

Transfer Dynamic Latent Variable Modeling for Quality Prediction of Multimode Processes

Chao Yang¹, Graduate Student Member, IEEE, Qiang Liu², Senior Member, IEEE, Yi Liu³, Member, IEEE, and Yiu-Ming Cheung⁴, Fellow, IEEE

Abstract—Quality prediction is beneficial to intelligent inspection, advanced process control, operation optimization, and product quality improvements of complex industrial processes. Most of the existing work obeys the assumption that training samples and testing samples follow similar data distributions. The assumption is, however, not true for practical multimode processes with dynamics. In practice, traditional approaches mostly establish a prediction model using the samples from the principal operating mode (POM) with abundant samples. The model is inapplicable to other modes with a few samples. In view of this, this article will propose a novel dynamic latent variable (DLV)-based transfer learning approach, called transfer DLV regression (TDLVR), for quality prediction of multimode processes with dynamics. The proposed TDLVR can not only derive the dynamics between process variables and quality variables in the POM but also extract the co-dynamic variations among process variables between the POM and the new mode. This can effectively overcome data marginal distribution discrepancy and enrich the information of the new mode. To make full use of the available labeled samples from the new mode, an error compensation mechanism is incorporated into the established TDLVR, termed compensated TDLVR (CTDLVR), to adapt to the conditional distribution discrepancy. Empirical studies show the efficacy of the proposed TDLVR and CTDLVR methods in several case studies, including numerical simulation examples and two real-industrial process examples.

Index Terms—Dynamic latent variable (DLV), LV regression, multimode processes, quality prediction, transfer learning.

I. INTRODUCTION

WITH the increasing development of process industries toward high integration, multimode processes have played an important role in responding to diverse market

demands and agile manufacturing. During the multimode operations, various products with different specifications or grades are produced by adjusting the proportion of ingredients or operating conditions. The quality variables are usually obtained by offline laboratory analysis, rather than online measurement. The grade changeover may result in large overshoots, long settling times, and increased off-grade products. Quality prediction is, therefore, essential to intelligent inspection, advanced process control, and operation optimization of the multimode processes. In general, the difficulties of establishing the first-principles model of the multimode processes, particularly the short-time mode with a few labeled samples, hinder the application of model-based approaches. By contrast, data-driven quality predictions have received much attention over the past decades [1], [2], [3], [4].

Data-driven quality predictions are mainly built from latent variable (LV) models that extract useful features by performing dimension reduction [5], [6], such as principal component (PC) regression [7], partial least square (PLS) [8], [9], canonical correlation analysis [10], slow feature analysis [11], [12]. To conduct complex nonlinear characteristics, kernel extension [13], Gaussian process model (GPM) [14], and least square support vector regression [15], as well as convolutional neural networks and long short-term memory networks in [16] and [17], have been applied for quality prediction of nonlinear processes [18], [19], [20]. The above methods are unimode modeling provided that training samples and testing samples follow the same distributions. This assumption is, however, not true for multimode processes. This complex characteristic motivates multimode modeling.

Multimodel approaches in [21] and just-in-time learning (JITL)-based approaches in [22] and [23] are popular for multimode processes. The multimodel approaches decompose a complex global model into multiple simple sub-models. However, they require abundant labeled samples collected in each operating mode to establish sub-models. In addition, mode division and matching are necessary. The JITL-based approaches use limited historical samples most similar to the ones of a new mode to establish quality prediction models. They are inapplicable when the number of labeled samples is insufficient for the new mode. Due to frequent changes in operating modes, it may be unable or expensive to acquire enough labeled samples for a specific mode.

As an alternative method, transfer learning (TL) in [24] and [25] has gained attention. TL approaches including deep

Manuscript received 5 March 2022; revised 19 September 2022, 2 December 2022, and 6 February 2023; accepted 4 April 2023. Date of publication 20 April 2023; date of current version 3 May 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 61991401, Grant U20A20189, and Grant 62161160338; in part by the National Natural Science Foundation of China, Research Grants Council, Joint Research Scheme, under Grant N_HKBU214/21; and in part by the 111 Project 2.0 under Grant B08015. (Corresponding author: Qiang Liu.)

Chao Yang and Qiang Liu are with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China (e-mail: yangc1109@163.com; liuq@mail.neu.edu.cn).

Yi Liu is with the Institute of Process Equipment and Control Engineering, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: yliuzju@zjut.edu.cn).

Yiu-Ming Cheung is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong SAR, China (e-mail: ymc@comp.hkbu.edu.hk).

Digital Object Identifier 10.1109/TNNLS.2023.3265762

2162-237X © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See <https://www.ieee.org/publications/rights/index.html> for more information.

TL [26], [27], [28], [29], [30] and multisource TL [14], [31], [32] can effectively handle cross-domain learning by reducing data distribution discrepancy in [33], [34], [35], and [36]. For a specific operating mode with a few labeled samples, TL provides an effective way to transfer useful information from other operating modes with abundant labeled samples for quality prediction. The principal operating mode (POM) with abundant labeled samples in multimode processes is considered the source domain, and the new mode with a few labeled samples is considered the target domain for the TL. Transfer component analysis (TCA) in [37] and correlation alignment (CORAL) in [38] are popular unsupervised TL methods, but they may extract the transferred features not relevant to the labels. Supervised TL methods have been carried out recently. For example, the paper [26] proposes a deep probabilistic transfer learning-based quality prediction with missing data. The paper [28] proposes a deep subdomain learning adaption network for quality prediction. The work of [39] proposes a domain transfer functional-link neural network for quality prediction of nonlinear processes. A GPM-based transfer learning method that adopts a weighted threshold to extract transferable information from multisource domains is proposed in [14]. Moreover, inspired by adversarial learning with good feature distribution representation ability, the paper [40] proposes a domain-adversarial neural network (DANN) to extract the shared features between different domains. Subsequently, the DANN-based quality prediction method has been proposed for a power plant in [41]. Also, a sample-based domain adaptation transfer learning method for quality prediction has been developed for multigrade chemical processes [42]. To further conduct distribution discrepancy among domains, an adversarial TL is developed in [43]. A domain-invariant PLS (di-PLS) with a domain regularization is developed to align feature distribution in the LV space [34]. Also, a heterogeneous transfer learning regression method called domain-adaptation joint PLS has been developed in [35] to handle dimension differences between the source domain and the target domain. To the best of our knowledge, however, most of the traditional TL-based quality prediction methods are developed based on static models.

The practical industrial processes normally operate dynamically whose dynamics come from the inertia units and closed-loop control. The dynamic processes generate auto-correlated and dynamically cross-correlated high-dimensional time series. Incorporating dynamic modeling in transfer learning is beneficial to quality prediction. Some recent work has been investigated to address the process dynamics in transfer learning, such as transfer linear dynamic system (TLDS) [44], transfer slow feature analysis (TSFA) [45], and transfer dynamic GPM [36]. While TLDS is an unsupervised TL method for process monitoring, TSFA and transfer dynamic GPM are supervised methods for quality prediction. TLDS develops a dynamic probabilistic LV model that captures the shared dynamic relations between the source domain and the target domain without using quality data. Among these supervised TL models, transfer dynamic GPM is a parameter-based TL method for quality prediction of nonlinear

dynamic processes by using a time-lagged augmentation matrix to capture dynamics. However, data augmentation not only encounters the curse of dimensionality problems leading to high computational complexity but also has difficulty in accurately extracting domain-shared dynamic features. TSFA obtains transferred dynamic features for quality prediction in a multisource domain scenario. However, TSFA weights multiple separate dynamic features extracted from different domains rather than extracting domain-shared dynamic features by feature distribution alignment.

The above TL methods cannot fully explore and transfer the co-dynamic variations with a reduced dimension hidden in the source domain and the target domain. Since dynamic processes are naturally driven by low-dimensional dynamic latent variations, the common (or transferred) dynamic LV (DLV) between the source domain and the target domain should be extracted from the reduced-dimensional subspace instead of the original data space. DLV modeling provides an effective way to extract low-dimensional dynamic variations. For example, dynamic PLS use augmentations of original variables to extract the DLV by the PLS algorithm. However, the augmented matrix introduces a complex model structure with a large number of parameters. In addition, the extracted LVs may not be necessarily dynamic and do not sequence in predictability. In order to derive a compact dynamic model with DLV sequenced in predictability, dynamic-inner PLS (DiPLS) and dynamic-inner canonical correlation analysis (DiCCA) have been developed [9], [46]. Nevertheless, DiPLS and DiCCA cannot avoid the performance degradation caused by insufficient labeled samples.

The efficient extraction of representative domain-invariant dynamic features is beneficial to the performance improvement of the quality prediction of multimode processes. However, the common DLV between various operating modes can be different from the most predictive DLV extracted by the above DLV methods. The possible difference makes the dynamic extension of transfer learning not straightforward. The work of [47] proposes to incorporate TL with dynamic PLS to deal with the concept drift problem with different data distributions. However, the transfer of dynamics among process data is not involved and the available labels of the new operating mode are not used. Transfer learning in the DLV framework for quality prediction of a new operating mode with a few labeled samples remains a challenge.

The major contributions of this article are twofold: 1) by a concurrent decomposition, the dynamics between process variables and quality variables in the POM, and the co-dynamic variations among process variables between the POM and the new mode are extracted to achieve a reduced marginal distribution discrepancy for quality prediction. 2) To make full use of the newly collected labeled samples in the new operating mode to address conditional distribution discrepancy with the POM, an error compensation mechanism is integrated into the TDLVR algorithm to propose a compensated TDLVR (CTDLVR) method. Note that the transfer from the dynamic operation of a steady state to another one for quality prediction is the major concern of this work. The process dynamics

mentioned in this work are autocorrelation and cross-dynamic relations among the process data, but not the ones during the nonsteady/transition stage.

The remainder of this article is organized as follows. Section II makes an overview of the relevant work of the DLV method. The proposed TDLVR algorithm and its implementation are presented in Section III. A new CTDLVR for conditional distribution discrepancy adaptation is proposed in Section IV. In Section V, the proposed methods are demonstrated by numerical simulation examples and two real-industrial processes. Finally, Section VI gives the conclusions.

II. OVERVIEW OF DYNAMIC LATENT VARIABLE MODELING

To extract reduced-dimensional DLVs from high-dimensional process data and quality data, DLV modeling methods, including DiPLS and DiCCA, were proposed in [9] and [46]. DiPLS not only extracts dynamic relations between the process scores and predicted quality scores but also keeps the consistent outer model structure with the inner model. The DLV is to extract a dynamic latent relation as

$$u_k = \beta_1 t_k + \beta_2 t_{k-1} + \cdots + \beta_s t_{k-s+1} + r_k \quad (1)$$

where t_k and u_k are the latent scores of process variables and quality variables, respectively. r_k and s are the residual and dynamic order, respectively. The latent scores t_k and u_k are obtained by projecting the original variables as

$$\begin{aligned} u_k &= \mathbf{y}_k^\top \mathbf{q} \\ t_k &= \mathbf{x}_k^\top \mathbf{w} \end{aligned} \quad (2)$$

where $\mathbf{x}_k \in \mathfrak{R}^m$ and $\mathbf{y}_k \in \mathfrak{R}^p$ represent the process data and quality data vectors at time instant k , respectively. \mathbf{w} and \mathbf{q} are the weights for process variables and quality variables, respectively. For each LV, the inner model prediction is described as

$$\begin{aligned} \hat{u}_k &= \mathbf{x}_k^\top \mathbf{w} \beta_1 + \cdots + \mathbf{x}_{k-s+1}^\top \mathbf{w} \beta_s \\ &= [\mathbf{x}_k^\top \quad \mathbf{x}_{k-1}^\top \quad \cdots \quad \mathbf{x}_{k-s+1}^\top] (\boldsymbol{\beta} \otimes \mathbf{w}) \end{aligned} \quad (3)$$

where $\boldsymbol{\beta} = (\beta_1, \dots, \beta_s)^\top$ and \otimes denotes the Kronecker product. DiPLS maximizes the covariance between u_k and \hat{u}_k , while DiCCA maximizes the correlation. The objective of DiPLS is as

$$\max_{\mathbf{q}, \mathbf{w}, \boldsymbol{\beta}} \frac{1}{N-s+1} \sum_{k=s}^N \mathbf{q}^\top \mathbf{y}_k [\mathbf{x}_k^\top \quad \mathbf{x}_{k-1}^\top \quad \cdots \quad \mathbf{x}_{k-s+1}^\top] (\boldsymbol{\beta} \otimes \mathbf{w}). \quad (4)$$

In Eq. (4), \mathbf{q} , \mathbf{w} , and $\boldsymbol{\beta}$ are used to project the original process data and quality data space into latent dynamic subspace in a reduced dimension. An iterative algorithm is used to solve the above-mentioned optimization problem. By doing this, the latent relations are extracted and used to calculate the predicted quality for a sample. The detailed algorithm of DiPLS and DiCCA can be found in [46].

