

Flow Adversarial Networks: Flowrate Prediction for Gas–Liquid Multiphase Flows Across Different Domains

Delin Hu^{ID}, Jinku Li^{ID}, Yinyan Liu^{ID}, and Yi Li^{ID}

Abstract—The solution of how to accurately and timely predict the flowrate of gas–liquid mixtures is the key to help petroleum and other related industries to reduce costs, improve efficiency, and optimize management. Although numerous studies have been carried out over the past decades, the problem is still significantly challenging due to the complexity of multiphase flows. This paper attempts to seek new possibilities for multiphase flow measurement and novel application scenarios for state-of-the-art machine learning (ML) techniques. Convolutional neural networks (CNNs) are applied to predict the flowrate of multiphase flows for the first time and can achieve promising performance. In addition, considering the difference between data distributions of training and testing samples and its negative impact on prediction accuracy of the CNN models on testing samples, we propose flow adversarial networks (FANs) that can distill both domain-invariant and flowrate-discriminative features from the raw input. The method is evaluated on dynamic experimental data of different multiphase flows on different flow conditions and operating environments. The experimental results demonstrate that FANs can effectively prevent the accuracy degradation caused by the gap between training and testing samples and have better performance than state-of-the-art approaches in the flowrate prediction field.

Index Terms—Convolutional neural network (CNN), domain adaptation, domain adversarial network, flowrate prediction, multiphase flow, regression, transfer learning, venturi tube.

NOMENCLATURE

$E(\cdot; \theta_e)$	feature extractor with parameters θ_e
$P(\cdot; \theta_p)$	label predictor with parameters θ_p
$D(\cdot; \theta_d)$	domain discriminator with parameters θ_d
p	time series of multi-channel pressure signals
p_s	time series of multi-channel pressure signals in source domain
p_t	time series of multi-channel pressure signals in target domain

Manuscript received July 6, 2018; revised October 31, 2018 and February 28, 2019; accepted March 11, 2019. This work was supported by the National Natural Science Foundation of China under Grant 61571252. (Corresponding author: Yi Li.)

D. Hu, Y. Liu, and Y. Li are with the Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China (e-mail: hdl16@mails.tsinghua.edu.cn; liyi@sz.tsinghua.edu.cn).

J. Li is with the Department of Automation, Tsinghua University, Beijing 100084, China.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNNLS.2019.2905082

\dot{m}_l	reference of average liquid mass flowrate
\dot{m}_{l_s}	reference of average liquid mass flowrate in source domain
\dot{m}_{l_t}	reference of average liquid mass flowrate in target domain
$\hat{\dot{m}}_l$	prediction of average liquid mass flowrate
$\dot{m}_{l_s}^m$	average liquid mass flowrate of multiphase flow in source domain
$\dot{m}_{l_s}^s$	average liquid mass flowrate of single-phase flow in source domain
$S(p, \dot{m}_l)$	joint distribution of p and \dot{m}_l in source domain
$T(p, \dot{m}_l)$	joint distribution of p and \dot{m}_l in target domain
$S(p)$	marginal distribution of p in source domain
$T(p)$	marginal distribution of p in target domain
f	feature vector
f_s	feature vector in source domain
f_t	feature vector in target domain
$\mathcal{F}_s(f)$	marginal distribution of f in source domain
$\mathcal{F}_t(f)$	marginal distribution of f in target domain
d	reference of domain label
\hat{d}	prediction of domain label

I. INTRODUCTION

Gas–liquid multiphase flows are commonly observed in various chemical reactors and related equipment of oil and gas fields. Accurately metering parameters of gas–liquid mixtures is quite important for these industries in terms of efficiency improvement, cost reduction, and management optimization. The traditional method needs to separate the components of the mixtures using separators at first, and then the flowrate of each phase can be measured by conventional single-phase flowmeters, respectively. However, there are numerous shortages and restrictions in practice. For instance, it cannot provide prediction values in time due to the fact that the process of multiphase flow separation usually

requires several hours or even longer. Moreover, the separators are expensive and take a lot of space, which further leads to a significant increment in costs, especially for offshore wells. Petroleum companies are therefore motivated to invest a great deal of funds in developing more advanced techniques for multiphase flow measurement, i.e., multiphase flow meters (MFM), which can estimate parameters of multiphase flows directly without separators [1]–[3].

Differential pressure devices including venturi tube and orifice plate flowmeters are widely used in single-phase flow measurement, and many studies about the application of these devices to predict the flowrate for multiphase flows have been carried out during the past decades. One typical method to measure the flowrate for the gas phase in gas–liquid mixtures is to compute the flowrate on the basis of differential pressure measurements and single-phase flowmeter principles at first, then find a correlation model to correct the error induced by the presence of the liquid phase. The same procedure can also be applied to meter the flowrate for the liquid phase. However, the establishment of correlation models normally requires simplifications for the dynamic states of multiphase flows due to the complexity and the lack of relevant fluid knowledge. Therefore, many correlation models are only applicable in the specific flow conditions and operating environments, which are consistent with the simplifications. For example, Lin [4], Xu *et al.* [5], Yuan *et al.* [6], and Xu *et al.* [7] proposed different semiempirical measurement models for wet gas, which is defined as the gas–liquid mixtures whose gas volume fraction (GVF) is equal to 95% or higher [8]. These models are invalid or cause great errors when metering gas–liquid multiphase flows under other flow conditions. The approach to broadening the application range and enhance measurement accuracy of this type of flowmeter has still been challenging and attracts the attention of numerous researchers all over the world.

Recently, as the development of machine learning (ML) and its successful applications in many other fields, more and more studies aiming to combine ML and multiphase flow measurement have been conducted. Timung and Mandal [9] employed probabilistic neural networks (NNs) to identify the flow pattern of gas–liquid mixtures through circular microchannel. Shaban and Tavoularis [10] applied elastic maps to analyze the probability density function (PDF) of differential pressure signals produced by the vertical upward water–air pipe flow for flow regime recognition. Wu *et al.* [11] designed an intelligent system based on backpropagation NNs (BPNNs), which can predict the flow pattern of the oil–water–gas multiphase flow according to characteristic vectors extracted from denoised differential pressure signals with fractal theory. Sun and Zhang [12] employed BPNNs to discern flow regimes through vortex flowmeter signals. In addition to flow pattern identification, ML techniques are also applied to measure many other parameters of multiphase flows. For instance, Shaban and Tavoularis [13] developed a new algorithm for flowrate prediction. In their method, independent component analysis and principal component analysis were used to reduce the dimensionality of feature vectors at first, and then those vectors were used as the input of

different BPNNs to predict flowrates in different flow patterns. Xu *et al.* [14] established the relation between signals from a throat-extended venturi tube and flowrates of gas–liquid mixtures via BPNNs and support vector machines (SVMs). Fan and Yan [15] extracted five characteristic parameters from conductance signals as the input of a BPNN model to estimate the flowrate of the water–air slug flow. Azizi *et al.* employed BPNNs to determine void fractions in [16] and water-in-liquid ratios in [17].

Although these methods have good performance on sensor data from dynamic multiphase flow experiments, it is still difficult to successfully apply them to actual industrial processes. The main reason is the difference between the data distributions of samples obtained on experimental platforms and samples in real industrial production. Reducing and eliminating the negative impact caused by the difference are the key to generalize the prediction models fitted by experimental data to real industrial applications.

On the one hand, the difference is partially contributed by different fluid mediums between experiments and real production. The mediums of the gas phases in most of the dynamic multiphase flow experiments are air. However, the mediums of the gas phases of the outputs from different oil or gas wells are not necessarily air. They could be natural gas, methane, carbon dioxide, and so on. Similarly, there is the same problem in the liquid phase. In terms of safety and economy, it is impossible that the components of multiphase flows in the laboratory are exactly the same as the outputs of oil or gas wells. In this paper, two strategies are presented to decrease the negative effect of the discrepancy, i.e., selecting an appropriate output objective for flowrate prediction models and flow adversarial networks (FANs).

On the other hand, improper experimental operations also make contributions to data difference. In the related work mentioned before, NNs were trained only with samples generated by stable single-phase flows, i.e., the openings of control valves of single-phase flows are kept as constants during the data collection. However, there is no guarantee that multiphase flows in real industrial processes are similar to multiphase flows mixed by stable single-phase flows in experiments. Aiming at this problem, we design a more reasonable experiment scheme and data acquisition method, so as to obtain training samples closer to practical applications.

This paper is highly motivated by the strong industrial demand of petroleum and other related companies. We expect to employ advanced ML techniques to help these companies to improve efficiency and reduce costs. The summary of our contributions is listed as follows.

- 1) We describe a novel application scenario (i.e., flowrate prediction for multiphase flows) for different ML techniques, including CNNs and domain adaptation.
- 2) We propose a domain adaptation method, FANs, to achieve flowrate prediction for multiphase flows cross different domains. This can effectively mitigate the negative impact on prediction accuracy caused by the gap between training and testing samples.
- 3) We conduct dynamic multiphase flow experiments on a semiindustrial experimental platform to collect data

of different multiphase flows in a large variety of flow conditions and operating environments.

- 4) We evaluate the effectiveness of CNNs, domain adversarial networks (DANs) and our method FANS using the multiphase flow experimental data set and observe that FANS can outperform the other two methods.

In the rest of this paper, the concepts of the ML techniques we employed are introduced in Section II, including convolutional NNs (CNNs), generative adversarial networks (GANs), and DANs. The details about the dynamic multiphase flow experiments and data acquisition method are discussed in Section III. The FANS are proposed in Section IV. Section V shows experimental results of different networks on different experimental data sets. The summary of this paper and future direction is presented in Section VI.

