Self-Tuning Transfer Dynamic Convolution Autoencoder for Quality Prediction of Multimode Processes With Shifts

Chao Yang, *Graduate Student Member, IEEE*, Qiang Liu, *Senior Member, IEEE*, Chen Wang, *Member, IEEE*, Jinliang Ding, *Senior Member, IEEE*, and Yiu-ming Cheung, *Fellow, IEEE*

***Abstract*—Process shift of multimode process involving data distribution and dynamic relation makes traditional transfer learning methods be intractable and even result in negative transfer. To tackle this issue, this article proposes a novel self-tuning transfer dynamic modeling method for quality prediction of multimode processes. First, in order to capture domain-invariant spatiotemporal (DIST) features, a transfer dynamic convolution autoencoder (TDCAE) with a feature decomposition structure is established. Meanwhile, a first-order vector autoregressive constraint is embedded to extract consistent inner dynamics for DIST features. Then, a shared regression network is established to extract the relations with quality variables. Furthermore, by making full use of private spatiotemporal information from target labeled samples in response to the process shift, the self-tuning TDCAE (STDCAE) aided by a fine-tuning strategy is established for online compensation. Finally, the efficacy of the proposed TDCAE and STDCAE is demonstrated by a comprehensive study of a three-phase flow facility process.**

***Index Terms*—Convolutional neural networks, deep autoencoder, dynamic process modeling, multimode processes, quality prediction, transfer learning (TL).**

Manuscript received 1 December 2023; revised 12 March 2024 and 20 April 2024; accepted 2 May 2024. Date of publication 31 May 2024; date of current version 5 September 2024. This work was supported in part by the National Natural Science Foundation of China under Grant U20A20189, Grant 61991401, Grant 62161160338, and Grant U23A20328, in part by the National Natural Science Foundation of China (NSFC)/Research Grants Council (RGC) Joint Research Scheme under Grant N_HKBU214/21, in part by the General Research Fund of RGC under Grant 12201321, Grant 12202622, and Grant 12201323, and in part by the RGC Senior Research Fellow Scheme under Grant SRFS2324-2S02. Paper no. TII-23-4832. *(Corresponding author: Qiang Liu.)*

Chao Yang, Qiang Liu, and Jinliang Ding are with the State Key Laboratory of Synthetical Automation for Process Industries (Northeastern University), Shenyang 110819, China (e-mail: 2010303@stu.neu.edu.cn; liuq@mail.neu.edu.cn; jlding@mail.neu.edu.cn).

Chen Wang is with the National Engineering Research Center for Big Data Software, Tsinghua University, Beijing 100190, China (e-mail: wang_chen@tsinghua.edu.cn).

Yiu-ming Cheung is with the Department of Computer Science, Hong Kong Baptist University, Kowloon Tong 999077, Hong Kong (e-mail: ymc@comp.hkbu.edu.hk).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TII.2024.3399932.

Digital Object Identifier 10.1109/TII.2024.3399932

Nomenclature

Abbreviations

CNN	Convolution neural network.
MMD	Maximum mean discrepancy.
DIST	Domain-invariant spatiotemporal.
SEN	Shared encoder network.
VAR	Vector autoregressive.
TDCAE	Transfer dynamic convolution autoencoder.
PTST	Private-target spatiotemporal.
PSST	Private-source spatiotemporal.
TERN	Target error regression network.
STDCAE	Self-tuning TDCAE.

Parameters

$\mathcal{K}$	Number of steady operating mode.
s	Length of time window.
$\mathbf{S}_{\mathcal{M}_i}$	Collected dataset for mode $\mathcal{M}_i$.
$\mathbf{X}_{\mathcal{M}_i}, \mathbf{Y}_{\mathcal{M}_i}$	Process and quality data of mode $\mathcal{M}_i$.
$\mathbf{X}^{\mathrm{c}}_{\mathcal{M}_i}, \mathbf{X}^{\mathrm{p}}_{\mathcal{M}_i}$	Shared and private part of process data for mode $\mathcal{M}_i$.
$\mathbf{X}^{s}_{\text{tar, tst}}$	Target domain test set.
$\mathbf{Y}^{\mathrm{c}}_{\mathcal{M}_i}, \mathbf{Y}^{\mathrm{p}}_{\mathcal{M}_i}$	Shared and private part of quality data for mode $\mathcal{M}_i$.
N_{src}	Number of source sequence samples.
N_{tar}	Number of target sequence samples.
$\mathbf{X}^{s,L}_{\text{tar}}$	Process variables of target labeled samples.
$\mathbf{Y}^{s,L}_{\text{tar}}$	Quality variables of target labeled samples.
$\hat{\mathbf{Y}}^{s,L,\mathrm{c}}_{\text{tar}}$	Predicted value of TDCAE for $\mathbf{X}^{s,L}_{\text{tar}}$.
$\mathbf{Y}^{s,L,\mathrm{p}}_{\text{tar}}$	Predicted error of TDCAE for $\mathbf{Y}^{s,L}_{\text{tar}}$.

I. Introduction

IN the last decade, data-driven quality prediction methods have been widely utilized in modern industrial processes due to the abilities to acquire relations between quality indicators and process measurements in a simple manner [1], [2]. In response to the diverse market demand, multimode processes have been widely used and data-driven quality prediction of multimode processes has received much attention recently [3], [4], [5], [6], [7]. The multimode characteristics that come from the switching of operation conditions and adjusting the ratio of raw materials can result in unfavorable data distribution discrepancies. Also, it is difficult to establish a reliable quality prediction model for the operating mode with limited labels while abundant labels are only applicable to specific operating modes. Data-driven quality prediction of multimode processes with scarcity of labels and data distribution discrepancies remains a challenge in academia and industry.

Recently, transfer learning (TL) [8], [9] has shown great potential in cross-domain learning scenarios. TL is to extract information from domains with abundant labeled dataset and reduce the distribution discrepancies across domains. In this regard, multimode process modeling can be viewed as a cross-domain learning task. Specifically, the operating mode with abundant labeled samples is taken as the source domain, and those new modes with limited or no labeled samples are taken as the target domain [4], [10], [11].

In the literature, deep transfer learning (DTL) [12], [13], [14], [15], [16], [17], [18], [19], [20] has been studied in developing quality prediction models for multimode processes due to the merit of good feature representation. The developed DTL quality prediction methods can be classified into three types, i.e., pretraining combined with fine-tuning methods [15], [16], unsupervised domain adaptation methods [4], [17], [18], and joint learning methods [14], [19]. In general, the performance of TL is highly relevant to the ability to reduce the distribution discrepancy by adopting appropriate metrics, e.g., maximum mean discrepancy (MMD) [21], and correlation alignment (CORAL) [22]. Furthermore, adversarial learning is an effective alternative to reduce distribution discrepancy between domains. It can be combined with TL to form adversarial transfer learning [13], [17], [23], [24]. Moreover, feature disentanglement technique in [25] and [26] is incorporated into TL to reduce the distribution discrepancy by alleviating the negative influence of private perturbed variations. For example, Hu et al. [26] proposed a deep feature disentanglement-based TL method for the remaining useful life prediction of bearings under different working conditions. However, the existing work only considers feature disentanglement with static relation. The process dynamics are not considered for quality prediction, while the domain-invariant dynamic feature extraction is still unsolved.

Due to the effect of process dynamics, the collected data are often multidimensional time series, which are autocorrelated and cross-correlated over time. It is essential to develop dynamic latent variable (DLV) models for dynamic process modeling [27]. Meanwhile, quality-relevant variations are driven by a reduced dimensional DLVs. DLV-based approaches have been developed for quality prediction [28], [29], [30]. However, these methods are linear methods. To model nonlinear dynamic relations, deep learning models for quality prediction have attracted much attention due to their advantage in handling complicated nonlinear characteristics [31]. In particular, deep DLV models [32], [33], [34], [35], [36], which incorporate with long short-term memory (LSTM) networks [37], [38], and convolution neural networks (CNN) [39], [40], [41], are investigated. Furthermore, DLV-based quality prediction has been extended to be applicable for multimode processes [34], [35]. One-dimensional CNN (1DCNN) has been applied for modeling time series in [42]. Also, the work of [43] develops a multidimensional CNN for the cement clinker production process.

During the switching of operation modes, process dynamics may shift over time. Recently, some work on TL aided by dynamic relation has been investigated [44], [45], [46], [47], which mainly focuses on reducing the dynamic discrepancy between different modes. Besides, some DTL methods [12], [13] rely on shared feature extractors composed of LSTM or CNN to extract domain-invariant spatiotemporal (DIST) features. The challenge of extracting invariant spatiotemporal features is addressed when the two objectives are met [44], [45]: 1) reducing marginal distribution discrepancy; 2) maintaining consistent process dynamic variations when labeled dataset are not available in the target domain. To the best of authors' knowledge, it is difficult to extract effective DIST features since the existing work cannot effectively exclude the negative effects of domain-private spatiotemporal features.

In multimode processes, data collected from different modes that operate in similar but different process mechanism may be different in data distribution and dynamics. The objective of extracting DIST features is, therefore, not only reducing the data distribution discrepancies while maintaining the consistency of process dynamics, but also excluding the negative effects of irrelevant variations. DIST feature extraction is crucial for quality prediction of the new mode with no abundant labels, especially for the early stage of the operating mode. In view of this, this article proposes a transfer dynamic convolution autoencoder (TDCAE) with a feature decomposition structure for quality prediction of multimode dynamic processes with shifts. Specifically, the 1DCNN is adopted as the basic architecture to extract the DIST features by a shared encoder network (SEN) separated from the private ones. Then, the maximum mean distance metric and discriminator network are adopted to reduce the distribution discrepancy between DIST features. Meanwhile, a first-order vector autoregressive (VAR) constraint is expected to extract consistent dynamics of DIST features. Finally, when a few labeled samples are available, self-tuning TDCAE (STDCAE) is proposed to utilize the private-target spatiotemporal (PTST) features to perform online compensation in response to process shifts of the target domain.

The major contributions of this work are two-fold.

1) A novel feature decomposition structure that can simultaneously consider marginal distribution discrepancy and process dynamics is proposed to extract invariant spatiotemporal features between modes for quality prediction of multimode processes.
2) A self-tuning error compensation mechanism is developed to alleviate the shift caused by conditional distribution discrepancy, which exploits separated private spatiotemporal features of a few labeled dataset in the target domain.