III. PROPOSED TDLVR FOR QUALITY PREDICTION OF MULTIMODE PROCESSES

A. TDLVR Objective

Suppose N_{src} samples are available for the POM (i.e., source domain) as $\{\mathbf{S}_{\text{src}}\} = \{\mathbf{X}_{\text{src}}, \mathbf{Y}_{\text{src}}\}$, where $\{\mathbf{X}_{\text{src}}\} = \{\mathbf{x}_{\text{src},i} \in \mathfrak{R}^m\}_{i=1}^{N_{\text{src}}}$ and $\{\mathbf{Y}_{\text{src}}\} = \{\mathbf{y}_{\text{src},i} \in \mathfrak{R}^p\}_{i=1}^{N_{\text{src}}}$ denote the process and quality variables with N_{src} samples, respectively. Meanwhile, the new mode (i.e., target domain) has unlabeled samples $\{\mathbf{S}_{\text{tar}}\} = \{\mathbf{X}_{\text{tar}}\}$, where $\{\mathbf{X}_{\text{tar}}\} = \{\mathbf{x}_{\text{tar},i} \in \mathfrak{R}^m\}_{i=1}^{N_{\text{tar}}}$ denotes the process variables with N_{tar} samples ($N_{\text{src}} \gg N_{\text{tar}}$). The original data matrices are represented as

$$\begin{aligned} \mathbf{X}_{\text{src}} &= [\mathbf{x}_{\text{src},1}, \mathbf{x}_{\text{src},2}, \dots, \mathbf{x}_{\text{src},N_{\text{src}}}]^\top \in \mathfrak{R}^{N_{\text{src}} \times m} \\ \mathbf{Y}_{\text{src}} &= [\mathbf{y}_{\text{src},1}, \mathbf{y}_{\text{src},2}, \dots, \mathbf{y}_{\text{src},N_{\text{src}}}]^\top \in \mathfrak{R}^{N_{\text{src}} \times p} \\ \mathbf{X}_{\text{tar}} &= [\mathbf{x}_{\text{tar},1}, \mathbf{x}_{\text{tar},2}, \dots, \mathbf{x}_{\text{tar},N_{\text{tar}}}]^\top \in \mathfrak{R}^{N_{\text{tar}} \times m}. \end{aligned} \quad (5)$$

For $i = 1, 2, \dots, s$, the following source domain submatrices are formed by time-lagged augmentation as

$$\begin{aligned} \mathbf{X}_{\text{src},i} &= [\mathbf{x}_{\text{src},i}, \dots, \mathbf{x}_{\text{src},i+N_{\text{src}}-s}]^\top \in \mathfrak{R}^{(N_{\text{src}}-s+1) \times m} \\ \mathbf{Z}_{\text{src},s} &= [\mathbf{X}_{\text{src},s}, \mathbf{X}_{\text{src},s-1}, \dots, \mathbf{X}_{\text{src},1}] \in \mathfrak{R}^{(N_{\text{src}}-s+1) \times (ms)} \\ \mathbf{Y}_{\text{src},s} &= [\mathbf{y}_{\text{src},s}, \mathbf{y}_{\text{src},s+1}, \dots, \mathbf{y}_{\text{src},N_{\text{src}}}]^\top. \end{aligned} \quad (6)$$

The target domain submatrices are formed in a similar way as

$$\begin{aligned} \mathbf{X}_{\text{tar},i} &= [\mathbf{x}_{\text{tar},i}, \dots, \mathbf{x}_{\text{tar},i+N_{\text{tar}}-s}]^\top \in \mathfrak{R}^{(N_{\text{tar}}-s+1) \times m} \\ \mathbf{Z}_{\text{tar},s} &= [\mathbf{X}_{\text{tar},s}, \mathbf{X}_{\text{tar},s-1}, \dots, \mathbf{X}_{\text{tar},1}] \in \mathfrak{R}^{(N_{\text{tar}}-s+1) \times (ms)}. \end{aligned} \quad (7)$$

In order to extract the common DLV from the POM to the new mode for quality prediction, the modeling objective is decomposed into three parts. The first objective is to extract the quality-relevant dynamics between process variables and quality variables in the POM. The second objective is to extract the co-dynamic variations of process variables between the POM and the new mode. However, the extracted quality-relevant dynamics may contain both POM-specific DLV variations and common DLV variations. The third objective is, therefore, to retain common DLV variations as much as possible. To sum up, the overall modeling framework is shown in Fig. 1.

The first objective is to extract dynamic latent relations with maximized co-variances between the process latent score $t_{\text{src},k}$ and the quality LV $u_{\text{src},k}$ in the source domain as

$$u_{\text{src},k} = \sum_{i=1}^s \beta_{1i} t_{\text{src},k-i+1} + r_{\text{src},k} \quad (8)$$

with the LVs as

$$\begin{aligned} u_{\text{src},k} &= \mathbf{y}_{\text{src},k}^\top \mathbf{q} \\ t_{\text{src},k} &= \mathbf{x}_{\text{src},k}^\top \mathbf{w} \end{aligned} \quad (9)$$

where $\mathbf{x}_{\text{src},k}$ and $\mathbf{y}_{\text{src},k}$ represent the process and quality data vectors at time k in the source domain, respectively. The inner model prediction for $u_{\text{src},k}$ can be described as

$$\begin{aligned} \hat{u}_{\text{src},k} &= \sum_{i=1}^s \beta_{1i} t_{\text{src},k-i+1} = \sum_{i=1}^s \beta_{1i} \mathbf{x}_{\text{src},k-i+1}^\top \mathbf{w} \\ &= [\mathbf{x}_{\text{src},k}^\top, \mathbf{x}_{\text{src},k-1}^\top, \dots, \mathbf{x}_{\text{src},k-s+1}^\top] (\boldsymbol{\beta}_1 \otimes \mathbf{w}) \end{aligned} \quad (10)$$

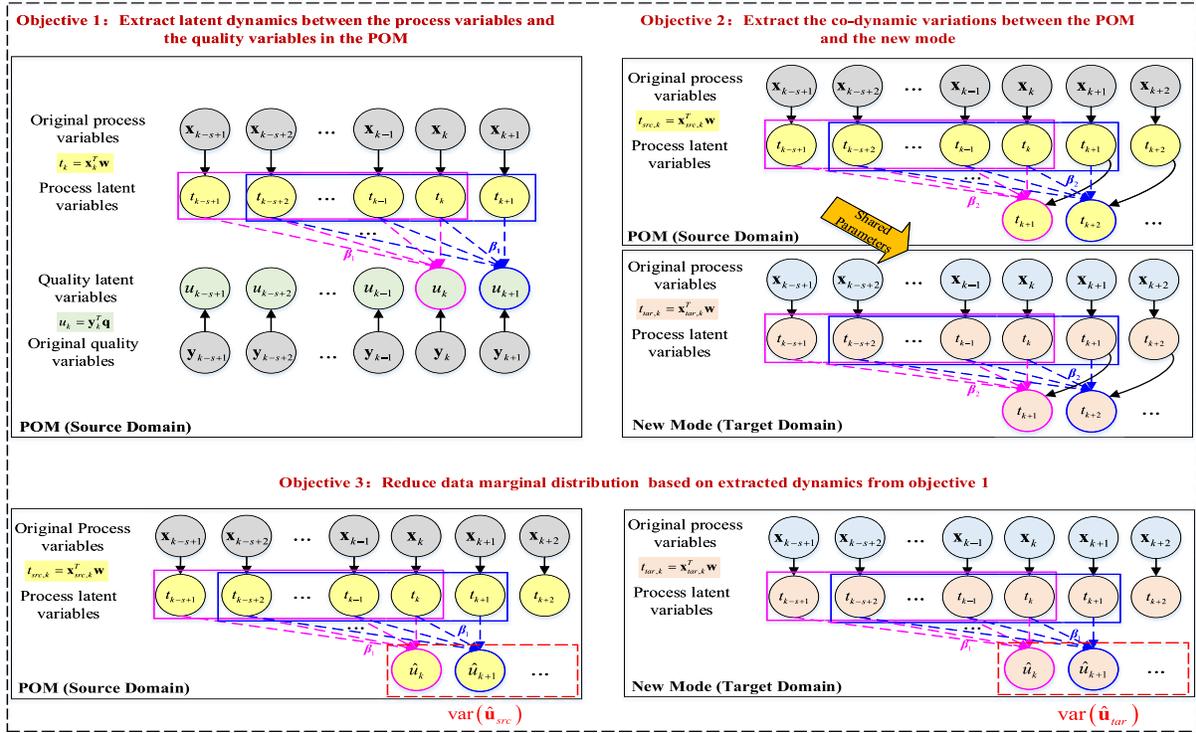


Fig. 1. Modeling framework of TDLVR.

where $\boldsymbol{\beta}_1 = (\beta_{11}, \dots, \beta_{1s})^\top$. DiCCA aims to maximize the correlation between $\hat{\mathbf{u}}_{src,k}$ and $u_{src,k}$, while DiPLS aims to maximize the covariance to enable the best predictability for quality variables. The DiPLS objective is utilized in this work and it is equivalent to minimizing the regularized distance between $\hat{\mathbf{u}}_{src,k}$ and $u_{src,k}$ as

$$\min_{\mathbf{q}, \mathbf{w}, \boldsymbol{\beta}_1} J_1 = \sum_{k=s}^{N_{src}} (u_{src,k} - \hat{u}_{src,k})^2. \quad (11)$$

The second objective is to extract co-dynamic variations between $t_{src,k+1}$ and $t_{tar,k+1}$. In the source domain, the dynamics of process variables using $\boldsymbol{\beta}_2$ can be described as

$$t_{src,k+1} = \sum_{i=1}^s \beta_{2i} t_{src,k-i+1} + r_{src,k+1}. \quad (12)$$

The same $\boldsymbol{\beta}_2$ is utilized to describe the dynamics within the process variables in the target domain as

$$t_{tar,k+1} = \sum_{i=1}^s \beta_{2i} t_{tar,k-i+1} + r_{tar,k+1}. \quad (13)$$

where $\boldsymbol{\beta}_2 = (\beta_{21}, \dots, \beta_{2s})^\top$. Both of the residuals $r_{src,k+1}$ and $r_{tar,k+1}$ are zero mean and uncorrelated in time. Since the labeled samples are not available for the target domain, it is not feasible to use DiPLS to model the dynamics between the process variables and quality variables for both the source domain and target domain models separately. To address this dynamic discrepancy, the same autoregression coefficients are used to model the domain-invariant dynamics among the process variables for the source domain and the target domain.

The inner model prediction for $t_{src,k+1}$ and $t_{tar,k+1}$ can be denoted as

$$\begin{aligned} \hat{t}_{src,k+1} &= \sum_{i=1}^s \beta_{2i} t_{src,k-i+1} = \sum_{i=1}^s \beta_{2i} \mathbf{x}_{src,k-i+1}^\top \mathbf{w} \\ \hat{t}_{tar,k+1} &= \sum_{i=1}^s \beta_{2i} t_{tar,k-i+1} = \sum_{i=1}^s \beta_{2i} \mathbf{x}_{tar,k-i+1}^\top \mathbf{w}. \end{aligned} \quad (14)$$

Remark 1: It is noted that the source domain and the target domain are similar but different, and the shared parameters are, thus, adopted to extract domain-invariant information for the multimode processes.

Similar to the DiCCA algorithm for one dataset in [46], both of the correlation between $t_{src,k+1}$ and $\hat{t}_{src,k+1}$, and the one between $t_{tar,k+1}$ and $\hat{t}_{tar,k+1}$ should be maximized to extract the co-dynamic variations as

$$\begin{aligned} \max_{\mathbf{w}, \boldsymbol{\beta}_2} J_2 &= \frac{\sum_{k=s}^{N_{src}} t_{src,k+1} \hat{t}_{src,k+1}}{\sqrt{\sum_{k=s}^{N_{src}} t_{src,k+1}^2} \sqrt{\sum_{k=s}^{N_{src}} \hat{t}_{src,k+1}^2}} \\ &+ \frac{\sum_{k=s}^{N_{tar}} t_{tar,k+1} \hat{t}_{tar,k+1}}{\sqrt{\sum_{k=s}^{N_{tar}} t_{tar,k+1}^2} \sqrt{\sum_{k=s}^{N_{tar}} \hat{t}_{tar,k+1}^2}} \\ &= \frac{\mathbf{t}_{src,s+1}^\top \hat{\mathbf{t}}_{src,s+1}}{\|\mathbf{t}_{src,s+1}\| \|\hat{\mathbf{t}}_{src,s+1}\|} + \frac{\mathbf{t}_{tar,s+1}^\top \hat{\mathbf{t}}_{tar,s+1}}{\|\mathbf{t}_{tar,s+1}\| \|\hat{\mathbf{t}}_{tar,s+1}\|}. \end{aligned} \quad (15)$$

In Eq. (15), the maximized correlations are to enable the best predictability for the latent process scores of the source domain and the target domain, respectively.