II. BACKGROUNDS OF RELATED MACHINE LEARNING TECHNIQUES

A. Convolutional Neural Networks

CNNs are a powerful tool to process data that comes in the form of multiple arrays [18]. The general structure of CNNs mainly includes two multilayer networks acting as different roles, namely, a feature extractor and a label predictor. The input of a CNN first needs to pass through the feature extractor, and then the feature maps representing the input can be obtained. The feature extractor usually consists of multiple convolutional layers and pooling layers. A convolutional layer is a set of filters that are actually matrixes or tensors whose elements are weights. Each result of the discrete convolution between a filter and each patch (local receptive field [19]) of the input of the layer is passed through a nonlinear activation function, and then all the consequences are organized as one feature map. It means that all units in one feature map share the same weights. The operation repeated on different filters can generate different feature maps. The combination of all the feature maps is the output of the convolutional layer. Fig. 1(a) shows a simple example of the convolutional operation.

The role of pooling layers, which usually follow convolutional layers, is to merge several neighboring features into one through typically taking their max or average [20]. This operation can reduce the dimension of feature representations and the sensitivity to small shifts and distortions [18]. Fig. 1(b) shows a simple example of the maxpooling operation. Multiple convolutional layers and pooling layers constitute the feature extractor, which can capture salient descriptions of the raw input.

The structure of the label predictor is much easier. It usually contains several fully connected layers and can complete classification and regression tasks according to the feature maps produced by the feature extractor.

Over the past decades, CNNs have achieved great success in many domains including computer vision, natural language processing, and so forth. A part of excellent architectures of CNNs can be found in [21]–[24].

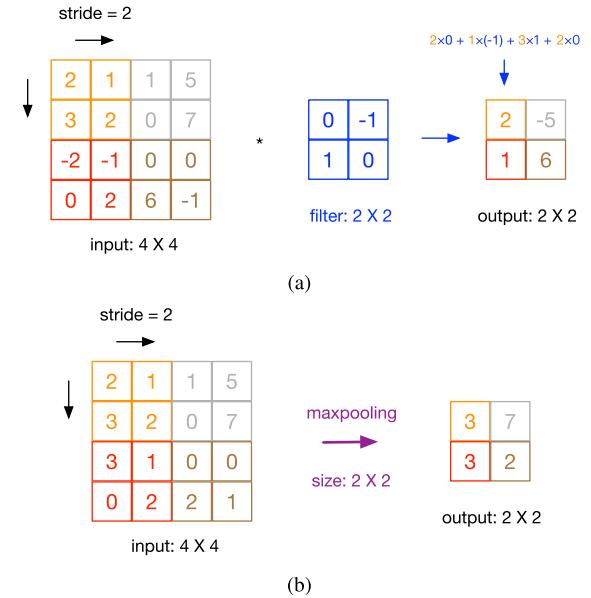


Fig. 1. Examples of convolutional and maxpooling operations. (a) Convolutional operation. (b) Maxpooling operation.

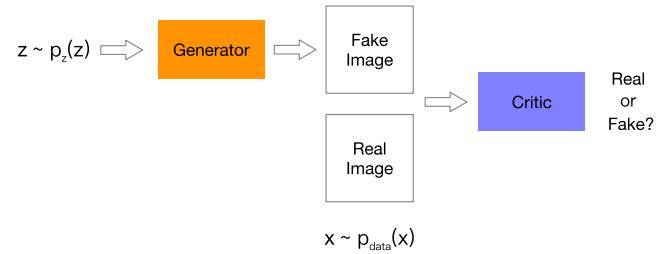


Fig. 2. General framework of GANs.

B. Generative Adversarial Networks

GANs proposed by Goodfellow *et al.* [25] are able to generate high quality and realistic images. The framework of GANs mainly contains two different parts (see Fig. 2), i.e., a generator G and a critic C . G can draw pictures according to the input vectors sampled from a random distribution, $z \sim p_z(z)$. C is employed to estimate the distance between the data distributions of pictures generated by G and the real ones. The approach to training C is actually the same as training a binary classifier. The class labels of real images can be defined as 1, and the labels for synthetic images can be defined as 0. Then, the parameters of C are iteratively updated by backpropagation to reduce the loss (cross entropy). When training G , the parameters of C are fixed. Then the parameters of G are adjusted to maximize the probability that C misclassifies the synthetic images as real ones. This step can be finished by backpropagation as well. As the two training processes are repeated, the pictures drawn by G will be more and more realistic. In fact, C and G play a two-player minimax game with value function $V(G, C)$ [25]

$$\min_G \max_C V(G, C) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log C(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - C(G(z)))] \quad (1)$$

where x is the real image and $P_{\text{data}}(x)$ is the distribution of x .

There is no doubt that GANs make great contributions to image generation and other related fields. However, there are also some weaknesses in the original framework of GANs, such as instability in training and generating numerous repetitive images [26]. In order to solve the problems, many related studies have been carried out and published recently, such as Wasserstein GANs (WGANs) [27], WGANs with gradient penalty (WGANs-GP) [28], deep convolutional GANs (DCGANs) [29], boundary equilibrium GAN (BEGAN) [30], loss sensitive GANs (LSGANs) [31], and so on.

C. Domain Adversarial Networks

In many ML applications, there is a common issue that the data distributions of training samples (the source domain) and samples encountered in practice (the target domain) are related but not exactly the same. The technique to lessen the negative impact of the problem on the performance of the learning model in the target domain is known as domain adaptation [32]. DANs are one of this kind of techniques and try to complete the domain adaptation task with the assistance of an adversarial architecture [33].

The general framework of DANs mainly includes three different multilayer networks, i.e., a feature extractor E , a label predictor P , and a domain discriminator D . E is able to extract features representing the raw input. D can determine whether the input comes from the source domain or target domain according to its features extracted by E . P can complete prediction tasks (classification or regression) on the basis of the same features.

The labeled samples in the source domain and the unlabeled samples in the target domain are both needed during training. D is trained as a binary classifier, and E attempts to extract domain-invariant features through fooling D . The roles of E and D are quite similar to the roles of G and C in GANs. A gradient reverse layer is employed to finish the adversarial training between D and E in DANs. In addition, the combination of E and P is expected to establish the relation between the raw input and the corresponding label. Therefore, parameters of E and P should be updated to minimize the difference between the reference and prediction values of labels as well [33], [34].

In DANs, E actually attempts to distill domain-invariant features through maximizing the probability that D makes mistakes. However, it does not mean distributions of source and target domain features will have high similarity. In our method, E is also expected to capture domain-invariant features for the better performance of prediction models on target domain samples. However, this goal is not achieved by directly maximizing the loss of D . Instead of that, E is required to generate source and target features, which can gain close scores from D , and more details are discussed in Section IV.

In addition to DANs, there are also other studies concerning domain adaptation with adversarial architectures. For instance, instead of finding domain-invariant features, adversarial discriminative domain adaptation (ADDA) [35] manages to generate target domain features that are similar to features generated by source domain samples. Then the source

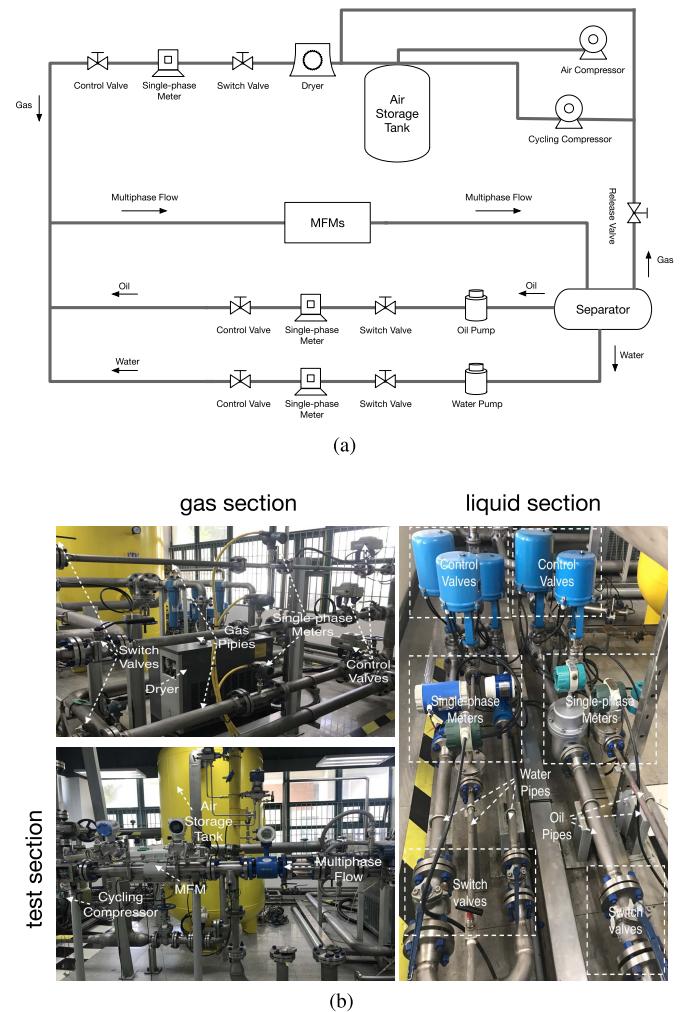


Fig. 3. Dynamic multiphase flow experimental facility. (a) Schematic. (b) Actual devices.

predictor can be used on target domain features directly to complete classification and regression tasks.

III. EXPERIMENTAL FACILITY AND DATA ACQUISITION

A. Experimental Facility

The dynamic experiments were conducted on multiphase flow test facility (see Fig. 3). The operating procedure of the facility is as follows.

- 1) In order to build an experimental environment as close as possible to the industrial environment (sealed pipes with pressure), the release valve should be closed and the air compressor should be activated to provide a reasonable static pressure for the whole system.
- 2) The oil and water single-phase flows are drawn from the separator by oil and water pumps, respectively, and the gas single-phase flow is supplied by the cycling compressor.
- 3) The parameters of all the single-phase flows are recorded by single-phase meters.
- 4) Changing the opening of the control valve can adjust the flowrate of the corresponding single-phase flow.

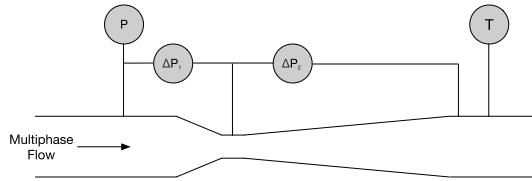


Fig. 4. Structure of the venturi tube.

