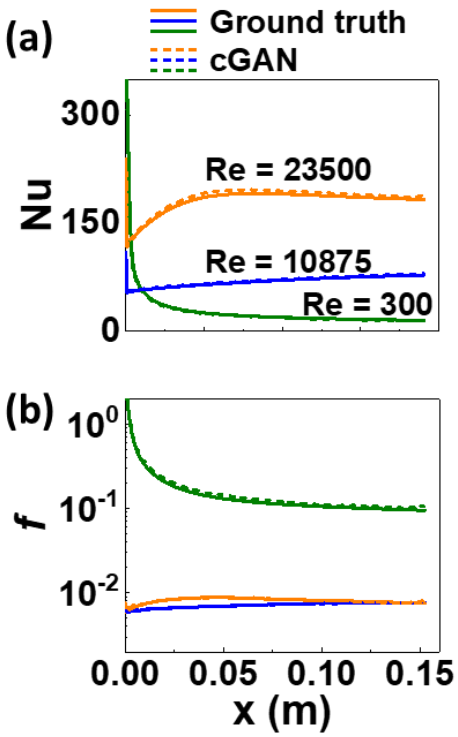

Figure 2. 6. (a) Local Nusselt Number and (b) Friction Factor Predicted by an Optimized cGAN at Unseen Re = 300, 10875, and 23500.

To further validate the accuracy of the cGAN model, a 10-fold cross-validation has been performed. The total dataset including 180 images was divided into ten subsets with each subset containing the same numbers of laminar-flow, transition-flow, and turbulent-flow samples. For each round of validation, one of the subsets was retained as testing data, and the other nine subsets were used as training data (Bowles et al. 2018; Kiyasseh et al. 2020; Cirillo, Abramian, and Eklund 2020; Maleki et al. 2020; Ghassemi, Shoeibi, and Rouhani 2020). Figure 2.7a shows the variation of MAX for different test datasets. The

box-and-whisker plot provides the median (50%), lower, upper, first-quartile (25%), and third quartile (75%) values of MAX. The red symbol indicates the mean value of MAX for each test dataset. Although the maximum values of MAX vary from 0.52°C to 10.12°C, the average of mean MAX is merely 1.6°C. Figures 2.7b and 2.7c present the variation of accuracies for Nu and *f*. The average of mean accuracy is 0.975 for Nu and 0.958 for *f*. The cross-validation shows that the accuracy of the cGAN model may vary with different training data, but the accuracy would be sufficient as long as an adequate size of training data is provided.

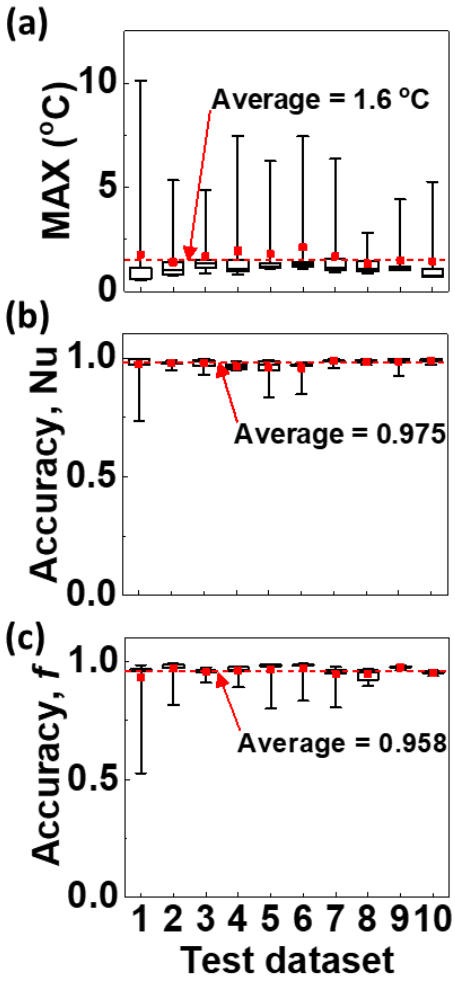

Figure 2. 7. Cross-validations for (a) MAX, (b) Accuracy of Inferred Nu, and (c) Accuracy of Inferred *f* with Different Test Datasets.

2.5. Model Testing with Unseen Geometries

The trained cGAN is able to infer the solution for unseen input geometries at a certain extent. To understand the capability of the cGAN, the cGAN with several modified input geometries including narrowed and widened channels were tested, and a 90-degree rotated channel (Fig. 2.8). The narrowed channel width is 70% of the original width and the widened channel width is 130% of the original width.

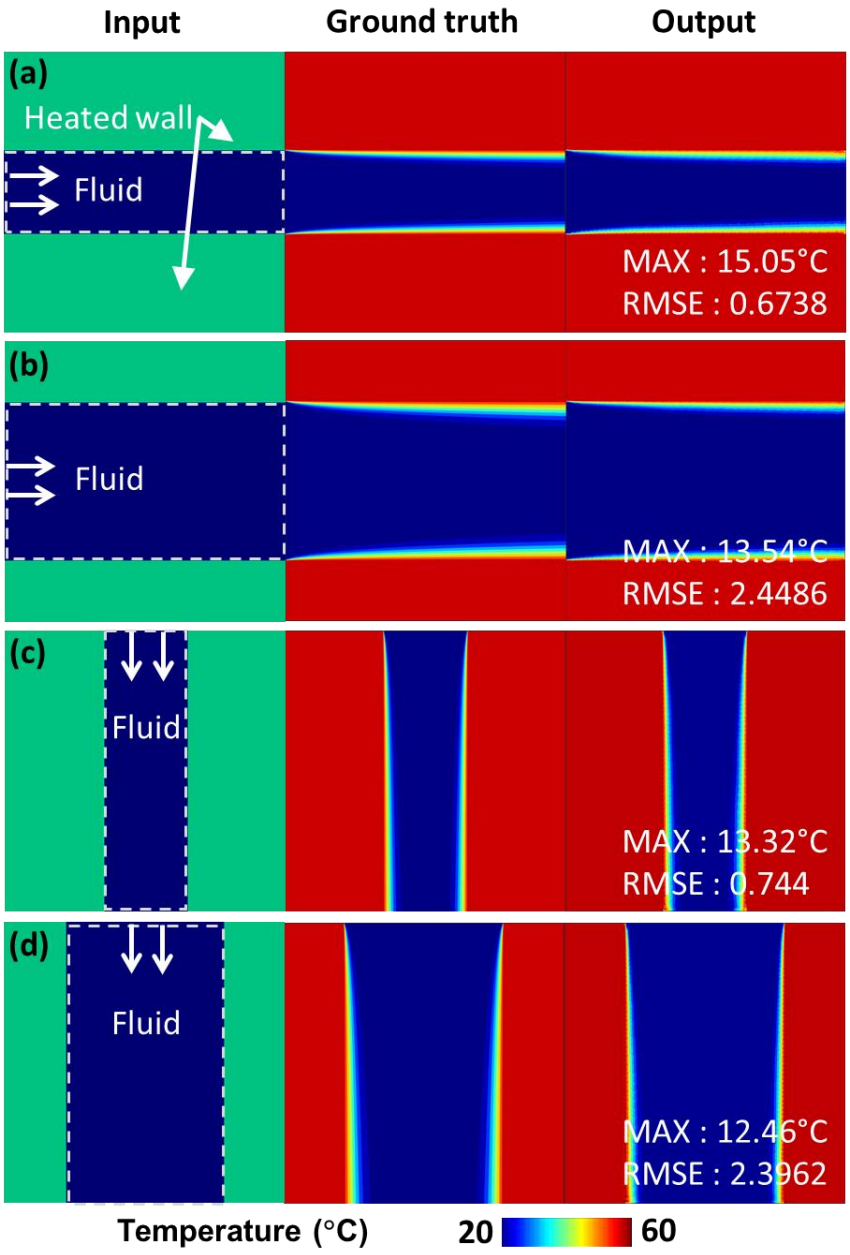

Figure 2. 8. Inputs, Ground Truths and Outputs for Unseen Channel Geometries with Re = 300: (a) Non-rotated Narrowed Channel, (b) Non-rotated Widened Channel, (c) Rotated Narrowed Channel, and (d) Rotated Widened Channel.

To facilitate the prediction of unseen geometries, the training data for the cGAN model via a simple image transformation technique was amplified. Data augmentation for GAN has been widely hired in previous research (Cheng et al. 2015; Frid-Adar et al. 2018; Ghassemi, Shoeibi, and Rouhani 2020) as an auxiliary method to enrich the training datasets in classification works. Figure 2.9 illustrates how an original input image was transformed into new data via three possible ways. At each training epoch, randomly selected original input and output images may be simply rotated 90° with probability. An image may be cropped and combined with its mirror image segment. The mirrored image may be also rotated 90°. If all the images in the training dataset are transformed and added to the dataset, 162 new samples, with either or both different channel widths and 90°-rotated channels, can be created at every training epoch. Thus, the total number of training samples may increase up to 162000 at an epoch of 1000. Through the data augmentation, amplifying the number of training samples without additional data preparations with the FVM was possible. Note that the computation time for image transformation is less than 1 second which is significantly shorter than the FVM runtime. However, randomly mirrored images may contain physically incorrect information depending on how the image is cropped and how an actual flow field is. Thus, to improve the cGAN accuracy, the data augmentation method must be refined.

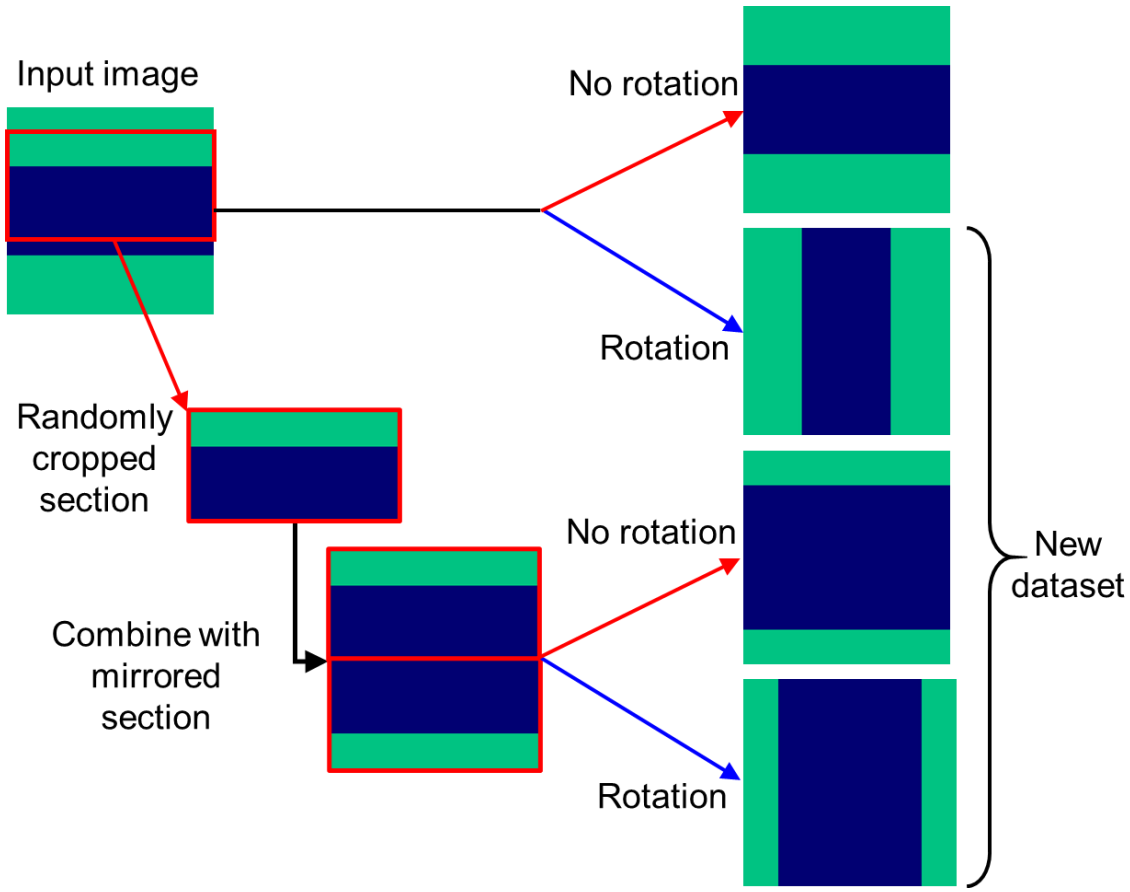

Figure 2. 9. Data Augmentation Process.

Figure 2.8 shows the temperature maps inferred by the cGAN for unseen channel geometries, i.e. narrowed, widened, and 90-degree rotated channel geometries at Re = 300. Overall, the cGAN predictions are close to the ground truth while exhibiting RMSE ≤ 3°C and MAX ≤ 15°C for the narrowed and widened channels, and RMSE ≤ 3°C and MAX ≤ 13°C for the rotated channels. When training the cGAN with an original dataset, MAX was as large as 44.5°C. However, after employing the dataset augmented by the simple image transformations, the cGAN provided 66% reduced MAX.

Figure 2.10 compares Nu and *f* of narrowed channels produced by the cGAN with the ground truth data. The accuracies for Nu and *f* predictions are 63.6% and 65%, respectively. If the narrowed channel is rotated, the accuracy for Nu reduces to 47.8%, but the accuracy for *f* maintains at a similar level (68.52%). Figure 2. 11 shows predicted Nu and *f* of widened channels. The accuracies for Nu and *f* approximations are 87.3% and 54%, respectively. However, if the widened channel is rotated, the accuracy for Nu reduces to 75%, but the accuracy for *f* increases to 63.79%.