Moreover, according to $\mathbf{X}_{\text{src},i}$ and $\mathbf{X}_{\text{tar},i}$ in (6) and (7), the subvectors of LVs can be denoted as

$$\begin{aligned}\mathbf{t}_{\text{src},i} &= \mathbf{X}_{\text{src},i} \mathbf{w} \in \mathfrak{R}^{N_{\text{src}}-s} \\ \mathbf{u}_{\text{src},i} &= \mathbf{Y}_{\text{src},i} \mathbf{q} \in \mathfrak{R}^{N_{\text{src}}-s} \\ \mathbf{t}_{\text{tar},i} &= \mathbf{X}_{\text{tar},i} \mathbf{w} \in \mathfrak{R}^{N_{\text{tar}}-s}\end{aligned}\quad (16)$$

where $\mathbf{t}_{\text{src},i}$, $\mathbf{u}_{\text{src},i}$, and $\mathbf{t}_{\text{tar},i}$ denote the time-lagged process scores from the source domain, quality scores from the source domain, and process scores from the target domain, respectively. The corresponding predictions of $\mathbf{t}_{\text{src},s}$, $\mathbf{u}_{\text{src},i}$, and $\mathbf{t}_{\text{tar},s}$ can be organized as

$$\begin{aligned}\hat{\mathbf{t}}_{\text{src},s+1} &= \sum_{i=1}^s \beta_{2i} \mathbf{t}_{\text{src},s-i+1} = \sum_{i=1}^s \beta_{2i} \mathbf{X}_{\text{src},s-i+1} \mathbf{w} \\ &= \mathbf{T}_{\text{src},s} \boldsymbol{\beta}_2 = \mathbf{X}_{\text{src},\beta_2} \mathbf{w} \\ \hat{\mathbf{u}}_{\text{src},s} &= \sum_{i=1}^s \beta_{1i} \mathbf{t}_{\text{src},s-i+1} = \sum_{i=1}^s \beta_{1i} \mathbf{X}_{\text{src},s-i+1} \mathbf{w} \\ &= \mathbf{T}_{\text{src},s} \boldsymbol{\beta}_1 = \mathbf{X}_{\text{src},\beta_1} \mathbf{w} \\ \hat{\mathbf{t}}_{\text{tar},s+1} &= \sum_{i=1}^s \beta_{2i} \mathbf{t}_{\text{tar},s-i+1} = \sum_{i=1}^s \beta_{2i} \mathbf{X}_{\text{tar},s-i+1} \mathbf{w} \\ &= \mathbf{T}_{\text{tar},s} \boldsymbol{\beta}_2 = \mathbf{X}_{\text{tar},\beta_2} \mathbf{w}.\end{aligned}\quad (17)$$

Some derived relations are given as

$$\boldsymbol{\beta} \otimes \mathbf{w} = (\mathbf{I} \otimes \mathbf{w}) \boldsymbol{\beta} = (\boldsymbol{\beta} \otimes \mathbf{I}) \mathbf{w} \quad (18)$$

$$\mathbf{X}_{\text{src},\beta_j} = \mathbf{Z}_{\text{src},s} (\boldsymbol{\beta}_j \otimes \mathbf{I}) = \sum_{i=1}^s \beta_{ji} \mathbf{X}_{\text{src},s-i+1}, \quad j = 1, 2$$

$$\mathbf{X}_{\text{tar},\beta_2} = \mathbf{Z}_{\text{tar},s} (\boldsymbol{\beta}_2 \otimes \mathbf{I}) = \sum_{i=1}^s \beta_{2i} \mathbf{X}_{\text{tar},s-i+1}$$

$$\mathbf{T}_{\text{src},s} = [\mathbf{t}_{\text{src},s}, \dots, \mathbf{t}_{\text{src},1}] = \mathbf{Z}_{\text{src},s} (\mathbf{I} \otimes \mathbf{w})$$

$$\mathbf{T}_{\text{tar},s} = [\mathbf{t}_{\text{tar},s}, \dots, \mathbf{t}_{\text{tar},1}] = \mathbf{Z}_{\text{tar},s} (\mathbf{I} \otimes \mathbf{w}). \quad (19)$$

The prediction in (17) can be reformulated as

$$\begin{aligned}\hat{\mathbf{t}}_{\text{src},s+1} &= \mathbf{Z}_{\text{src},s} (\boldsymbol{\beta}_2 \otimes \mathbf{w}) = \mathbf{T}_{\text{src},s} \boldsymbol{\beta}_2 = \mathbf{X}_{\text{src},\beta_2} \mathbf{w} \\ \hat{\mathbf{u}}_{\text{src},s} &= \mathbf{Z}_{\text{src},s} (\boldsymbol{\beta}_1 \otimes \mathbf{w}) = \mathbf{T}_{\text{src},s} \boldsymbol{\beta}_1 = \mathbf{X}_{\text{src},\beta_1} \mathbf{w} \\ \hat{\mathbf{t}}_{\text{tar},s+1} &= \mathbf{Z}_{\text{tar},s} (\boldsymbol{\beta}_2 \otimes \mathbf{w}) = \mathbf{T}_{\text{tar},s} \boldsymbol{\beta}_2 = \mathbf{X}_{\text{tar},\beta_2} \mathbf{w}.\end{aligned}\quad (20)$$

Using (11) and (15), the dynamic relations between the process variables and the quality variables, and co-dynamic variations between the two domains are well addressed.

Remark 2: The main purpose of (12)–(15) is to find the co-dynamic variations among process variables between the source domain and the target domain using the same coefficients. By contrast, simply augmenting the source domain data and the target domain data and applying DiPLS is inappropriate from the following two aspects: 1) DiPLS can only learn dynamic relations while assuming the source domain and the target domain follow the same data distribution. For these two domains with different data distributions and dynamic relations, it is not feasible to use DiPLS directly on the augmented source domain and target domain data. 2) We expect to extract the co-dynamic variations between the source domain and the target domain, but only process data with no or a few labels in the target domain are available. This situation

makes DiPLS inapplicable to learning the dynamic relations between process data and quality data in the target domain.

The third objective is to eliminate data distribution discrepancy based on the extracted dynamics between the process variables and the quality variables. The inner model in (10) can be expressed as

$$\begin{aligned}\hat{\mathbf{u}}_{\text{src},s} &= \mathbf{X}_{\text{src},\beta_1} \mathbf{w} \\ \hat{\mathbf{u}}_{\text{tar},s} &= \mathbf{X}_{\text{tar},\beta_1} \mathbf{w}.\end{aligned}\quad (21)$$

The Euclidean norm of the variance difference between $\hat{\mathbf{u}}_{\text{src},k}$ and $\hat{\mathbf{u}}_{\text{tar},k}$ is adopted as a regularization term to reduce data distribution discrepancy. The objective can be written as

$$\begin{aligned}\min_{\mathbf{q}, \mathbf{w}, \boldsymbol{\beta}_1} J_3 &= \|\text{var}(\hat{\mathbf{u}}_{\text{src},s}) - \text{var}(\hat{\mathbf{u}}_{\text{tar},s})\|^2 \\ &= \left\| \mathbf{w}^\top \left(\frac{\mathbf{X}_{\text{src},\beta_1}^\top \mathbf{X}_{\text{src},\beta_1}}{N_{\text{src}} - s - 1} - \frac{\mathbf{X}_{\text{tar},\beta_1}^\top \mathbf{X}_{\text{tar},\beta_1}}{N_{\text{tar}} - s - 1} \right) \mathbf{w} \right\|^2.\end{aligned}\quad (22)$$

Using the expressions in (16), (19), and (21), the first objective can be reorganized to minimize $\|\mathbf{u}_{\text{src},s} - \hat{\mathbf{u}}_{\text{src},s}\|^2 = \|\mathbf{Y}_{\text{src},s} \mathbf{q} - \mathbf{X}_{\text{src},\beta_1} \mathbf{w}\|^2$. In summary, these three objectives are combined into a unified objective as

$$\begin{aligned}\min_{\mathbf{q}, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2} J &= \frac{\delta}{2} \|\mathbf{Y}_{\text{src},s} \mathbf{q} - \mathbf{X}_{\text{src},\beta_1} \mathbf{w}\|^2 \\ &\quad - (\mathbf{w}^\top \mathbf{X}_{\text{src},s+1}^\top \mathbf{T}_{\text{src},s} \boldsymbol{\beta}_2 + \mathbf{w}^\top \mathbf{X}_{\text{tar},s+1}^\top \mathbf{T}_{\text{tar},s} \boldsymbol{\beta}_2) \\ &\quad + \frac{1}{2} \left\| \mathbf{w}^\top \left(\frac{\mathbf{X}_{\text{src},\beta_1}^\top \mathbf{X}_{\text{src},\beta_1}}{N_{\text{src}} - s - 1} - \frac{\mathbf{X}_{\text{tar},\beta_1}^\top \mathbf{X}_{\text{tar},\beta_1}}{N_{\text{tar}} - s - 1} \right) \mathbf{w} \right\|^2 \\ \text{s.t. } &\|\mathbf{q}\| = 1, \|\boldsymbol{\beta}_1\| = \|\boldsymbol{\beta}_2\| = 1, \|\mathbf{X}_{\text{src},s+1} \mathbf{w}\| = 1 \\ &\|\mathbf{T}_{\text{src},s} \boldsymbol{\beta}_2\| = 1, \|\mathbf{X}_{\text{tar},s+1} \mathbf{w}\| = 1, \|\mathbf{T}_{\text{tar},s} \boldsymbol{\beta}_2\| = 1\end{aligned}\quad (23)$$

where δ is a tradeoff coefficient.

B. Latent Variable Extraction

Lagrange multipliers are used to solve the above optimization problem in (23). Define

$$\begin{aligned}L &= \frac{\delta}{2} \|\mathbf{Y}_{\text{src},s} \mathbf{q} - \mathbf{X}_{\text{src},\beta_1} \mathbf{w}\|^2 \\ &\quad - (\mathbf{w}^\top \mathbf{X}_{\text{src},s+1}^\top \mathbf{T}_{\text{src},s} \boldsymbol{\beta}_2 + \mathbf{w}^\top \mathbf{X}_{\text{tar},s+1}^\top \mathbf{T}_{\text{tar},s} \boldsymbol{\beta}_2) \\ &\quad + \frac{1}{2} \left\| \mathbf{w}^\top \left(\frac{\mathbf{X}_{\text{src},\beta_1}^\top \mathbf{X}_{\text{src},\beta_1}}{N_{\text{src}} - s - 1} - \frac{\mathbf{X}_{\text{tar},\beta_1}^\top \mathbf{X}_{\text{tar},\beta_1}}{N_{\text{tar}} - s - 1} \right) \mathbf{w} \right\|^2 \\ &\quad + \frac{\lambda_q}{2} (1 - \mathbf{q}^\top \mathbf{q}) - \frac{\lambda_{\beta_1}}{2} (1 - \boldsymbol{\beta}_1^\top \boldsymbol{\beta}_1) - \frac{\lambda_{\beta_2}}{2} (1 - \boldsymbol{\beta}_2^\top \boldsymbol{\beta}_2) \\ &\quad - \frac{\gamma}{2} (1 - \mathbf{w}^\top \mathbf{X}_{\text{src},s+1}^\top \mathbf{X}_{\text{src},s+1} \mathbf{w}) \\ &\quad - \frac{\gamma}{2} (1 - \mathbf{w}^\top \mathbf{X}_{\text{tar},s+1}^\top \mathbf{X}_{\text{tar},s+1} \mathbf{w}) \\ &\quad - \frac{\gamma}{2} (1 - \boldsymbol{\beta}_2^\top \mathbf{T}_{\text{src},s}^\top \mathbf{T}_{\text{src},s} \boldsymbol{\beta}_2) \\ &\quad - \frac{\gamma}{2} (1 - \boldsymbol{\beta}_2^\top \mathbf{T}_{\text{tar},s}^\top \mathbf{T}_{\text{tar},s} \boldsymbol{\beta}_2).\end{aligned}\quad (24)$$

The third term of the right side in (24) can be represented as $\phi(\mathbf{X}_{\text{src},\beta_1}^\top; \mathbf{X}_{\text{tar},\beta_1}^\top; \mathbf{w})$. Taking derivatives of L with respect

to \mathbf{q} , \mathbf{w} , β_1 , and β_2 , and setting to zero, the following relations are extracted similar to the work in [9]:

$$\frac{\partial L}{\partial \mathbf{q}} = \delta \mathbf{Y}_{\text{src},s}^\top (\mathbf{Y}_{\text{src},s} \mathbf{q} - \mathbf{X}_{\text{src},\beta_1} \mathbf{w}) - \lambda_q \mathbf{q} = \mathbf{0} \quad (25)$$