The rest of this article is organized as follows. We make an overview of related works on Autoencoder, 1DCNN, and VAR for dynamic modeling in Section II. Then, Section III gives the motivation and framework of the proposed TDCAE. Subsequently, an online compensation method called STDCAE is presented in Section IV. In Section V, the effectiveness of the proposed methods is demonstrated through a three-phase flow facility process (TPFF). Finally, Section VI concludes this article.

II. Overview of Related Works

A. Autoencoder

Autoencoder is one kind of deep models that can preserve dominant low-dimensional features for high dimension dataset and has gained popularity in recent years for quality prediction [48], nondestructive testing [49], etc. It has a symmetrical network structure consisting of an input layer, a hidden layer, and an output layer, which is trained by reconstruction loss. Here, we let the collected dataset $\mathbf{S} = \{\mathbf{X}\} = \{(\boldsymbol{x}_i), i = 1, \ldots, N\}$ with N samples, where $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{ip}]^\top \in \mathbb{R}^p$, p is the number of process variables. The input sample is encoded and decoded sequentially in the encoder and the decoder as follows:

$$\boldsymbol{h}_i = S_e\left(\mathbf{W}_e \cdot \boldsymbol{x}_i + \mathbf{b}_e\right) \tag{1}$$

$$\tilde{\boldsymbol{x}}_i = S_d\left(\mathbf{W}_d \cdot \boldsymbol{h}_i + \mathbf{b}_d\right) \tag{2}$$

where $\boldsymbol{h}_i = [h_{i1}, h_{i2}, \ldots, h_{iu}] \in \mathbb{R}^u$ denotes u encoded latent features, $\tilde{\boldsymbol{x}}_i = [\tilde{x}_{i1}, \tilde{x}_{i2}, \ldots, \tilde{x}_{ip}] \in \mathbb{R}^p$ is the reconstructed process variables, $\mathbf{W}_e \in \mathbb{R}^{u\times p}$ and $\mathbf{b}_e \in \mathbb{R}^u$ denote weight matrix and bias vector of the encoder, respectively, $\mathbf{W}_d \in \mathbb{R}^{p\times u}$ and $\mathbf{b}_d \in \mathbb{R}^p$ denote weight matrix and bias vactor of the decoder, respectively. S_e and S_d denote the nonlinear activation functions, respectively.

B. One-Dimensional CNN

1DCNN inherits the property of weight sharing, which reduces the number of learning parameters and accelerates the convergence [42]. It has a one-dimensional convolution kernel only focusing on receptive field information along the temporal direction. For $\mathbf{S}^s = \{\mathbf{X}^s\} = \{(\boldsymbol{x}_i^s), i = 1, \ldots, N\}$, where $\boldsymbol{x}_i^s = (\boldsymbol{x}_{s+i-1}, \boldsymbol{x}_{s+i-2}, \ldots, \boldsymbol{x}_i) \in \mathbb{R}^{p\times s}$ with a time window size of s, the extracted features by convolution operation is described as

$$\boldsymbol{h}_i^s(m) = \sum_{j=1}^{p} C_{1v}\left(\boldsymbol{x}_i^s(j)\right), \quad \begin{matrix} i = (1, \ldots, N) \\ m = (1, \ldots, M) \end{matrix} \tag{3}$$

where $C_{1v} \in \mathbb{R}^{1\times v}$ is a one-dimensional convolution operation with a length size of v ($v < s$), $\boldsymbol{h}_i^s(m)$ denotes the mth feature map of the ith window of samples. M convolution kernels are adopted to extract features.

C. VAR for Dynamic Modeling

A class of explicit DLV models [27], [28] is developed by introducing the VAR model in the latent subspace. They can drive some DLVs with noninteracting relations with each other. It is assumed that $\mathbf{S} = \{\mathbf{X}\} = \{(\boldsymbol{x}_k), k = 1, \ldots, N+s\}$ with $N+s$ samples are available, where $\boldsymbol{x}_k \in \mathbb{R}^p$ is the sample vector at time instant k. For each DLV, its latent dynamic relations depending on the past lagged data are described as

$$t_k = \boldsymbol{x}_k^\top \mathbf{w} \tag{4}$$

$$t_k = \beta_1 t_{k-1} + \beta_2 t_{k-2} + \cdots + \beta_s t_{k-s} + r_k \tag{5}$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_s)^\top$ satisfies $\|\boldsymbol{\beta}\|^2 = 1$, t_k is the score of process variables, $\mathbf{w} \in \mathbb{R}^{p\times 1}$ is the weight vector determined by maximizing predictability of the projected latent variables (LVs) [27]. s is the length of time window selected according to the dynamic order. r_k is the residual that denotes the predictive error.

III. Proposed TDCAE Algorithm

A. Motivation and Basic Framework of TDCAE

$\mathcal{K}$ steady operating modes of a multimode process are denoted as $\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_\mathcal{K}\}$. Only a few modes can collect abundant labeled datasets, while most modes cannot. Given that each mode is different but similar, it is assumed that each mode dataset $\mathbf{S}_{\mathcal{M}_i} = \{\mathbf{X}_{\mathcal{M}_i}, \mathbf{Y}_{\mathcal{M}_i}\}$, where $i \in \{1, 2, \ldots, \mathcal{K}\}$, can be decomposed into the shared part and the private part as

$$\begin{aligned}\{\mathbf{X}_{\mathcal{M}_i}, \mathbf{Y}_{\mathcal{M}_i}\} &= \left\{\mathbf{X}^{\mathrm{c}}_{\mathcal{M}_i} + \mathbf{X}^{\mathrm{p}}_{\mathcal{M}_i}, \mathbf{Y}^{\mathrm{c}}_{\mathcal{M}_i} + \mathbf{Y}^{\mathrm{p}}_{\mathcal{M}_i}\right\} \\ &= \left\{\mathbf{X}^{\mathrm{c}}_{\mathcal{M}_i}, \mathbf{Y}^{\mathrm{c}}_{\mathcal{M}_i}\right\} + \left\{\mathbf{X}^{\mathrm{p}}_{\mathcal{M}_i}, \mathbf{Y}^{\mathrm{p}}_{\mathcal{M}_i}\right\}\end{aligned} \tag{6}$$

where $\mathbf{X}^{\mathrm{c}}_{\mathcal{M}_i}$ and $\mathbf{Y}^{\mathrm{c}}_{\mathcal{M}_i}$ denote the shared part with respect to the input and output, and $\mathbf{X}^{\mathrm{p}}_{\mathcal{M}_i}$ and $\mathbf{Y}^{\mathrm{p}}_{\mathcal{M}_i}$ are the private part.

In TL, the first term of the right side in (6) is expected to be captured and aligned, which facilitates cross-domain learning, while the second term should be discarded due to its potentially negative effects. Therefore, a novel TDCAE using feature decomposition is proposed in this work. For ease of understanding, using the time window of size s for serialization, the sequence samples of source domain and target domain are, respectively, organized as $\{\mathbf{X}^s_{\text{src}}, \mathbf{Y}^s_{\text{src}}\} = \{(\boldsymbol{x}^s_{\text{src},i}, \boldsymbol{y}^s_{\text{src},i})\}_{i=1}^{N_{\text{src}}}$, where N_{src} is the number of source sequence samples, and $\{\mathbf{X}^s_{\text{tar}}\} = \{(\boldsymbol{x}^s_{\text{tar},i})\}_{i=1}^{N_{\text{tar}}}$, where N_{tar} is the number of target sequence samples. The framework of the proposed TDCAE algorithm is shown in Fig. 1.

1) Shared Encoder Network: The SEN $E_{\mathrm{c}}(\cdot;\theta_{\mathrm{c}})$ is to extract l DIST features between the source domain and target domain. For each input sample $\boldsymbol{x}^s_{\text{src},i}$ and $\boldsymbol{x}^s_{\text{tar},i}$, the modeling process is described as

$$\boldsymbol{h}^{s,\mathrm{c}}_{\text{src},i} = E_{\mathrm{c}}(\boldsymbol{x}^s_{\text{src},i}) \tag{7}$$

$$\boldsymbol{h}^{s,\mathrm{c}}_{\text{tar},i} = E_{\mathrm{c}}(\boldsymbol{x}^s_{\text{tar},i}) \tag{8}$$

where $\boldsymbol{h}^{s,\mathrm{c}}_{\text{src},i} \in \mathbb{R}^{s\times l}$ and $\boldsymbol{h}^{s,\mathrm{c}}_{\text{tar},i} \in \mathbb{R}^{s\times l}$ denote DIST features from the source domain and target domain, respectively.

2) Private Encoder Network: Different from the SEN, the private encoder network includes two parts, i.e., private-source encoder network $E_{\text{ps}}(\cdot;\theta_{\text{ps}})$ and private-target encoder network $E_{\text{pt}}(\cdot;\theta_{\text{pt}})$, which are to extract l PSST and PTST features concerning the source domain or target domain, respectively. The corresponding modeling process is described as

$$\boldsymbol{h}^{s,\mathrm{p}}_{\text{src},i} = E_{\text{ps}}(\boldsymbol{x}^s_{\text{src},i}) \tag{9}$$

$$\boldsymbol{h}^{s,\mathrm{p}}_{\text{tar},i} = E_{\text{pt}}(\boldsymbol{x}^s_{\text{tar},i}) \tag{10}$$

where $\boldsymbol{h}^{s,\mathrm{p}}_{\text{src},i} \in \mathbb{R}^{s\times l}$ and $\boldsymbol{h}^{s,\mathrm{p}}_{\text{tar},i} \in \mathbb{R}^{s\times l}$ denote the private spatiotemporal features concerning the source domain and target domain, respectively.