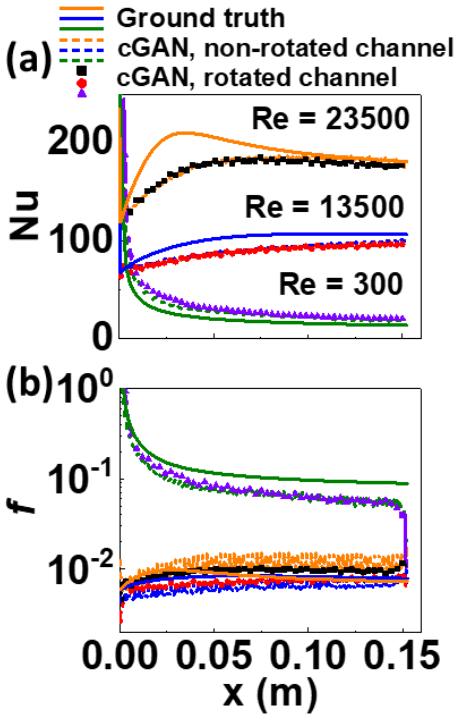

Figure 2. 10. (a) Local Nusselt Number and (b) Friction Factor Predicted for Unseen Narrowed Channels.

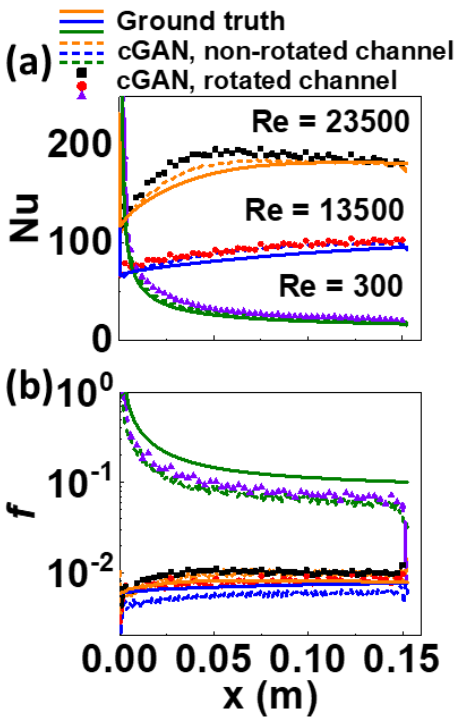

Figure 2. 11. (a) Local Nusselt Number and (b) Friction Factor Predicted for Unseen Widened Channels.

Even with the simple data augmentation, the cGAN was able to approximate the convection properties for unseen geometrical inputs with an accuracy > 50%. Especially, the augmentation by image rotations enabled the cGAN to predict even for the channels with arbitrary angular orientations. Although the accuracy for new geometries is greatly smaller than the accuracy for trained geometry, the cGAN predictions seem still useful for a rough and rapid estimation without solving a numerical model.

2.6. Conclusions

A deep learning model for forced convection heat transfer problems based on conditional generative adversarial networks (cGAN) was developed. The cGAN was

trained by a set of graphical inputs containing the geometric and flow conditions and graphical outputs representing the convection properties. A single, trained cGAN model successfully predicted the distributions of temperature, Nu and f of a heated internal channel flow over a wide range of Re (Re = 100 to 27750). To achieve a high accuracy, the dataset size, training epoch and the hyperparameter λ of the cGAN were optimized. The optimized cGAN model exhibited an accuracy $\leq$ 97.6% for the Nu estimation and root mean square error (RMSE) < 0.3°C maximum absolute error (MAX) $\leq$ 2.7°C for the temperature approximation. The inference ability of the cGAN model was further validated through a 10-fold cross-validation test. When combined with a data augmentation technique, the capability of cGAN model for unseen channel geometries also demonstrated. After trained with an amplified dataset, the cGAN was able to predict unseen channel geometries such as widened, narrowed, and rotated channels. For the new channel geometries, the cGAN inferred Nu and f with an accuracy $\leq$ 87.3% and temperature distribution with RMSE $\leq$ 3°C and MAX $\leq$ 13°C.

The presented cGAN convection model will enable to rapidly approximate the spatial distributions of convection properties, e.g., temperature, Nu, f, in a 2D domain, if the input information is provided in a 2D image format. Although this method was demonstrated for a simple 2D steady-state convection problem, this approach can be readily extended to a variety of problems involving complex surfaces like rough surfaces and extended surfaces. Moreover, if rapid and repetitive estimations of convection properties in geometrically complex systems are needed over a wide range of flow conditions, the cGAN convection model can serve as a good alternative to the traditional numerical simulation techniques.

3. DEEP GENERATIVE MODELING OF 3D UNSTEADY STATE HEAT CONVECTION PROCESS

3.1. Introduction

Recently, there is a growing interest to explore the deep learning approaches for modeling transport processes, which has been mostly carried out using computational fluid dynamics (CFD). Deep learning algorithms can readily identify the features embedded in unstructured data such as images, texts, and signals, which eliminates the feature extraction processes in data-driven modeling. CFD techniques are widely used and effective, but extensive computational resources and computation time are required to solve for the large-scale, spatiotemporal dynamic problems. If simulation and/or experimental data for transport processes are available, the deep learning models are expected to be trained to solve for the complex transport problems without necessitating significant computational costs, and serve as surrogate models.

Various deep learning techniques have been used to model heat transfer processes. Conditional generative adversarial networks (cGAN) were first used to infer the steady-state temperature fields in a two-dimensional (2D) domain when conduction was the only heat transfer mechanism (Farimani, Gomes, and Pande 2017). In this work, the training dataset was prepared using a finite difference method. The cGAN model was able to generate resulting temperature fields when the model inputs, which were 2D images representing the initial temperature distributions, were given. The cGAN was also trained to simultaneously approximate the temperature, velocity, and pressure fields of a natural convection process occurring in a 2D square cavity when the model input was provided as five-channel 2D images containing the initial and boundary conditions (Jiang and Farimani

2020). Recently, the cGAN technique was used to infer the 2D distributions of temperature, Nusselt number, and friction factor of a channel in a heated channel over Reynolds number ranging from 100 to 27750 (M. Kang and Kwon 2022). When combined with a data augmentation technique, the cGAN model was able to predict for unseen fluid channel geometries such as narrowed, widened, and rotated channels, which demonstrated the nonlinear interpolation and extrapolation capabilities of the cGAN technique. The previous demonstrations used the cGAN to generate images containing physical properties that are resulted from certain input conditions. Traditionally, convolutional neural network (CNN) has been a common deep learning technique for image processing and analysis, but CNN requires a careful design of a loss function to accomplish a good performance (Isola et al. 2016). To facilitate the deep learning based image generation tasks and to increase the accuracy, GAN was proposed to automatically adapt the loss function to the data (Goodfellow et al. 2014), and has gained popularity.

Apart from the cGAN technique, several other deep learning techniques have been employed for heat transfer modeling. CNN (Sharma et al. 2018) and autoencoder (AE) (Edalatifar et al. 2020) were trained to predict the steady-state temperature distributions in a 2D domain when conduction was the only heat transfer mechanism. Similar to GAN, both models successfully reconstructed the input images containing boundary conditions to output images. In these works, the CNN model employed a 2D heat equation as a loss function (Sharma et al. 2018) and the autoencoder model used mean squared errors or mean of maximum squared errors as its loss function (Edalatifar et al. 2020). In another demonstration, CNN was trained to predict the 2D distribution of local heat flux in a fully developed turbulent channel flow when the model inputs were the 2D images of wall-shear

stresses and pressure fluctuations (Kim and Lee 2020). In this work, the loss function was defined as the sum of mean squared error and regularization loss. Recently, physics-informed neural networks (PINN) for one-dimensional (1D) and 2D heat conduction problems were reported (Zobeiry and Humfeld 2021). By defining the loss function as the errors in conduction heat transfer equation, boundary and initial conditions, deep neural networks (DNN) were able to predict the temperature distributions in a 2D domain without the need of the pre-generated training dataset. DNN are also commonly used for regressing discrete numeric outputs from discrete numeric inputs. In a previous work (Sundar et al. 2020), two-layer artificial neural networks learned to predict the fouling resistances of a cross-flow heat exchanger when six operating condition parameters were provided.

Despite the recent progress in deep learning techniques for modeling the heat transfer processes, it is unclear if the deep learning techniques will still be effective for inferring complex phenomena such as transient, multi-mode conjugate, or various turbulent-flow heat transfer processes. If the deep learning surrogate models are able to solve for the complex heat transfer processes faster than numerical simulations, those models will be useful when designing and optimizing the 2D or three-dimensional (3D) transport systems. Therefore, to test the potential of deep learning, this chapter investigates the deep conditional generative adversarial networks that can predict for the 3D transient laminar mixed convection phenomena. In mixed convection, free and forced convection effects are comparable to each other, leading to buoyancy-driven secondary flows, as well as unstable and inhomogeneous property fields. The flow characteristics of mixed convection are known to enhance heat transfer up to 4 to 5 folds as compared to pure forced convection (Osborne and Incropera 1985; Ostrach and Kamotani 1975; Yasuo and Yutaka

1966; Bergles and Simonds 1971; McComas and Eckert 1966), thereby drawing substantial attention for use in engineering and scientific applications such as heat exchangers (Oztop and Al-Salem 2012; Andrzejczyk and Muszynski 2017; Ghorbani et al. 2010), radiation energy collectors (Selmi, Al-Khawaja, and Marafia 2008; Edalatpour and Solano 2017), plants (Banna, Pietri, and Zeghmati 2004), groundwater and geothermal systems (Smith 2004; Kumari 2001).

3.2. Mixed Convection Problem and Data Generation

A mixed convection flow in a 3D channel with a width (W) of 60mm, height (H) of 15mm and length (L) of 130mm is considered as shown in Fig. 3.1. At an inlet, water at 35°C (T_i) enters the channel with a uniform velocity distribution and the Reynolds number (Re) of 100. The bottom wall temperature (T_b) is maintained at 43°C, and the other walls are thermally insulated. Under the prescribed boundary conditions, the Rayleigh number (Ra) is 3.9×10^6 and the Richardson number (Ri) is 88.8.

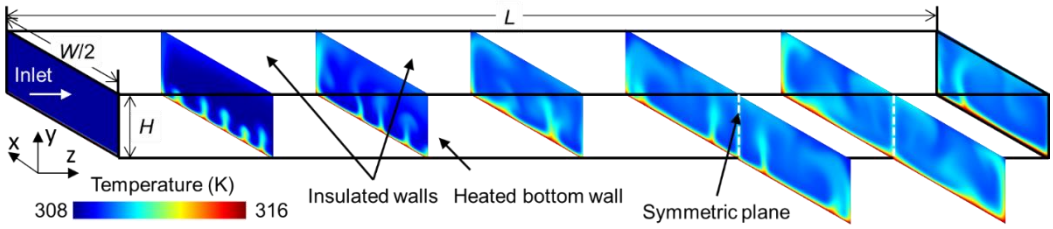

Figure 3. 1. Temperature Distribution at Selected Cross-sections in a Rectangular Channel Induced by Mixed Convection Heat Transfer.

The transient flow and temperature fields are simulated by a finite volume model (FVM) implemented in commercial CFD software (ANSYS Fluent). To reduce the

computational cost, only the half-section of the channel is modeled by employing the *yz*-plane at $x = 0$ as a symmetric plane. In this work, the residual for convergence is 10^{-4} for the continuity equation, 10^{-5} for the momentum and energy equations. The simulation model employs a time step size of 0.05s and the number of time steps of 7200. The model determines the density (ρ) and dynamic viscosity (μ) of water as a function of the temperature without employing the Boussinesq approximation. Other water properties such as specific heat (C_p), Prandtl number (Pr), and thermal conductivity (k) are assumed as constants at the mean temperature of the water inlet and bottom wall.

The solutions of the FVM are the temperature distributions in the simulation domain at discrete time steps, which are employed to create training and testing datasets for the cGAN model. The datasets consist of 4320, 2D, 64 × 128-resolution temperature images obtained across *xy*-planes at 360 time steps. Two types of datasets are tested with different bit depths: one with 8-bit RGB images (256 colors with 3 channels) and the other with 16-bit grayscale images (thousands of colors with a single channel). In the 8-bit RGB images, the jet colormap is used for mapping the temperature distribution between 35°C and 43°C. In the grayscale images, the intensity corresponds to the absolute temperature. For each dataset, randomly selected 90% images are used for training and 10% images are employed for testing.

3.3. Deep Learning Methodology

The cGAN (Mirza and Osindero 2014) was trained to rapidly predict the flow temperature distribution. Among various deep learning algorithms for image processing such as CNN (Dai, Lu, and Wu 2014) and variational autoencoder (VAE) (Kingma and

Welling 2013), cGAN is selected due to its superior capability in image generation. CNN is the class of artificial neural networks that has been popular for classifying, recognizing, analyzing, or segmenting images. However, it is not easy to generate high-quality target images using CNN without designing sophisticated loss functions. VAE is an artificial neural network that maps the characteristics of input images into a latent space and reconstructs images from vectors in the latent space. Despite the stability in training, the outputs of VAE are not as close to the ground truth images as the outputs of the cGAN.

The generator network (GN) of cGAN learns to create an image close to the ground truth image for fooling the discriminator network (DN). Figure 3.2(a) presents the GN architecture. The input to GN is a 1 × 8 conditional vector that describes the channel geometry, thermal and hydraulic boundary conditions, fluid properties, time (t), and the z location in channel, i.e., input (c) = [channel aspect ratio (AR), Re, Pr, Ri, T_i, T_b, t, z]. The fully-connected layer (G1) feeds and reshapes the input vector into an input layer with a shape of 1 × 1 × 512 tensor. Then, the input layer goes through seven blocks (i.e., from G2 to G8) where each block consists of three layers including transpose convolutional, batch normalization, and ReLU (Rectified Linear Unit) (Krizhevsky, Sutskever, and Hinton 2017) activation layers. The transpose convolutional layer performs an inverse convolution operation, and generates an output feature map that has greater dimensions than the input. The batch normalization (Ioffe and Szegedy 2015) layer transforms the means and variances of the layer inputs to make the learning process stable and fast. In this work, a batch normalization layer is not included in G1, since GN performs well by doing this. Among various available activation functions such as sigmoid, ReLU, and leaky ReLU, GN employs ReLU due to computation efficiency. Finally, the output layer (G9) uses the

Tanh activation function, and generates a flow temperature image with a desired size of 64 × 128.