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{w}} = & -\delta \mathbf{X}_{\text{src},\beta_1}^\top (\mathbf{Y}_{\text{src},s} \mathbf{q} - \mathbf{X}_{\text{src},\beta_1} \mathbf{w}) \\ & - \mathbf{X}_{\text{src},s+1}^\top \mathbf{X}_{\text{src},\beta_2} \mathbf{w} - \mathbf{X}_{\text{src},\beta_2}^\top \mathbf{X}_{\text{src},s+1} \mathbf{w} \\ & - \mathbf{X}_{\text{tar},s+1}^\top \mathbf{X}_{\text{tar},\beta_2} \mathbf{w} - \mathbf{X}_{\text{tar},\beta_2}^\top \mathbf{X}_{\text{tar},s+1} \mathbf{w} \\ & + \phi'_w(\mathbf{X}_{\text{src},\beta_1}^\top; \mathbf{X}_{\text{tar},\beta_1}^\top; \mathbf{w}) \mathbf{w} \\ & + \gamma \mathbf{X}_{\text{src},s+1}^\top \mathbf{X}_{\text{src},s+1} \mathbf{w} + \gamma \mathbf{X}_{\text{tar},s+1}^\top \mathbf{X}_{\text{tar},s+1} \mathbf{w} \\ & + \gamma \mathbf{X}_{\text{src},\beta_2}^\top \mathbf{X}_{\text{src},\beta_2} \mathbf{w} + \gamma \mathbf{X}_{\text{tar},\beta_2}^\top \mathbf{X}_{\text{tar},\beta_2} \mathbf{w} = \mathbf{0} \end{aligned} \quad (26)$$

$$\begin{aligned} \frac{\partial L}{\partial \beta_1} = & -\delta \mathbf{T}_{\text{src},s}^\top (\mathbf{Y}_{\text{src},s} \mathbf{q} - \mathbf{T}_{\text{src},s} \beta_1) + \lambda_\beta \beta_1 \\ & + \phi'_{\beta_1}(\mathbf{X}_{\text{src},\beta_1}^\top; \mathbf{X}_{\text{tar},\beta_1}^\top; \mathbf{w}) \beta_1 = \mathbf{0} \end{aligned} \quad (27)$$

$$\begin{aligned} \frac{\partial L}{\partial \beta_2} = & -\mathbf{T}_{\text{src},s}^\top \mathbf{X}_{\text{src},s+1} \mathbf{w} - \mathbf{T}_{\text{tar},s}^\top \mathbf{X}_{\text{tar},s+1} \mathbf{w} \\ & + \lambda_\beta \beta_2 + \gamma \mathbf{T}_{\text{src},s}^\top \mathbf{T}_{\text{src},s} \beta_2 \\ & + \gamma \mathbf{T}_{\text{tar},s}^\top \mathbf{T}_{\text{tar},s} \beta_2 = \mathbf{0}. \end{aligned} \quad (28)$$

Equation (25) can be simplified as

$$\delta \mathbf{Y}_{\text{src},s}^\top (\mathbf{Y}_{\text{src},s} \mathbf{q} - \mathbf{X}_{\text{src},\beta_1} \mathbf{w}) = \lambda_q \mathbf{q}. \quad (29)$$

In (26), the expression $\phi'_w(\mathbf{X}_{\text{src},\beta_1}^\top; \mathbf{X}_{\text{tar},\beta_1}^\top; \mathbf{w})$ can be further denoted as

$$\begin{aligned} \mathbf{D}_1 = & \frac{\mathbf{X}_{\text{src},\beta_1}^\top \mathbf{X}_{\text{src},\beta_1}}{N_{\text{src}} - s - 1} - \frac{\mathbf{X}_{\text{tar},\beta_1}^\top \mathbf{X}_{\text{tar},\beta_1}}{N_{\text{tar}} - s - 1} \\ & \phi'_w(\mathbf{X}_{\text{src},\beta_1}^\top; \mathbf{X}_{\text{tar},\beta_1}^\top; \mathbf{w}) = \mathbf{D}_1 \mathbf{w} \mathbf{w}^\top \mathbf{D}_1. \end{aligned} \quad (30)$$

Substituting (30) into (26) and denoting

$$\begin{aligned} \Pi = & \mathbf{D}_1 \mathbf{w} \mathbf{w}^\top \mathbf{D}_1 - \mathbf{X}_{\text{src},s+1}^\top \mathbf{X}_{\text{src},\beta_2} - \mathbf{X}_{\text{src},\beta_2}^\top \mathbf{X}_{\text{src},s+1} \\ & - \mathbf{X}_{\text{tar},s+1}^\top \mathbf{X}_{\text{tar},\beta_2} - \mathbf{X}_{\text{tar},\beta_2}^\top \mathbf{X}_{\text{tar},s+1} \\ & + \gamma \mathbf{X}_{\text{src},s+1}^\top \mathbf{X}_{\text{src},s+1} + \gamma \mathbf{X}_{\text{tar},s+1}^\top \mathbf{X}_{\text{tar},s+1} \\ & + \gamma \mathbf{X}_{\text{src},\beta_2}^\top \mathbf{X}_{\text{src},\beta_2} + \gamma \mathbf{X}_{\text{tar},\beta_2}^\top \mathbf{X}_{\text{tar},\beta_2} \end{aligned} \quad (31)$$

we have the following relation from (26):

$$\mathbf{w} = \left(\frac{1}{\delta} \Pi + \mathbf{X}_{\text{src},\beta_1}^\top \mathbf{X}_{\text{src},\beta_1} \right)^{-1} \mathbf{X}_{\text{src},\beta_1}^\top \mathbf{u}_{\text{src},s}. \quad (32)$$

In (27), the expression $\phi'_{\beta_1}(\mathbf{X}_{\text{src},\beta_1}^\top; \mathbf{X}_{\text{tar},\beta_1}^\top; \mathbf{w})$ can be further denoted as

$$\begin{aligned} \mathbf{D}_2 = & \frac{\mathbf{T}_{\text{src},s}^\top \mathbf{T}_{\text{src},s}}{N_{\text{src}} - s - 1} - \frac{\mathbf{T}_{\text{tar},s}^\top \mathbf{T}_{\text{tar},s}}{N_{\text{tar}} - s - 1} \\ & \phi'_{\beta_1}(\mathbf{X}_{\text{src},\beta_1}^\top; \mathbf{X}_{\text{tar},\beta_1}^\top; \mathbf{w}) = \mathbf{D}_2 \beta_1 \beta_1^\top \mathbf{D}_2. \end{aligned} \quad (33)$$

In order to calculate β_1 and β_2 , the relations between λ_q , γ , and λ_β are derived in the following.

Premultiplying (29) by \mathbf{q}^\top , we have

$$\delta \mathbf{q}^\top \mathbf{Y}_{\text{src},s}^\top (\mathbf{Y}_{\text{src},s} \mathbf{q} - \mathbf{X}_{\text{src},\beta_1} \mathbf{w}) = \lambda_q \mathbf{q}^\top \mathbf{q} = \lambda_q. \quad (34)$$

Similarly, premultiplying (26)–(28) by \mathbf{w}^\top , β_1^\top , and β_2^\top , we have

$$\lambda_q + 2\Theta_1 - \Theta_2 - \gamma = 0 \quad (35)$$

$$\lambda_q - \lambda_\beta - \Theta_2 = 0 \quad (36)$$

$$\Theta_1 - \lambda_\beta - \gamma = 0 \quad (37)$$

where $\Theta_1 = \mathbf{w}^\top (\mathbf{X}_{\text{src},s+1}^\top \mathbf{X}_{\text{src},\beta_2} \mathbf{w} + \mathbf{X}_{\text{tar},s+1}^\top \mathbf{X}_{\text{tar},\beta_2} \mathbf{w})$, $\Theta_2 = \phi(\mathbf{X}_{\text{src},\beta_1}^\top; \mathbf{X}_{\text{tar},\beta_1}^\top; \mathbf{w})$, which leads to $3\lambda_\beta + \gamma = 0$. Using this relation, γ in (31) can be replaced by $-3\lambda_\beta$. Equations (27) and (28) can be calculated as

$$\beta_1 = (\mathbf{D}_2 \beta_1 \beta_1^\top \mathbf{D}_2 + \delta \mathbf{T}_{\text{src},s}^\top \mathbf{T}_{\text{src},s} + \lambda_\beta \mathbf{I})^{-1} \mathbf{T}_{\text{src},s}^\top \mathbf{u}_{\text{src},s} \quad (38)$$

$$\beta_2 = \frac{1}{\lambda_\beta} (\mathbf{I} - 3(\mathbf{T}_{\text{src},s}^\top \mathbf{T}_{\text{src},s} + \mathbf{T}_{\text{tar},s}^\top \mathbf{T}_{\text{tar},s}))^{-1} (\mathbf{T}_{\text{src},s}^\top \mathbf{T}_{\text{src},s+1} + \mathbf{T}_{\text{tar},s}^\top \mathbf{T}_{\text{tar},s+1}). \quad (39)$$

In order to be consistent with the dynamic outer model, for the source domain, a regression model is built to describe the relation between $\mathbf{u}_{\text{src},s}$ and $(\mathbf{t}_{\text{src},s}, \mathbf{t}_{\text{src},s-1}, \dots, \mathbf{t}_{\text{src},1})$.

$$\mathbf{u}_{\text{src},s} = \alpha_1 \mathbf{t}_{\text{src},s} + \alpha_2 \mathbf{t}_{\text{src},s-1} + \dots + \alpha_s \mathbf{t}_{\text{src},1} + \mathbf{r}_{\text{src},s} \quad (40)$$

where $\mathbf{r}_{\text{src},s}$ is a regression error that should be minimized.

The solution of (48) is, thus, obtained as

$$\boldsymbol{\alpha} = (\mathbf{T}_{\text{src},s}^\top \mathbf{T}_{\text{src},s})^{-1} \mathbf{T}_{\text{src},s}^\top \mathbf{u}_{\text{src},s} \quad (41)$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_s]^\top$. The predicted quality scores $\hat{\mathbf{u}}_{\text{src},s}$ can be represented as

$$\begin{aligned} \hat{\mathbf{u}}_{\text{src},s} &= \mathbf{T}_{\text{src},s} \boldsymbol{\alpha} \\ &= \mathbf{T}_{\text{src},s} (\mathbf{T}_{\text{src},s}^\top \mathbf{T}_{\text{src},s})^{-1} \mathbf{T}_{\text{src},s}^\top \mathbf{u}_{\text{src},s}. \end{aligned} \quad (42)$$

Remark 3: If $\mathbf{Y}_{\text{src},s}$ has only one output to be modeled, $\mathbf{q} = 1$.

Similar to the work of [34], δ in (32) is obtained by the following equation with the weight \mathbf{w} computed from the DiPLS algorithm:

$$\delta = \frac{\mathbf{w}^\top \Pi \mathbf{w}}{\mathbf{w}^\top (\mathbf{X}_{\text{src},\beta_1}^\top \mathbf{u}_{\text{src},s} - \mathbf{X}_{\text{src},\beta_1}^\top \mathbf{X}_{\text{src},\beta_1} \mathbf{w})}. \quad (43)$$

C. TDLVR-Based Quality Prediction

After modeling the outer and inner structures of TDLVR, the loading vectors \mathbf{p}_{src} for \mathbf{X}_{src} and \mathbf{p}_{tar} for \mathbf{X}_{tar} can be calculated by minimizing the Frobenius norm of the corresponding residuals between \mathbf{X}_{src} and $\mathbf{t}_{\text{src}} \mathbf{p}_{\text{src}}^\top$ and between \mathbf{X}_{tar} and $\mathbf{t}_{\text{tar}} \mathbf{p}_{\text{tar}}^\top$. The detailed expressions are as

$$\mathbf{p}_{\text{src}} = \mathbf{X}_{\text{src}}^\top \mathbf{t}_{\text{src}} / \mathbf{t}_{\text{src}}^\top \mathbf{t}_{\text{src}} \quad (44)$$

$$\mathbf{p}_{\text{tar}} = \mathbf{X}_{\text{tar}}^\top \mathbf{t}_{\text{tar}} / \mathbf{t}_{\text{tar}}^\top \mathbf{t}_{\text{tar}}. \quad (45)$$

It is noted that labeled samples can only be available in the source domain, rather than the target domain. The loading vector \mathbf{q}_{src} for $\mathbf{Y}_{\text{src},s}$ in the source domain can be obtained in a similar way as

$$\mathbf{q}_{\text{src}} = \mathbf{Y}_{\text{src},s}^\top \hat{\mathbf{u}}_{\text{src},s} / \hat{\mathbf{u}}_{\text{src},s}^\top \hat{\mathbf{u}}_{\text{src},s}. \quad (46)$$

\mathbf{X}_{src} , \mathbf{X}_{tar} , and $\mathbf{Y}_{\text{src},s}$ are deflated by the following:

$$\mathbf{X}_{\text{src}} := \mathbf{X}_{\text{src}} - \mathbf{t}_{\text{src}} \mathbf{p}_{\text{src}}^\top \quad (47)$$

$$\mathbf{X}_{\text{tar}} := \mathbf{X}_{\text{tar}} - \mathbf{t}_{\text{tar}} \mathbf{p}_{\text{tar}}^\top \quad (48)$$

$$\mathbf{Y}_{\text{src},s} := \mathbf{Y}_{\text{src},s} - \hat{\mathbf{u}}_{\text{src},s} \mathbf{q}_{\text{src}}^\top. \quad (49)$$

The next latent component of TDLVR can be extracted from the deflated \mathbf{X}_{src} , \mathbf{X}_{tar} , and $\mathbf{Y}_{\text{src},s}$ by conducting the same procedure of outer structure modeling, inner structure modeling, and

deflation iteratively. l possibly DLVs are extracted to represent the most common variations of \mathbf{X}_{src} and $\mathbf{Y}_{\text{src},s}$ with \mathbf{X}_{tar} .