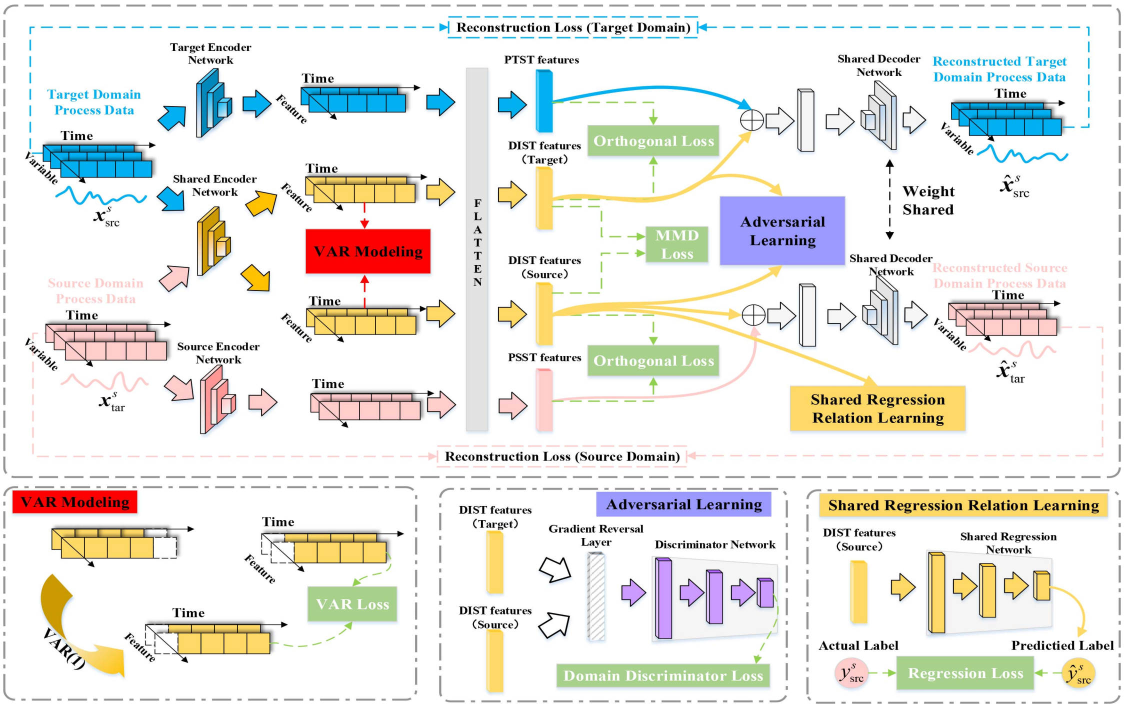

Fig. 1. Modeling framework of TDCAE for cross-domain quality prediction.

3) Shared Decoder Network: To extract effective DIST, PSST, and PTST features, their combined form is reconstructed back to the original space as much as possible through a shared decoder network $D_{\text{ds}}(\cdot, \theta_{\text{ds}})$. The reconstructed loss of mean square error (MSE) is calculated as

$$\mathcal{Q}_{\text{rec}} = \frac{1}{N_{\text{src}}} \sum_{i=1}^{N_{\text{src}}} \left\| \boldsymbol{x}_{\text{src},i}^{s} - \hat{\boldsymbol{x}}_{\text{src},i}^{s} \right\|_F^2 + \frac{1}{N_{\text{tar}}} \sum_{i=1}^{N_{\text{tar}}} \left\| \boldsymbol{x}_{\text{tar},i}^{s} - \hat{\boldsymbol{x}}_{\text{tar},i}^{s} \right\|_F^2 \tag{11}$$

where $\hat{\boldsymbol{x}}_{\text{src},i}^{s} = D_{\text{ds}}(\boldsymbol{h}_{\text{src},i}^{s,\text{com}})$ and $\hat{\boldsymbol{x}}_{\text{tar},i}^{s} = D_{\text{ds}}(\boldsymbol{h}_{\text{tar},i}^{s,\text{com}})$. $\boldsymbol{h}_{\text{src},i}^{s,\text{com}} = \boldsymbol{h}_{\text{src},i}^{s,\text{c}} + \boldsymbol{h}_{\text{src},i}^{s,\text{p}}$ and $\boldsymbol{h}_{\text{tar},i}^{s,\text{com}} = \boldsymbol{h}_{\text{tar},i}^{s,\text{c}} + \boldsymbol{h}_{\text{tar},i}^{s,\text{p}}$ denote the combined spatiotemporal features with respect to the source and target domain.

4) Shared Regression Network: The source DIST features $\boldsymbol{h}_{\text{src},i}^{s,\text{c}}$ are expected to achieve maximum predictability for $\boldsymbol{y}_{\text{src},i}^{s}$. Before feeding into the shared regression network $f_{\text{cr}}(\cdot;\theta_{\text{cr}})$, $\boldsymbol{h}_{\text{src},i}^{s,\text{c}}$ is flattened to a vector form $\boldsymbol{u}_{\text{src},i}^{s,\text{c}} \in \mathbb{R}^{sl\times 1}$. Therefore, the prediction loss of MSE is calculated as

$$\mathcal{Q}_{\text{reg}} = \frac{1}{N_{\text{src}}} \sum_{i=1}^{N_{\text{src}}} \left\| \boldsymbol{y}_{\text{src},i}^{s} - \hat{\boldsymbol{y}}_{\text{src},i}^{s} \right\|^2 \tag{12}$$

where $\hat{\boldsymbol{y}}_{\text{src},i}^{s} = f_{\text{cr}}(\boldsymbol{u}_{\text{src},i}^{s,\text{c}})$ denotes the predicted quality.

5) Domain Discriminator Network: The objective of the domain discriminator network $G_{\text{d}}(\cdot;\theta_{\text{d}})$ is to make the flatten DIST features $\boldsymbol{u}_{\text{src},i}^{s,\text{c}}$ and $\boldsymbol{u}_{\text{tar},i}^{s,\text{c}}$ indistinguishable. Meanwhile, a gradient reversal layer, which is the identity function with reverse gradient is adopted to make the domain discriminator and the SEN trained in an adversarial manner. The adversarial loss can be maximized and minimized concerning θ_{d} and θ_{c}, which is defined as

$$\mathcal{Q}_{\text{d}} = \sum_{i=1}^{N_{\text{src}}+N_{\text{tar}}} \left\{ d_i \log \hat{d}_i + (1-d_i)\log(1-\hat{d}_i) \right\} \tag{13}$$

where $d_i \in \{0,1\}$ denotes the actual label of the ith sample. When $i \le N_{\text{src}}$, $\hat{d}_i = G_{\text{d}}(\boldsymbol{u}_{\text{src},i}^{s,\text{c}})$ is predicted label of the source domain, otherwise $\hat{d}_i = G_d(\boldsymbol{u}_{\text{tar},i}^{s,\text{c}})$ is predicted label of the target domain.

Remark 1: Note that both the SEN and private encoder network are composed of two 1-D convolution layers with batch normalization. The shared decoder network is composed of two 1-D deconvolution layers, and the shared regression network and discriminator network are composed of two fully connected layers (FCL).

B. Distribution Alignment and Separation

The MMD, as a common distribution distance metric function, is to further align flattened DIST features. The MMD loss between $\boldsymbol{u}_{\text{src},i}^{s,\text{c}}$ and $\boldsymbol{u}_{\text{tar},i}^{s,\text{c}}$ is calculated as

$$\mathcal{Q}_{\text{mmd}} = \left\| \sum_{i=1}^{N_{\text{src}}} \phi(\boldsymbol{u}_{\text{src},i}^{s,\text{c}}) - \sum_{i=1}^{N_{\text{tar}}} \phi(\boldsymbol{u}_{\text{tar},i}^{s,\text{c}}) \right\|_{\mathcal{H}}^2 \tag{14}$$

where $\phi(\cdot)$ denotes the mapping function, $\mathcal{H}$ is the reproducing kernel Hilbert space.

Meanwhile, the DIST features should be separated from the private ones in each domain. For all samples of the source domain, the DIST and PSST features are flattened to $\mathbf{U}_{\text{src}}^{s,\text{c}} = [\boldsymbol{u}_{\text{src},1}^{s,\text{c}}, \ldots, \boldsymbol{u}_{\text{src},i}^{s,\text{c}}, \ldots, \boldsymbol{u}_{\text{src},N_{\text{src}}}^{s,\text{c}}]^\top \in \mathbb{R}^{N_{\text{src}} \times sl}$ and $\mathbf{U}_{\text{src}}^{s,\text{p}} = [\boldsymbol{u}_{\text{src},1}^{s,\text{p}}, \ldots, \boldsymbol{u}_{\text{src},i}^{s,\text{p}}, \ldots, \boldsymbol{u}_{\text{src},N_{\text{src}}}^{s,\text{p}}]^\top \in \mathbb{R}^{N_{\text{src}} \times sl}$. Similarly, the DIST and PTST features of the target domain are also flattened to $\mathbf{U}_{\text{tar}}^{s,\text{c}} = [\boldsymbol{u}_{\text{tar},1}^{s,\text{c}}, \ldots, \boldsymbol{u}_{\text{tar},i}^{s,\text{c}}, \ldots, \boldsymbol{u}_{\text{tar},N_{\text{tar}}}^{s,\text{c}}]^\top \in \mathbb{R}^{N_{\text{tar}} \times sl}$ and $\mathbf{U}_{\text{tar}}^{s,\text{p}} = [\boldsymbol{u}_{\text{tar},1}^{s,\text{p}}, \ldots, \boldsymbol{u}_{\text{tar},i}^{s,\text{p}}, \ldots, \boldsymbol{u}_{\text{tar},N_{\text{tar}}}^{s,\text{p}}]^\top \in \mathbb{R}^{N_{\text{tar}} \times sl}$. The orthogonality loss constraint is introduced between the SEN and the private encoder network in each domain, which is calculated as

$$\mathcal{Q}_{\text{orth}} = \left\| \left(\mathbf{U}_{\text{src}}^{s,\text{c}}\right)^\top \mathbf{U}_{\text{src}}^{s,\text{p}} \right\|_F^2 + \left\| \left(\mathbf{U}_{\text{tar}}^{s,\text{c}}\right)^\top \mathbf{U}_{\text{tar}}^{s,\text{p}} \right\|_F^2 \tag{15}$$

where $\|\cdot\|_F^2$ denotes the Frobenius norm.