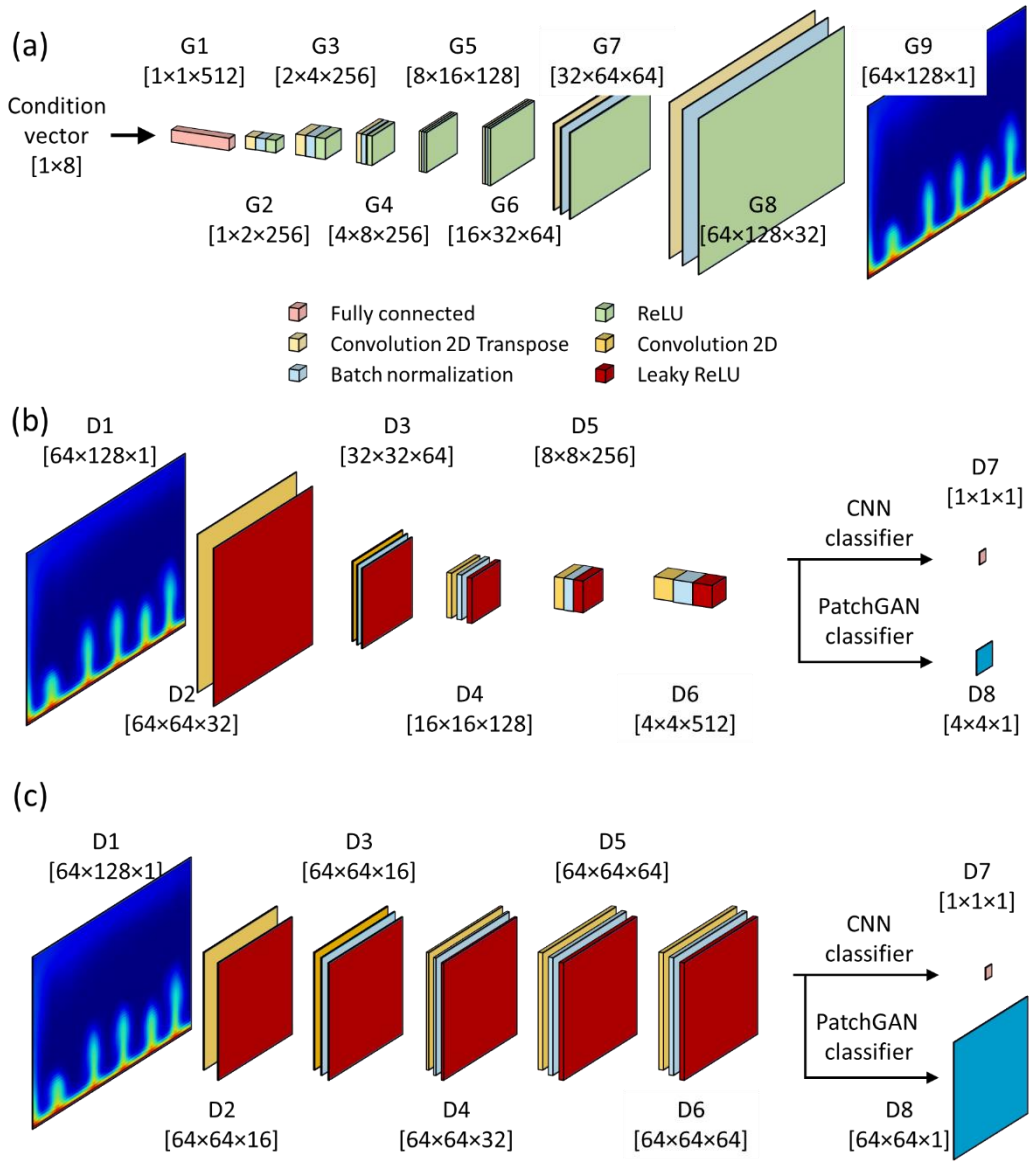

Figure 3. 2. cGAN Architectures. (a) Generator Architecture. (b) Discriminator Architectures with Strided Convolutions. (c) Discriminator Architectures without Strided Convolutions.

The discriminator network (DN) learns to distinguish ground truth images from the fake images that GN creates. There are several common types of DN developed for GANs. In the original GAN (Goodfellow et al. 2014), the DN is defined by multilayer perceptrons, which provides the classifier with a scalar probability that the input image is from the ground truth group rather than the GN. In recent GANs, the multilayer perceptrons were replaced by CNN to improve stability in training. In the deep convolutional GAN (DCGAN) (Radford, Metz, and Chintala 2016), the DN uses a modified CNN that replaces the pooling layers with strided convolutions, and uses the batch normalization. In PatchGAN (Isola et al. 2016), the classifier receives small patches of images, and classifies the images, which requires a smaller number of parameters in DN than the traditional GAN classifier.

In this work, four types of discriminator architectures are tested to identify a DN suitable for the data obtained from complex convection phenomena. Figure 3.2(b) presents the first and second architectures that reduce the input image into an output tensor. The first DN architecture, denoted as a CNN classifier, generates a single probability value by employing a flatten and fully-connected layer in the last block (D7). The second DN architecture, denoted as a PatchGAN classifier, generates a 4×4 image (D8) with each pixel representing the guess for a 64×128 patch of the input image. Figure 3.2(c) illustrates the third and fourth DN architectures that do not employ strided convolutions in order to maintain the feature map size after D1. The third architecture generates a single probability (D7), and the fourth architecture generates a 64×64 image (D8) as an output. By combining the GN and prescribed DNs, four different types of cGAN models are created and compare their performances to understand the effect of DN architecture. The cGAN models are denoted as Models 1 to 4 where Model 1 uses the CNN classifier with strided

convolutions; Model 2 uses the CNN classifier without strided convolutions, Model 3 uses the PatchGAN classifier with strided convolutions; and Model 4 uses the PatchGAN classifier without strided convolutions. In addition, Model 5 is prepared to understand the effect of *L1* loss. All the models employ the ADAM (adaptive moment estimation) optimizer (Kingma and Ba 2014) with a learning rate of 0.0002 and β_1 of 0.5.

The objective of a cGAN combines the objectives of the GN and DN. The GN produces a flow temperature map (Y) from a provided input vector (c) that looks similar to the CFD simulation result (X), and tries to deceive the DN. The DN tries to distinguish X from Y. Thus, one of the cGAN objectives is formulated as

$$L_{cGAN}(GN,DN) = E_{c,X}[\log DN(c,X)] + E_{c,Y}[\log\{1-DN(c,Y)\}] \tag{3.1}$$

where $DN(c, X)$ represents the probability that the DN receives c and X, then classifies the input image as X. $DN(c, Y)$ represents the probability that the DN receives c and Y, and classifies the input image as X. The DN tries to maximize the expectation values of $DN(c, X)$ and $1 - DN(c, Y)$. However, the GN tries to minimize the expectation values in Eq. (3.1).

To improve the ability of GN, *L1* loss is often added as a GN objective. The $L1$ loss measures the mean absolute pixel difference between X and Y. Thus, if the GN is trained to minimize the *L1* loss, the produced images become clearer than the results without the *L1* objective. The *L1* loss is formulated as below.

$$L_{L1}(GN) = E_{c,X,Y}[\|Y - X\|] \tag{3.2}$$

Thus, the final objective of a cGAN is described as

$$L_{Total} = min_{GN} max_{DN} L_{cGAN}(GN,DN) + \lambda L_{L1}(GN) \tag{3.3}$$

where the hyperparameter λ is a weight for L_{L1}(GN). Note that for Model 5, the objective is L_{Total} is equal to L_{cGAN} only.

The hyperparameters are optimized such as training epoch and λ for balancing the training quality and computation duration. The training epoch needs to be sufficiently large to reduce the training losses to the desirable levels. The training results obtained with epochs of 500, 1000, 2000, and 3000 were compared. For Model 4, with the increase of epoch, the training duration increased from 1.5 hours to 8.8 hours, and the mean temperature difference (ΔT) between the ground truths and generated images decreased from 0.22K to 0.093K. Figure 3.3 shows the training losses of Models 2 and 4 as a function of epoch. For both models, L_{DN} gradually decreases and reaches 10^{-3} over the epoch. However, L_{cGAN} is slightly smaller in Model 4 than Model 2, indicating that the DN architectures with or without strided convolutions train the GN to different levels. For both models, L_{L1} and L_{Total} continue to decrease with the increase of epoch. Considering the training duration and ΔT, All the subsequent model training was conducted with an epoch of 3000. The details of hyperparameter optimization are in the supplementary material.

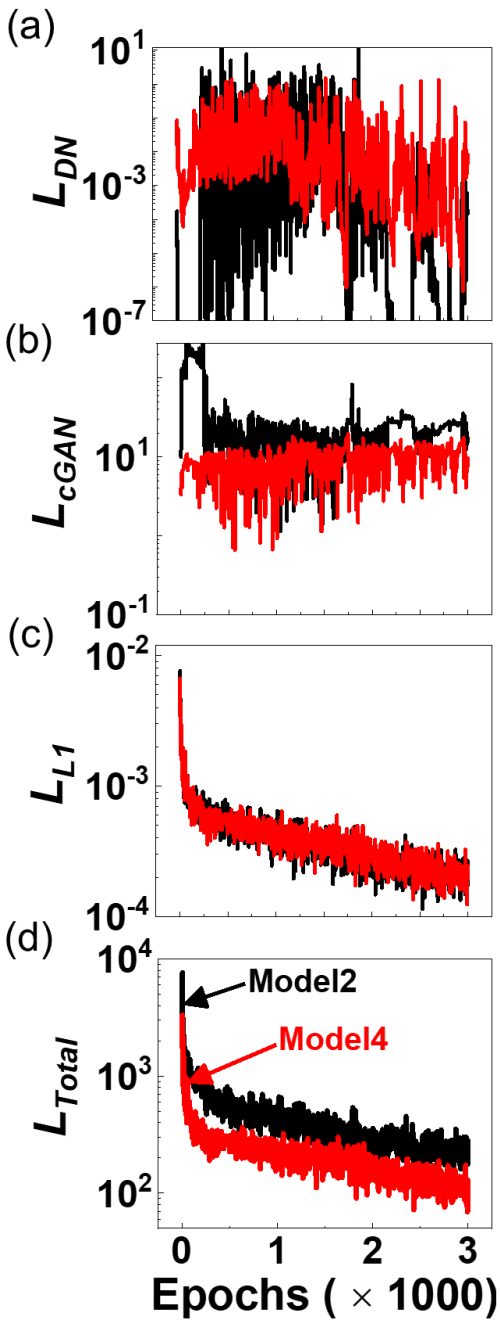

Figure 3. 3. Convergence of Loss Functions During Training: (a) Discriminator Loss, (b) Generator Loss, (c) *L1* Loss, and (d) Total Losses in Generator Training.

The hyperparameter λ balances the two objectives for the generator training. If λ increases, the GN focuses on minimizing the mean absolute difference between X and Y,

and improving the low-frequency accuracy. Many temperature images in our convection problem include low-frequency features, i.e., small temperature gradients across unit image pixels. Thus, in this study, λ is raised to sufficiently weigh *L1* loss over L_{cGAN} and to minimize the model errors. The value of λ that minimizes ΔT is identified as 5×10^5 for Model 1 and 4, and 10^6 for Model 2 and 3. For Model 5, λ is 0. With the optimal values of λ, the mean prediction error ΔT that is averaged over all the channel locations and time steps is less than 0.1K for all the models except Model 5.

3.4. Model Testing

The trained cGANs serve as surrogate models for the mixed convection that can infer the temperature map at an arbitrary channel location *z* and time *t*. Figure 3.4 shows the temperature maps at an early stage, t = 12s, for selected *z* locations, which are not seen by the models during training. Models 1 - 4 successfully generate the characteristic development of convection transverse rolls in mixed convection flow. The temperature maps show similar features raised by buoyancy driven secondary flows at an early stage. However, Model 5 fails to approximate correct temperature maps owing to the incomplete objective of the GN. At t = 12s, the prediction error ΔT that is averaged over all the channel locations ranges merely 0.15 - 0.24K for Models 1 to 4, but is 1.91K for Model 5.

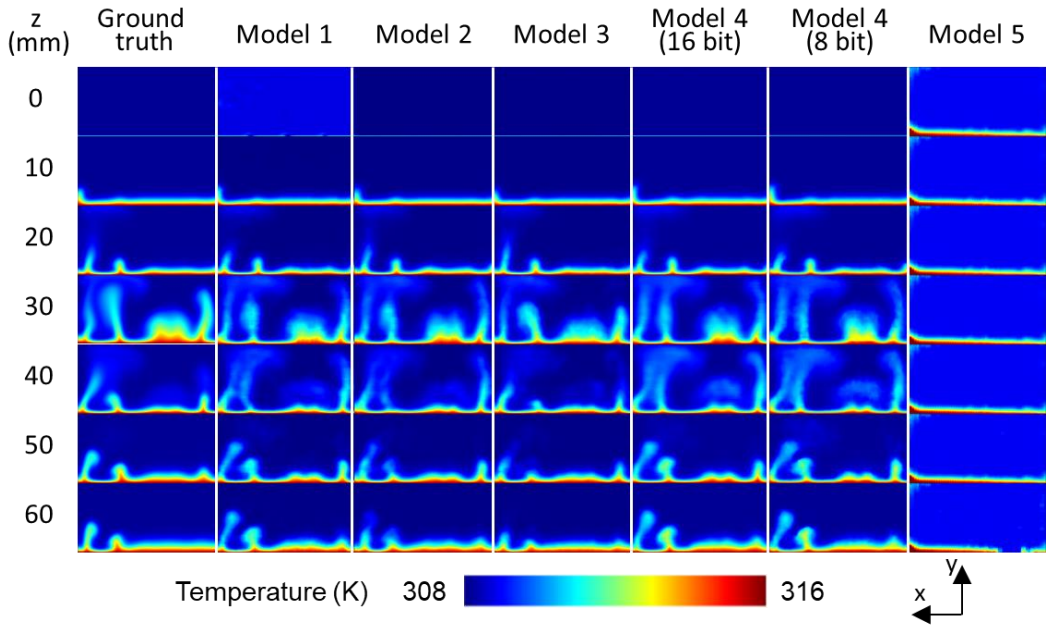

Figure 3. 4. Temperature Distributions at Selected Cross-sections in a Heated Channel Predicted at t = 12s by cGAN Models and FVM (Denoted as Ground Truth).