After performing TDLVR, \mathbf{X}_{src} , \mathbf{X}_{tar} , and $\mathbf{Y}_{\text{src},s}$ are decomposed as

$$\begin{aligned}\mathbf{X}_{\text{src}} &= \sum_{i=1}^l \mathbf{t}_{\text{src},i} \mathbf{p}_{\text{src},i}^{\top} + \mathbf{E}_{\text{src}} = \mathbf{T}_{\text{src}} \mathbf{P}_{\text{src}}^{\top} + \mathbf{E}_{\text{src}} \\ \mathbf{X}_{\text{tar}} &= \sum_{i=1}^l \mathbf{t}_{\text{tar},i} \mathbf{p}_{\text{tar},i}^{\top} + \mathbf{E}_{\text{tar}} = \mathbf{T}_{\text{tar}} \mathbf{P}_{\text{tar}}^{\top} + \mathbf{E}_{\text{tar}} \\ \mathbf{Y}_{\text{src},s} &= \sum_{i=1}^l \hat{\mathbf{u}}_{\text{src},s,i} \mathbf{q}_{\text{src},i}^{\top} + \mathbf{F}_{\text{src}} = \hat{\mathbf{U}}_{\text{src},s} \mathbf{Q}_{\text{src}}^{\top} + \mathbf{F}_{\text{src}} \quad (50)\end{aligned}$$

where $\hat{\mathbf{U}}_{\text{src},s} = [\hat{\mathbf{u}}_{\text{src},s,1}, \hat{\mathbf{u}}_{\text{src},s,2}, \dots, \hat{\mathbf{u}}_{\text{src},s,l}] \in \mathfrak{R}^{(N_{\text{src}}-s) \times l}$, $\hat{\mathbf{u}}_{\text{src},s,i} = \mathbf{T}_{\text{src},s,i} \boldsymbol{\alpha}_i$, $\mathbf{T}_{\text{src},s,i} = [\mathbf{t}_{\text{src},s,i}, \mathbf{t}_{\text{src},(s-1),i}, \dots, \mathbf{t}_{\text{src},1,i}] \in \mathfrak{R}^{(N_{\text{src}}-s) \times s}$, and $\mathbf{T}_{\text{tar},s,i} = [\mathbf{t}_{\text{tar},s,i}, \mathbf{t}_{\text{tar},(s-1),i}, \dots, \mathbf{t}_{\text{tar},1,i}] \in \mathfrak{R}^{(N_{\text{tar}}-s) \times s}$. \mathbf{E}_{src} and \mathbf{E}_{tar} are the process data residuals for the source domain and the target domain, respectively. \mathbf{F}_{src} are the quality data residuals for the source domain.

From (50), predicted quality is represented as

$$\hat{\mathbf{Y}}_{\text{tar,tst},s} = \hat{\mathbf{U}}_{\text{tar,tst},s} \mathbf{Q}_{\text{src}}^{\top} \quad (51)$$

where $\hat{\mathbf{U}}_{\text{tar,tst},s}$ can be obtained from $\mathbf{T}_{\text{tar,tst},s}$, $\mathbf{T}_{\text{tar,tst},s} = \mathbf{X}_{\text{tar,tst},s} \mathbf{R}$, and $\mathbf{R} = \mathbf{W}(\mathbf{P}_{\text{src}}^{\top} \mathbf{W})^{-1}$. Hence, $\hat{\mathbf{Y}}_{\text{tar,tst},s}$ can be predicted from $\mathbf{X}_{\text{tar,tst}}$ directly. The parameters \mathbf{Q}_{src} and $\boldsymbol{\alpha}_i$ learned from the source domain are used for quality prediction of the target domain. The proposed TDLVR aims to reduce marginal distribution differences using the unlabeled process data samples in the target domain when labeled samples are not available in the target domain and the marginal distribution difference is the dominant factor of data distribution. The detailed TDLVR algorithm is summarized in Algorithm 1. It can be seen that the nonlinear iterative partial least-square (NIPALS) algorithm is used to solve the optimization problem in (23). The convergence of the NIPALS algorithm is achieved as demonstrated in [48]. The iteration is considered to converge when the two-norm of the difference between the extracted process data score and the one of the last iteration is fewer than a threshold or when the number of iterations reaches a predefined maximum. s is determined by prior knowledge. l and λ_{β} are selected by grid search.

IV. COMPENSATED TDLVR USING AVAILABLE LABELED SAMPLES IN TARGET DOMAIN

To further address the conditional probability distribution discrepancy (CPDD) between domains, a new CTDVLR algorithm using the newly collected labeled samples in the target domain is proposed.

A. Motivation and Strategy for Solving CPDD

For transfer learning, the distribution discrepancy that mainly describes the joint probability distribution $P(\mathbf{X}, \mathbf{Y})$ is inconsistent between the source domain and the target domain, i.e., $P(\mathbf{X}_{\text{src}}, \mathbf{Y}_{\text{src}}) \neq P(\mathbf{X}_{\text{tar}}, \mathbf{Y}_{\text{tar}})$. In fact, the joint probability distribution discrepancy (JPDD) can be further divided into two parts: marginal probability distribution discrepancy (MPDD), i.e., $P(\mathbf{X}_{\text{src}}) \neq P(\mathbf{X}_{\text{tar}})$ and CPDD, i.e.,

Algorithm 1 Proposed TDLVR Method

Input: $\mathbf{X}_{\text{src}} \in \mathfrak{R}^{N_{\text{src}} \times m}$, $\mathbf{Y}_{\text{src}} \in \mathfrak{R}^{N_{\text{src}} \times 1}$; $\mathbf{X}_{\text{tar}} \in \mathfrak{R}^{N_{\text{tar}} \times m}$; $\mathbf{X}_{\text{tar,tst}} \in \mathfrak{R}^{N_{\text{tar,tst}} \times m}$; l ; s ; λ_{β} .
Output: \mathbf{W} , \mathbf{P}_{src} , \mathbf{Q}_{src} and $\{\boldsymbol{\alpha}_i\}_{i=1}^l$.

Offline Training

- 1: Normalize \mathbf{X}_{src} , \mathbf{Y}_{src} , \mathbf{X}_{tar} , and $\mathbf{X}_{\text{tar,tst}}$. Construct the corresponding augmented matrix $\mathbf{Z}_{\text{src},s}$, $\mathbf{Z}_{\text{tar},s}$, and $\mathbf{Y}_{\text{src},s}$.
- 2: Initialize $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ with $[1, 0, \dots, 0]^{\top}$ and $\mathbf{u}_{\text{src},s}$ as some column of $\mathbf{Y}_{\text{src},s}$.
- 3: **for** $i = 1, \dots, l$ LVs **do**
- 4: Initialize \mathbf{w} with a random unit vector.
- 5: **Outer modeling:** Iterate the following relations until convergence.
 - 6: a. Obtain δ by Eq. (43);
 - 7: b. Calculate \mathbf{w} from Eq. (32) and let $\mathbf{w} = \mathbf{w}/\|\mathbf{w}\|$;
 - 8: c. Construct $\mathbf{T}_{\text{src},s} = [\mathbf{t}_{\text{src},s}, \dots, \mathbf{t}_{\text{src},1}]$ with

$$\mathbf{t}_{\text{src}} = \mathbf{X}_{\text{src}} \mathbf{w};$$
 - 9: d. Construct $\mathbf{T}_{\text{tar},s} = [\mathbf{t}_{\text{tar},s}, \dots, \mathbf{t}_{\text{tar},1}]$ with

$$\mathbf{t}_{\text{tar}} = \mathbf{X}_{\text{tar}} \mathbf{w};$$
 - 10: e. Find \mathbf{q} from Eq. (29) and let $\mathbf{q} = \mathbf{q}/\|\mathbf{q}\|$;
 - 11: f. Obtain $\mathbf{u}_{\text{src},s} = \mathbf{Y}_{\text{src},s} \mathbf{q}$;
 - 12: g. Calculate $\boldsymbol{\beta}_1$ from Eq. (38) and let $\boldsymbol{\beta}_1 = \boldsymbol{\beta}_1/\|\boldsymbol{\beta}_1\|$;
 - 13: h. Calculate $\boldsymbol{\beta}_2$ from Eq. (39) and let $\boldsymbol{\beta}_2 = \boldsymbol{\beta}_2/\|\boldsymbol{\beta}_2\|$;
- 14: **Inner modeling:** Build a linear model between $\mathbf{T}_{\text{src},s}$ and $\mathbf{u}_{\text{src},s}$ by

$$\boldsymbol{\alpha} = (\mathbf{T}_{\text{src},s}^{\top} \mathbf{T}_{\text{src},s})^{-1} \mathbf{T}_{\text{src},s}^{\top} \mathbf{u}_{\text{src},s};$$
- 15: Calculate $\hat{\mathbf{u}}_{\text{src},s} = \mathbf{T}_{\text{src},s} \boldsymbol{\alpha}$;
- 16: **Deflation**
- 17: Calculate $\mathbf{p}_{\text{src}} = \mathbf{X}_{\text{src}}^{\top} \mathbf{t}_{\text{src}} / \mathbf{t}_{\text{src}}^{\top} \mathbf{t}_{\text{src}}$;
- 18: Deflate $\mathbf{X}_{\text{src}} := \mathbf{X}_{\text{src}} - \mathbf{t}_{\text{src}} \mathbf{p}_{\text{src}}^{\top}$;
- 19: Calculate $\mathbf{p}_{\text{tar}} = \mathbf{X}_{\text{tar}}^{\top} \mathbf{t}_{\text{tar}} / \mathbf{t}_{\text{tar}}^{\top} \mathbf{t}_{\text{tar}}$;
- 20: Deflate $\mathbf{X}_{\text{tar}} := \mathbf{X}_{\text{tar}} - \mathbf{t}_{\text{tar}} \mathbf{p}_{\text{tar}}^{\top}$;
- 21: Calculate $\mathbf{q}_{\text{src}} = \mathbf{Y}_{\text{src},s}^{\top} \hat{\mathbf{u}}_{\text{src},s} / \hat{\mathbf{u}}_{\text{src},s}^{\top} \hat{\mathbf{u}}_{\text{src},s}$;
- 22: Deflate $\mathbf{Y}_{\text{src},s} := \mathbf{Y}_{\text{src},s} - \hat{\mathbf{u}}_{\text{src},s} \mathbf{q}_{\text{src}}^{\top}$;
- 23: Store $\mathbf{W}(:, i) = \mathbf{w}$; $\mathbf{P}_{\text{src}}(:, i) = \mathbf{p}_{\text{src}}$; $\boldsymbol{\alpha}_i = \boldsymbol{\alpha}$;
- 24: $\mathbf{Q}_{\text{src}}(:, i) = \mathbf{q}_{\text{src}}$;
- 25: **end for**
- 26: **return** \mathbf{W} , \mathbf{P}_{src} , \mathbf{Q}_{src} and $\{\boldsymbol{\alpha}_i\}_{i=1}^l$ used for online prediction.

$P(\mathbf{Y}_{\text{src}}|\mathbf{X}_{\text{src}}) \neq P(\mathbf{Y}_{\text{tar}}|\mathbf{X}_{\text{tar}})$. Some transfer learning methods, known as unsupervised domain adaptation, focus on the situation when the MPDD is a major contributor to the JPDD, provided that the CPDD is small to be omitted [49], [50]. In practice, there may be a significant CPDD. To address the CPDD for the TL methods, it is common to incorporate a compensation mechanism such as offset approach [33], and LV represented conditional distribution alignment [51]. In this work, the online compensation model in the TDLVR framework is proposed. Notably, the proposed TDLVR in Section III can conduct the MPDD but not the CPDD since no labeled samples in the target domain are involved in the training stage.

To address the CPDD well, an error compensation model is established using available labeled samples in the target domain. The output of the error compensation model and the one of TDLVR are combined to establish a CTDLVR. On the basis of the reduction of MPDD by TDLVR, CTDLVR outperforms TDLVR in eliminating CPDD to some extent.

B. Proposed CTDLVR

Suppose sufficient labeled training samples are available in the POM as $\{\mathbf{X}_{\text{src}}^L, \mathbf{Y}_{\text{src}}^L\}$. Meanwhile, sufficient unlabeled training samples $\{\mathbf{X}_{\text{tar}}^U\}$ and the limited labeled training samples $\{\mathbf{X}_{\text{tar}}^L, \mathbf{Y}_{\text{tar}}^L\}$ are available in the target domain, respectively. During the CTDLVR modeling, TDLVR in Section III is first used to extract the common dynamics between labeled training samples in the source domain and unlabeled training samples in the target domain as

$$f_{\text{TDLVR}}(\cdot) : (\mathbf{X}_{\text{src}}^L; \mathbf{X}_{\text{tar}}^U) \rightarrow \mathbf{Y}_{\text{src}}^L \quad (52)$$

where $f_{\text{TDLVR}}(\cdot)$ is learned by using the proposed TDLVR.