- 5) The single-phase flows are mixed to form the multiphase flow, and then the mixture passes through the MFM under test.
- 6) The multiphase flow will eventually enter the separator to be separated into single-phase flows and reused.

A venturi tube (see Fig. 4) with four different sensors was mounted at the place of the “MFM” shown in Fig. 3(a) to collect information about the multiphase flow passing the horizontal pipe. The dynamic pressure P , the differential pressure of the convergent section ΔP_1 , the differential pressure of the divergent section ΔP_2 , and the temperature T were measured and recorded. The range of measurements of ΔP_1 and ΔP_2 is from 0 to 62.2 kPa. The device was manufactured in accordance with ISO standards, and more details concerning the structure and parameters are shown in [36].

B. Data Acquisition

The standard values of the instantaneous parameters for the multiphase flow are not available during experiments, because there are no accurate and reliable MFMs on the market. The only available information is the data recorded by the single-phase meters. Considering within a period S , the total volume (v_l^s) of the liquid single-phase flow passing the single-phase meter is quite close to the total volume (v_l^m) of the liquid phase in the multiphase flow passing the venturi tube. The problem should be converted to predict the average flowrate of the multiphase flow over the period S , rather than the instantaneous flowrate. The upper bound of the absolute value of the absolute error (ε_v) between the average flowrates for the liquid phase and the liquid single-phase flow is shown as follows:

$$|\varepsilon_v| = \left| \frac{v_l^s}{S} - \frac{v_l^m}{S} \right| \leq \frac{\pi r^2 l}{S} \quad (2)$$

where r is the radius for the cross section of the pipe and l is the length of the experimental pipe between the single-phase meter and the venturi tube.

Note that, in fact, only in some extreme cases, which are not likely to happen in practice, $|\varepsilon_v|$ is equal to $\pi r^2 l / S$. In actual multiphase flow experiments, $|\varepsilon_v|$ should be far less than $\pi r^2 l / S$. $\pi r^2 l$ can be regarded as a constant, so the upper bound of $|\varepsilon_v|$ decreases as S increases. Therefore, we can use v_l^s to approximate v_l^m when S is large enough. In this paper, S is selected as 5 min.

The dynamic experiments were divided into three groups in this paper according to the medium of the liquid phase. One group is the water-air two-phase flow, and the other two groups are the oil-air two-phase flow and the oil-water-air

TABLE I
EXPERIMENTAL CONDITIONS

Medium	Liquid Density (kg/m ³)	Static Pressure (MPa)	Liquid Mass Flowrate (kg/h)	Gas Mass Flowrate (kg/h)
water-air	1000	0.4	0 ~ 6900	0 ~ 350
		0.6	0 ~ 6900	0 ~ 350
oil-air	830	0.4	0 ~ 5200	0 ~ 310
		0.6	0 ~ 5200	0 ~ 310
oil-water-air	830 + 170γ	0.5	0 ~ 6000	0 ~ 60

TABLE II
FORMULAS OF DIFFERENT KINDS OF FLOWRATES

	water-air	oil-air	oil-water-air
mass(\dot{m}_l)	$\frac{v_w^s \rho_w}{S}$	$\frac{v_o^s \rho_o}{S}$	$\frac{v_w^s \rho_w + v_o^s \rho_o}{S}$
momentum(\dot{M}_l)	$\frac{v_w^s \sqrt{\rho_w}}{S}$	$\frac{v_o^s \sqrt{\rho_o}}{S}$	$\frac{v_w^s \sqrt{\rho_w} + v_o^s \sqrt{\rho_o}}{S}$
volume(\dot{V}_l)	$\frac{v_w^s}{S}$	$\frac{v_o^s}{S}$	$\frac{v_w^s + v_o^s}{S}$

three-phase flow. The dynamic multiphase flow experiments were conducted on the facility shown in Fig. 3(b). The details concerning experimental conditions are shown in Table I, where γ is the water-in-liquid ratio ($\gamma = v_{\text{water}}/v_{\text{liquid}}$).

The flowrate of each single-phase flow was changed every 2 min in the two-phase experiments, and every 3 min in the three-phase experiments. The data for the multiphase flows formed by stable single-phase flows and transient single-phase flows should be collected during the experiments. Note that the operation is different from the multiphase flow experiments conducted in previous work. In order to investigate the properties of multiphase flows systematically and comprehensively, such as the mechanism behind different flow patterns, the interaction between different phases and so forth, most of the previous experiments observed and recorded multiphase flow data only after the flowrates of single-phase flows were stable before mixing. However, from the perspective of flowrate prediction, stable single-phase flows are not indispensable. The experiment method in this paper cannot only obtain training samples closer to real industrial production, but also improve the efficiency of data collection.

The time series of multichannel pressure signals (P , ΔP_1 , and ΔP_2), which can characterize the flowrate of the multiphase flow passing the venturi tube, was selected as the input of our prediction model in this paper. The sampling frequencies of sensors are 6 Hz. The length of the period S is 300 s. Consequently, each time series is a 1800×3 matrix. Average flowrates of liquid single-phase flows are calculated as reference labels. Table II lists the computational formulas for different types of flowrates. Fig. 5 shows the examples concerning the time series of multichannel normalized pressure signals (p). All the liquid mass flowrates of the examples shown in Fig. 5 are about 3000 kg/h.

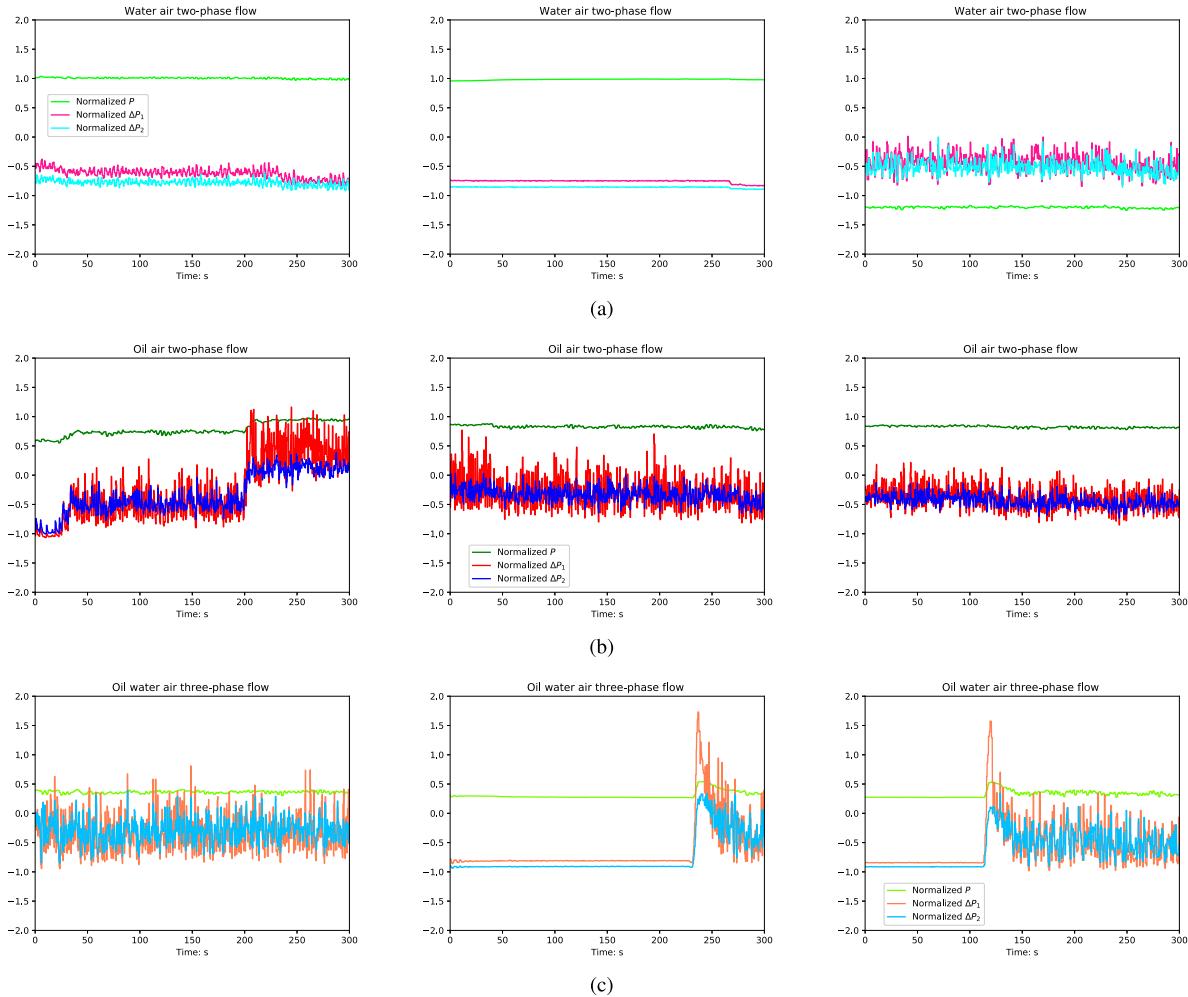


Fig. 5. Examples of the time series of multichannel normalized pressure signals. (a) Three examples generated by the water-air two-phase flow. (b) Three examples from the oil-air two-phase flow. (c) Examples are from the oil-water-air three-phase flow. The average mass flowrate of the liquid phase for each example is approximately 3000 kg/h, but they look pretty different due to the difference among flowrates of gas phases, mediums of liquid phases, flow patterns, and so forth.

The entire experiments took about 100 h.¹ Approximately, 3500 samples of the water-air two-phase flow, 3500 samples of the oil-air two-phase flow, and 2500 samples of the oil-water-air three-phase flow were collected. Partial overlap between different samples is acceptable. For instance, it is possible that the sample A and the sample B are generated by the experimental data from the 300th s to the 600th s and the data from the 550th s to the 850th s, respectively. Although these two samples overlap in the period between the 550th s and the 600th s, they can be treated as two different samples.