C. Consistent Inner Dynamic by VAR

To make DIST features have consistent inner dynamics, a first-order VAR model, called VAR(1), is introduced to guide the learning of SEN. Considering that $\boldsymbol{h}_{\text{src},i}^{s,\text{c}}$ and $\boldsymbol{h}_{\text{tar},i}^{s,\text{c}}$ have time dimension size of s, the last value of their time dimension is discarded to form $\boldsymbol{h}_{\text{src},i}^{(1:s-1),\text{c}}$ and $\boldsymbol{h}_{\text{tar},i}^{(1:s-1)\text{c}}$, followed by the first value of their time dimension is discarded to form $\boldsymbol{h}_{\text{src},i}^{(2:s),\text{c}}$ and $\boldsymbol{h}_{\text{tar},i}^{(2:s),\text{c}}$. Next, $\boldsymbol{h}_{\text{src},i}^{(1:s-1),\text{c}}$, $\boldsymbol{h}_{\text{tar},i}^{(1:s-1)\text{c}}$, $\boldsymbol{h}_{\text{src},i}^{(2:s),\text{c}}$, and $\boldsymbol{h}_{\text{tar},i}^{(2:s),\text{c}}$ are flattened to vector form $\boldsymbol{u}_{\text{src},i}^{(1:s-1),\text{c}}$, $\boldsymbol{u}_{\text{tar},i}^{(1:s-1)\text{c}}$, $\boldsymbol{u}_{\text{src},i}^{(2:s),\text{c}}$, and $\boldsymbol{u}_{\text{tar},i}^{(2:s),\text{c}}$. Using VAR(1) model $f_{\text{var}}(\cdot;\theta_{\text{var}})$ driven by a fully connected network, a shared autoregressive relation between $\boldsymbol{u}_{\text{src},i}^{(1:s-1),\text{c}}$ and $\boldsymbol{u}_{\text{src},i}^{(2:s),\text{c}}$, and the one between $\boldsymbol{u}_{\text{tar},i}^{(1:s-1),\text{c}}$ and $\boldsymbol{u}_{\text{tar},i}^{(2:s),\text{c}}$ are established and the MSE loss is calculated as

$$\mathcal{Q}_{\text{var}} = \frac{1}{N_{\text{src}}} \sum_{i=1}^{N_{\text{src}}} \left\| \boldsymbol{u}_{\text{src},i}^{(2:s),\text{c}} - f_{\text{var}}(\boldsymbol{u}_{\text{src},i}^{(1:s-1),\text{c}}) \right\|^2 + \frac{1}{N_{\text{tar}}} \sum_{i=1}^{N_{\text{tar}}} \left\| \boldsymbol{u}_{\text{tar},i}^{(2:s),\text{c}} - f_{\text{var}}(\boldsymbol{u}_{\text{tar},i}^{(1:s-1),\text{c}}) \right\|^2 \tag{16}$$

where $f_{\text{var}}(\boldsymbol{u}_{\text{src},i}^{(1:s-1),\text{c}})$ and $f_{\text{var}}(\boldsymbol{u}_{\text{tar},i}^{(1:s-1),\text{c}})$ are the predicted value of VAR(1) with respect to the source domain and target domain.

D. Training Loss of TDCAE Algorithm

In TDCAE, all submodules are jointly trained and optimized through Adam optimization until convergence [50]. The loss function of TDCAE can be summarized as

$$\mathcal{Q}_{\text{TDCAE}} = \mathcal{Q}_{\text{reg}} + \lambda_{\text{rec}}\mathcal{Q}_{\text{rec}} + \lambda_{\text{mmd}}\mathcal{Q}_{\text{mmd}} + \lambda_{\text{d}}\mathcal{Q}_{\text{d}} + \lambda_{\text{orth}}\mathcal{Q}_{\text{orth}} + \lambda_{\text{var}}\mathcal{Q}_{\text{var}} \tag{17}$$

where λ_{rec}, λ_{d}, λ_{mmd}, λ_{orth}, and λ_{var} are tradeoff parameters for various loss functions. The network parameters of TDCAE are updated as follows:

$$\begin{aligned}
\theta_{\text{c}} &\leftarrow \theta_{\text{c}} - \mu \begin{pmatrix} \nabla_{\theta_{\text{c}}}\mathcal{Q}_{\text{reg}} + \lambda_{\text{rec}}\nabla_{\theta_{\text{c}}}\mathcal{Q}_{\text{rec}} + \lambda_{\text{mmd}}\nabla_{\theta_{\text{c}}}\mathcal{Q}_{\text{mmd}} \\ -\lambda_{\text{d}}\nabla_{\theta_{\text{c}}}\mathcal{Q}_{\text{d}} + \lambda_{\text{orth}}\nabla_{\theta_{\text{c}}}\mathcal{Q}_{\text{orth}} + \lambda_{\text{var}}\nabla_{\theta_{\text{c}}}\mathcal{Q}_{\text{var}} \end{pmatrix} \\
\theta_{\text{ps}} &\leftarrow \theta_{\text{ps}} - \mu\left(\lambda_{\text{rec}}\nabla_{\theta_{\text{ps}}}\mathcal{Q}_{\text{rec}} + \lambda_{\text{orth}}\nabla_{\theta_{\text{ps}}}\mathcal{Q}_{\text{orth}}\right) \\
\theta_{\text{pt}} &\leftarrow \theta_{\text{pt}} - \mu\left(\lambda_{\text{rec}}\nabla_{\theta_{\text{pt}}}\mathcal{Q}_{\text{rec}} + \lambda_{\text{orth}}\nabla_{\theta_{\text{pt}}}\mathcal{Q}_{\text{orth}}\right) \\
\theta_{\text{ds}} &\leftarrow \theta_{\text{ds}} - \mu\nabla_{\theta_{\text{ds}}}\mathcal{Q}_{\text{rec}} \\
\theta_{\text{cr}} &\leftarrow \theta_{\text{cr}} - \mu\nabla_{\theta_{\text{cr}}}\mathcal{Q}_{\text{reg}} \\
\theta_{\text{d}} &\leftarrow \theta_{\text{d}} - \mu\nabla_{\theta_{\text{d}}}\mathcal{Q}_{\text{d}} \\
\theta_{\text{var}} &\leftarrow \theta_{\text{var}} - \mu\nabla_{\theta_{\text{var}}}\mathcal{Q}_{\text{var}}
\end{aligned} \tag{18}$$

where μ denotes the learning rate. The implementation details of the TDCAE are summarized in Algorithm 1.

Algorithm 1: The Proposed TDCAE.

Input : Source domain sequence labeled dataset $\{\mathbf{X}_{\text{src}}^s, \mathbf{Y}_{\text{src}}^s\}$ with N_{src} samples, target domain unlabeled dataset $\{\mathbf{X}_{\text{tar}}^s\}$ with N_{tar} samples.
Output: The optimal modules $E_{\text{c}}(\cdot;\theta_{\text{c}})$, $f_{\text{cr}}(\cdot;\theta_{\text{cr}})$, and $E_{\text{pt}}(\cdot;\theta_{\text{pt}})$.

The following hyperparameters are determined by trial-and-error method. N_e: number of training epochs, μ: learning rate, k: batch size, s: time window size, λ_{rec}: the tradeoff parameter of reconstruction loss, λ_{d}: the tradeoff parameter of adversarial loss, λ_{mmd}: the tradeoff parameter of MMD loss, λ_{orth}: the tradeoff parameter of orthogonality loss, and λ_{var}: the tradeoff parameter of VAR loss.
Initialize all parameters of TDCAE model $\boldsymbol{\theta}_{\text{TDCAE}} = \{\theta_{\text{c}}, \theta_{\text{ps}}, \theta_{\text{pt}}, \theta_{\text{ds}}, \theta_{\text{cr}}, \theta_{\text{d}}, \theta_{\text{var}}\}$.
for $i \leftarrow 1$ **to** N_e **do**
  Calculate the mini-batch number n_k of an epoch by: $n_k = \min\{\lfloor N_{\text{src}}/k \rfloor, \lfloor N_{\text{tar}}/k \rfloor\}$.
  for $j \leftarrow 1$ **to** n_k **do**
    Randomly sample mini-batch source and target dataset.
    / ====== Source domain ====== /
    Extract mini-batch DIST features by Eq. (7).
    Extract mini-batch PSST features by Eq. (9).
    Reconstruct the combination of DIST and PSST features back to the input space.
    / ====== Target domain ====== /
    Extract mini-batch DIST features by Eq. (8).
    Extract mini-batch PTST features by Eq. (10)
    Reconstruct the combination of DIST and PSST features back to the input space.
    / ====== Loss calculation ====== /
    Calculate the reconstruction loss $\mathcal{Q}_{\text{rec}}$ by Eq. (11).
    Calculate the prediction loss $\mathcal{Q}_{\text{reg}}$ by Eq. (12).
    Calculate the adversarial loss $\mathcal{Q}_{\text{d}}$ by Eq. (13).
    Calculate the MMD loss $\mathcal{Q}_{\text{mmd}}$ by Eq. (14).
    Calculate the orthogonality loss $\mathcal{Q}_{\text{orth}}$ by Eq. (15).
    Calculate the VAR loss $\mathcal{Q}_{\text{var}}$ by Eq. (16).
    / ====== Optimization and update ====== /
    The weighted training loss $\mathcal{Q}_{\text{TDCAE}}$ from Eq. (17) is minimized by the Adam optimizer.
    Update the parameters $\boldsymbol{\theta}_{\text{TDCAE}}$ by Eq. (18).
  end
end
Save network structure and optimal parameters of $E_{\text{c}}(\cdot;\theta_{\text{c}})$, $f_{\text{cr}}(\cdot;\theta_{\text{cr}})$, and $E_{\text{pt}}(\cdot;\theta_{\text{pt}})$ for quality prediction.

Remark 2: The distribution of process data in each mode not only varies over time but also differs. This results in a marginal distribution discrepancy with different process dynamics. As demonstrated in the work of [44] and [45], the switching of operating modes and inconsistent process inner dynamics can degrade the modeling performance. The marginal distribution discrepancy is addressed by minimizing the maximum mean distance and confusing discriminator, i.e., the third and fourth terms in (17). The extraction of consistent process dynamic variations is achieved by using the same weight coefficients for

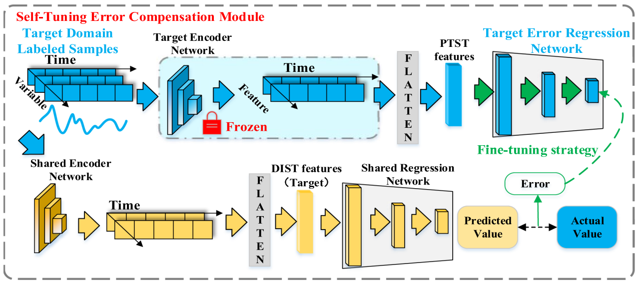

Fig. 2. STDCAE modeling scheme for online compensation.

two domains in (16). Different from the existing work in [44] and [45], the proposed method not only incorporates the deep models for effective feature extraction but also excludes the negative of invariant spatiotemporal features.

IV. STDCAE for Online Compensation

In this section, STDCAE is proposed to provide an error compensation for TDCAE using a small number of labeled dataset from the target domain.