TDLVR is used as the main model to perform quality predicting based on $\mathbf{X}_{\text{tar}}^L$

$$\hat{\mathbf{Y}}_{\text{tar}}^L = f_{\text{TDLVR}}(\mathbf{X}_{\text{tar}}^L) \quad (53)$$

where $\hat{\mathbf{Y}}_{\text{tar}}^L$ represents predicted labels in the target domain. Herein, the CPDD can be referred to as the offset between label values. To overcome the difficulty of TDLVR in handling the CPDD, a DiPLS-based error compensation model is established.

The errors between the predicted labels and the actual labels are first calculated as

$$\hat{\mathbf{Z}}_{\text{tar}}^L = \mathbf{Y}_{\text{tar}}^L - \hat{\mathbf{Y}}_{\text{tar}}^L \quad (54)$$

where $\hat{\mathbf{Z}}_{\text{tar}}^L$ represents the offset value. In fact, the offset value can be regarded as a specific part of the target domain. Therefore, the input feature of the limited labeled data in the target domain $\mathbf{X}_{\text{tar}}^L$ and the offset value $\hat{\mathbf{Z}}_{\text{tar}}^L$ is reorganized as a new training dataset $\{\mathbf{X}_{\text{tar}}^L, \hat{\mathbf{Z}}_{\text{tar}}^L\}$ for establishing the DiPLS-based error compensation model. The model can be described as

$$f_{\text{DiPLS}}(\cdot) : \mathbf{X}_{\text{tar}}^L \rightarrow \hat{\mathbf{Z}}_{\text{tar}}^L \quad (55)$$

where $f_{\text{DiPLS}}(\cdot)$ is the mapping function learned by applying the DiPLS algorithm in [9] on the training dataset $\{\mathbf{X}_{\text{tar}}^L, \hat{\mathbf{Z}}_{\text{tar}}^L\}$.

Finally, the output of the CTDLVR model consists of two parts, including the main prediction by TDLVR and the error prediction by DiPLS. The predicted output of the testing data is described as

$$\begin{aligned} \hat{\mathbf{Y}}_{\text{tar,tst}} &= f_{\text{CTDLVR}}(\mathbf{X}_{\text{tar,tst}}) \\ &= f_{\text{TDLVR}}(\mathbf{X}_{\text{tar,tst}}) + f_{\text{DiPLS}}(\mathbf{X}_{\text{tar,tst}}) \end{aligned} \quad (56)$$

where $f_{\text{CTDLVR}}(\cdot)$ is the mapping function of CTDLVR.

V. CASE STUDIES

In this section, numerical simulation examples and two industrial process examples are applied to demonstrate the effectiveness of the proposed TDLVR and CTDLVR.

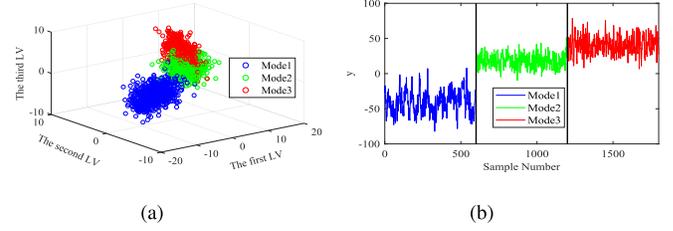


Fig. 2. Data distribution discrepancy for multimode processes. (a) Three-dimensional distribution scatter of process LVs. (b) Data discrepancy of quality variables.

The proposed methods are compared with traditional LV models including PLS and DiPLS, and TL models, including di-PLS, dynamic di-PLS, TCA, CORAL, DANN, Wasserstein distance guided representation learning (WDGRL) [29], and margin disparity discrepancy (MDD) [30]. The dynamic di-PLS is a dynamic version of di-PLS by using data time augmentation.

A. Numerical Simulation Example

The process data \mathbf{X} and quality data \mathbf{Y} are generated by a dynamic process as

$$\begin{aligned} \mathbf{t}_k &= \mathbf{A}_1 \mathbf{t}_{k-1} - \mathbf{A}_2 \mathbf{t}_{k-2} + \mathbf{t}_k^* + \mathbf{f}_k \\ \mathbf{x}_k &= \mathbf{P} \mathbf{t}_k + \mathbf{e}_k \\ \mathbf{y}_k &= \mathbf{C}_1^\top \mathbf{x}_k + \mathbf{C}_2^\top \mathbf{x}_{k-1} + \mathbf{v}_k \\ \mathbf{t}_k^* &= \mathbf{t} \mathbf{0}_k, \quad k \geq 3 \end{aligned} \quad (57)$$

$$\mathbf{P} = \begin{bmatrix} 0.5586 & 0.2042 & 0.6370 \\ 0.2007 & 0.0492 & 0.4492 \\ 0.0874 & 0.6062 & 0.0664 \\ 0.9332 & 0.5463 & 0.3743 \\ 0.2594 & 0.0958 & 0.2491 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 0.7451 & 1.9939 \\ 0.4928 & 0.7728 \\ 0.7329 & 1.0146 \\ 0.47383 & 1.1563 \\ 0.5652 & 1.2307 \end{bmatrix}$$

where $\mathbf{e}_k \in \mathbb{R}^3 \sim N(0, 0.01)$, $\mathbf{f}_k \in \mathbb{R}^3 \sim N(0, 0.01)$, $\mathbf{v}_k \in \mathbb{R}^1 \sim N(0, 0.01)$, $\mathbf{C} = [\mathbf{C}_1 \mathbf{C}_2]$, and $\mathbf{t} \mathbf{0}_k$ follows different Gaussian distributions for three modes, respectively. Meanwhile, \mathbf{A}_1 and \mathbf{A}_2 in one mode are different from those in another mode. The parameters are listed in Table I.

A total of 1800 samples are collected from three modes with every 600 samples. The number of LVs is chosen as 3 for Mode 1 (i.e., M1), Mode 2 (i.e., M2), and Mode 3 (i.e., M3), respectively. The LVs follow Gaussian distributions, i.e., $(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, $(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, and $(\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$. In addition, to import intermodel varying dynamics, time-lag matrices \mathbf{A}_1 and \mathbf{A}_2 are set. In contrast to M1, only \mathbf{A}_1 varies in M2, while \mathbf{A}_2 remains as it is. Different from Mode 1 and Mode 2, \mathbf{A}_1 and \mathbf{A}_2 all vary in M3. As a result, the 3-D distribution scatter of process LVs and data discrepancy of quality variables are shown in Fig. 2. It can be observed that the 3-D distributions are significantly different for the three modes.

The first 300 samples for each mode are selected as training datasets, and the remaining 300 samples are regarded as testing datasets. In addition, one of the three modes is randomly selected as the source domain and the rest as the target domain. For each scenario, all baseline methods and the

TABLE I
PARAMETERS FOR SIMULATED MULTIMODE PROCESSES

Mode No.	A_1			A_2			$t\mathbf{0}_k \sim (\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), i = (1, 2, 3)$			
							$\boldsymbol{\mu}$	$\boldsymbol{\Sigma}$		
Mode 1	0.4389	0.1210	-0.0862	-0.2998	-0.1905	-0.2669	-1.3	3.5	1.2	1.89
	-0.2966	-0.1550	0.2274	-0.0204	-0.1585	-0.2950	-1.7	1.2	1.26	0.46
	0.3538	-0.6573	0.4239	0.1461	-0.0755	0.3749	-1.1	1.89	0.46	0.64
Mode 2	0.2389	0.1210	-0.0862	-0.2998	-0.1905	-0.2669	1.0	3.4	1.3	0.66
	-0.2966	-0.0550	0.2274	-0.0204	-0.1585	-0.2950	0.0	1.3	1.2	-0.05
	0.0154	-0.6573	0.0424	0.1461	-0.0755	0.3749	3.0	0.66	-0.05	0.78
Mode 3	0.2334	-0.0836	0.2573	-0.1998	-0.0905	-0.2669	1.0	1.49	2.3	-0.7
	0.1783	-0.1590	0.3416	-0.2141	-0.1585	-0.3950	-2.0	2.3	1.25	-0.96
	0.4345	-0.4573	0.3239	-0.1461	0.6755	-0.3749	1.2	-0.7	-0.96	0.66

TABLE II

COMPARISON OF PREDICTION PERFORMANCE OF TDLVR, CTDLVR, AND TRADITIONAL APPROACHES FOR NUMERICAL SIMULATION EXAMPLES

Methods	MSE/MAE						Mean MSE/MAE/PCC
	M1 → M2	M1 → M3	M2 → M1	M2 → M3	M3 → M1	M3 → M2	
PLS	85.70 / 7.51	78.80 / 7.19	121.06 / 9.03	112.46 / 8.60	177.50 / 10.70	133.03 / 9.24	118.09 / 8.71 / 0.60
DiPLS [9]	12.47 / 2.79	7.21 / 2.19	24.33 / 4.11	42.64 / 5.39	92.49 / 7.84	49.28 / 5.54	38.07 / 4.64 / 0.87
CORAL [38]	87.07 / 7.55	77.41 / 7.14	126.38 / 9.25	114.69 / 8.70	177.34 / 10.70	131.22 / 9.19	119.02 / 8.76 / 0.55
TCA [37]	137.56 / 9.55	166.84 / 10.38	317.51 / 14.25	175.24 / 10.70	232.35 / 12.26	72.54 / 6.62	183.67 / 10.63 / 0.21
WDGRL [29]	62.04 / 6.22	53.66 / 5.89	90.49 / 7.60	73.30 / 6.99	83.89 / 7.23	89.31 / 7.42	75.45 / 6.89 / 0.68
MDD [30]	65.32 / 6.36	56.10 / 6.09	97.76 / 7.91	84.97 / 7.48	89.34 / 7.52	107.37 / 8.08	83.48 / 7.24 / 0.65
DANN [40]	60.34 / 6.32	67.06 / 6.57	89.53 / 7.46	105.37 / 7.99	101.60 / 8.02	76.43 / 7.15	83.39 / 7.25 / 0.56
di-PLS [34]	2.44E+04 / 156.13	1.77E+05 / 421.13	3.43E+05 / 585.65	1.66E+05 / 407.67	1.60E+05 / 400.19	8.19E+03 / 90.08	1.47E+05 / 343.48 / 0.16
dynamic di-PLS	68.23 / 8.12	28.62 / 4.98	126.56 / 11.02	44.49 / 5.90	22.56 / 4.26	26.63 / 4.61	52.85 / 6.48 / 0.87
TDLVR (Ours)	4.51 / 1.71	4.88 / 1.80	5.85 / 1.97	1.84 / 1.13	17.88 / 3.39	7.59 / 2.23	7.09 / 2.04 / 0.98
CTDLVR (Ours)	3.32 / 1.44	3.45 / 1.48	3.05 / 1.42	0.77 / 0.71	12.45 / 2.93	5.62 / 1.86	4.78 / 1.64 / 0.98

proposed method are evaluated on the same target testing dataset. PLS and DiPLS only utilize the source domain labeled dataset to establish the quality prediction model, while all transfer learning methods, including CORAL, TCA, DANN, di-PLS, WDGRL, MDD, dynamic di-PLS, and TDLVR, utilize both the source domain labeled data and the target domain unlabeled data as the training dataset. Notably, it is a common practice for transfer learning to use unlabeled samples of the target domain as part of the training dataset to reduce distribution differences. CTDLVR, as an online version of TDLVR, utilizes a small number of labeled samples from the training dataset of the target domain to achieve error compensation. In this work, the number of labeled training samples available from the target domain for CTDLVR is selected as 15. In addition, the number of LVs or PCs l is set as 3 for all of the methods. For DiPLS, dynamic di-PLS, TDLVR, and CTDLVR, time-lag coefficient or dynamic order s is set as 2. λ_β of CTDLVR and TDLVR is chosen from $[-10e^3, \dots, -10e^{-3}, -10e^{-4}, 10e^{-4}, 10e^{-3}, \dots, 10e^3]$ by grid search.

One mode from {M1, M2, M3} is selected as the source domain with the remaining ones as the target domain, respectively. The mean squared error (MSE), mean absolute error (MAE), and mean Pearson correlation coefficient (PCC) between the predicted value and actual value of six mode pairs are used to evaluate the prediction performance. The prediction results on testing data are listed in Table II. Several

insights can be observed: 1) the results of two static methods, i.e., PLS and di-PLS, are significantly worse than those of the proposed TDLVR and CTDLVR. In particular, di-PLS has a negative effect due to its inability of capturing dynamic relations and is even worse than PLS. 2) Compared with PLS, DiPLS can improve performance, but it still performs worse than TDLVR. 3) Dynamic di-PLS can effectively improve the performance of di-PLS, but it is unstable. For example, dynamic di-PLS performs better than DiPLS only when M3 is chosen as the source domain, and M1 and M2 are the target domains. By contrast, TDLVR and CTDLVR show better performance in all scenarios. Finally, the prediction performances are further evaluated by mean MSE, mean MAE, and mean PCC, as shown in the last column of Table II. The results indicate that the proposed TDLVR and CTDLVR achieve lower MSE and MAE, and higher PCC compared with the traditional methods. Meanwhile, the predicted values and absolute prediction errors for M1→M2 and M1→M3 are shown in Fig. 3. It is seen that the proposed TDLVR and CTDLVR track the actual value of the quality variable well.