IV. FLOW ADVERSARIAL NETWORKS

A. Assumption

We assume that source domain samples $(\mathbf{p}_s, \dot{m}_{l_s})$ are generated from a distribution $\mathcal{S}(\mathbf{p}, \dot{m}_l)$, and target domain samples $(\mathbf{p}_t, \dot{m}_{l_t})$ are generated from another distribution $\mathcal{T}(\mathbf{p}, \dot{m}_l)$. The two distributions are related but not exactly the same. $\mathcal{S}(\mathbf{p})$ and $\mathcal{T}(\mathbf{p})$ are marginal distributions of \mathbf{p}_s and \mathbf{p}_t .

¹Our data are available at <https://pan.baidu.com/s/1xBNqaShAB6SUdPc5vL9JA>

Our goal in this paper is to learn a flowrate prediction model which can accurately predict \dot{m}_{l_t} according to \mathbf{p}_t . The prediction model usually includes two different parts, i.e., a feature extractor $E(\cdot; \theta_e)$ and a label predictor $P(\cdot; \theta_p)$. The input signal \mathbf{p} first needs to pass through the feature extractor $E(\cdot; \theta_e)$ and is mapped to the feature vector f , i.e., $f = E(\mathbf{p}; \theta_e)$. $P(\cdot; \theta_p)$ can predict the label (flowrate) \dot{m}_l according to the feature vector f , i.e., $\hat{\dot{m}}_l = P(f; \theta_p)$. Our objective is to find optimal parameters θ_e^* and θ_p^* , minimizing the error between the real target label \dot{m}_{l_t} and the prediction $\hat{\dot{m}}_{l_t}$

$$(\theta_e^*, \theta_p^*) = \arg \min_{\theta_e, \theta_p} \mathbb{E}_{(\mathbf{p}_t, \dot{m}_{l_t}) \sim \mathcal{T}(\mathbf{p}, \dot{m}_l)} [\dot{m}_{l_t} - P(E(\mathbf{p}_t; \theta_e); \theta_p)]^2. \quad (3)$$

In an ideal case, this goal can be achieved by supervised learning using target domain samples $(\mathbf{p}_t, \dot{m}_{l_t})$. However, in real applications, it is impossible or extremely pricey to collect sufficient reference information to train a deep architecture. In other words, the available information during training is labeled source samples $(\mathbf{p}_s, \dot{m}_{l_s})$, unlabeled target samples \mathbf{p}_t , and domain labels d of samples. d is a binary variable

which indicates the domain of \mathbf{p} . If \mathbf{p} is a source sample, i.e., $\mathbf{p} \sim \mathcal{S}(\mathbf{p})$, $d = 0$. Otherwise, $\mathbf{p} \sim \mathcal{T}(\mathbf{p})$, and $d = 1$. θ_e and θ_p need to be determined by those information. Actually, it can be regarded as a domain adaptation problem. We need to train a prediction model with source samples $(\mathbf{p}_s, \dot{m}_{l_s})$, and explore how to apply that model to the target domain. In this section, we present FANs to address the issue.

B. Formulation

Source domain samples \mathbf{p}_s and target domain samples \mathbf{p}_t can be, respectively, mapped to source domain feature vectors f_s and target domain feature vectors f_t through the feature extractor $E(\mathbf{p}; \theta_e)$. f_s and f_t subject to the source domain feature distribution \mathcal{F}_s and the target domain feature distribution \mathcal{F}_t , respectively, i.e., $f_s \sim \mathcal{F}_s(f)$ and $f_t \sim \mathcal{F}_t(f)$. If $E(\mathbf{p}; \theta_e)$ were merely learned with source domain samples, it would be difficult for $E(\mathbf{p}; \theta_e)$ to capture correct descriptions of target domain samples due to the difference between data distributions of source and target domains. Consequently, there is a gap between $\mathcal{F}_s(f)$ and $\mathcal{F}_t(f)$. This gap will cause a decline in target domain prediction accuracy when the label predictor $P(f; \theta_p)$ determines the flowrate according to f . If we can narrow the difference between $\mathcal{F}_s(f)$ and $\mathcal{F}_t(f)$, the negative impact will definitely be mitigated. Thus, we expect $E(\mathbf{p}; \theta_e)$ to produce both domain-invariant and flowrate-discriminative features. In other words, $\mathcal{F}_s(f)$ and $\mathcal{F}_t(f)$ should be as similar as possible. Meanwhile, $P(f; \theta_p)$ can predict the flowrate accurately on the basis of f .

We introduce a domain discriminator $D(\cdot; \theta_d)$ to help $E(\mathbf{p}; \theta_e)$ distill domain-invariant features. The input of $D(\cdot; \theta_d)$ is the features f , and the output is the prediction (\hat{d}) of the domain label, i.e., $\hat{d} = D(f; \theta_d)$. Similar to GANs, $D(f; \theta_d)$ actually can measure the distance between $\mathcal{F}_s(f)$ and $\mathcal{F}_t(f)$. If the domain label \hat{d} produced by $D(f; \theta_d)$ is the same as the real domain label d , the gap between f_s and f_t is significant enough to be noticed by $D(f; \theta_d)$. On the contrary, if $D(f; \theta_d)$ cannot classify source domain features f_s and target domain features f_t correctly, the gap is not remarkable for it.

The general framework of FANs includes three parts, i.e., $E(\mathbf{p}; \theta_e)$, $P(f; \theta_p)$, and $D(f; \theta_d)$. Each part plays a different role. First of all, $D(f; \theta_d)$ is expected to distinguish source and target domain features. We can achieve this goal via updating θ_d to minimize the loss function \mathcal{L}_d^* which is defined as follows:

$$\begin{aligned} \mathcal{L}_d^*(\mathbf{p}_s, \mathbf{p}_t; \theta_e, \theta_d) &= \mathbb{E}_{\mathbf{p}_s \sim \mathcal{S}(\mathbf{p})} [D(E(\mathbf{p}_s; \theta_e); \theta_d)] \\ &\quad - \mathbb{E}_{\mathbf{p}_t \sim \mathcal{T}(\mathbf{p})} [D(E(\mathbf{p}_t; \theta_e); \theta_d))] \\ &\quad + \eta \mathbb{E}_{\hat{f} \sim \mathcal{F}_{\hat{f}}(f)} [(\|\nabla_{\hat{f}} D(\hat{f}; \theta_d)\|_2 - 1)^2]. \end{aligned} \quad (4)$$

We can also train $D(f; \theta_d)$ with the cross entropy loss as a binary classifier. However, in that case, $D(f; \theta_d)$ is actually to measure the Jensen-Shannon divergence between $\mathcal{F}_s(f)$ and $\mathcal{F}_t(f)$. This would lead to instability and difficulty in training [26]. Instead of that, we employ the loss function of WGANs [27], [28], which measures the Wasserstein distance

between $\mathcal{F}_s(f)$ and $\mathcal{F}_t(f)$ and can, therefore, reduce instability in training.

In addition, we expect $P(f; \theta_p)$ can determine the average liquid flowrate according to f . Therefore, its loss function \mathcal{L}_p can be defined as follows:

$$\begin{aligned} \mathcal{L}_p(\mathbf{p}_s, \dot{m}_{l_s}; \theta_e, \theta_p) &= \mathbb{E}_{(\mathbf{p}_s, \dot{m}_{l_s}) \sim \mathcal{S}(\mathbf{p}, \dot{m}_l)} \\ &\quad \left[\max \left((P(E(\mathbf{p}; \theta_e); \theta_p) - \dot{m}_{l_s})^2 - \delta, 0 \right) \right] \end{aligned} \quad (5)$$

where \mathcal{L}_p is similar to the loss function of support vector regression (SVR) [37]. We choose this form of loss function mainly because \dot{m}_{l_s} is close but not exactly equal to the real mass flowrate ($\dot{m}_{l_s}^m$) of the liquid phase in the multiphase flow. In fact, \dot{m}_{l_s} , here, is the flowrate of the liquid single-phase flow before mixing, i.e., $\dot{m}_{l_s}^s$. As we discussed in Section III, there is an error between $\dot{m}_{l_s}^m$ and $\dot{m}_{l_s}^s$, but it is less than a very small upper bound. Therefore, we can replace $\dot{m}_{l_s}^m$ with $\dot{m}_{l_s}^s$. However, if the prediction \hat{m}_l was strictly required to be exactly the same as $\dot{m}_{l_s}^s$ in training, the prediction model would be easy to overfit.

Finally, we need to design the loss function for $E(\mathbf{p}; \theta_e)$. $E(\mathbf{p}; \theta_e)$ is expected to distill domain-invariant and flowrate-discriminative features at the same time, and its loss function, therefore, should contain two different parts. First, $E(\mathbf{p}; \theta_e)$ can improve the similarity between $\mathcal{F}_s(f)$ and $\mathcal{F}_t(f)$ by fooling $D(f; \theta_d)$. However, the approach to training the generator $G(\cdot; \theta_g)$ in WGANs is not appropriate to domain adaptation. In the case of image generation, the change of the parameters of $G(\cdot; \theta_g)$ will not affect the data distribution of real images. Therefore, the real images are not used when training $G(\cdot; \theta_g)$. θ_g is updated to maximize the discriminator's scores for the generated pictures. In other words, the distribution of synthetic images is modified to get close to the distribution of real images which is fixed. However, in the case of domain adaptation, θ_e has great impacts on both $\mathcal{F}_s(f)$ and $\mathcal{F}_t(f)$. $\mathcal{F}_t(f)$ will change as we adjust θ_e to modify $\mathcal{F}_s(f)$. Hence, both of $\mathcal{F}_s(f)$ and $\mathcal{F}_t(f)$ need to be considered when $E(\cdot; \theta_e)$ is optimized. Therefore, we define \mathcal{L}_d^\diamond as follows:

$$\mathcal{L}_d^\diamond(\mathbf{p}_s, \mathbf{p}_t; \theta_e, \theta_d) = \left\{ \mathbb{E}_{\mathbf{p}_s \sim \mathcal{S}(\mathbf{p})} [D(E(\mathbf{p}_s; \theta_e); \theta_d)] - \mathbb{E}_{\mathbf{p}_t \sim \mathcal{T}(\mathbf{p})} [D(E(\mathbf{p}_t; \theta_e); \theta_d))] \right\}^2. \quad (6)$$

The difference between $\mathbb{E}_{\mathbf{p}_s \sim \mathcal{S}(\mathbf{p})} [D(E(\mathbf{p}_s; \theta_e); \theta_d)]$ and $\mathbb{E}_{\mathbf{p}_t \sim \mathcal{T}(\mathbf{p})} [D(E(\mathbf{p}_t; \theta_e); \theta_d))]$ indicates the dissimilarity between $\mathcal{F}_s(f)$ and $\mathcal{F}_t(f)$. Therefore, minimizing \mathcal{L}_d^\diamond as a function of θ_e yields maximizing the similarity between $\mathcal{F}_s(f)$ and $\mathcal{F}_t(f)$.