The image contrast determined by the temperature gradient affects the model accuracy. Figure 3.5 shows the model predictions during a stable stage, t = 342s. The trained models are able to infer the elongated transverse convection rolls and the secondary flow development over the entire channel region. As compared to the initial stage of the mixed convection, the spatial temperature variation during a stable stage is apparently small due to the flow mixing induced by the secondary flows. For example, at z = 60 mm, the Michelson image contrast reduces from 0.0128 to 0.0113 when comparing the ground truths at t = 12s and 342s. The Michelson image contrast of each temperature map is calculated by $(T_{max} - T_{min})/(T_{max} + T_{min})$, where T_{max} and T_{min} are the maximum and minimum temperatures. I observe that the models tend to generate blurred features, e.g., indistinct boundaries of convection rolls at z = 60 mm if the Michelson image contrast is

approximately below 0.011. Overall, the PatchGAN classifier performed better than the CNN classifier. At t = 342s, the prediction error ΔT that is averaged over all the channel locations is 0.135K for Model 1, 0.125K for Model 2, 0.106K for Model 3, and 0.132K for Model 4. Qualitatively, Model 4 generates the image features with the best clarity. Based on the quantitative and qualitative comparison, Model 4 is considered as the most suitable one for our convection problem.

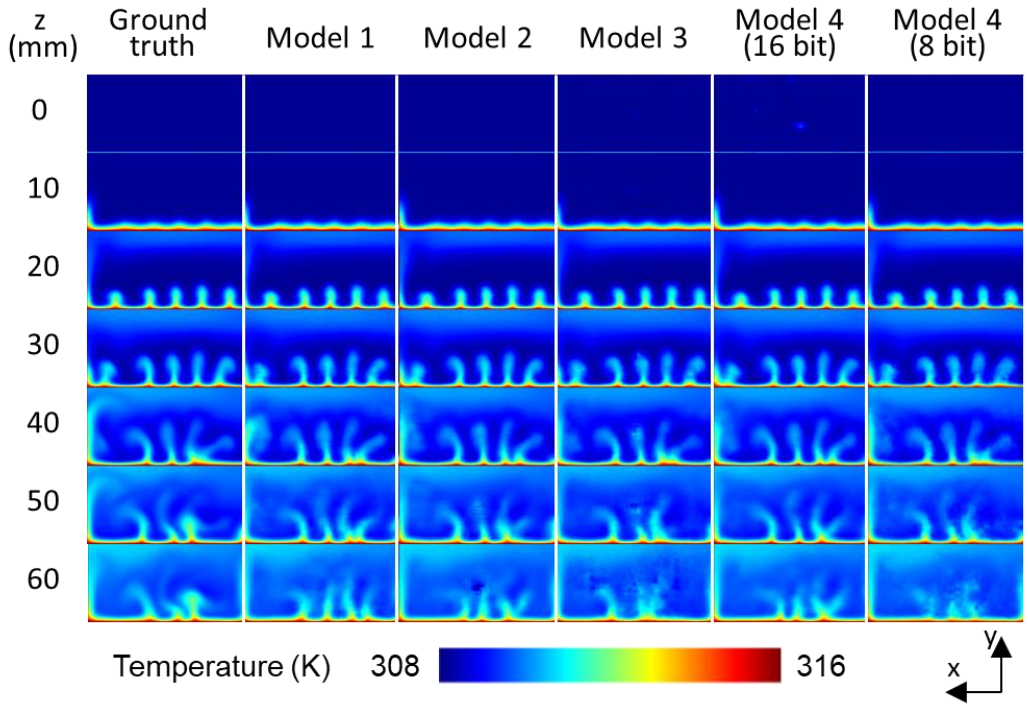

Figure 3. 5. Temperature Distributions at Selected Cross-sections in a Heated Channel at t = 342s Predicted by cGAN Models and FVM (Denoted as Ground Truth).

When the cGAN is trained with the identical training hyperparameters, the depth and the number of channels of input images can become important factors. The two columns for Model 4 in Fig. 3.4 and 3.5 compare the model predictions when the model is

trained either with 8-bit, 3-channel RGB images or 16-bit, 1-channel grayscale images. Qualitatively, the model trained with the single-channel images sharply produces the image features even for the temperature maps with low contrasts, i.e., temperature maps at $z \geq$ 40mm. The single-channel image-based training would be easier to achieve a higher accuracy, since it requires a substantially smaller number of training parameters than the three-channel image-based training. The single-channel image-based training required merely 513 parameters for the generator training whereas the three-channel image-based training required 1539 parameters. Moreover, the single-channel images required 1.6% smaller data space than the three-channel images.

The cGAN exhibits higher accuracy for the simpler temperature maps occurring during the stable stage. Figure 3.6 compares the temperature predicted by Models 2, 4 and FVM at all pixels of the selected temperature maps. Figures 3.6(a) and 3.6(b) correspond to Model 2 and Fig. 3.6(c) and (d) correspond to Model 4. The coefficient of determination R^2 tends to be greater at a smaller z, i.e., near the channel entrance region, where the image features are simpler with higher contrast than at a greater z. When the temperature ranges are not consistent between the temperature maps, the model accuracy can be more fairly evaluated by R^2 than the absolute scale error such as ΔT. R^2 is greater at t = 342s where the image feature variation over time is less than the initial flow development stage at t = 12s. The time-averaged R^2 for entire time steps at z = 50mm is 0.952 for Model 2 and 0.953 for Model 4. At several time steps as shown in Fig. 3.6, Model 4 exhibits lower R^2 than Model 2. According to the time-averaged R^2, the PatchGAN classifier and the CNN classifier without strided convolutions present a similar accuracy.

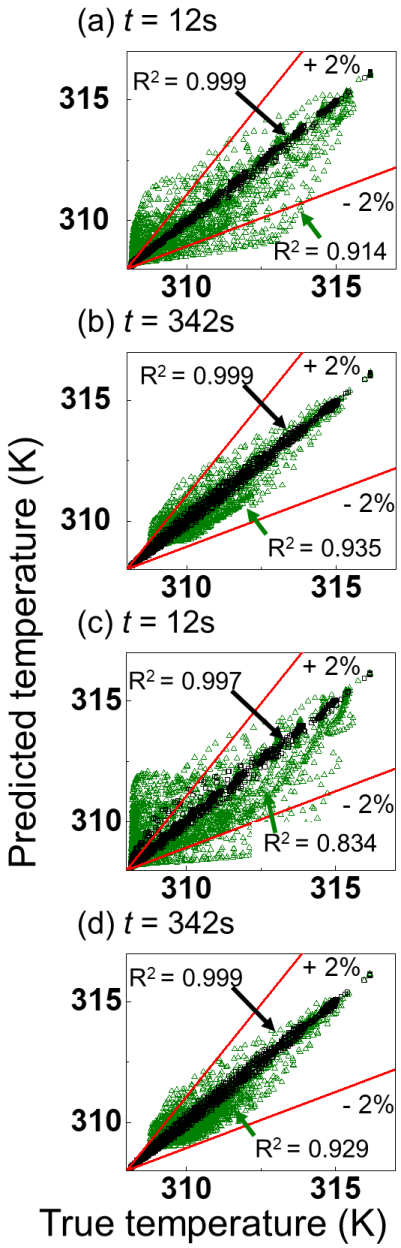

Figure 3. 6. Comparison of Temperature Predicted by cGAN and FVM. (a) Model 2 Prediction at t = 12s, (b) Model 2 Prediction at t = 342s, (c) Model 4 Prediction at t = 12s, and (d) Model 4 Prediction at t = 342s.

Figure 3.7 shows the temperature maps predicted by Model 4 at unseen channel locations from z = 5mm to z = 55mm. Figures 3.7(a) and 7(b) present the model predictions

either during the early unstable stage at t = 12s and the stable stage at t = 342s. Even for the unseen channel locations, the trained cGAN successfully approximates the development of convection rolls at all time steps. Similar to the previous model test conducted for various t, the model predicts better for the temperature maps with a higher contrast. The Michelson image contrast in the ground truth, that is averaged for all selected channel locations, is 0.0114 at t = 12s and 0.0109 at t = 342s, indicating that the image contrast is poor when the temperature field has been mixed and homogenized. Consequently, Model 4 exhibited greater location-averaged R^2 at t = 12s (R^2 = 0.907) than at t = 342s (R^2 = 0.862). Based on the dependence of the cGAN accuracy on the image contrast, I can expect that the cGAN will perform better if the color range of the temperature maps are adjusted for the different stages of the convection process.

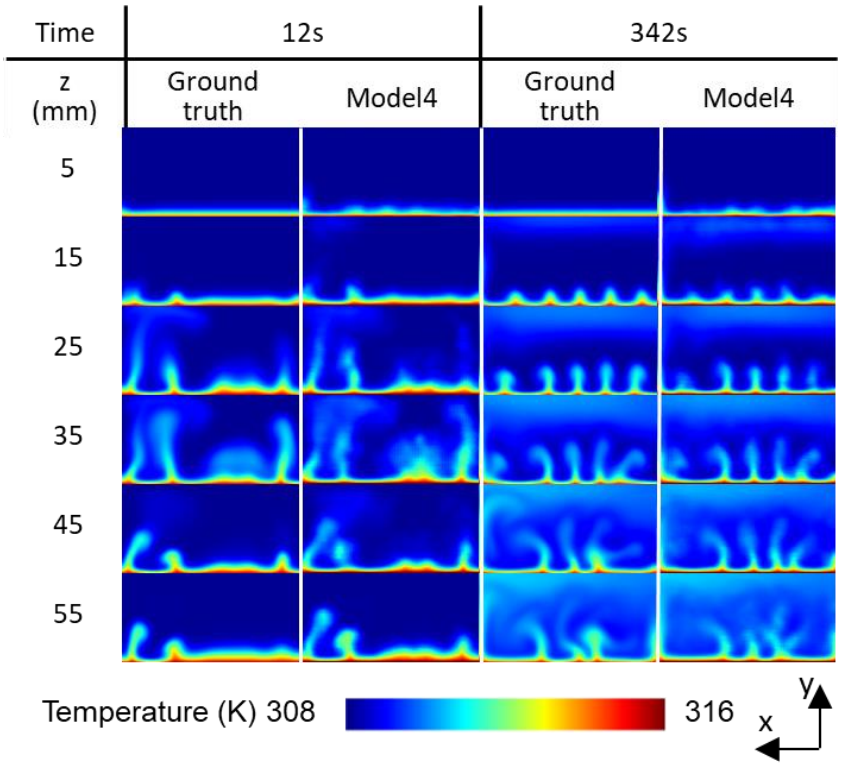

Figure 3. 7. Temperature Distributions at Unseen Cross-sections in a Heated Channel Predicted by Model 4 and FVM (Denoted as Ground Truth).

3.5. Conclusions

This study investigated a cGAN based surrogate model for rapidly approximating the temperature maps of a 3D transient mixed convection process. The trained cGAN was able to infer the temperature map at an arbitrary channel location and time when an input conditional vector provided the numerical information about the channel geometry, thermal and hydraulic boundary conditions, location and time. For the training of the cGAN, four architectures of the discriminator network were compared, i.e., PatchGAN classifier and CNN classifier with or without strided convolutions. All four cGAN models accurately predicted the temperature maps with a mean prediction error ΔT less than 0.1K. When comparing the model outputs qualitatively, Model 4 produced the image features, i.e., shapes of convection rolls, with the best clarity, indicating that the PatchGAN classifier without the strided convolutions is suitable for generating the complex temperature features of the unstable mixed convection processes. The cGAN accuracy depended on the image contrast, rate of spatiotemporal variation of the temperature, and the number of channels. When the Michelson image contrast was less than 0.011 and when the temperature features rapidly varied over a time step, the model tended to generate blurred features. Training the model with single-channel images helped to sharply produce the complex image features even for the temperature maps with low contrasts. This chapter demonstrated the potential of a deep learning approach as a rapid surrogate model for complex transport processes. The developed cGAN surrogate model may cover a wider range of flow conditions and channel geometries if additional training data is provided. The cGAN surrogate model will help the researchers who need to design and optimize the heat transfer systems without using the expensive and compute-intensive numerical simulations.

4. MACHINE LEARNING CLASSIFICATION OF THE MASS CONVECTION PROCESS

4.1. Introduction

Understanding the laminar-turbulent transition process in flow channels or tubes is one of fundamental problems in fluid dynamics. In 1883, Reynolds conducted experiments to investigate the critical Reynolds number related to where and when eddies start to appear(Reynolds 1883). In practice, such information is important, because the flow properties such as pressure drop, mass transfer coefficient, or convective heat transfer coefficient radically change in different flow regimes. However, an accurate analysis of the laminar-turbulent transition process in custom-designed, geometrically complicated channels or tubes has been very difficult task due to the issues in device fabrication and data analysis. Flow visualization has been mostly limited to tubes or channels made of glass or highly transparent plastic. Due to the difficulty and cost, manufacturing of fluidic devices with complex geometries or small-scale features has been challenging. Furthermore, even when the internal flows are successfully visualized, extracting and analyzing the information from the flow videos have been time-consuming and inaccurate tasks. Mostly people analyze merely a few image frames out of the entire video. Such data extraction method usually results in significant loss of information embedded in the raw data, but has been inevitably adopted for the efficiency in data analysis.