The dynamics of the three operation modes are further described by DiPLS. The corresponding visualization results of weighting coefficients β are shown in Fig. 4. It can be seen that the weighting coefficients learned by DiPLS are different from each other for each mode. This situation directly indicates the dynamics differences between the three modes and the need for cross-domain modeling. In addition,

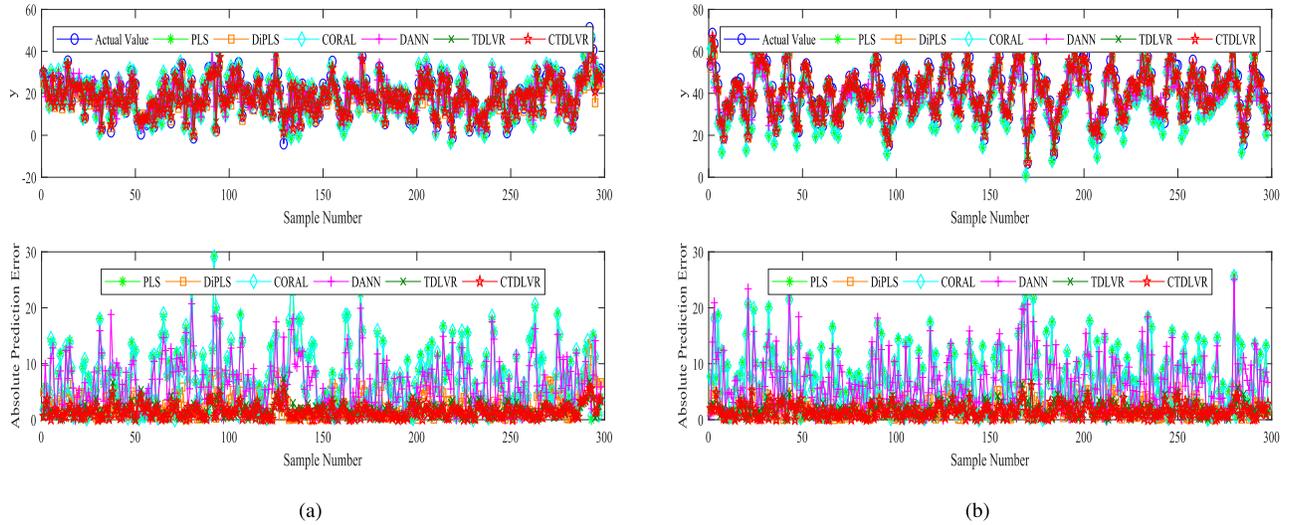


Fig. 3. Predicted values and the corresponding absolute prediction errors using TDLVR, CTDLVR, and other approaches in numerical simulation examples. (The left part of \rightarrow denotes the source domain, with the right part of \rightarrow denotes the target domain). (a) M1 \rightarrow M2. (b) M1 \rightarrow M3.

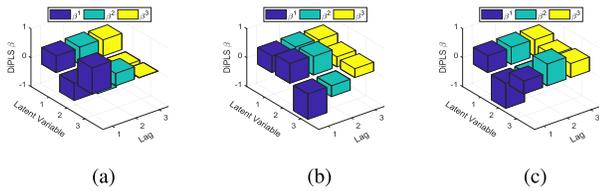


Fig. 4. Visualization of weighting coefficients β using DiPLS in three different modes. (a) M1. (b) M2. (c) M3.

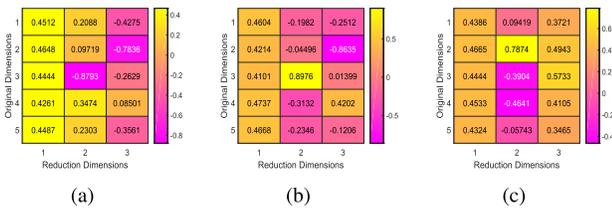


Fig. 5. Heatmap visualization of projection matrix W using DiPLS. (a) M1. (b) M2. (c) M3.

the heatmap visualization results of the projection matrix of the DiPLS are shown in Fig. 5. It can be seen that the projection matrix differences make it difficult to handle the cross-domain modeling for DiPLS. The proposed TDLVR inherits the interpretability of DiPLS and is able to describe the dynamics between modes. For TDLVR, the corresponding visualization results of two types of weighting coefficients β_1 and β_2 are shown in Fig. 6. It can be seen that the weighting coefficients can learn the appropriate dynamic information between domains. β_1 represents the dynamics between process variables and quality variables in the source domain, and β_2 represents co-dynamic variations within process variables between the source domain and the target domain. Meanwhile, the heatmaps of the projection matrix of TDLVR are shown in Fig. 7. It can be observed that different projection matrices are obtained to find a suitable projection in each scenario. For example, for M1 \rightarrow M2 and M2 \rightarrow M1, the effect of weighting coefficient β_2 is to select a suitable projection that may not

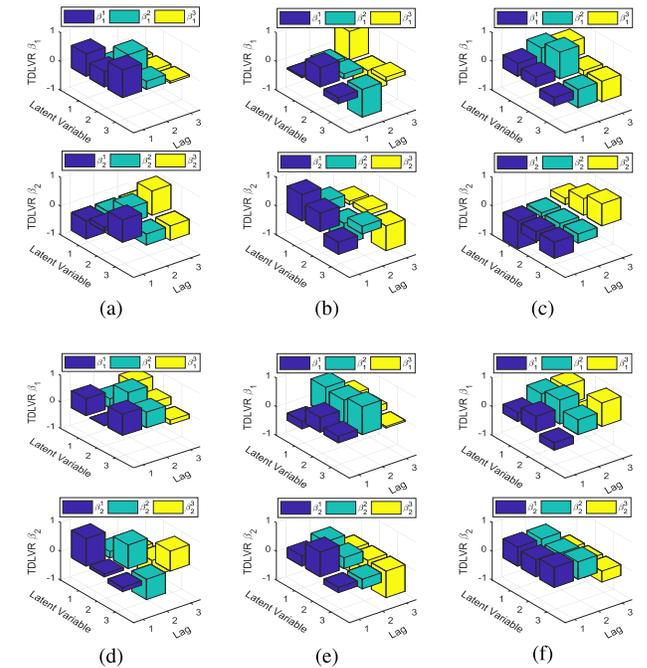


Fig. 6. Visualization of two types of weighting coefficients β_1 and β_2 using TDLVR. (a) M1 \rightarrow M2. (b) M2 \rightarrow M1. (c) M3 \rightarrow M1. (d) M1 \rightarrow M3. (e) M2 \rightarrow M3. (f) M3 \rightarrow M2.

be the same. However, for weighting coefficient β_1 , in the new projection space, the dynamics between process variables and quality variables are different from those of DiPLS. For example, for M1 \rightarrow M2 and M1 \rightarrow M3, by comparing with β of DiPLS from Fig. 4, β_1 of TDLVR can retain similar dynamics between two different domains to some extent.

B. Industrial Polyethylene Example

An industrial polyethylene process in [43], as a typical multimode process with dynamics, is used to demonstrate the proposed TDLVR and CTDLVR methods. The investigated

TABLE III

COMPARISON OF PREDICTION PERFORMANCE OF TDLVR, CTDLVR, AND TRADITIONAL APPROACHES FOR AN INDUSTRIAL POLYETHYLENE EXAMPLE

Methods	MSE/MAE						Mean MSE/MAE/PCC
	M1 → M2	M1 → M3	M2 → M1	M2 → M3	M3 → M1	M3 → M2	
PLS	1.81E+02 / 10.06	7.18E+03 / 64.53	17.00E+02 / 30.98	9.39E+03 / 71.74	0.92E+03 / 26.86	2.75E+02 / 10.44	2.23E+03 / 35.77 / 0.23
DiPLS [9]	1.28E+02 / 7.81	7.14E+03 / 67.12	5.80E+02 / 18.95	5.31E+03 / 49.61	1.23E+03 / 30.99	2.48E+02 / 13.67	1.65E+03 / 31.37 / 0.21
CORAL [38]	2.01E+02 / 10.76	5.69E+03 / 57.58	8.54E+02 / 25.23	3.84E+03 / 52.13	1.48E+03 / 32.86	3.35E+02 / 12.98	1.57E+03 / 31.92 / 0.21
TCA [37]	1.23E+02 / 8.45	7.68E+03 / 75.13	9.63E+02 / 23.48	5.49E+03 / 61.60	0.75E+03 / 24.14	1.34E+02 / 9.51	1.64E+03 / 33.75 / 0.30
WDGRL [29]	1.59E+02 / 9.13	5.00E+03 / 55.25	4.40E+02 / 17.18	3.33E+03 / 48.50	0.68E+03 / 19.43	6.64E+02 / 19.27	1.71E+03 / 28.13 / 0.19
MDD [30]	1.33E+02 / 8.34	4.34E+03 / 52.92	6.03E+02 / 18.20	3.40E+03 / 43.37	0.66E+03 / 19.31	3.72E+02 / 14.98	1.58E+03 / 26.19 / 0.16
DANN [40]	1.05E+02 / 7.39	6.38E+03 / 59.84	8.39E+02 / 23.52	4.50E+03 / 48.13	1.36E+03 / 29.38	8.32E+02 / 19.59	1.76E+03 / 31.31 / 0.22
di-PLS [34]	1.25E+02 / 8.47	6.93E+03 / 63.04	4.89E+02 / 18.13	5.09E+03 / 49.89	0.89E+03 / 25.48	2.17E+02 / 10.85	1.50E+03 / 29.31 / 0.22
dynamic di-PLS	1.41E+02 / 8.20	5.08E+03 / 54.72	6.67E+02 / 21.85	4.82E+03 / 49.00	1.42E+03 / 35.22	2.13E+02 / 12.07	1.48E+03 / 30.18 / 0.16
TDLVR (Ours)	0.75E+02 / 6.90	2.01E+03 / 34.46	3.24E+02 / 12.73	3.17E+03 / 40.99	0.51E+03 / 19.31	1.13E+02 / 8.31	0.72E+03 / 20.45 / 0.30
CTDLVR (Ours)	0.68E+02 / 6.06	1.77E+03 / 35.16	2.83E+02 / 12.34	2.88E+03 / 38.34	0.41E+03 / 16.88	0.71E+02 / 6.24	0.62E+03 / 19.17 / 0.46

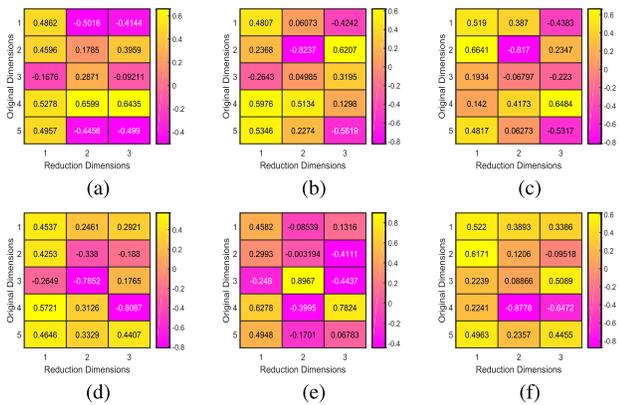


Fig. 7. Heatmap visualization results of projection matrix W using TDLVR to transfer shared information between modes. (a) M1→M2. (b) M2→M1. (c) M3→M1. (d) M1→M3. (e) M2→M3. (f) M3→M2.

process consists of three steady-state grades with different characteristics and transitions [52]. The melt index (MI), as the key product quality variable, can only be measured by offline laboratory analysis in around 6–8 h. Without online analyzers for MI, off-grade products and materials are produced inevitably in practice. Consequently, the proposed TDLVR and CTDLVR are applied to predict the MI online. In this study, three steady-state grades corresponding to medium (30–50), low (10–20), and high (240–350) levels of the MI are selected. A total of 331 samples are collected from the three grades, which are referred to as Mode 1 (i.e., M1), Mode 2 (i.e., M2), and Mode 3 (i.e., M3), respectively. Finally, 95 labeled samples in M1, 90 labeled samples in M2, and 145 labeled samples in M3 are utilized for verification.

Data preprocessing, including mean interpolation and three-sigma criterion, is first performed to deal with missing values and outliers. Subsequently, the collected data are divided into two parts: the training samples and the testing samples. In M1, the number of training samples is 45 and the remaining 50 is the test dataset; in M2, the number of the training samples is 45 and the remaining 45 is the test dataset; and in M3, the number of training samples is 80 and the remaining 65 is the test dataset. In addition, one of M1, M2, and M3 is randomly selected as the source domain with relatively sufficient labeled training data, while the remaining is the target domain. The

grade of the process switches frequently with each operating for a short period. Accordingly, the number of quality measurements available in each grade is scarce.