In general, the distribution discrepancy between modes is divided into the marginal distribution discrepancy and conditional distribution discrepancy [6], [20], [44]. The conditional distribution discrepancy can produce prediction error of TDCAE for the new operating mode. To address this, after TDCAE is first adopted to perform prediction for a few labeled dataset from the new mode, the corresponding prediction errors that contain the private information of the target domain are obtained. Then, the extracted PTST dynamic features from the TDCAE are compensated to obtain the private regression relation of the target domain. This private regression relation achieves online error compensation for TDCAE. As mentioned in Section III, the proposed TDCAE aims to extract invariant information between domains with the assistance of unlabeled samples from the target domain. From (6), the invariant regression relation of the target domain can be predicted through TDCAE, while the private regression relation is not well considered. In practice, however, a few labeled samples $\{\mathbf{X}_{\text{tar}}^{s,L}, \mathbf{Y}_{\text{tar}}^{s,L}\}$ are available in the target domain. Although it is difficult for the target domain to train a reliable model, the private variations may be significant for performance improvement. Therefore, an STDCAE is developed by introducing private information to perform online compensation. The detailed modeling scheme is shown in Fig. 2. Specifically, after training TDCAE, the SEN $E_{\text{c}}(\cdot;\theta_{\text{c}})$, the private-target encoder network $E_{\text{pt}}(\cdot;\theta_{\text{pt}})$, and the shared regression network $f_{\text{cr}}(\cdot;\theta_{\text{cr}})$ are frozen to select trainable parameters. TDCAE can provide a predicted label $\hat{\mathbf{Y}}_{\text{tar}}^{s,L,\text{c}}$ for process dataset $\mathbf{X}_{\text{tar}}^{s,L}$ from the target domain. It is noted that TDCAE only extracts DIST features to perform prediction for invariant regression relation between domains. The prediction error between $\hat{\mathbf{Y}}_{\text{tar}}^{s,L,\text{c}}$ and $\mathbf{Y}_{\text{tar}}^{s,L}$ is calculated as

$$\begin{aligned}\mathbf{Y}_{\text{tar}}^{s,L,\text{p}} &= \mathbf{Y}_{\text{tar}}^{s,L} - \hat{\mathbf{Y}}_{\text{tar}}^{s,L,\text{c}} = \mathbf{Y}_{\text{tar}}^{s,L} - f_{\text{TDCAE}}(\mathbf{X}_{\text{tar}}^{s,L}) \\ &= \mathbf{Y}_{\text{tar}}^{s,L} - f_{\text{cr}}(\mathbf{U}_{\text{tar}}^{s,L,\text{c}}) = \mathbf{Y}_{\text{tar}}^{s,L} - f_{\text{cr}}(E_{\text{c}}(\mathbf{X}_{\text{tar}}^{s,L})) \quad (19)\end{aligned}$$

where $f_{\text{TDCAE}}(\cdot;\boldsymbol{\theta}_{\text{TDCAE}})$ denotes invariant regression relation of TDCAE using source domain labeled dataset $\{\mathbf{X}_{\text{src}}^{s}, \mathbf{Y}_{\text{src}}^{s}\}$ and target domain unlabeled dataset $\{\mathbf{X}_{\text{tar}}^{s}\}$. $\mathbf{U}_{\text{tar}}^{s,L,\text{c}}$ denotes the flattened DIST features from the target domain. $\hat{\mathbf{Y}}_{\text{tar}}^{s,L,\text{c}}$ is the predicted value of TDCAE, which is equivalent to the shared information of label value. $\mathbf{Y}_{\text{tar}}^{s,L,\text{p}}$ denotes the predicted error, which contains the private information of label value.

Subsequently, the extracted flattened PTST features $\mathbf{U}_{\text{tar}}^{s,L,\text{p}}$ by private-target encoder network and the predicted error are reorganized as a new training dataset $\{\mathbf{U}_{\text{tar}}^{s,L,\text{p}}, \mathbf{Y}_{\text{tar}}^{s,L,\text{p}}\}$ to train a target error regression network (TERN) by fine-tuning strategy to minimize the MSE. The private regression relation is described as

$$f_{\text{TERN}}(\cdot;\theta_{\text{TERN}}) := \mathbf{U}_{\text{tar}}^{s,L,\text{p}} \rightarrow \mathbf{Y}_{\text{tar}}^{s,L,\text{p}} \quad (20)$$

where $f_{\text{TERN}}(\cdot;\theta_{\text{TERN}})$ is the mapping function of target error compensation network.

Finally, the predicted result of STDCAE on target domain test dataset $\mathbf{X}_{\text{tar,tst}}^{s}$ is a linear combination of TDCAE and TERN, which is described as

$$\begin{aligned}\hat{\mathbf{Y}}_{\text{tar,tst}}^{s} &= f_{\text{STDCAE}}(\mathbf{X}_{\text{tar,tst}}^{s}) \\ &= f_{\text{cr}}\left(E_{\text{c}}\left(\mathbf{X}_{\text{tar,tst}}^{s}\right)\right) + f_{\text{TERN}}\left(E_{\text{pt}}\left(\mathbf{X}_{\text{tar,tst}}^{s}\right)\right) \quad (21)\end{aligned}$$

where $f_{\text{STDCAE}}(\cdot)$ is the regression relation of STDCAE.

V. Case Study

In this section, the effectiveness of the proposed TDCAE and STDCAE methods is demonstrated through the TPFF process.

A. Description of TPFF Process

The TPFF process [51] is a dynamic process, which aims to regulate and measure the flow rate of air, water, and oil in a pressurized facility. In the supplement material file, Fig. S1 shows the TPFF process, including a gas–liquid separator, a three-phase separator, numerous coalesces, and storage tanks connected by pipelines of different specifications and geometries. In the test area, there are pipelines with various bore sizes and geometries, as well as a gas and liquid two-phase separator located on a high platform. The fluid mixtures made up of air, water, and oil, can be provided as the input of the TPFF process at required rates, which are further separated through the three-phase separator at ground level. The whole process produces 24 process variables and the sampling rate is 1 Hz. In addition, multiple steady operating modes can be switched by regulating the setpoints of waterflow and airflow rate. It is, therefore, a dynamic multimode process with a shift between different modes, suitable for the evaluation of the proposed TDCAE and STDCAE methods. In this work, the pressure of the three-phase separator is chosen as the quality variable, and a total of related 16 process variables are the inputs. A detailed description of the quality variable and chosen process variables are listed in Table SI of the supplement material file. Furthermore, we investigate three operation modes with abundant samples, denoted by $\{\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3\}$, and the corresponding parameters are listed in Table I. There are 1000, 800, 950 labels in $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$, respectively.

TABLE I
PARAMETERS FOR THREE MODES

Mode	Waterflow rate (Kg/s)	Airflow rate (m^3/s)	Number of samples
$\mathcal{M}_1$	2	0.0208	1000
$\mathcal{M}_2$	2	0.0417	800
$\mathcal{M}_3$	3.5	0.0208	950

B. Experimental Details

For the comparison across different modes, one mode among $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$ is randomly selected as the source domain, while the remaining two are designated as target domains. In the source domain, all collected dataset are utilized as training dataset, whereas in the target domain, the initial 200 data samples are employed as training dataset and the remaining ones are reserved as test dataset. Consequently, a total of six transfer scenarios are executed. Within each scenario, traditional dynamic modeling approaches, such as dynamic inner partial least squares (DiPLS) [28], LSTM [37], and 1DCNN [42], along with several TL baselines including Deep CORAL [22], domain adversarial neural network (DANN) [17], transfer dynamic latent variable regression (TDLVR) [44], domain adaptation regression by aligning inverse gram matrices (DARE-GRAM) [18], and conditional distribution deep adaptation in regression (CDAR) [12], are investigated as comparison methods. All methods are assessed using the same test dataset. Here, traditional dynamic modeling approaches trained by solely labeled training samples of the source domain are to validate the necessity of transferring invariant dynamic variation. Some domain adaptation TL baselines, such as Deep CORAL, DANN, TDLVR, and DARE-GRAM, are chosen to validate the benefits of DIST feature extraction in the proposed TDCAE. The CDAR method that solely considers conditional distribution discrepancies is chosen to validate the benefit of the proposed method that also takes into account in the marginal distribution discrepancies. STDCAE, as an improvement of TDCAE, serves to verify the effectiveness of online compensation mechanism for alleviating conditional distribution discrepancy. It is noted that 200 labeled training samples from the target domain are only utilized for STDCAE and CDAR, not TDCAE and other TL baselines.

For fair comparisons, the relevant parameters for all methods should be set to be the same. The length of the time window is selected as $s = 4$ by trial-and-error method. The number of LVs of DiPLS is set to 2. The TDLVR has the same LVs as the DiPLS. LSTM has a hidden layer containing 30 neurons, and 1DCNN has two one-dimensional convolutional layers with kernel size 3 and channel size $\{30, 5\}$, respectively, two batch normalization layers, and two FCL with size $\{20, 10\}$. The Deep CORAL, DANN, DARE-GRAM, and CDAR employ the same network structures as the 1DCNN. For the proposed TDCAE, the SEN and two private encoder networks have the same convolutional structures as 1DCNN. Meanwhile, the shared decoder network has a symmetric structure with the private encoder network, and the shared regression network and the discriminator network have the same FCL structure with the 1DCNN. In the proposed STDCAE, the TERN also has the same FCL structure. The rectification linear unit (ReLU) and Tanh activation functions are used corresponding to the convolutional layer and the FCL, respectively. During training stage of TDCAE, the penalty parameters λ_{rec}, λ_{mmd}, λ_{var}, and λ_{d} are searched from $\{0.001, 0.01, 0.1, 1\}$, as well as λ_{orth} is searched from $\{0.0025, 0.005, 0.0075, 0.01\}$. Moreover, the batch size is set to 64, the number of training epochs is 50, the Adam optimizer with a weight decay of 0.0001 is utilized to train the abovementioned deep models, and its learning rate is set to 0.005 from $\{0.001, 0.005, 0.01\}$. In STDCAE, the epochs of fine-tuning the TERN model is set to 10 and its learning rate is set to 0.001.