Recently, machine learning (ML) has been employed in a wide range of areas such as medicine, health care, robotics, stock market where useful information (e.g., patterns) must be extracted out of large-sized data in a rapid fashion(Erickson et al. 2017; Bell 2015; Smart and Kaelbling 2002; Beam and Kohane 2018). Now the impact of ML has also been

growing over scientific research due to the reduced computation cost required to perform ML(Brenner, Eldredge, and Freund 2019; Brunton, Noack, and Koumoutsakos 2020). Particularly, fluid mechanics is one of the fields that start to rigorously adopt ML. In fluid mechanics, a challenge has been to effectively and accurately analyze massive amount of data acquired from experiments, field measurements, and numerical simulations. As many fluid systems exhibit complex dynamic phenomena, large amount of data is inevitably produced while investigating the systems.

Several recent publications explore how ML is able to assist data analysis and to tackle challenging tasks in fluid mechanics field. Previous studies employed artificial neural network (ANN) (Kreitzer, Hanchak, and Byrd 2013; Rosa et al. 2010) or deep learning algorithm (Du et al. 2018) to identify the flow regime of two-phase flows. For modeling the hydraulic or thermal properties of two-phase flows, prior knowledge of the flow regime is essential, which, however, can mostly be determined by analyzing multiple parameters. By simultaneously processing more than 100 input parameters, ANN was able to achieve 99% accuracy for determining the regimes of two-phase R-134a flow(Kreitzer, Hanchak, and Byrd 2013). Artificial neural network was also useful when regressing nonlinear correlations involving multiple variables. In a previous work(Joss and Müller 2019), ANN was used to obtain a correlation that predicted the normal boiling point of organic compounds from a dataset for more than 6000 compounds. The ANN model accomplished the coefficient of correlation R^2 of 0.89 while R^2 of normal linear regression was 0.84(Joss and Müller 2019). More recently, deep-learning algorithm performed a challenging task that was to extract the velocity and pressure fields directly from qualitative flow images(Raissi, Yazdani, and Karniadakis 2020). Navier-Stokes equations were

encoded into the neural network algorithm, which enabled to convert the qualitative information (i.e., fluid concentration) to quantitative information (i.e., velocity and pressure). While the previous works show that ML offers considerable promise to fluid mechanics, the field of ML for fluid mechanics is yet considered very immature and there remain many unresolved problems. So far, only a few ML algorithms has been tested for fluid mechanics problems such as neural network. Neural network is a powerful and versatile algorithm but requires relatively large-size dataset and sophisticated settings. In common experiments, simple ML algorithms such as random forests(Breiman 2001) and support vector machine, would be adequate, since they exhibit robust performance while necessitating tuning of a small number of hyperparameters and comparatively small-size dataset.

This chapter explores how ML improves the interpretation of the unsteady flow phenomena by automating the flow image analysis. Particularly, this study focuses on the laminar-turbulent transition process occurring in three-dimensionally (3D) printed tubes. The random forests (RF) algorithm is first trained by the flow images taken of the tubes, and predicts the stages of the laminar-turbulent transition process or the onset locations of flow regime transition. The hyperparameters of RF are tuned to enhance the prediction accuracy.

4.2. Methodology Overview

Figure 4.1 briefly illustrates the overview of ML assisted flow analysis combined with 3D printing sample preparation. First, transparent flow devices of interest are prototyped by 3D printing. Then, ML dataset, flow images, are acquired through flow

visualization. To reduce the dimensionality and noise in datasets, image preprocessing is often necessary. When employing a supervised ML algorithm, the flow images need to be manually classified and labeled considering the objectives of flow analysis. Once datasets and data labels are properly formatted for an ML algorithm, those information is used for model training and testing. During training and testing hyperparameters are tuned such that ML model exhibits reasonable performance. Finally, the ML model is further validated through the test on a secondary, unbiased dataset.

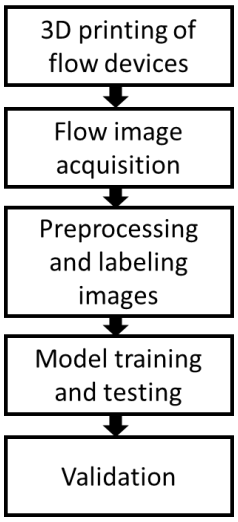

Figure 4. 1. Methodology Flowchart

Figure 4.2 shows an experimental setup that was used to acquire the flow images. Water at 25^{o}C first was flown through a filter that removed particles greater than 5μm. A flowmeter with an accuracy of ± 4% controlled the volumetric flow rate from 0.063 liters per minute (lpm) to 0.505 lpm (equivalent range of Re = 230 - 1843). To visualize the flow, a syringe pump injected dye solution (blue food coloring 91%, isopropyl alcohol 9%) with the same velocity into the water flow. For the dye injection, a needle was inserted into the water line. The needle tip was bent horizontally and aligned to the water center line. Then,

a high-speed camera acquired flow images at a resolution of 1280 × 720 pixels with 240 frames per second (fps).

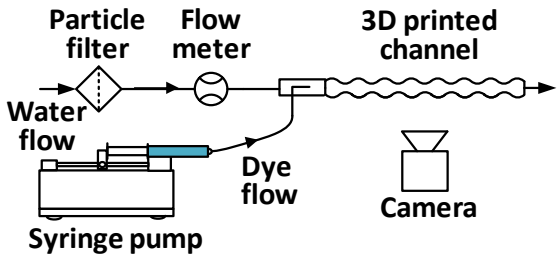

Figure 4. 2. Flow Visualization Setup.

For the flow visualization, two semi-transparent tubes were fabricated by a commercially available digital light processing (DLP) 3D printer (Perfactory P4K, EnvisionTEC). Polymer 3D printing is capable of prototyping parts of high geometric complexity with smooth surfaces(Kwon et al. 2020; 2019). When combined with a transparent material, it is possible to print a variety of transparent fluidic devices with extremely low costs(Shallan et al. 2014; Romanov et al. 2018; Aycock, Hariharan, and Craven 2017). Figure 4.3(a) shows a 3D printed, circular straight tube with diameter, D, of 6.5mm, wall thickness, t, of 0.5mm, and length, L of 140mm. Figure 4.3(b) shows another tube with sinusoidally wavy profile in circular cross-section. The wavy tube has minimum diameter, D_{min}, of 3mm, average diameter, D_{avg}, of 6.5mm, wavelength, λ, of 14mm, wave amplitude, A, of 1.75mm, t of 0.5mm, and L of 140mm. Such aspect ratio of the wavy tube has been widely used in literature, since it was empirically demonstrated to offer balanced heat transfer augmentation and pressure drop penalty(Wang and Vanka 1995; Tatsuo Nishimura, Ohori, and Kawamura 1984; Tatsuo Nishimura et al. 1985; Guzmán and Amon

1994; T. Nishimura et al. 2003). The 3D printer successfully replicated the CAD models while there were slight differences in size (< 0.7%). For each tube, consumed printing material was merely around $2 and printing time was about 45 hours under in-house setup, which were extremely cheap compared to other fabrication techniques. Figure 4.3(c) shows the moment when the 3D printer finalizes the printing of a wavy tube.

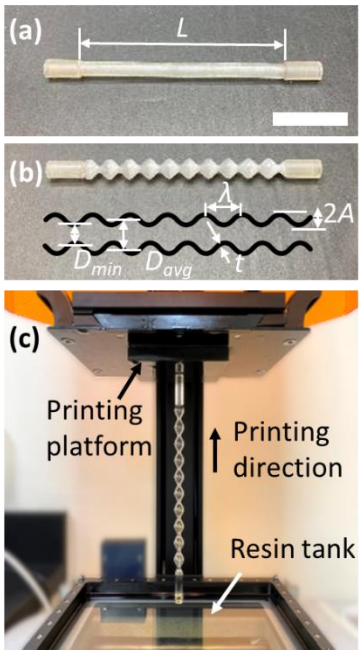

Figure 4. 3. Images of (a) 3D printed Straight Tube, (b) Wavy Tube, and (c) DLP Printer. The Scale Bar Represents 5cm.

4.3. Machine Learning

4.3.1. Data Preparation

Preprocessing of data, i.e., dimensionality reduction and data cleaning, is an essential step to efficiently run ML(Brenner, Eldredge, and Freund 2019). Figure 4.4 shows the preprocessing steps. First, flow images were sampled at every 1/30 seconds from the videos originally taken at 240 fps. With the flow speed range used in this work, the fact

that the change of flow pattern was not significant per every 1/240 seconds was observed. In such situation, downsampling reduces the data amount, and leads to filtering out flow images with different flow patterns. Then, the images were cropped to cover only the tube regions [Fig. 4.4(a)]. This step not only eliminates the noise in data that would exist outside the tube region, but also reduces the data dimension. The cropped images contained reduced number of pixels (1280 × 122 pixels). The dimensionality of the data can be further reduced by converting the images to grayscale [Fig. 4.4(b)]. Original images had color composed of red, green, and blue values. In grayscale images, the vector information contained per pixel are reduced to a scalar value, i.e., grayscale magnitude between 0 and 255. Last, the contrast between dye and surroundings was raised by whitening the pixels corresponding to the dye flow [Fig. 4.4(c)]. The fact that the pixels of dye region had much lower grayscale magnitudes than the surroundings was observed. Thus, the dye pixels were readily filtered, and their grayscale magnitudes were set to 255. After the preprocessing, an image matrix with a size of 1280 × 122 was transformed to a 1 × 156160 single-row array. Then, a ML feature matrix was created by stacking a series of image arrays. For example, if dataset was containing 210-image information, then the size of feature matrix became 210 × 156160.

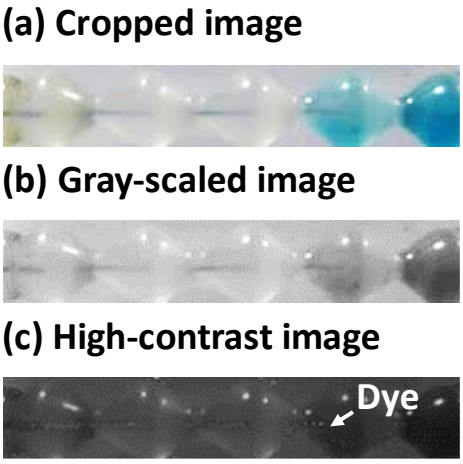

Figure 4. 4. Image Pre-processing for Machine Learning Modeling. (a) Cropped, (b) Gray-scaled, and (c) High-contrast Images of a Wavy Tube.

4.3.2. Data Labeling

The images were classified based on either the stages of the laminar-turbulent transition process or the locations where the transition started to occur. For the flow transition stage classification, the stages were split to five types based on the flow regimes presented in the tube: laminar regime only (label 0), laminar and transitional regimes (label 1), laminar, transitional, and turbulent regimes (label 2), transitional and turbulent regimes (label 3), and turbulent regime only (label 4). Figure 4.5 shows the representative label images of the straight tube. In label-0 images, dye flows straightly without fluctuations. In label-1 images, flow is laminar near the entrance, and oscillatory flow intermittently appears from the middle region. In label-2 images, turbulent flow is shown near the tube exit, where dye spreads throughout the tube entirely. In label-3 images, the flow is unsteady and unstable from the tube entrance. Lastly, in label-4 images, the flow is completely turbulent anywhere in the tube. Similarly, Fig. 4.6 shows the label images of the wavy tube.

In wavy tube, the flow transition occurred at lower Re (Re = 691) than the straight tube (Re = 1152). Such early flow transition is characteristic of the wavy tubes(T. Nishimura et al. 2003). Due to the strong flow instability, label-1 images were not found in the wavy tube.

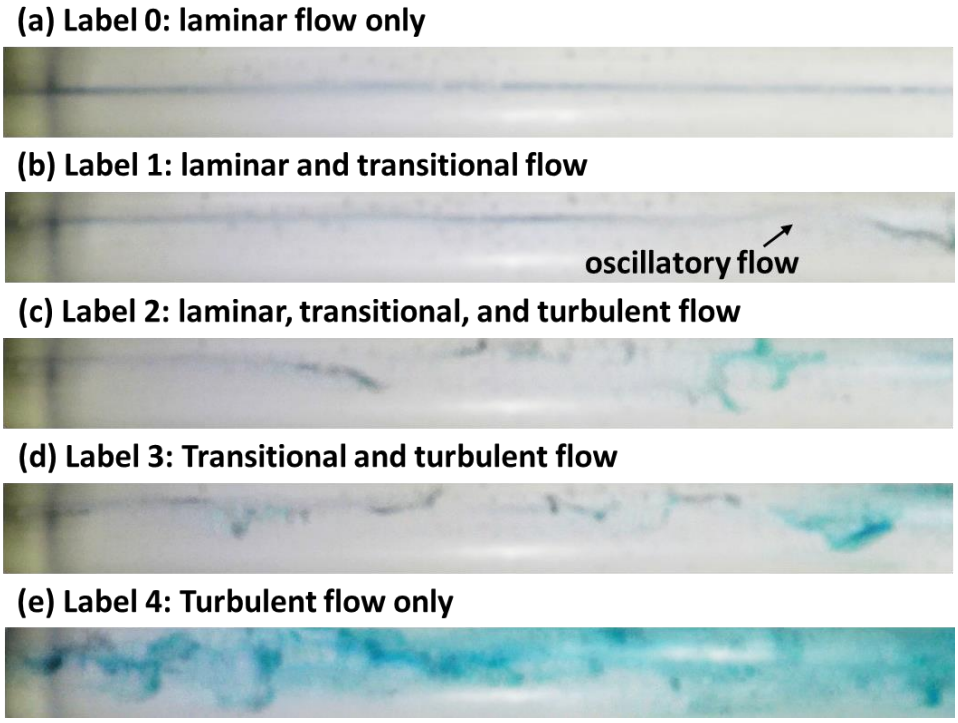

Figure 4. 5. Flow Regime Labels for Straight Channel.