Furthermore, the features produced by $E(\mathbf{p}; \theta_e)$ should also be flowrate discriminative, and this goal can simply be achieved by minimizing \mathcal{L}_p . Therefore, the complete loss function of $E(\mathbf{p}; \theta_e)$ can be written as below, and the

Algorithm 1 FANs, We Recommend RMSprop for Parameters Optimization

Require: The gradient penalty coefficient η , the domain coefficient λ , the number of domain discriminator iterations per loop n_d , the batch size m , RMSprop hyperparameters α , ℓ_d , ℓ_e and ℓ_p .

Require: initial feature extractor $E(\cdot; \theta_{e0})$, initial label predictor $P(\cdot; \theta_{p0})$, initial domain discriminator $D(\cdot; \theta_{d0})$.

```

1: while  $\theta$  has not converged do
2:   for each  $i \in [1, n_d]$  do
3:     for each  $j \in [1, m]$  do
4:       Sample source data  $p_s \sim \mathcal{S}(p)$ , target data  $p_t \sim T(p)$ , a random number  $\beta \sim U[0, 1]$ .
5:        $f_s \leftarrow E(p_s; \theta_e)$ 
6:        $f_t \leftarrow E(p_t; \theta_e)$ 
7:        $\hat{f} \leftarrow \beta f_t + (1 - \beta) f_s$ 
8:        $\mathcal{L}_{d_j}^* \leftarrow D(f_s; \theta_d) - D(f_t; \theta_d) + \eta (\|\nabla_{\hat{f}} D(\hat{f}; \theta_d)\|_2 - 1)^2$  {Eq. (4)}
9:     end for
10:     $\theta_d \leftarrow \text{RMSprop}(\nabla_{\theta_d} \frac{1}{m} \sum_{j=1}^m \mathcal{L}_{d_j}^*, \theta_d, \ell_d, \alpha)$ 
11:  end for
12:  Sample a batch of unlabeled target data  $\{p_{t_j}\}_{j=1}^m \sim T(p)$ , a batch of labeled source data  $\{(p_{s_j}, \dot{m}_{l_{s_j}})\}_{j=1}^m \sim \mathcal{S}(p, \dot{m}_l)$ .
13:   $\{f_{s_j}\}_{j=1}^m \leftarrow \{E(p_{s_j}; \theta_e)\}_{j=1}^m$ 
14:   $\{f_{t_j}\}_{j=1}^m \leftarrow \{E(p_{t_j}; \theta_e)\}_{j=1}^m$ 
15:   $\mathcal{L}_p \leftarrow \frac{1}{2m} \sum_{j=1}^m [\max((P(f_{s_j}; \theta_p) - \dot{m}_{l_{s_j}})^2 - \delta, 0)]$  {Eq. (5)}
16:   $\mathcal{L}_d^\diamond \leftarrow \left[ \frac{1}{m} \sum_{j=1}^m D(f_{s_j}; \theta_d) - \frac{1}{m} \sum_{j=1}^m D(f_{t_j}; \theta_d) \right]^2$  {Eq. (6)}
17:   $\mathcal{L}_e \leftarrow \mathcal{L}_p + \lambda \mathcal{L}_d^\diamond$  {Eq. (7)}
18:   $\theta_p \leftarrow \text{RMSprop}(\nabla_{\theta_p} \mathcal{L}_p, \theta_p, \ell_p, \alpha)$ 
19:   $\theta_e \leftarrow \text{RMSprop}(\nabla_{\theta_e} \mathcal{L}_e, \theta_e, \ell_e, \alpha)$ 
20: end while

```

pseudo-code² of FANs is given in Algorithm 1

$$\begin{aligned} \mathcal{L}_e(p_s, p_t, \dot{m}_l; \theta_e, \theta_p, \theta_d) = & \mathcal{L}_p(p_s, \dot{m}_{l_s}; \theta_e, \theta_p) \\ & + \lambda \mathcal{L}_d^\diamond(p_s, p_t; \theta_e, \theta_d). \quad (7) \end{aligned}$$

C. Similarity and Difference Between FANs and DANs

The basic ideas of FANs and DANs are pretty similar. Both of them try to find domain-invariant and label-discriminative features via adversarial learning with a domain discriminator $D(f; \theta_d)$. However, there are two major differences. First, the loss function of $D(f; \theta_d)$ in DANs is defined as

$$\begin{aligned} \mathcal{L}_d(p_s, p_t; \theta_e, \theta_d) = & -\mathbb{E}_{p_s \sim \mathcal{S}(p)} [\log(1 - D(E(p_s; \theta_e); \theta_d))] \\ & -\mathbb{E}_{p_t \sim T(p)} [\log D(E(p_t; \theta_e); \theta_d)]. \quad (8) \end{aligned}$$

It is different from our method, and probably causes instability and difficulty in training [26]. More importantly, $D(f; \theta_d)$

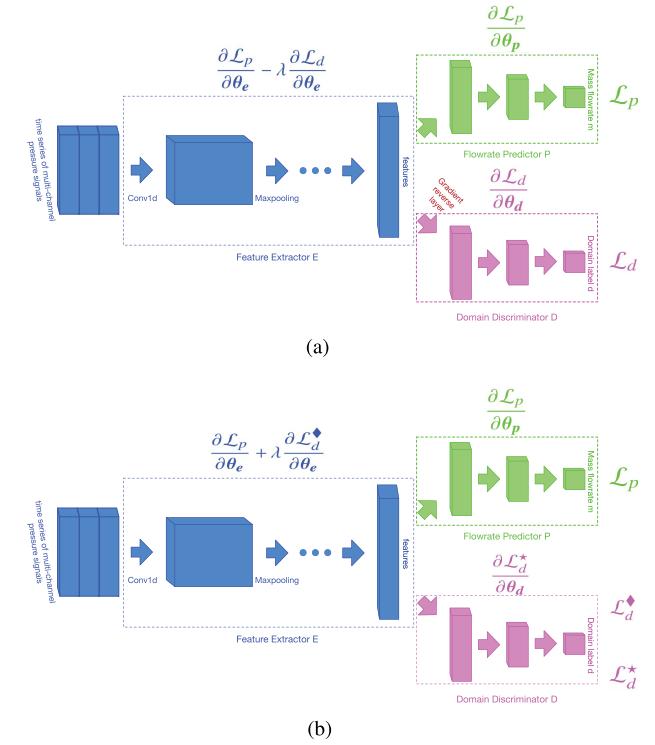


Fig. 6. Overviews of our proposed FANs and the original DANs. In both the two methods, the procedure to predict the flowrate for multiphase flows is the same. The time series of multichannel pressure signals p first passes through the feature extractor $E(p; \theta_e)$, which contains multiple 1-D convolutional layers and maxpooling layers, and then we can get the feature vector f . The domain discriminator $D(f; \theta_d)$ and the label predictor $P(f; \theta_p)$ can determine the domain label and the liquid mass flowrate based on f . However, parameters of the prediction models are updated according to different loss functions. (a) In DANs, $D(f; \theta_d)$ is connected with $E(p; \theta_e)$ via a gradient reverse layer. $E(p; \theta_e)$ tries to capture domain-invariant features through directly maximizing the loss of $D(f; \theta_d)$. (b) In FANs, we employ two different loss functions \mathcal{L}_d^* and \mathcal{L}_d^\diamond for training $D(f; \theta_d)$ and training $E(p; \theta_e)$, respectively. Instead of maximizing the probability that $D(f; \theta_d)$ makes mistakes, we expect source and target features produced by $E(p; \theta_e)$ can obtain close outputs after passing $D(f; \theta_d)$.

is connected with the feature extractor $E(p; \theta_e)$ via a gradient reverse layer in the DAN architecture (see Fig. 6). This layer is used to provide the reverse gradient of the loss of $D(f; \theta_d)$, i.e., $-\partial \mathcal{L}_d / \partial \theta_e$, when training $E(p; \theta_e)$. In other words, $E(p; \theta_e)$ attempts to produce domain-invariant features by directly maximizing the loss of $D(f; \theta_d)$. However, it is not reasonable. $D(f; \theta_d)$ will completely misclassify f_s as class 1 and f_t as class 0, when \mathcal{L}_d is taken to the maximum. This does not mean that $\mathcal{F}_s(f)$ and $\mathcal{F}_t(f)$ have high similarity. In fact, $D(f; \theta_d)$ only needs to add a minus sign to the output, then the accuracy of domain classification will return to 100%. The objective of DANs is shown as follows:

$$\theta_d^* = \arg \min_{\theta_d} \mathcal{L}_d(p_s, p_t; \theta_e, \theta_d) \quad (9)$$

$$\begin{aligned} (\theta_e^*, \theta_p^*) = & \arg \min_{\theta_e, \theta_p} \mathcal{L}_p(p_s, \dot{m}_{l_s}; \theta_e, \theta_p) \\ & - \lambda \mathcal{L}_d(p_s, p_t; \theta_e, \theta_d). \quad (10) \end{aligned}$$

In our method, the gradient reverse layer is removed. We do not require $E(p; \theta_e)$ to generate f_s and f_t that $D(f; \theta_d)$ will completely classify into the incorrect categories. Instead,