TABLE II
COMPARISON OF MODE EFFICIENCY BETWEEN THE PROPOSED TDCAE, STDCAE, AND OTHER APPROACHES

Methods	Parameters	Training time(s)	Model size(kb)	FLOPs
DiPLS [28]	36	0.018	14	N/A
LSTM [37]	5791	11.470	25	24030
1DCNN [42]	2636	4.680	16	8730
Deep CORAL [22]	2636	3.790	17	8730
DANN [17]	2868	5.120	62	8590
TDLVR [44]	40	0.481	18	N/A
DARE-GRAM [18]	2636	2.810	20	8730
CDAR [12]	5272	5.31	17	9340
TDCAE (Ours)	10354	15.800	62	34255
STDCAE (Ours)	10575	15.960	65	34465

C. Model Efficiency, Convergence, and Parametric Sensitivity

For simplicity, taking the scenario of $\mathcal{M}_1 \rightarrow \mathcal{M}_2$ as an illustration, four common metrics, including parameters, training time, model size, and floating point operations per second (FLOPs) are adopted to evaluate the model efficiency. It is noted that the FLOPs is only used to evaluate deep models rather than DiPLS and TDLVR models. The comparison result of model efficiency is listed in Table II. It can be seen that DiPLS and TDLVR are attributed to the shortest training time with fewer model parameters. The main reason is that they are linear models. Compared with LSTM, 1DCNN shows a superiority in four metrics. All TL methods adopt the same network structure as 1DCNN. Deep CORAL and DARE-GRAM have comparable model efficiency with 1DCNN, while DANN has a lower model efficiency due to the introduction of a discriminant network. The reason for the decreased model efficiency of CDAR is to establish separate networks for the source domain and target domain. Although the proposed TDCAE and STDCAE have a more complex model due to a parallel learning framework composed of multiple subnetwork modules, the model efficiency is still acceptable. For example, the training time is around 16 s. Subsequently, to demonstrate the convergence of TDCAE under multiple loss constraints, the training loss curve of the proposed TDCAE in the scenario of $\mathcal{M}_1 \rightarrow \mathcal{M}_2$ is shown in Fig. S2 of the supplement material file. It is clear that these losses are correlated with each other and can be efficiently optimized to achieve the convergence of TDCAE. Furthermore, the parametric sensitivity is conducted for each loss constraint of the TDCAE. The impact of each parameter variation on TDCAE performance in the scenario of $\mathcal{M}_1 \rightarrow \mathcal{M}_2$ is shown in Fig. S3 of the supplement material file. It is noted that smaller values of λ_{rec} and λ_{var} will weaken the desired effects of

TABLE III
PREDICTION PERFORMANCE COMPARISON OF TDCAE, STDCAE, AND CONVENTIONAL METHODS FOR THE TPFF PROCESS

Methods	RMSE($\times 10^{-4}$) / MAE($\times 10^{-4}$) / R^2		
	$\mathcal{M}_1 \rightarrow \mathcal{M}_2$	$\mathcal{M}_1 \rightarrow \mathcal{M}_3$	$\mathcal{M}_2 \rightarrow \mathcal{M}_1$
DiPLS [28]	24.535 ± 0.000 / 16.033 ± 0.000 /–2.413 ±0.000	3.112±0.000 / 2.581±0.000 / 0.257±0.000	3.031±0.000 / 2.457±0.000 / 0.323±0.000
LSTM [37]	17.929 ± 2.292 / 15.043 ± 1.861 /–0.846 ±0.452	2.315±0.236 / 1.815±0.160 / 0.585±0.087	3.031±0.328 / 2.467±0.289 / 0.317±0.151
1DCNN [42]	13.748 ± 3.684 / 10.840 ± 2.957 /–0.133 ±0.600	2.196±0.217 / 1.725±0.185 / 0.627±0.075	2.972±0.135 / 2.422±0.106 / 0.349±0.059
Deep CORAL [22]	10.651 ± 1.337 / 8.198 ± 0.762 / 0.349 ±0.163	2.067±0.373 / 1.597±0.290 / 0.663±0.125	2.831±0.368 / 2.280±0.324 / 0.402±0.155
DANN [17]	8.871 ± 1.170 / 7.236 ± 1.005 / 0.548 ±0.112	1.926±0.373 / 1.503±0.306 / 0.707±0.107	2.697±0.180 / 2.202±0.142 / 0.463±0.072
TDLVR [44]	6.142 ± 0.000 / 4.812 ± 0.000 / 0.786 ±0.000	1.892±0.000 / 1.586±0.000 / 0.725±0.000	2.427±0.000 / 1.982±0.000 / 0.566±0.000
DARE-GRAM [18]	9.910 ± 1.114 / 7.992 ± 0.972 / 0.437 ±0.124	1.913±0.197 / 1.472±0.120 / 0.717±0.059	2.538±0.080 / 2.035±0.068 / 0.525±0.030
CDAR [12]	14.632 ± 1.996 / 10.581 ± 1.391 /–0.232 ±0.352	1.782±0.151 / 1.447±0.112 / 0.755±0.042	2.716±0.149 / 2.193±0.139 / 0.456±0.059
TDCAE (Ours)	5.212 ± 0.347 / 4.082 ± 0.288 / 0.846 ±0.020	1.174±0.113 / 0.937±0.082 / 0.893±0.020	2.294±0.045 / 1.885±0.034 / 0.613±0.015
STDCAE (Ours)	**5.064 ± 0.398** / **3.941 ± 0.337** / **0.854 ±0.022**	**1.136±0.100** / **0.917±0.067** / **0.900±0.017**	**2.226±0.050** / **1.824±0.048** / **0.635±0.017**
Methods	RMSE($\times 10^{-4}$) / MAE($\times 10^{-4}$) / R^2		
	$\mathcal{M}_2 \rightarrow \mathcal{M}_3$	$\mathcal{M}_3 \rightarrow \mathcal{M}_1$	$\mathcal{M}_3 \rightarrow \mathcal{M}_2$
DiPLS [28]	2.020 ± 0.000 / 1.628 ± 0.000 / 0.687 ±0.000	2.544±0.000 / 2.197±0.000 / 0.523±0.000	16.814±0.000 / 11.882±0.000 /–0.603±0.000
LSTM [37]	3.036 ± 0.436 / 2.304 ± 0.302 / 0.281 ±0.205	2.687±0.408 / 2.191±0.348 / 0.459±0.165	12.908±1.432 / 10.913±1.136 / 0.046±0.201
1DCNN [42]	2.397 ± 0.232 / 1.961 ± 0.176 / 0.556 ±0.086	2.604±0.409 / 2.107±0.388 / 0.491±0.164	11.688±2.240 / 8.191±1.046 / 0.203±0.290
Deep CORAL [22]	2.471 ± 0.160 / 1.945 ± 0.096 / 0.530 ±0.061	2.172±0.327 / 1.686±0.217 / 0.646±0.115	10.870±1.457 / 8.442±0.965 / 0.320±0.175
DANN [17]	1.958 ± 0.159 / 1.571 ± 0.132 / 0.704 ±0.048	2.085±0.250 / 1.678±0.237 / 0.676±0.076	9.328±1.187 / 7.369±0.750 / 0.500±0.126
TDLVR [44]	1.989 ± 0.000 / 1.673 ± 0.000 / 0.696 ±0.000	1.990±0.000 / 1.690±0.000 / 0.709±0.000	7.531±0.000 / 6.298±0.000 / 0.678±0.000
DARE-GRAM [18]	2.280 ± 0.146 / 1.801 ± 0.097 / 0.600 ±0.049	2.257±0.153 / 1.807±0.128 / 0.624±0.051	9.957±0.317 / 7.908±0.289 / 0.437±0.035
CDAR [12]	2.240 ± 0.186 / 1.839 ± 0.173 / 0.613 ±0.064	2.326±0.214 / 1.903±0.201 / 0.599±0.071	8.636±1.018 / 6.980±0.908 / 0.573±0.099
TDCAE (Ours)	1.408 ± 0.098 / 1.125 ± 0.082 / 0.847 ±0.022	1.713±0.064 / 1.342±0.029 / 0.784±0.016	6.402±0.413 / 5.104±0.273 / 0.767±0.030
STDCAE (Ours)	**1.368 ± 0.107** / **1.101 ± 0.103** / **0.856 ±0.023**	**1.680±0.059** / **1.335±0.031** / **0.792±0.015**	**6.153±0.422** / **4.854±0.308** / **0.785±0.031**

The bold values represent the best performance.

reconstruction loss and the VAR loss in TDCAE, which correspondingly decrease the ability to extract the representative spatiotemporal features of the original data and to maintain the consistent inner dynamic variations. The remaining three parameters, i.e., λ_{mmd}, λ_{d}, and λ_{ortho}, are set to unify the corresponding loss to similar magnitude of the regression loss, whereby avoiding imbalanced learning.

D. Results and Discussion

A total of three performance indicators, including root mean squared error (RMSE), mean absolute error (MAE), and coefficient of determination (R^2), are adopted. In all scenarios, five independent experiments are conducted for the proposed TDCAE, STDCAE, and comparison methods. The prediction performance comparison of TDCAE, STDCAE, and other methods is listed in Table III. Meanwhile, the corresponding prediction performance boxplots of the proposed TDCAE, STDCAE, and other TL approaches are shown in Fig. S4 of the supplement material file. Some insights can be drawn, as follows.

1) Several traditional dynamic methods, including DiPLS, LSTM, and 1DCNN, inevitably suffer from performance degradation caused by distribution discrepancy between modes.
2) Three unsupervised domain adaptation TL baselines, including Deep CORAL, DANN, and DARE-GRAM, achieve better prediction performance by reducing distribution discrepancies but are still hindered since they can neither effectively exclude the negative effects of domain-private information nor consider consistent dynamic variations between modes.
3) By virtue of effectively capturing consistent dynamic variations between modes, TDLVR is generally superior to Deep CORAL, DANN, and DARE-GRAM, except in the scenario of $\mathcal{M}_2 \rightarrow \mathcal{M}_3$ where it is worse than DANN. However, as a linear transfer method, it has shortcomings in dealing with nonlinearity. Also, it cannot exclude the negative effects of domain-private information for knowledge transfer.
4) As a pretraining and fine-tuning TL method, CDAR only outperforms Deep CORAL, DANN, and DARE-GRAM in a few scenarios. The main reason is that it only considers conditional distribution discrepancy and ignores marginal distribution discrepancy.
5) The proposed TDCAE and STDCAE in all scenarios show a better performance for the evaluation indices than all baselines.

On the one hand, compared with the existing DTL methods including Deep CORAL, DANN, DARE-GRAM, and CDAR that do not consider the DIST feature extraction with consistent inner dynamics, TDCAE has a significant benefit in quality prediction from DIST features. On the other hand, the proposed STDCAE achieves a better prediction performance compared to TDCAE that validates the effect of self-tuning error compensation mechanism. Taking the scenario of $\mathcal{M}_1 \rightarrow \mathcal{M}_2$ as an example, STDCAE can improve the performance indices of RMSE, MAE, and R^2 by 2.84%, 3.45%, and 0.95%, respectively. Supplement material file, Fig. S5 displays the average predicted values of TDCAE, STDCAE, and the other approaches in two scenarios of $\mathcal{M}_1 \rightarrow \mathcal{M}_2$ and $\mathcal{M}_1 \rightarrow \mathcal{M}_3$. It can be seen that the proposed TDCAE and STDCAE exhibit better tracking of the pressure variable.