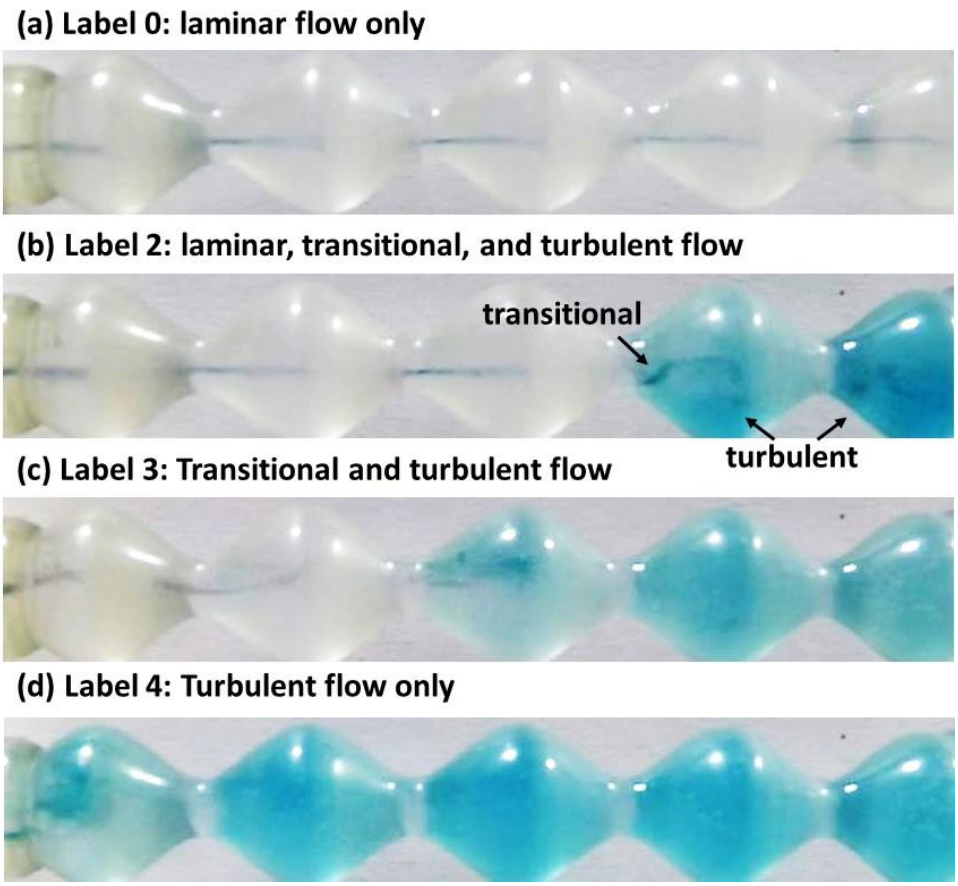

Figure 4. 6. Flow Regime Labels for Wavy Channel.

For the wavy tube, another type of classification was performed based on the transition onset location as shown in Fig. 4.7. The label number corresponds to the wave section where the flow transition starts to occur. For example, label 0 means that transition does not occur. Label *n* refers to the transition occurring from n-*th* section. All classification labels are summarized in Table 1.

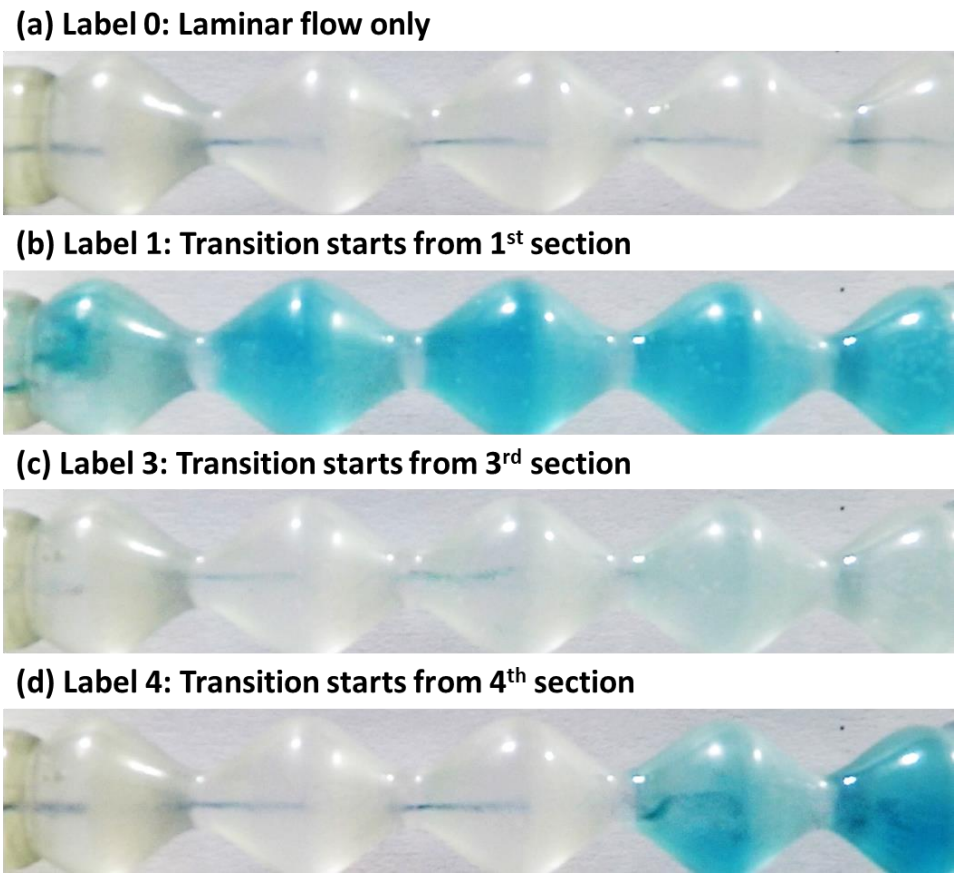

Figure 4. 7. Labels for Transition Onset Location in a Wavy Channel. (a) Label 0: Transition Does Not Occur; (b) Label 1: Transition Occurs from 1st Section; (c) Label 3: Transition Occurs from 3rd Section; (d) Label 4: Transition Occurs from 4th Section.

Table 4. 1. Two Label Systems Used for Image Classification: (a) Flow Regime Labels (Top) and (b) Labels for Flow Transition Onset Location (Bottom).

(a) Flow regime labels	
0	Laminar flow only
1	Laminar and transitional flow
2	Laminar, transitional, and turbulent flow
3	Transitional, and turbulent flow
4	Turbulent flow only
(b) Labels for flow transition onset location	
0	Laminar flow only
1	Transition starts from 1st section
3	Transition starts from 3rd section
4	Transition starts from 4th section

4.3.3. Classifiers

For the image classification, this work particularly employed a general-purpose classifier, random forests (RF) algorithm(Breiman 2001). This algorithm operates as an ensemble of several randomized decision trees, and makes decision by averaging the predictions made by the trees. Important hyperparmeters for RF include the depth of each tree (max_depth), seed number used by a random number generator (random_state), and the number of trees in the forest (n_estimator). Due to the simplicity of the algorithm and an open-source implementation in scikit-learn(Pedregosa et al. 2011), RF has been one of the popular ML algorithms.

To understand how the training dataset affects the performance of the RF algorithm, 3 different datasets were employed to train the classifiers based on the flow transition stage

labels. Three datasets correspond to the flow images acquired from 1) the straight tube only, 2) the wavy tube only, and 3) both the straight and wavy tubes. Table 2 lists the information of training and testing datasets. The RF classifier trained by each dataset was denoted as classifier 1, 2, or 3 (see Table 2). In addition, the RF classifier was also trained for the transition onset locations. This classifier was denoted as classifier 4.

Table 4. 2. Training and Testing Datasets, and Accuracy Scores of RF Classifiers.

		Classifier 1	**Classifier 2**	**Classifier 3**	**Classifier 4**
	Classification criterion	Flow transition stage			Transition onset location
	Tube type of datasets	Straight tube	Wavy tube	Straight, wavy tubes	Wavy tube
I.Training	**Size of dataset (frames)**	630	630	1260	630
	Accuracy score	1	1	1	0.97
II.Test	**Size of dataset (frames)**	210	210	420	210
	Accuracy score	1	0.95	0.97	0.96

4.3.4. Model Training and Testing

A protocol for model training and testing is as follows. The image dataset is randomly divided into training (75%) and testing (25%) sets. After training, the RF classifier is evaluated by the accuracy score(Pedregosa et al. 2011) that represents the ratio of correct predictions to the total number of predictions. A confusion matrix is also inspected to visualize the performance, and to understand where the errors originate from.

If the training result is not satisfactory, then hyperparameters are tuned to improve the accuracy score. Once the training result becomes well-fitting, the RF classifier is again assessed with the testing dataset.

Following the aforementioned protocol, all classifiers were evaluated on training datasets, resulting in accuracy scores greater than 0.96 (Table 2). Among all the classifiers, classifier 4 was the most challenging to achieve high accuracy. Table 3 shows the confusion matrix of classifier 4 on the training dataset. The confusion matrix shows that the classifier 4 mislabelled some of the label-3 and label-4 images. To improve the classification accuracy, two hyperparameters, i.e., max_depth and random_state, were tuned while fixing n_estimator as 11. Figure 4.8(a) shows how the accuracy score of classifier 4 on training dataset varies with hyperparameters. It was observed that max_depth ≥ 3 was required to accurately classify label-3 images. Although large max_depth is likely to give a high accuracy score, it must be also considered that increasing max_depth leads to extended computation time and increasing the risk of overfitting. After the tuning, the accuracy score of classifier 4 was enhanced to 0.97 on training dataset and 0.96 on testing dataset.

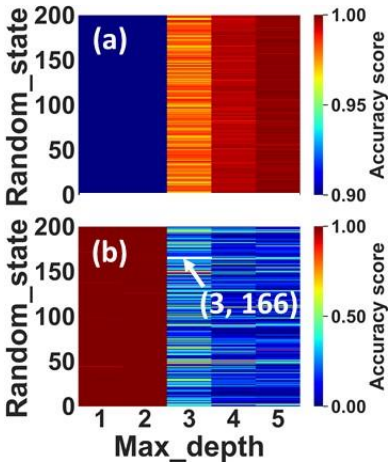

Figure 4. 8. Hyperparameter Tuning of Classifier 4 on (a) Training and (b) Validation Datasets. Accuracy Scores Are Measured as a Function of Max_depth and Random_state.

Table 4. 3. Confusion Matrix of Classifier 4 on Training Dataset

		Predicted label			
		0	**1**	**3**	**4**
Actual label	**0**	157	0	0	0
	1	0	158	0	0
	3	0	0	89	5
	4	0	0	15	206

4.3.5. Validation

To understand the versatilities of the trained RF classifiers, models were validated using a second, unbiased datasets. Table 4 lists the validation datasets for all classifiers, where ST and WT represent straight tube and wavy tube, respectively. To acquire validation data, flow images were taken at different Reynolds numbers as compared to training data. Each validation dataset consisted of 210 flow images. Furthermore, wavy tube images were used to evaluate classifier 1, that is trained by straight tube only.

Similarly, the validation dataset for wavy tube classifier 2 contained straight tube images. As expected, the accuracy score of classifier 1 on transitional flows in the wavy tube was extremely low, 0, because the image patterns of transitional flows in straight and wavy tubes apparently differ. Due to the same reason, the accuracy score of classifier 2 on transitional flows in the straight tube was extremely low, 0.23. To overcome such underfitting issue, sufficient amount of data for various tube shapes must be added in training set. For instance, classifier 3, trained by both straight and wavy tube images, was highly accurate for all validation datasets.

Table 4. 4. Validation Datasets and Accuracy Scores of RF Classifiers.

		Classifier 1			**Classifier 2**			**Classifier 3**			**Classifier 4**		
Dataset	**Tube type**	ST	WT		ST	WT		ST	WT		WT		
	Re	691	230	691	1152	230	1843	691	230	1843	230	691	1843
Accuracy score		1	1	0	0.23	1	0.98	0.99	1	1	1	0.97	1

Validation may lead to a second round of hyperparameters tuning. The accuracy score of classifier 4 on validation set was sensitive to hyperparameters, which necessitated another round of hyperparameter tuning as shown in Fig. 4.8(b). The fact that reducing max_depth increased the accuracy score was obwerved, indicating that the trained model was overfitting. To avoid the overfitting problem and to achieve high accuracy from both training and validation sets, a best combination of max_depth and random_state was

searched from Fig. 4.8(a) and (b). Consequently, max_depth of 3 and random_state of 166 were chosen, and improved the accuracy score on the validation set to 0.97.

4.4. Discussion

ML model is capable of analyzing the flow images faster and more accurately than humans. Such capability of ML classifier may unlock new opportunities in fluid research or flow system applications. For example, Fig. 4.9(a) and (b) show flow regime label variations over time in straight and wavy tubes that are automatically identified by RF classifiers. Classifier 1 predicted for the straight tube and classifier 2 predicted for the wavy tube. Each curve contains information of 210 flow images which were classified up to 2.4 seconds only. These plots finely show how flow pattern dynamically varies along time at a certain flow condition. At Re =1382 in the straight tube [Fig. 4.9(a)], flow regime switches between laminar, transitional and turbulent, and eventually maintains transitional turbulent flow. At Re = 1382 in the wavy tube [Fig. 4.9(b)], flow regime frequently switches between transitional turbulent and turbulent, indicating that the flow is more unstable than the one in straight tube. Figure 4.9(c) shows how the label for transition onset location in the wavy tube changes with time. Classifier 4 revealed that the transition onset location oscillates between the 3^{rd} and 4^{th} sections at Re = 921. Such high-rate, time-dependent analysis on flow images is a challenging and extremely time-consuming task, if this task is done manually. The automated and high-throughput flow analysis enables to accurately understand the transient flow physics in fluidic devices.