²Our codes are available at https://github.com/hudelin19/idan_multiphaseflow

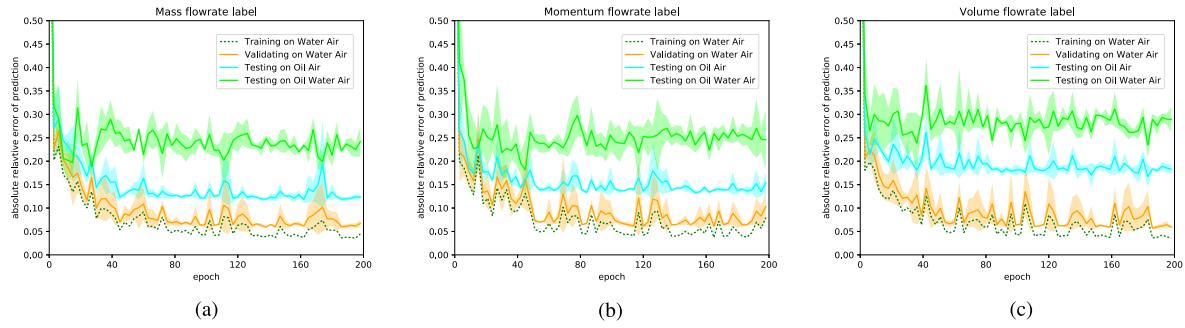


Fig. 7. Learning and testing curves of CNNs with different output objectives. (a)–(c) Learning and testing curves with \dot{m}_l as the output, \dot{M}_l as the output, and \dot{v}_l as the output. The results are the average for three runs, and the shadow represents the standard deviation. The horizontal axis of each graph represents the numbers of epochs, and the vertical axis represents the mean absolute relative error.

TABLE III
NETWORK SPECIFICATION

	Layer	Number of Neurons	Activation Function
$E(\mathbf{p}; \theta_e)$	conv1	64	SELU
	conv2	64	SELU
	max-pool1		MAXPOOL
	conv3	128	SELU
	conv4	128	SELU
	max-pool2		MAXPOOL
	conv5	256	SELU
	conv6	256	SELU
	conv7	256	SELU
	max-pool3		MAXPOOL
	conv8	512	SELU
	conv9	512	SELU
	conv10	512	SELU
$P(f; \theta_p)$	max-pool4		MAXPOOL
	conv11	512	SELU
	conv12	512	SELU
	conv13	512	SELU
	max-pool5		MAXPOOL
$D(f; \theta_d)$	fc1	512	SELU
	fc2	64	SELU
	fc3	1	LINEAR

we expect f_s and f_t can yield similar and even the same outputs after passing $D(f; \theta_d)$. The objective of FANs is shown as follows:

$$\theta_d^* = \arg \min_{\theta_d} \mathcal{L}_d^*(\mathbf{p}_s, \mathbf{p}_t; \theta_e, \theta_d) \quad (11)$$

$$\theta_p^* = \arg \min_{\theta_p} \mathcal{L}_p(\mathbf{p}_s, \dot{m}_l; \theta_e, \theta_p) \quad (12)$$

$$\begin{aligned} \theta_e^* = \arg \min_{\theta_e} & \mathcal{L}_p(\mathbf{p}_s, \dot{m}_l; \theta_e, \theta_p) \\ & + \lambda \mathcal{L}_d^*(\mathbf{p}_s, \mathbf{p}_t; \theta_e, \theta_d). \end{aligned} \quad (13)$$

V. RESULTS AND DISCUSSION

On the basis of the dynamic experimental data collected from the multiphase flow test facility discussed in Section III, we carried out a series of experiments to evaluate our method. The specification of networks we used in all experiments is shown in Table III.

CNN prediction models consist of $E(\mathbf{p}; \theta_e)$ and $P(f; \theta_p)$. DANs and FANs contain all the three parts in the table. However, the output of $D(f; \theta_d)$ in DANs needs to pass a sigmoid function, because $D(f; \theta_d)$ is regarded as a binary classifier in this case. In following experiments, we chose SELU as the activation function and initialized the parameters in accordance with [38]. The size of each convolutional filter in the table is 3, and the stride 1. The size of each max-pool is 2, and the stride is 2 as well.

A. CNNs With Different Output Objectives

First, we want to test that whether choosing different types of flowrates as the output of the same CNN would affect the performance. Therefore, we trained different CNN prediction models on water-air two-phase flow samples with different kinds of flowrates as outputs, i.e., the mass flowrate \dot{m}_l , the momentum flowrate \dot{M}_l , and the volume flowrate \dot{v}_l . Then we tested each prediction model on oil-air two-phase flow samples and oil-water-air three-phase flow samples. The learning and testing curves are shown in Fig. 7.

The results show that no matter what kind of flowrate is selected as the output, the performance of the CNN prediction model on training samples is quite close. The mean absolute relative error for each CNN can eventually converge around 5% on training samples and 9% on validation samples. However, the performance on testing samples varies according to the objective of the output. The mean absolute relative error on the oil-air two-phase flow data set is 13% for the \dot{m}_l output, 15% for the \dot{M}_l output, and close to 20% for the \dot{v}_l output. The same metric on the oil-water-air three-phase flow data set is 22% for the \dot{m}_l output, 25% for the \dot{M}_l output, and even more than 28% for the \dot{v}_l output.

This phenomenon is related to the physical property of the differential pressure signals. However, to our surprise, it somewhat conflicts with the formula of traditional venturi single-phase flowmeters, which is shown as follows [36]:

$$\dot{m}_l = \frac{\epsilon C}{\sqrt{1 - \beta^4}} \pi r^2 \beta^2 \sqrt{2 \rho_l \Delta P_1} \quad (14)$$

where ϵ is the expansion factor of the fluid, C is the discharge coefficient of the venturi tube, r is the radius of the throat and β is the ratio between diameters of the throat and pipe.

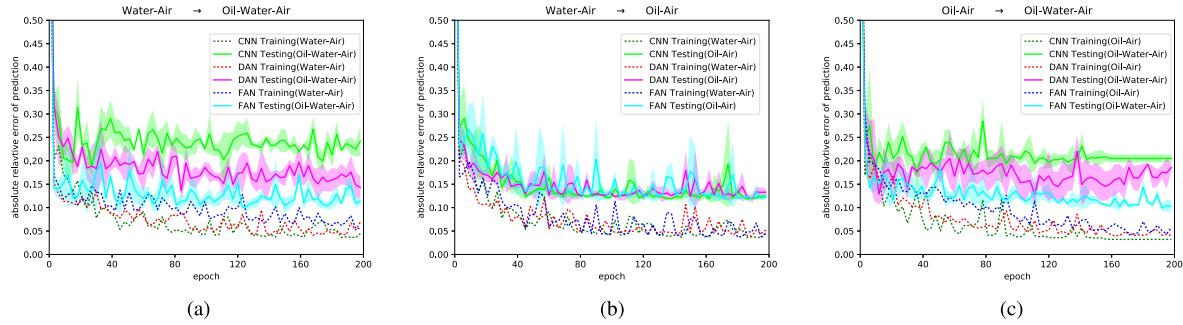


Fig. 8. Learning and testing curves of FANs, DANs, and CNNs in different transfer scenarios. (a) Performance of different algorithms in $S_w \rightarrow T_{ow}$, (b) $S_w \rightarrow T_o$, and (c) $S_o \rightarrow T_{ow}$. The results are the average for three runs, and the shadow represents the standard deviation. The horizontal axis of each graph represents the numbers of epochs, and the vertical axis represents the mean absolute relative error.

According to (14), ΔP_1 directly reflects instantaneous momentum flowrate M_l rather than instantaneous mass flowrate m_l . A similar conclusion can be drawn by dimensional analysis. However, the CNN with the mass flowrate output has the strongest generalization ability rather than the CNN with the momentum flowrate output. This result is probably contributed by two factors. First of all, the mechanism of multiphase flows is much more complicated than single-phase flows. The conclusions based on single-phase flow theories are not necessarily applicable to multiphase flows. In addition, CNNs mainly determine the average flowrate by detecting the patterns consisting of time-adjacent multichannel pressure signals. For the single-phase prediction formula, an instantaneous differential pressure signal can deduce an instantaneous flowrate. The relationships among neighboring pressure signals in the time domain are not considered. Therefore, the two methods are essentially different.

From the testing curves of CNNs with different kinds of flowrates as output objectives, we observe that the difference of the liquid phase densities has a negative impact on the performance of the prediction model, and the issue can be mitigated to some extent by selecting the type of the flowrate to predict. The mass flowrate \dot{m}_l output can achieve the best performance in comparison to the other two kinds of flowrates. Therefore, in the following experiments, all models were trained to predict the mass flowrate \dot{m}_l rather than \dot{M}_l and \dot{v}_l .

B. Results of Different Domain Adaptation Tasks

Next, in order to evaluate our proposed domain adaptation method, we consider three different transfer scenarios, i.e., transferring from the water-air two-phase flow to the oil-water-air three-phase flow $S_w \rightarrow T_{ow}$, transferring from the water-air two-phase flow to the oil-air two-phase flow $S_w \rightarrow T_o$, and transferring from the oil-air two-phase flow to the oil-water-air three-phase flow $S_o \rightarrow T_{ow}$. The learning and testing curves are shown in Fig. 8. Note that we only take into account the oil-water-air three-phase flow samples whose mass flowrate is less than 4800 kg/h in $S_o \rightarrow T_{ow}$. It is mainly because regression models are not good at extrapolation. The maximum flowrate in training samples is approximately 5200 kg/h, but is close to 6000 kg/h in testing samples. The

testing samples with large mass flowrate should be ignored; otherwise, extrapolation will take place in testing.