To further demonstrate the feasibility of the proposed TDCAE, the distribution visualization of the first two LVs between the source domain and target domain by TDCAE and other

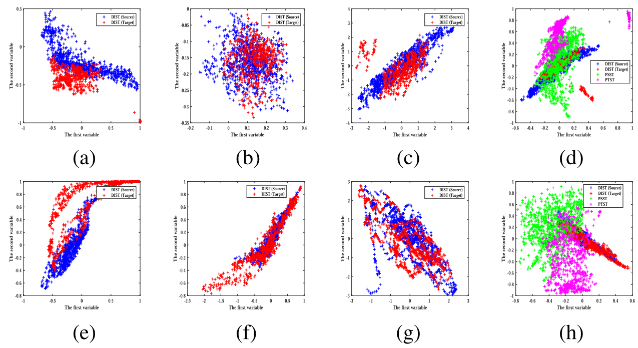

Fig. 3. Distribution visualization between the source domain and target domain by TDCAE and other approaches: (Top four panel) scenario of $\mathcal{M}_1 \rightarrow \mathcal{M}_2$ and (bottom four panel) scenario of $\mathcal{M}_1 \rightarrow \mathcal{M}_3$. (a) DANN. (b) Deep CORAL. (c) TDLVR. (d) TDCAE. (e) DANN. (f) Deep CORAL. (g) TDLVR. (h) TDCAE.

TABLE IV
AVERAGE ABLATION RESULTS OF TDCAE IN TWO SCENARIOS

Methods	Average RMSE($\times 10^{-4}$) / MAE($\times 10^{-4}$) / R^2	
	$\mathcal{M}_1 \rightarrow \mathcal{M}_2$	$\mathcal{M}_1 \rightarrow \mathcal{M}_3$
w/o VAR and MMD	6.777 / 5.364 / 0.739	1.586 / 1.190 / 0.805
w/o VAR	6.080 / 4.861 / 0.784	1.324 / 1.037 / 0.865
w/o MMD	5.745 / 4.569 / 0.813	1.284 / 0.992 / 0.873
TDCAE (all parts)	**5.212** / **4.082** / **0.846**	**1.174** / **0.937** / **0.893**

The bold values represent the best performance.

approaches, including Deep CORAL, DANN, and TDLVR are shown in Fig. 3. It can be seen that Deep CORAL, DANN, and TDLVR can only focus on the alignment of the shared information but not separate the private information between domains. The negative effect of private information will hinder the transfer ability of the prediction model to some extent. By contrast, the TDCAE not only aligns shared information between domains but also separates private information and renders it orthogonal to shared information as much as possible.

E. Ablation Experiment

The ablation experiment of TDCAE on VAR and MMD losses is conducted in two scenarios of $\mathcal{M}_1 \rightarrow \mathcal{M}_2$ and $\mathcal{M}_1 \rightarrow \mathcal{M}_3$. By five independent experiments, the average ablation results of TDCAE are listed in Table IV. Meanwhile, their performance comparison histogram is shown in Fig. 4. In Fig. 4 and Table IV, "w/o" represents "without" indicating that the corresponding strategy is ignored. The following observations are summarized as follows.

1) The "w/o VAR and MMD" exhibits the worst performance in TDCAE, yet it still outperforms DANN, Deep CORAL, and DARE GRAM, underscoring the importance of excluding private variations.
2) The "w/o VAR" reveals that TDCAE does not consider the maintenance of consistent inner dynamics between domains, highlighting the necessity of consistent dynamic variations between domains.
3) The "w/o MMD" exhibits poorer performance compared to TDCAE (all parts), indicating that MMD distance can further mitigate distribution discrepancy between domains.

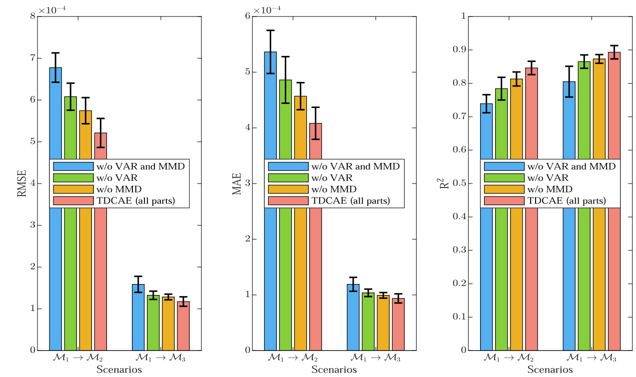

Fig. 4. Performance comparison histogram of TDCAE in two scenarios.

As a result, the proposed TDCAE achieves superior prediction accuracy by jointly considering VAR and MMD losses, making it an effective way for the extraction of DIST features.

VI. CONCLUSION

In this article, a new TL modeling framework based on dynamic convolution autoencoder, including TDCAE and STDCAE, has been successfully developed for online prediction of quality indicators for multimode processes with shifts. Specifically, the proposed TDCAE provides a feature decomposition framework and achieves an effective separation of shared and private information so as to eliminate the negative effect of TL to some extent. Meanwhile, the first-order VAR model and MMD distance metric are embedded to capture consistent inner dynamic variations and reduce distribution discrepancy between modes, respectively. Furthermore, the proposed STDCAE exploits their private information and establishes a TERN for online compensation in response to the shift of the target domain, which can effectively improve the TDCAE.

Future work will focus on the development of dynamic TL algorithms that incorporate continuous memory mechanisms to avoid catastrophic forgetting of historical transfer tasks. Moreover, it would be interesting to develop a domain generalization TL-based online quality prediction method for multimode processes when the target domain unlabeled dataset are not available.

REFERENCES

[1] F. A. Souza, R. Araújo, and J. Mendes, "Review of soft sensor methods for regression applications," *Chemometrics Intell. Lab. Syst.*, vol. 152, pp. 69–79, 2016.
[2] Q. Sun and Z. Ge, "A survey on deep learning for data-driven soft sensors," *IEEE Trans. Ind. Informat.*, vol. 17, no. 9, pp. 5853–5866, Sep. 2021.
[3] X. Shen, J. Liu, J. Sun, L. Jiang, H. Zhao, and H. Zhang, "SSCT-Net: A semisupervised circular teacher network for defect detection with limited labeled multiview MFL samples," *IEEE Trans. Ind. Informat.*, vol. 19, no. 10, pp. 10114–10124, Oct. 2023, doi: 10.1109/TII.2022.3232764.
[4] Y. Liu, C. Yang, M. Zhang, Y. Dai, and Y. Yao, "Development of adversarial transfer learning soft sensor for multigrade processes," *Ind. Eng. Chem. Res.*, vol. 59, no. 37, pp. 16330–16345, 2020.
[5] B. Shen, L. Yao, Z. Yang, and Z. Ge, "Mode information separated β-VAE regression for multimode industrial process soft sensing," *IEEE Sensors J.*, vol. 23, no. 9, pp. 10231–10240, May 2023, doi: 10.1109/JSEN.2023.3261330.

[6] X. Zhang, C. Song, J. Zhao, and D. Xia, “Gaussian mixture continuously adaptive regression for multimode processes soft sensing under time-varying virtual drift,” *J. Process Control*, vol. 124, pp. 1–13, 2023.

[7] K. Zhang, K. Peng, S. Zhao, and F. Wang, “A novel feature-extraction-based process monitoring method for multimode processes with common features and its applications to a rolling process,” *IEEE Trans. Ind. Informat.*, vol. 17, no. 9, pp. 6466–6475, Sep. 2021.

[8] 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.

[9] L. Zhang and X. Gao, “Transfer adaptation learning: A decade survey,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 1, pp. 23–44, Jan. 2024, doi: 10.1109/TNNLS.2022.3183326.

[10] Y. Liu, C. Yang, K. Liu, B. Chen, and Y. Yao, “Domain adaptation transfer learning soft sensor for product quality prediction,” *Chemometrics Intell. Lab. Syst.*, vol. 192, 2019, Art. no.103813.

[11] L. L. T. Chan and J. Chen, “Gaussian process model based multi-source labeled data transfer learning for reducing cost of modeling target chemical processes with unlabeled data,” *Control Eng. Pract.*, vol. 117, Dec. 2021, Art. no. 104941.

[12] X. Liu, Y. Li, Q. Meng, and G. Chen, “Deep transfer learning for conditional shift in regression,” *Knowl.-Based Syst.*, vol. 227, Sep. 2021, Art. no. 107216.

[13] X. Fang, G. Gong, G. Li, L. Chun, W. Li, and P. Peng, “A hybrid deep transfer learning strategy for short term cross-building energy prediction,” *Energy*, vol. 215, Jan. 2021, Art. no. 119208.

[14] Z. Chai, C. Zhao, B. Huang, and H. Chen, “A deep probabilistic transfer learning framework for soft sensor modeling with missing data,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 7598–7609, Dec. 2022.

[15] K. Jiang, Z. Jiang, Y. Xie, D. Pan, and W. Gui, “Prediction of multiple molten iron quality indices in the blast furnace ironmaking process based on attention-wise deep transfer network,” *IEEE Trans. Instrum. Meas.*, vol. 71, Jun. 2022, Art. no. 2512114.

[16] Y. He, L. Chen, and Q. Zhu, “Quality regularization based semisupervised adversarial transfer model with unlabeled data for industrial soft sensing,” *IEEE Trans. Ind. Inform.*, vol. 20, no. 2, pp. 1190–1197, Feb. 2024.

[17] Y. Ganin et al., “Domain-adversarial training of neural networks,” *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2096–2030, 2016.

[18] I. Nejjar, Q. Wang, and O. Fink, “DARE-GRAM: Unsupervised domain adaptation regression by aligning inverse gram matrices,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 11744–11754.

[19] X. Zhang, C. Song, J. Zhao, Z. Xu, and X. Deng, “Deep subdomain learning adaptation network: A sensor fault-tolerant soft sensor for industrial processes,” *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, Dec. 2022, doi: 10.1109/TNNLS.2022.3231849.

[20] X. Zhang, C. Song, J. Zhao, and Z. Xu, “Deep Gaussian mixture adaptive network for robust soft sensor modeling with a closed-loop calibration mechanism,” *Eng. Appl. Artif. Intell.*, vol. 122, Jun. 2023, Art. no. 106124.