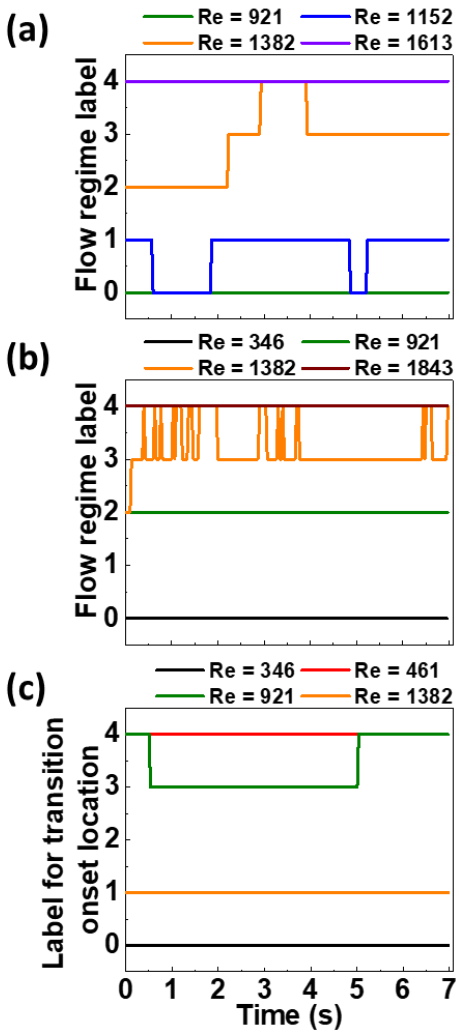

Figure 4. 9. Machine Learning Classification of Flow Images by RF Models. Flow Regime Variation with Time in (a) Straight Tube (Classifier 1), and (b) Wavy Tube (Classifier 2), (c) Transition Onset Location Variation with Time in Wavy Tube (Classifier 4).

Despite the great potential of the RF algorithm, ML assisted flow analysis can become further autonomous or will be able to extract new perspective information if advanced ML techniques are employed. For example, unsupervised deep learning algorithms such as neural network, generative adversarial network, are able to

autonomously extract features and identify unknown complex patterns, although they require larger amount of dataset and greater effort in structure settings than other simple algorithms. This effort will make ML assisted flow analysis smarter and more practical by reducing the required efforts in feature engineering and data labeling process.

4.5. Conclusions

This work explored a high-throughput and accurate flow analysis method enabled by ML with 3D printing technology. Laminar-turbulent transition process was analyzed in a semi-transparent 3D wavy tube and a straight tube, which were prepared by DLP 3D printing with low cost. Flow images were recorded by dye injection technique, and used for ML datasets. Simple random forests models classified the images based on the flow regime and transition onset location. With both straight and wavy tubes flow images, the trained RF classifier exhibited an accuracy of > 0.97 on training datasets and > 0.95 on testing datasets. However, when completely different tube designs, e.g., wavy tube images for the classifier trained only by straight tube images or vice versa, were provided, the RF model showed low accuracy < 0.23, indicating the importance of sufficient and appropriate dataset. When classifying the images based on transition onset location, efforts for tuning hyperparameters, i.e, max_depth and random_state, were crucial to avoid underfitting or overfitting issues. Finally, the trained RF models demonstrated high-throughput flow analysis by automatically classifying flow images within ~0.01 second per an image. This study demonstrates the potential of ML approach combined with 3D printing that enables to gather and process a large amount of fluid mechanics data.

5. CONCLUSION AND FUTURE WORK

5.1. Conclusions

This work explored the potential of machine learning in heat and mass convection phenomena. The machine learning model automatically inferred the thermal properties (e.g., Nusselt number, friction factor, and temperature distributions) without iterative numerical calculation under various boundary conditions and unseen geometries. Moreover, the fluidic devices were printed by the 3D printer for experiments, then the fluid images attained from the experiments automatically classified by flow transition stage and transition onset location.

First, this research developed the first deep learning model for forced convection heat transfer problems based on conditional generative adversarial networks (cGAN). To train cGAN networks, a set of graphical inputs containing the geometric and flow conditions and graphical outputs representing the convection properties were prepared. A trained cGAN model successfully inferred the distribution of temperature, Nu, and f with various widths of the channel and over a wide range of Re (Re = 100 to 27750) at the inlet. The cGAN was optimized through hyperparameter tuning, e.g., the data size, training epoch, and λ to attain high accuracy. It achieved an accuracy ≤ 0.97 for the Nu estimation and a maximum absolute error (MAX) ≤ 3.3K for the temperature approximation. A simple data augmentation process also was added in the training process to apply this trained cGAN to unseen various channel geometries, e.g., widened, narrowed, and rotated. This process resulted in improving the model accuracy by 70%.

For identify the cGAN architecture that works for transient convection, this study investigated a cGAN based surrogate model for rapidly approximating the temperature maps of a 3D transient mixed convection process. The trained cGAN was able to infer the temperature map at an arbitrary channel location and time when an input conditional vector provided the numerical information about the channel geometry, thermal and hydraulic boundary conditions, location and time. For the training of the cGAN, four architectures of the discriminator network were compared, i.e., PatchGAN classifier and CNN classifier with or without strided convolutions. All four cGAN models accurately predicted the temperature maps with a mean prediction error ΔT less than 0.1K. When comparing the model outputs qualitatively, Model 4 produced the image features, i.e., shapes of convection rolls, with the best clarity, indicating that the PatchGAN classifier without the strided convolutions is suitable for generating the complex temperature features of the unstable mixed convection processes. The cGAN accuracy depended on the image contrast, rate of spatiotemporal variation of the temperature, and the number of channels. When the Michelson image contrast was less than 0.011 and when the temperature features rapidly varied over a time step, the model tended to generate blurred features. Training the model with single-channel images helped to sharply produce the complex image features even for the temperature maps with low contrasts. This work demonstrated the potential of a deep learning approach as a rapid surrogate model for complex transport processes.

Lastly, machine learning was hired to analyze and predict flow characteristics in the 3D printed tube experimentally. The flow regimes and locations where the transition started in the printed fluidic devices were analyzed and predicted by the machine learning, random forests models. The 3D printed tubes with great dimensional accuracy were

prepared for flow visualization, and flow images were captured by a high-speed camera. Three kinds of classifiers were hired according to their purpose, e.g., classifying flow regimes in a straight tube, a wavy tube, and classifying the location of the onset transition in the wavy tube. The maximum accuracy score increased up to 0.97 through hyperparameter tuning.

5.2. Future Work

The following work will discuss how combining machine learning and optimization algorithms can result in a new design methodology. Optimization algorithms enabled reliable predictions in the prescribed physical conditions. Machine learning proved to have a strong inference ability in predicting the output with the aid of multiple stacked convolutional neural networks without iterative numerical computations. Thus, a new design methodology will combine the advantages of optimization algorithms and machine learning to suggest time-efficient and automated design algorithms. It can also be applied to various design problems to find optimal designs for things such as heat exchangers, electronics in satellites, or spacecraft, which require reliable thermal management to handle a large amount of heat while keeping tight control of the temperature under limited space. In the optimization algorithm process with machine learning, it is possible to generate designs close to optimum designs using machine learning rather than the randomly generated technique used in the conventional optimization algorithm.

The new design methodology, combined optimization algorithms, and machine learning will lead to the generation of nearly optimal designs in a timely and automated manner, as shown in Fig. 5.1.

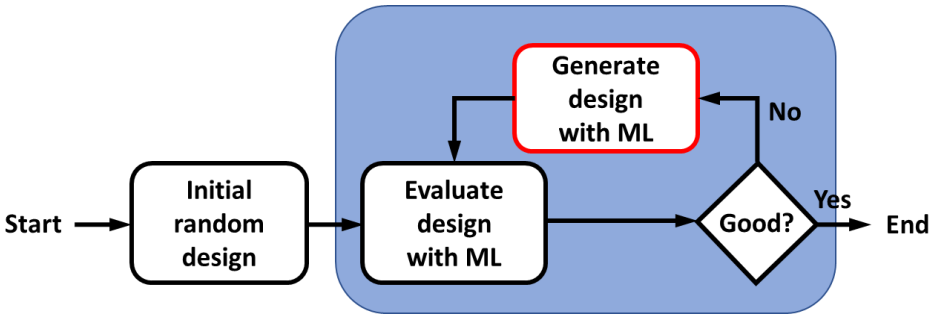

Figure 5. 1. Machine Learning Based Optimization Algorithm

Next, machine learning could quickly provide purpose-built nanofluids. Nanofluid is a fluid dispersed by adding nanoparticles with great thermal conductivity to the fluid. It can provide greater thermal conductivity than pure fluid. However, since the thermal conductivity of nanofluids does not always proportionally increase with the concentration of nanoparticles, the concentration of nanoparticles with the greatest thermal conductivity is important. Moreover, this concentration varies depending on the material and shape of the nanoparticles, and it can only be found through repeated experiments. Therefore, if we use machine learning, we will be able to quickly find the shape and concentration of nanoparticles with the greatest thermal conductivity.

Lastly, machine learning will make it possible to infer the boundary conditions inversely from the temperature distributions. If the temperature distributions are continuously monitored in precision production processes such as those with semiconductors, pharmaceuticals, chemicals, and food production, when an abnormal

temperature occurs, it can help to quickly find and repair the part where temperature control has failed.

REFERENCES

Andrzejczyk, Rafal, and Tomasz Muszynski. 2017. “Thermodynamic and Geometrical Characteristics of Mixed Convection Heat Transfer in the Shell and Coil Tube Heat Exchanger with Baffles.” *Applied Thermal Engineering* 121 (5): 115–25.

Arjovsky, Martin, Soumith Chintala, and Léon Bottou. 2017. “Wasserstein GAN.” *ArXiv:1701.07875*, January.

Aycock, Kenneth I., Prasanna Hariharan, and Brent A. Craven. 2017. “Particle Image Velocimetry Measurements in an Anatomical Vascular Model Fabricated Using Inkjet 3D Printing.” *Experiments in Fluids* 58 (11): 1–8.

Banna, M., L. Pietri, and B. Zeghmati. 2004. “Turbulent Mixed Convection of Heat and Water Vapor Transfers in a Two-Dimensional Vegetation Canopy.” *Heat and Mass Transfer/Waerme- Und Stoffuebertragung* 40 (10): 757–68.

Beam, Andrew L., and Isaac S. Kohane. 2018. “Big Data and Machine Learning in Health Care.” *Journal of the American Medical Association* 319 (13): 1317–18.

Bell, Jason. 2015. *Machine Learning : Hands-on for Developers and Technical Professionals*. Indianapolis, Indiana: Wiley.

Bergles, A. E., and R. R. Simonds. 1971. “Combined Forced and Free Convection for Laminar Flow in Horizontal Tubes with Uniform Heat Flux.” *International Journal of Heat and Mass Transfer* 14 (12): 1989–2000.

Bowles, Christopher, Roger Gunn, Alexander Hammers, and Daniel Rueckert. 2018. “GANsfer Learning: Combining Labelled and Unlabelled Data for GAN Based Data Augmentation.” *ArXiv:1811.10669*.

Breiman, Leo. 2001. “Random Forests.” *Machine Learning* 45 (1): 5–32.

Brenner, M. P., J. D. Eldredge, and J. B. Freund. 2019. “Perspective on Machine Learning for Advancing Fluid Mechanics.” *Physical Review Fluids* 4 (10): 100501.

Brunton, Steven L., Bernd R. Noack, and Petros Koumoutsakos. 2020. “Machine Learning for Fluid Mechanics.” *Annual Review of Fluid Mechanics* 52 (1): 477–508.

Cai, Shengze, Zhicheng Wang, Sifan Wang, Paris Perdikaris, and George Em Karniadakis. 2021. “Physics-Informed Neural Networks for Heat Transfer Problems.” *Journal of Heat Transfer* 143 (6): 060801.

Cheng, Jun, Wei Huang, Shuangliang Cao, Ru Yang, Wei Yang, Zhaoqiang Yun, Zhijian Wang, and Qianjin Feng. 2015. “Enhanced Performance of Brain Tumor

Classification via Tumor Region Augmentation and Partition." Edited by Daoqiang Zhang. *PLOS ONE* 10 (10): e0140381.

Choi, Yunjey, Minje Choi, Munyoung Kim, Jung Woo Ha, Sunghun Kim, and Jaegul Choo. 2017. "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation." In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 8789–97. IEEE Computer Society.

Cirillo, Marco Domenico, David Abramian, and Anders Eklund. 2020. "Vox2Vox: 3D-GAN for Brain Tumour Segmentation." *ArXiv:2003.13653*, March.

Dai, Jifeng, Yang Lu, and Ying Nian Wu. 2014. "Generative Modeling of Convolutional Neural Networks." *ArXiv:1412.6296*, December.

Deng, Li, and Dong Yu. 2013. "Deep Learning: Methods and Applications." *Foundations and Trends in Signal Processing*. Now Publishers Inc.

Du, Meng, Hongyi Yin, Xiaoyan Chen, and Xinqiang Wang. 2018. "Oil-in-Water Two-Phase Flow Pattern Identification from Experimental Snapshots Using Convolutional Neural Network." *IEEE Access* 7: 6219–25.