In $S_w \rightarrow T_{ow}$ and $S_o \rightarrow T_{ow}$ scenarios, CNNs have the smallest relative error on source domain samples, at about 5% in both two cases, but the largest relative error on target domain samples, at around 22% and 20%, respectively. It is mainly because CNNs learn numerous domain-specific features, which can help to improve the accuracy on source domain samples but decrease the performance on target domain samples. On the contrary, FANs produce the most significant relative error on source domain samples, at around 7% in those two experiments, but the slightest relative error on target domain samples, at around 10%. The gap between source domain samples and target domain samples is only 3%, which is much less than its counterpart in CNNs. This demonstrates that the domain discriminator enables the feature extractor to learn domain-invariant features rather than domain-specific features. DANs obtain an intermediate position on both source and target samples in terms of the relative error. They can also capture domain-invariant features, but FANs can achieve better performance.

In the $S_w \rightarrow T_o$ scenario, the learning and testing curves of both three methods are quite close. The FAN and DAN prediction models do not generate any remarkable improvement in this case compared to the CNN. The reason is that the data distributions of source samples and target samples are very similar in this scenario. The testing error of the CNN on target domain samples is around 13%, which is only 4% higher than the error on source validation samples (see Fig. 7). It means that the error can only drop by less than 4% even if the data distributions of source and target samples are exactly the same. In fact, the samples of the water-air two-phase flow and the samples of the oil-air two-phase flow can be approximately considered as being generated by the same distribution in our experiments. It is mainly because the experimental conditions were very close including the number of the medium in the liquid phase, the static pressure, the range of the gas mass flowrate, the liquid viscosity, and so on, when we carried out dynamic flow experiments to collect data. In addition, even though the difference between the liquid densities may lead to a decline in prediction accuracy on testing samples, selecting the mass flowrate as the output objective of the prediction model can alleviate the issue to

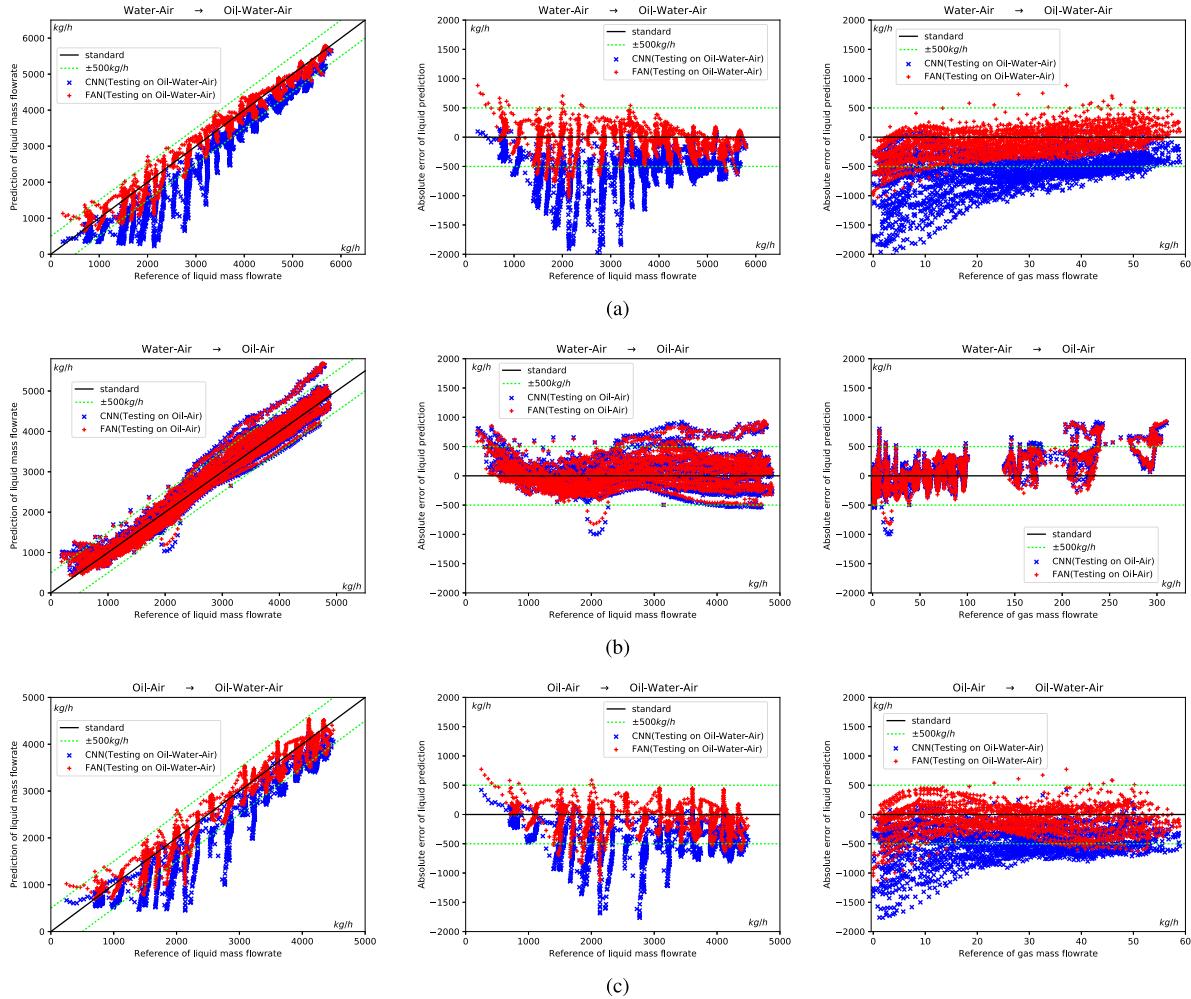


Fig. 9. Prediction results of FANs and CNNs on target domain samples. (a) $\mathcal{S}_w \rightarrow \mathcal{T}_{ow}$. (b) $\mathcal{S}_w \rightarrow \mathcal{T}_o$. (c) $\mathcal{S}_o \rightarrow \mathcal{T}_{ow}$. The horizontal axis of each left graph in (a)–(c) represents the reference of the liquid mass flowrate \dot{m}_l , and the vertical axis represents the prediction of the liquid mass flowrate \hat{m}_l . The meaning of the horizontal axis in each middle graph is the same as the left one, but the vertical axis is the absolute error between the prediction and the reference. Each right graph shows the relationship between the gas mass flowrate \dot{m}_g and the absolute error.

some extent. Therefore, the CNN is able to achieve satisfactory prediction accuracy in this case, and domain adaptation seems not to be essential.

The prediction results of FANs and CNNs are shown in Fig. 9. In general, with domain adaptation, the prediction accuracy rises, the prediction error band narrows and the model becomes more reliable. However, it does not mean that all predicted values are closer to the ground truth. For example, as illustrated in Fig. 9(a), the outputs of the FAN for several samples whose flowrate is near 300 kg/h are much worse than the estimations of the CNN. For further analysis, we plot Fig. 10 to show the improvement after domain adaptation. Actually, the improvement can be interpreted as the decrement, from the distance between CNN prediction and reference to the distance between FAN prediction and reference. If the decrement is positive, it means that the absolute error of the FAN is less than that of the CNN. It should be regard as a positive improvement. If the decrement is negative, it means that the absolute error of the FAN is greater than that of the CNN. It should be regard as a negative improvement.

From Fig. 10, we observe that most of target domain samples gain a positive improvement after domain adaptation. To be specific, more than 92% target domain samples can obtain better predicted values in $\mathcal{S}_w \rightarrow \mathcal{T}_{ow}$, and more than 88% in $\mathcal{S}_o \rightarrow \mathcal{T}_{ow}$. Even in $\mathcal{S}_w \rightarrow \mathcal{T}_o$, the number of target domain samples with a positive improvement exceeds a half. Furthermore, we also notice that positive improvements have greater numerical values than negative improvements in general. For instance, as illustrated in Fig. 10(a), most of the negative improvements are lower than 500 kg/h, while numerous positive improvements are higher than 500 kg/h and even up to 1500 kg/h. Therefore, the accuracy of the prediction model sees a significant rise after domain adaptation.

Moreover, Fig. 9 also demonstrates that there is no significant reduction in prediction accuracy of our method with respect to different dynamic experimental conditions. For example, the system static pressure in oil–water–air dynamic flow experiments is different from water–air and oil–air experiments. However, our prediction model is able to achieve comparable performance in both the three-phase flow data set

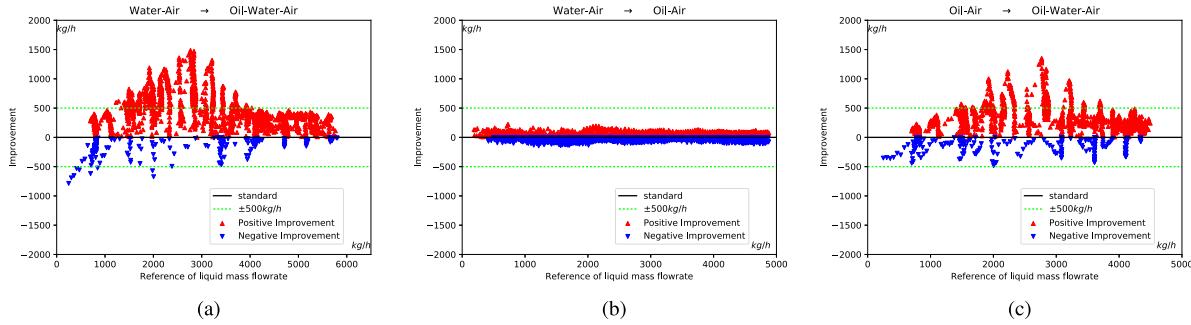


Fig. 10. Improvement of FANs on target domain samples compared to CNNs. The horizontal axis of each graph is the reference of the liquid mass flowrate \dot{m}_l , and the vertical axis is actually the improvement after domain adaptation, which is defined as the decrement, from the distance (absolute error) between CNN prediction and reference to the distance (absolute error) between FAN prediction and reference. (a) $S_w \rightarrow T_{ow}$. (b) $S_w \rightarrow T_o$. (c) $S_o \rightarrow T_{ow}$.