[21] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, “Domain adaptation via transfer component analysis,” *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[22] B. Sun and K. Saenko, “Deep CORAL: Correlation alignment for deep domain adaptation,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 443–450.

[23] H. S. Farahani, A. Fatehi, A. Nadali, and M. A. Shoorehdeli, “Domain adversarial neural network regression to design transferable soft sensor in a power plant,” *Comput. Ind.*, vol. 132, Nov. 2021, Art. no. 103489.

[24] D. Li, Y. Liu, D. Huang, C. F. Lui, and M. Xie, “Development of an adversarial transfer learning based soft sensor in industrial systems,” *IEEE Trans. Instrum. Meas.*, vol. 72, Jul. 2023, Art. no. 3000610, doi: 10.1109/TIM.2023.3291771.

[25] H. Hwang, G.-H. Kim, S. Hong, and K.-E. Kim, “Variational interaction information maximization for cross-domain disentanglement,” in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2020, pp. 22479–22491.

[26] T. Hu, Y. Guo, L. Gu, Y. Zhou, Z. Zhang, and Z. Zhou, “Remaining useful life prediction of bearings under different working conditions using a deep feature disentanglement based transfer learning method,” *Rel. Eng. Syst. Saf.*, vol. 219, Mar. 2022, Art. no. 108265.

[27] S. J. Qin, Y. Dong, Q. Zhu, J. Wang, and Q. Liu, “Bridging systems theory and data science: A unifying review of dynamic latent variable analytics and process monitoring,” *Annu. Rev. Control*, vol. 50, pp. 29–48, 2020.

[28] Y. Dong and S. J. Qin, “Regression on dynamic PLS structures for supervised learning of dynamic data,” *J. Process Control*, vol. 68, pp. 64–72, 2018.

[29] Q. Zhu, S. J. Qin, and Y. Dong, “Dynamic latent variable regression for inferential sensor modeling and monitoring,” *Comput. Chem. Eng.*, vol. 137, Jun. 2020, Art. no. 106809.

[30] B. Xu and Q. Zhu, “Concurrent auto-regressive latent variable model for dynamic anomaly detection,” *J. Process Control*, vol. 108, pp. 1–11, Dec. 2021.

[31] C. Ou et al., “Quality-driven regularization for deep learning networks and its application to industrial soft sensors,” *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, Feb. 2022, doi: 10.1109/TNNLS.2022.3144162.

[32] X. Yuan et al., “Multiscale dynamic feature learning for quality prediction based on hierarchical sequential generative network,” *IEEE Sensors J.*, vol. 23, no. 17, pp. 19561–19570, Sep. 2023.

[33] L. Feng, C. Zhao, and Y. Sun, “Dual attention-based encoder–decoder: A customized sequence-to-sequence learning for soft sensor development,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3306–3317, Aug. 2021.

[34] L. Yao, B. Shen, L. Cui, J. Zheng, and Z. Ge, “Semi-supervised deep dynamic probabilistic latent variable model for multimode process soft sensor application,” *IEEE Trans. Ind. Informat.*, vol. 19, no. 4, pp. 6056–6068, Apr. 2023.

[35] J. Xu and Z. Cai, “Gaussian mixture deep dynamic latent variable model with application to soft sensing for multimode industrial processes,” *Appl. Soft Comput.*, vol. 114, Jan. 2022, Art. no. 108092.

[36] S. Zhang and T. Qiu, “A dynamic-inner convolutional autoencoder for process monitoring,” *Comput. Chem. Eng.*, vol. 158, 2022, Art. no. 107654.

[37] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[38] X. Yuan, L. Li, and Y. Wang, “Nonlinear dynamic soft sensor modeling with supervised long short-term memory network,” *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3168–3176, May 2020.

[39] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[40] X. Yuan, S. Qi, Y. A. Shardt, Y. Wang, C. Yang, and W. Gui, “Soft sensor model for dynamic processes based on multichannel convolutional neural network,” *Chemometrics Intell. Lab. Syst.*, vol. 203, 2020, Art. no. 104050.

[41] K. Wang, C. Shang, L. Liu, Y. Jiang, D. Huang, and F. Yang, “Dynamic soft sensor development based on convolutional neural networks,” *Ind. Eng. Chem. Res.*, vol. 58, no. 26, pp. 11521–11531, 2019.

[42] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, “1D convolutional neural networks and applications: A survey,” *Mech. Syst. Signal Process.*, vol. 151, 2021, Art. no. 107398.

[43] X. Jiang and Z. Ge, “Augmented multidimensional convolutional neural network for industrial soft sensing,” *IEEE Trans. Instrum. Meas.*, vol. 70, Apr. 2021, Art. no. 2508410.

[44] C. Yang, Q. Liu, Y. Liu, and Y. M. Cheung, “Transfer dynamic latent variable modeling for quality prediction of multimode processes,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 5, pp. 6061–6074, May 2024, doi: 10.1109/TNNLS.2023.3265762.

[45] K. Wang, W. Zhou, Y. Mo, X. Yuan, Y. Wang, and C. Yang, “New mode cold start monitoring in industrial processes: A solution of spatial–temporal feature transfer,” *Knowl.-Based Syst.*, vol. 248, 2022, Art. no. 108851.

[46] Z. Zhao, G. Yan, M. Ren, L. Cheng, Z. Zhu, and Y. Pang, “Dynamic transfer partial least squares for domain adaptive regression,” *J. Process Control*, vol. 118, pp. 55–68, Oct. 2022.

[47] J. Xie, B. Huang, and S. Dubljevic, “Transfer learning for dynamic feature extraction using variational Bayesian inference,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 11, pp. 5524–5535, Nov. 2022.

[48] X. Yuan, B. Huang, Y. Wang, C. Yang, and W. Gui, “Deep learning-based feature representation and its application for soft sensor modeling with variable-wise weighted SAE,” *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3235–3243, Jul. 2018.

[49] K. Liu, M. Zheng, Y. Liu, J. Yang, and Y. Yao, “Deep autoencoder thermography for defect detection of carbon fiber composites,” *IEEE Trans. Ind. Informat.*, vol. 19, no. 5, pp. 6429–6438, May 2023.

[50] C. Chen, Q. Dou, H. Chen, J. Qin, and P. A. Heng, “Unsupervised bidirectional cross-modality adaptation via deeply synergistic image and feature alignment for medical image segmentation,” *IEEE Trans. Med. Imag.*, vol. 39, no. 7, pp. 2494–2505, Jul. 2020.

[51] C. Ruiz-Cárcel, Y. Cao, D. Mba, L. Lao, and R. Samuel, “Statistical process monitoring of a multiphase flow facility,” *Control Eng. Pract.*, vol. 42, pp. 74–88, Sep. 2015.

Chao Yang (Graduate Student Member, IEEE) received the M.Eng. degree in power engineering and engineering thermophysics from the Zhejiang University of Technology, Hangzhou, China, in 2019. He is currently working toward the Ph.D. degree in control science and engineering with the State Key Laboratory of Synthetical Automation for Process Industry, Northeastern University, Shenyang, China.

Since 2024, he has been visiting the University of Surrey, Guildford, U.K., sponsored by the China Scholarship Council to exchange. His current research interests include deep learning, transfer learning, process monitoring, fault diagnosis, and process data analytics.

Qiang Liu (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in control theory and engineering from Northeastern University, Shenyang, China, in 2003, 2006, and 2012, respectively.

He is currently a Full Professor with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, China. He has authored or coauthored more than 70 peer-reviewed papers. His research interests include big data analytics, machine learning, statistical process monitoring, and fault diagnosis of complex industrial processes.

Dr. Liu was the recipient of the excellent doctor degree dissertation award from the Liaoning Province of China and the Outstanding Young Scholar of Liaoning Revitalization Talents Program, China. Two papers were selected as one of the F5000-Top academic papers in Chinese top-quality SCI tech Journals in the years of 2019 and 2022, respectively.

Chen Wang (Member, IEEE) received the B.S. and M.S. degrees in computer science from Fudan University, Shanghai, China, in 2003 and 2006, respectively. He is currently working toward the D.Eng. (part-time) in software engineering from Tsinghua University, Beijing, China.

He is currently a Senior Researcher and CTO with National Engineering Research Center for Big Data Software, Tsinghua University. Before this, he was a Senior Manager and Research Staff Member with Information Management Research Group, IBM Research, Beijing, China. He is currently the Head of Industrial Big Data Working Group, National IT Standardization Committee (TC28), China, and the Deputy Secretary-General of Alliance of Industrial Internet, China. His research interests include industrial big data and data management.

Jinliang Ding (Senior Member IEEE) received the B.S., M.S., and Ph.D. degrees in control theory and control engineering from Northeastern University, Shenyang, China, in 2001, 2004, and 2012, respectively.

He is currently a Professor with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University. He has authored or coauthored more than 200 refereed journal and international conference papers, and has invented or coinvented more than 50 patents. His research interests include modeling, plant-wide control and optimization for the complex industrial systems, machine learning, industrial artificial intelligence, computational intelligence, and its application.

Dr. Ding was the recipient of numerous awards, including the Young Scholars Science and Technology Award of China in 2016, the National Science Fund for Distinguished Young Scholars in 2015, the National Technological Invention Award in 2013, the Natural Science Award of Liaoning Province in 2022, and the First-Prize of Science and Technology Awards from the Ministry of Education in 2006, 2012, and 2018. Additionally, one of his articles in Control Engineering Practice won the Best Paper Award from 2011 to 2013. He is currently an Associate Editor for IEEE Transactions on Evolutionary Computation, IEEE Transactions on Emerging Topics in Computational Intelligence, and IEEE Transactions on Circuits and Systems—Part II: Express Briefs.

Yiu-ming Cheung (Fellow, IEEE) received the Ph.D. degree in artificial intelligence from the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong, in 2000.

He is currently a Chair Professor of the Department of Computer Science, Hong Kong Baptist University, Hong Kong. His research interests include machine learning and visual computing, as well as their applications in data science, pattern recognition, and optimization.

Dr. Cheung is also a Fellow of AAAS, IET, and BCS. He has been the Editor-in-Chief of IEEE Transactions on Emerging Topics in Computational Intelligence since 2023. Also, he is currently an Associate Editor for several prestigious journals, including IEEE Transactions on Cybernetics, IEEE Transactions on Cognitive and Developmental Systems, IEEE Transactions on Neural Networks and Learning Systems (2014–2020), and *Pattern Recognition*, to name a few.