Edalatifar, Mohammad, Mohammad Bagher Tavakoli, Mohammad Ghalambaz, and Farbod Setoudeh. 2020. "Using Deep Learning to Learn Physics of Conduction Heat Transfer." *Journal of Thermal Analysis and Calorimetry 2020 146:3* 146 (3): 1435–52.

Edalatpour, Mojtaba, and Juan P. Solano. 2017. "Thermal-Hydraulic Characteristics and Exergy Performance in Tube-on-Sheet Flat Plate Solar Collectors: Effects of Nanofluids and Mixed Convection." *International Journal of Thermal Sciences* 118 (August): 397–409.

Erickson, Bradley J., Panagiotis Korfiatis, Zeynettin Akkus, and Timothy L. Kline. 2017. "Machine Learning for Medical Imaging." *Radiographics* 37 (2): 505–15.

Farimani, Amir Barati, Joseph Gomes, and Vijay S. Pande. 2017. "Deep Learning the Physics of Transport Phenomena." *ArXiv:1709.02432*, September.

ziger, Joel H., Milovan Perić, and Robert L. Street. 2020. *Computational Methods for Fluid Dynamics*. *Computational Methods for Fluid Dynamics*. Springer International Publishing.

Frid-Adar, Maayan, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. 2018. "GAN-Based Synthetic Medical Image Augmentation for Increased CNN Performance in Liver Lesion Classification." *Neurocomputing* 321

(December): 321–31.

Gawehn, Erik, Jan A. Hiss, and Gisbert Schneider. 2016. “Deep Learning in Drug Discovery.” *Molecular Informatics* 35 (1): 3–14.

Ghassemi, Navid, Afshin Shoeibi, and Modjtaba Rouhani. 2020. “Deep Neural Network with Generative Adversarial Networks Pre-Training for Brain Tumor Classification Based on MR Images.” *Biomedical Signal Processing and Control* 57 (March): 101678.

Ghorbani, N., H. Taherian, M. Gorji, and H. Mirgolbabaei. 2010. “Experimental Study of Mixed Convection Heat Transfer in Vertical Helically Coiled Tube Heat Exchangers.” *Experimental Thermal and Fluid Science* 34 (7): 900–905.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. “Generative Adversarial Nets.”

Gulrajani, Ishaan, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. 2017. “Improved Training of Wasserstein GANs.” *Advances in Neural Information Processing Systems*. Vol. 30.

Guzmán, A. M., and C. H. Amon. 1994. “Transition to Chaos in Converging-Diverging Channel Flows: Ruelle-Takens-Newhouse Scenario.” *Physics of Fluids* 6 (6): 1994–2002.

Ioffe, Sergey, and Christian Szegedy. 2015. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.” In *Proceedings of the 32nd International Conference on Machine Learning*, 448–56.

Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, Alexei A Efros, and Berkeley Ai Research. 2016. “Image-to-Image Translation with Conditional Adversarial Networks.” *ArXiv:1611.07004*.

Jiang, Changlin, and Amir Barati Farimani. 2020. “Deep Learning Convective Flow Using Conditional Generative Adversarial Networks.” *ArXiv:2005.06422*, May.

Joss, Lisa, and Erich A. Müller. 2019. “Machine Learning for Fluid Property Correlations: Classroom Examples with MATLAB.” *Journal of Chemical Education* 96 (4): 697–703.

Kang, Munku, and Beomjin Kwon. 2022. “Deep Learning of Forced Convection Heat Transfer.” *Journal of Heat Transfer* 144 (2): 021801.

Kang, Seokho, and Kyunghyun Cho. 2019. “Conditional Molecular Design with Deep

Generative Models." *Journal of Chemical Information and Modeling* 59 (1): 43–52.

Karras, Tero, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017. "Progressive Growing of GANs for Improved Quality, Stability, and Variation." *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, October.

Karras, Tero, Samuli Laine, and Timo Aila. 2018. "A Style-Based Generator Architecture for Generative Adversarial Networks." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43 (12): 4217–28.

Karras, Tero, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2019. "Analyzing and Improving the Image Quality of StyleGAN." *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, December, 8107–16.

Kim, Junhyuk, and Changhoon Lee. 2020. "Prediction of Turbulent Heat Transfer Using Convolutional Neural Networks." *Journal of Fluid Mechanics* 882 (A18).

Kingma, Diederik P., and Jimmy Lei Ba. 2014. "Adam: A Method for Stochastic Optimization." *ArXiv:1412.6980*.

Kingma, Diederik P., and Max Welling. 2013. "Auto-Encoding Variational Bayes." *ArXiv:1312.6114*.

Kiyasseh, Dani, Girmaw Abebe Tadesse, Le Nguyen Thanh Nhan, Le Van Tan, Louise Thwaites, Tingting Zhu, and David Clifton. 2020. "PlethAugment: GAN-Based PPG Augmentation for Medical Diagnosis in Low-Resource Settings." *IEEE Journal of Biomedical and Health Informatics* 24 (11): 3226–35.

Kreitzer, Paul J., Michael Hanchak, and Larry Byrd. 2013. "Flow Regime Identification of Horizontal Two Phase Refrigerant R-134a Flow Using Neural Networks." In *ASME International Mechanical Engineering Congress and Exposition, Proceedings (IMECE)*. Vol. 7 B. American Society of Mechanical Engineers (ASME).

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. 2017. "ImageNet Classification with Deep Convolutional Neural Networks." *Communications of the ACM* 60 (6): 84–90.

Kumari, M. 2001. "Variable Viscosity Effects on Free and Mixed Convection Boundary-Layer Flow from a Horizontal Surface in a Saturated Porous Medium - Variable Heat Flux." *Mechanics Research Communications* 28 (3): 339–48.

Kwon, Beomjin, Thomas Foulkes, Tianyu Yang, Nenad Miljkovic, and William P. King.

2020. “Air Jet Impingement Cooling of Electronic Devices Using Additively Manufactured Nozzles.” *IEEE Transactions on Components, Packaging and Manufacturing Technology* 10 (2): 220–29.

Kwon, Beomjin, Leon Liebenberg, Anthony M. Jacobi, and William P. King. 2019. “Heat Transfer Enhancement of Internal Laminar Flows Using Additively Manufactured Static Mixers.” *International Journal of Heat and Mass Transfer* 137 (July): 292–300.

Lecun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. “Deep Learning.” *Nature* 521 (7553): 436–44.

Lee, Sangseung, and Donghyun You. 2019. “Data-Driven Prediction of Unsteady Flow over a Circular Cylinder Using Deep Learning.” *Journal of Fluid Mechanics* 879 (November): 217–54.

Maleki, Farhad, Nikesh Muthukrishnan, Katie Ovens, Caroline Reinhold, and Reza Forghani. 2020. “Machine Learning Algorithm Validation From Essentials to Advanced Applications and Implications for Regulatory Certification and Deployment.” *Neuroimaging Clinics* 30 (4): 433–45.

McClure, Emma R., and Van P. Carey. 2021. “Genetic Algorithm and Deep Learning to Explore Parametric Trends in Nucleate Boiling Heat Transfer Data.” *Journal of Heat Transfer* 143 (12): 121602.

McComas, S. T., and E. R.G. Eckert. 1966. “Combined Free and Forced Convection in a Horizontal Circular Tube.” *Journal of Heat Transfer* 88 (2): 147–52.

Mikolov, Tomáš, Anoop Deoras, Daniel Povey, Lukáš Burget, and Jan Černocký. 2011. “Strategies for Training Large Scale Neural Network Language Models.” In *Proceedings of the IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, 196–201. Waikoloa, HI, December 11-15, 2011, INSPEC Accession No. 12577641.

Mirza, Mehdi, and Simon Osindero. 2014. “Conditional Generative Adversarial Nets.” *ArXiv:1411.1784*.

Nishimura, T., Y. N. Bian, Y. Matsumoto, and K. Kunitsugu. 2003. “Fluid Flow and Mass Transfer Characteristics in a Sinusoidal Wavy-Walled Tube at Moderate Reynolds Numbers for Steady Flow.” *Heat and Mass Transfer* 39 (3): 239–48.

Nishimura, Tatsuo, Yoshiji Ohori, Yoshihiko Kajimoto, and Yuji Kawamura. 1985. “Mass Transfer Characteristics in a Channel with Symmetric Wavy Wall for Steady Flow.” *Journal of Chemical Engineering of Japan* 18 (6): 550–55.

Nishimura, Tatsuo, Yoshiji Ohori, and Yuji Kawamura. 1984. “Flow Characteristics in a Channel with Symmetric Wavy Wall for Steady Flow.” *Journal of Chemical Engineering of Japan* 17 (5): 466–71.

Osborne, D. G., and F. P. Incropera. 1985. “Laminar, Mixed Convection Heat Transfer for Flow between Horizontal Parallel Plates with Asymmetric Heating.” *International Journal of Heat and Mass Transfer* 28 (1): 207–17.

Ostrach, S., and Y. Kamotani. 1975. “Heat Transfer Augmentation in Laminar Fully Developed Channel Flow by Means of Heating from Below.” *Journal of Heat Transfer* 97 (2): 220–25.

Oztop, Hakan F., and Khaled Al-Salem. 2012. “A Review on Entropy Generation in Natural and Mixed Convection Heat Transfer for Energy Systems.” *Renewable and Sustainable Energy Reviews* 16 (1): 911–20.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, et al. 2011. “Scikit-Learn: Machine Learning in Python.” *Journal of Machine Learning Research* 12: 2825–30.

Radford, Alec, Luke Metz, and Soumith Chintala. 2016. “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks.” *ArXiv:1511.06434*, November.

Raissi, Maziar, Alireza Yazdani, and George Em Karniadakis. 2020. “Hidden Fluid Mechanics: Learning Velocity and Pressure Fields from Flow Visualizations.” *Science* 367 (6481): 1026–30.

Reynolds, Osborne. 1883. “An Experimental Investigation of the Circumstances Which Determine Whether the Motion of Water Shall Be Direct or Sinuous, and of the Law of Resistance in Parallel Channels.” *Philosophical Transactions of the Royal Society of London* 174 (December): 935–82.

Romanov, Valentin, Raheel Samuel, Marzieh Chaharlang, Alexander R. Jafek, Adam Frost, and Bruce K. Gale. 2018. “FDM 3D Printing of High-Pressure, Heat-Resistant, Transparent Microfluidic Devices.” *Analytical Chemistry* 90 (17): 10450–56.

Rosa, E. S., R. M. Salgado, T. Ohishi, and N. Mastelari. 2010. “Performance Comparison of Artificial Neural Networks and Expert Systems Applied to Flow Pattern Identification in Vertical Ascendant Gas-Liquid Flows.” *International Journal of Multiphase Flow* 36 (9): 738–54.

Selmi, Mohamed, Mohammed J. Al-Khawaja, and Abdulhamid Marafia. 2008. “Validation of CFD Simulation for Flat Plate Solar Energy Collector.” *Renewable*

Energy 33 (3): 383–87.

Shallan, Aliaa I., Petr Smejkal, Monika Corban, Rosanne M. Guijt, and Michael C. Breadmore. 2014. “Cost-Effective Three-Dimensional Printing of Visibly Transparent Microchips within Minutes.” *Analytical Chemistry* 86 (6): 3124–30.

Sharma, Rishi, Amir Barati, Farimani Bioengineering, Joe Gomes Bioengineering, Peter Eastman Bioengineering, and Vijay Pande Bioengineering. 2018. “Weakly-Supervised Deep Learning of Heat Transport via Physics Informed Loss.” *ArXiv:1807.11374*.

Smart, William D., and Leslie Pack Kaelbling. 2002. “Effective Reinforcement Learning for Mobile Robots.” *Proceedings-IEEE International Conference on Robotics and Automation* 4: 3404–10.

Smith, Anthony J. 2004. “Mixed Convection and Density-Dependent Seawater Circulation in Coastal Aquifers.” *Water Resources Research* 40 (8): 8309.

Sundar, Sreenath, Manjunath C. Rajagopal, Hanyang Zhao, Gowtham Kuntumalla, Yuquan Meng, Ho Chan Chang, Chenhui Shao, et al. 2020. “Fouling Modeling and Prediction Approach for Heat Exchangers Using Deep Learning.” *International Journal of Heat and Mass Transfer* 159 (October): 120112.

Tien Bui, Dieu, Nhat Duc Hoang, Francisco Martínez-Álvarez, Phuong Thao Thi Ngo, Pham Viet Hoa, Tien Dat Pham, Pijush Samui, and Romulus Costache. 2020. “A Novel Deep Learning Neural Network Approach for Predicting Flash Flood Susceptibility: A Case Study at a High Frequency Tropical Storm Area.” *Science of The Total Environment* 701 (January): 134413.

Valter, Per, Peter Lindgren, and Ramjee Prasad. 2018. “Advanced Business Model Innovation Supported by Artificial Intelligence and Deep Learning.” *Wireless Personal Communications* 100 (1): 97–111.

Wang, G., and S. P. Vanka. 1995. “Convective Heat Transfer in Periodic Wavy Passages.” *International Journal of Heat and Mass Transfer* 38 (17): 3219–30.

Yang, Li, Wei Dai, Yu Rao, and Minking K. Chyu. 2019. “Optimization of the Hole Distribution of an Effusively Cooled Surface Facing Non-Uniform Incoming Temperature Using Deep Learning Approaches.” *International Journal of Heat and Mass Transfer* 145 (December): 118749.

Yasuo, Mori, and Uchida Yutaka. 1966. “Forced Convective Heat Transfer between Horizontal Flat Plates.” *International Journal of Heat and Mass Transfer* 9 (8): 803–17.

Zhu, Jun-Yan, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks Monet Photos.” *ArXiv:1703.10593,* 2223–32.

Zobeiry, Navid, and Keith D. Humfeld. 2021. “A Physics-Informed Machine Learning Approach for Solving Heat Transfer Equation in Advanced Manufacturing and Engineering Applications.” *Engineering Applications of Artificial Intelligence* 101 (May): 104232.