TABLE IV
REGRESSION METRICS

		MAPE ¹	MAE ²	RMSE ³
$\mathcal{S}_w \rightarrow \mathcal{T}_{ow}$	CNN	22.58%	583.61	673.89
	DAN	10.41%	268.91	342.99
	FAN	8.36%	194.99	245.42
$\mathcal{S}_w \rightarrow \mathcal{T}_o$	CNN	11.44%	202.40	268.28
	DAN	10.90%	191.39	248.74
	FAN	11.07%	199.69	260.39
$\mathcal{S}_o \rightarrow \mathcal{T}_{ow}$	CNN	20.38%	487.20	568.26
	DAN	10.81%	231.77	294.00
	FAN	9.09%	205.77	260.39

¹ Mean Absolute Percentage Error.

² Mean Absolute Error.

³ Root Mean Square Error.

and the two-phase flow data set. Our model also performs well in a large variety of flow patterns, GVF_s, and water-in-liquid ratios. Most of state-of-the-art techniques in the multiphase flow measurement field cannot achieve this result.

Table IV evaluates different prediction models with three regression metrics, including mean absolute percentage error (MAPE), mean absolute error (MAE), and root mean square error (RMSE). Our method outperforms DANs and CNNs in $\mathcal{S}_w \rightarrow \mathcal{T}_{ow}$ and $\mathcal{S}_o \rightarrow \mathcal{T}_{ow}$ domain adaptation tasks with regard to all the three metrics. Even in $\mathcal{S}_w \rightarrow \mathcal{T}_o$, our method also has a slight improvement compared with the CNN.

VI. CONCLUSION

In this paper, we have constructed feature maps of multiphase flows with the time series of multi-channel pressure signals, and successfully used them to train CNN models for flowrate prediction. Moreover, the accuracy degradation caused by the gap between testing and training samples that are probably generated by different multiphase flows in different operating environments, and flow conditions are also taken into account. We propose FANs on the basis of DANs to mitigate this issue. Compared with the methods that have been reported in the past, which can merely provide reliable prediction for some specific flows (such as wet gas), our method can be applied to a much wider range of flow conditions and operating environments. The experimental results suggest our prediction model can achieve promising testing accuracy even

the testing and training samples are different due to different experimental conditions during the data collection. In other words, the proposed method has strong generalization ability and practical application potential.

In the near future, we plan to install our devices in a real industrial production environment to evaluate our method on practical multiphase flow samples, which is much more challenging. In addition, in order to further strengthen our model, we also consider to design a novel deep architecture to integrate signals from multimodal sensors.

REFERENCES

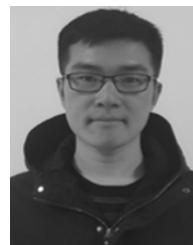
- [1] R. Thorn, G. A. Johansen, and E. A. Hammer, "Recent developments in three-phase flow measurement," *Meas. Sci. Technol.*, vol. 8, no. 7, pp. 691–701, 1997.
 - [2] *Natural Gas-Wet Gas Flow Measurement in Natural Gas Operations*, Standard ISO/TR 12748, 2015.
 - [3] R. Thorn, G. A. Johansen, and B. T. Hjertaker, "Three-phase flow measurement in the petroleum industry," *Meas. Sci. Technol.*, vol. 24, no. 1, 2013, Art. no. 012003.
 - [4] Z. H. Lin, "Two-phase flow measurements with sharp-edged orifices," *Int. J. Multiphase Flow*, vol. 8, no. 6, pp. 683–693, 1982.
 - [5] Y. Xu, C. Yuan, Z. Long, Q. Zhang, Z. Li, and T. Zhang, "Research the wet gas flow measurement based on dual-throttle device," *Flow Meas. Instrum.*, vol. 34, pp. 68–75, Dec. 2013.
 - [6] C. Yuan, Y. Xu, T. Zhang, J. Li, and H. Wang, "Experimental investigation of wet gas over reading in venturi," *Exp. Thermal Fluid Sci.*, vol. 66, pp. 63–71, Sep. 2015.
 - [7] L. Xu, W. Zhou, X. Li, and M. Wang, "Wet-gas flow modeling for the straight section of throat-extended venturi meter," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 6, pp. 2080–2087, Jun. 2011.
 - [8] *Measurement of Wet Gas Flow by Means of Pressure Differential Devices Inserted in Circular Cross-Section Conduits*, Standard ISO/TR 11583, 2012.
 - [9] S. Timung and T. K. Mandal, "Prediction of flow pattern of gas–liquid flow through circular microchannel using probabilistic neural network," *Appl. Soft Comput.*, vol. 13, no. 4, pp. 1674–1685, 2013.
 - [10] H. Shaban and S. Tavoularis, "Identification of flow regime in vertical upward air–water pipe flow using differential pressure signals and elastic maps," *Int. J. Multiphase Flow*, vol. 61, pp. 62–72, May 2014.
 - [11] H. Wu, F. Zhou, and Y. Wu, "Intelligent identification system of flow regime of oil–gas–water multiphase flow," *Int. J. Multiphase Flow*, vol. 27, no. 3, pp. 459–475, 2001.
 - [12] Z. Sun and H. Zhang, "Neural networks approach for prediction of gas–liquid two-phase flow pattern based on frequency domain analysis of vortex flowmeter signals," *Meas. Sci. Technol.*, vol. 19, no. 1, 2008, Art. no. 015401.
 - [13] H. Shaban and S. Tavoularis, "Measurement of gas and liquid flow rates in two-phase pipe flows by the application of machine learning techniques to differential pressure signals," *Int. J. Multiphase Flow*, vol. 67, pp. 106–117, Dec. 2014.

- [14] L. Xu, W. Zhou, X. Li, and S. Tang, "Wet gas metering using a revised venturi meter and soft-computing approximation techniques," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 3, pp. 947–956, Mar. 2011.
- [15] S. Fan and T. Yan, "Two-phase air–water slug flow measurement in horizontal pipe using conductance probes and neural network," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 2, pp. 456–466, Feb. 2014.
- [16] S. Azizi, E. Ahmadloo, and M. M. Awad, "Prediction of void fraction for gas–liquid flow in horizontal, upward and downward inclined pipes using artificial neural network," *Int. J. Multiphase Flow*, vol. 87, pp. 35–44, Dec. 2016.
- [17] S. Azizi, M. M. Awad, and E. Ahmadloo, "Prediction of water holdup in vertical and inclined oil–water two-phase flow using artificial neural network," *Int. J. Multiphase Flow*, vol. 80, pp. 181–187, Apr. 2016.
- [18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
- [19] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *J. Physiol.*, vol. 148, pp. 574–591, Oct. 1959.
- [20] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
- [21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 25th Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.
- [22] K. Simonyan and A. Zisserman. (Sep. 2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: <https://arxiv.org/abs/1409.1556>
- [23] C. Szegedy *et al.*. (Sep. 2014). "Going deeper with convolutions." [Online]. Available: <https://arxiv.org/abs/1409.4842>
- [24] K. He, X. Zhang, S. Ren, and J. Sun. (2016). "Identity mappings in deep residual networks." [Online]. Available: <http://adsabs.harvard.edu/abs/2016arXiv160305027H>
- [25] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.* 27, 2014, pp. 2672–2680.
- [26] M. Arjovsky and L. Bottou. (2017). "Towards principled methods for training generative adversarial networks." [Online]. Available: <https://arxiv.org/abs/1701.04862>
- [27] M. Arjovsky, S. Chintala, and L. Bottou. (2017). "Wasserstein GAN." [Online]. Available: <https://arxiv.org/abs/1701.07875>
- [28] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. (2017). "Improved training of wasserstein GANs." [Online]. Available: <https://arxiv.org/abs/1704.00028>
- [29] A. Radford, L. Metz, and S. Chintala. (Nov. 2015). "Unsupervised representation learning with deep convolutional generative adversarial networks." [Online]. Available: <https://arxiv.org/abs/1511.06434>
- [30] D. Berthelot, T. Schumm, and L. Metz. (2017). "BEGAN: Boundary equilibrium generative adversarial networks." [Online]. Available: <https://arxiv.org/abs/1703.10717>
- [31] G.-J. Qi. (2017). "Loss-sensitive generative adversarial networks on Lipschitz densities." [Online]. Available: <https://arxiv.org/abs/1701.06264>
- [32] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
- [33] Y. Ganin and V. Lempitsky. (2015). "Unsupervised domain adaptation by backpropagation," [Online]. Available: <https://arxiv.org/abs/1409.7495>
- [34] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 59, pp. 1–35, 2016.
- [35] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. CVPR*, 2017, pp. 7167–7176.
- [36] *Measurement of Fluid Flow by Means of Pressure Differential Devices Inserted in Circular Cross-Section Conduits Running Full—Part 4: Venturi Tubes*, Standard ISO 5167-4, 2003.
- [37] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, Aug. 2004.
- [38] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Proc. 30th Adv. Neural Inf. Process. Syst.*, 2017, pp. 971–980.



Delin Hu received the B.E. degree in automation from the College of Electrical Engineering and Information Technology, Sichuan University, Sichuan, China, in 2015. He is currently pursuing the M.E. degree in control science and engineering with the Department of Automation, Tsinghua University, Beijing, China.

His current research interests include machine learning (ML), sensor data mining, and industrial process measurement and monitoring.



Jinku Li received the B.E. degree from the School of Automation, Beijing Institute of Technology, Beijing, China, in 2015. He is currently pursuing the Ph.D. degree with the Department of Automation, Tsinghua University, Beijing.

His current research interests include multiphase flow, flow measurement and instrumentation, and machine learning (ML) techniques.



Yinyan Liu received the B.E. degree from the Department of Automation, School of Control and Computer Engineering, North China Electric Power University, Beijing, China, in 2011, and the M.E. degree from the Department of Automation, Tsinghua University, Beijing, in 2018.

Her current research interests include electrical tomography, intelligent sensor, and deep learning.



Yi Li received the B.E. degree from the University of Central Lancashire, Preston, U.K., in 2004, and the Ph.D. degree from The University of Manchester, Manchester, U.K., in 2008.

He is currently an Associate Professor with the Graduate School at Shenzhen, Tsinghua University, Beijing, China. His current research interests include electrical tomography, data fusion, gas–liquid multiphase flow measurement, and image reconstruction.