In the scenario where M1 is the source domain with M2 and M3 are the target domains, the MI predictions and absolute prediction errors of six approaches are shown in Fig. 8. The probability density functions (PDFs) of the first LV using DiPLS and the proposed TDLVR are shown in Fig. 9. It can be seen that the distribution of one mode is obviously different from the one of another mode using DiPLS. The distribution difference leads to prediction accuracy degradation of DiPLS. Meanwhile, the proposed TDLVR effectively aligns the distribution of LVs. The MSE and MAE of all scenarios and the mean PCC are listed in Table III. It can be seen that the proposed TDLVR and CTDLVR achieve lower MSE and MAE and higher mean PCC, indicating better prediction performances than PLS, DiPLS, CORAL, TCA, DANN, WDGRL, MDD, di-PLS, and dynamic di-PLS. Furthermore, owing to error compensation, CTDLVR is mostly superior to TDLVR.

The mean MSE and MAE metrics are further utilized for comprehensive evaluations. DiPLS is mostly better than PLS, although DiPLS shows poor performance in the scenario of M3 to M1 caused by weak dynamics. Compared with PLS and DiPLS, di-PLS achieves better prediction performance. Dynamic di-PLS with comparable performance to di-PLS is unstable in handling weak dynamics. It is unreasonable for dynamic di-PLS to use time lag matrices directly to deal with dynamic characteristics while ignoring the dynamic discrepancy between domains. From the results, TDLVR performs much better than other approaches in all scenarios. Unlike dynamic di-PLS, TDLVR makes use of weight coefficients to describe dynamics between process variables and quality variables, as well as the co-dynamic variations of process variables between domains.

To verify the time-lag impacts on model performance, DiPLS, dynamic di-PLS, and TDLVR are compared with different time lag coefficients. The results are shown in Fig. 10. The performance of each method fluctuates with the time lag coefficient, but TDLVR is much better than the other two methods. In addition, TDLVR is more stable and reliable than dynamic di-PLS. In contrast, dynamic di-PLS is inaccurate and worse than DiPLS in some situations. In this experiment,

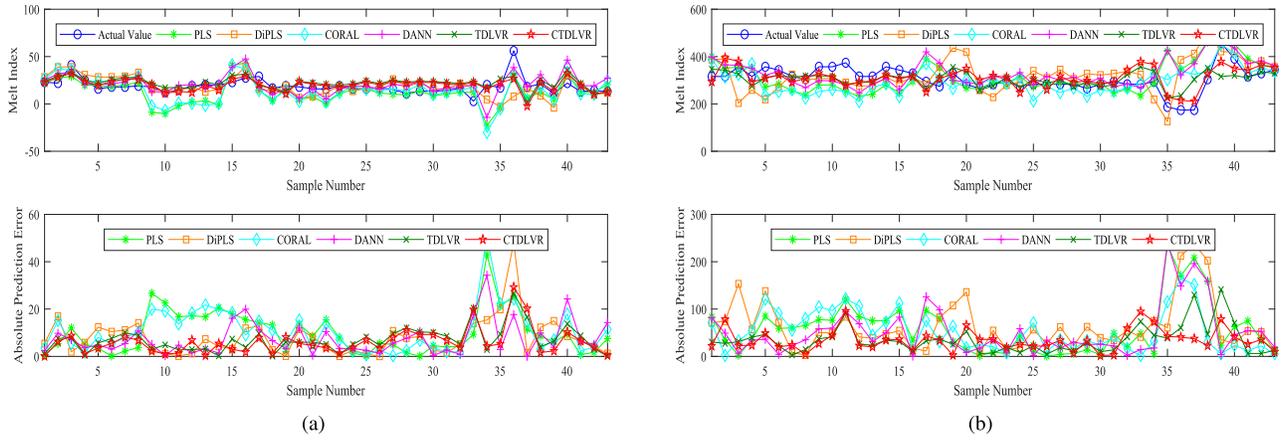


Fig. 8. Predicted results and their corresponding absolute prediction errors using TDLVR, CTDLVR, and other approaches. (a) $M1 \rightarrow M2$. (b) $M1 \rightarrow M3$.

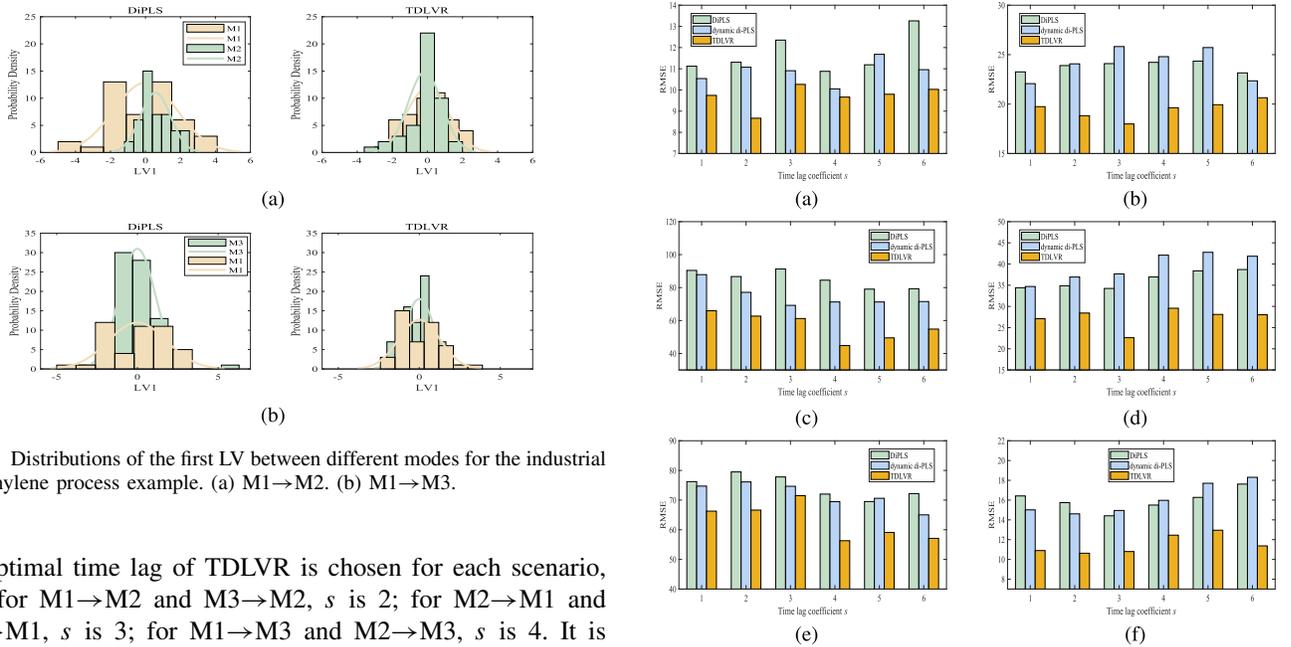


Fig. 9. Distributions of the first LV between different modes for the industrial polyethylene process example. (a) $M1 \rightarrow M2$. (b) $M1 \rightarrow M3$.

the optimal time lag of TDLVR is chosen for each scenario, e.g., for $M1 \rightarrow M2$ and $M3 \rightarrow M2$, s is 2; for $M2 \rightarrow M1$ and $M3 \rightarrow M1$, s is 3; for $M1 \rightarrow M3$ and $M2 \rightarrow M3$, s is 4. It is noted that a large time lag is not desired due to data volume constraints. All prediction results demonstrate that the proposed TDLVR achieves the best predictions. It is noted that CTDLVR inherits the merits of TDLVR.

C. Debutanizer Column Example

The debutanizer dataset [53] contains seven process variables, including top temperature, top pressure, reflux flow, flow to next process, VI tray temperature, bottom temperature 1, bottom temperature 2, and one quality variable, i.e., bottom butane concentration. This dataset has significant concept drift and data distribution differences. The coherent dynamics of this process can be adopted to evaluate the effectiveness of the proposed TDLVR. The first 1100 samples are selected as the source domain and the rest 1394 samples as the target domain. The number of LVs and dynamic order is selected to be the same as those used in [47]. Meanwhile, due to the slow sampling rate of quality variables, only 1/10 of samples in the target domain are adopted for performance evaluation. The comparison results of MSE, MAE, and PCC on the target

Fig. 10. Comparison of prediction performance for TDLVR, dynamic di-PLS, and DiPLS approaches for the polyethylene process example. (a) $M1 \rightarrow M2$. (b) $M2 \rightarrow M1$. (c) $M1 \rightarrow M3$. (d) $M3 \rightarrow M1$. (e) $M2 \rightarrow M3$. (f) $M3 \rightarrow M2$.

TABLE IV
COMPARISON OF PREDICTION PERFORMANCE OF TDLVR AND TRADITIONAL APPROACHES FOR A DEBUTANIZER COLUMN EXAMPLE

Methods	MSE	MAE	PCC
PLS	0.038	0.152	0.523
DiPLS [9]	0.020	0.111	0.756
CORAL [38]	0.022	0.113	0.572
TCA [37]	0.021	0.108	0.638
WDGRL [29]	0.023	0.110	0.510
MDD [30]	0.023	0.114	0.527
DANN [40]	0.023	0.112	0.689
di-PLS [34]	0.022	0.108	0.540
dynamic di-PLS	0.014	0.090	0.782
TDLVR(ours)	0.008	0.067	0.851

domain are listed in Table IV. It can be seen that some transfer methods, including CORAL, TCA, WDGRL, MDD, DANN, and di-PLS, are better than the traditional PLS and worse than

traditional DiPLS. Furthermore, dynamic di-PLS outperforms di-PLS, because the former considers data dynamics. The proposed TDLVR achieves a more accurate prediction than all the above-mentioned methods.

VI. CONCLUSION

In this work, a novel transfer DLV modeling scheme, including TDLVR and CTDLVR, has been developed for quality prediction of multimode processes with limited labeled samples available in the new mode. Compared with the traditional TL and dynamic modeling approaches, the proposed TDLVR achieves significantly improved predictions by extracting the dynamics between process variables and quality variables in the POM, the co-dynamic variations between the POM and a new mode, and overcoming the marginal distribution discrepancy between the two modes. Making full use of a small number of labeled samples from the new mode, the proposed CTDLVR with an error compensation model has effectively reduced the prediction errors of TDLVR due to conditional distribution discrepancy.

REFERENCES

- Q. Q. Sun and Z. Q. Ge, "A survey on deep learning for data-driven soft sensors," *IEEE Trans. Ind. Informat.*, vol. 17, no. 9, pp. 5853–5866, Sep. 2021.
- G. Napoli and M. G. Xibilia, "Soft sensor design for a topping process in the case of small datasets," *Comput. Chem. Eng.*, vol. 35, no. 11, pp. 2447–2456, Nov. 2011.
- M. Seera, C. P. Lim, C. K. Loo, and H. Singh, "Power quality analysis using a hybrid model of the fuzzy min–max neural network and clustering tree," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2760–2767, Dec. 2016.
- L. Ren, Z. Meng, X. Wang, R. Lu, and L. T. Yang, "A wide-deep-sequence model-based quality prediction method in industrial process analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3721–3731, Sep. 2020.
- Q. Shi, Y.-M. Cheung, Q. Zhao, and H. Lu, "Feature extraction for incomplete data via low-rank tensor decomposition with feature regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1803–1817, Jun. 2019.
- W. Jing, Z. Zhong-Qiu, H. Xuegang, C. Yiu-ming, W. Meng, and W. Xindong, "Online group feature selection," 2014, *arXiv:1404.4774*.
- E. Zamprogna, M. Barolo, and D. E. Seborg, "Optimal selection of soft sensor inputs for batch distillation columns using principal component analysis," *J. Process Control*, vol. 15, no. 1, pp. 39–52, Feb. 2005.
- X. Kong and Z. Ge, "Deep PLS: A lightweight deep learning model for interpretable and efficient data analytics," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 11, 2022, doi: [10.1109/TNNLS.2022.3154090](https://doi.org/10.1109/TNNLS.2022.3154090).
- Y. Dong and J. Qin, "Regression on dynamic PLS structures for supervised learning of dynamic data," *J. Process Control*, vol. 68, pp. 64–72, Aug. 2018.
- C. O. Sakar and O. Kursun, "Discriminative feature extraction by a neural implementation of canonical correlation analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 1, pp. 164–176, Jan. 2017.
- C. Shang, B. Huang, F. Yang, and D. Huang, "Probabilistic slow feature analysis-based representation learning from massive process data for soft sensor modeling," *AIChE J.*, vol. 61, no. 12, pp. 4126–4139, 2015.
- P. Song and C. Zhao, "Slow down to go better: A survey on slow feature analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Sep. 7, 2022, doi: [10.1109/TNNLS.2022.3201621](https://doi.org/10.1109/TNNLS.2022.3201621).
- Y.-M. Cheung and H. Zeng, "Local kernel regression score for selecting features of high-dimensional data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 12, pp. 1798–1802, Dec. 2009.
- 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.
- P. Zhou, D. Guo, H. Wang, and T. Chai, "Data-driven robust M-LS-SVR-based NARX modeling for estimation and control of molten iron quality indices in blast furnace ironmaking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4007–4021, Sep. 2018.
- X. Jiang and Z. Ge, "Augmented multidimensional convolutional neural network for industrial soft sensing," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–10, 2021.
- 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. 99, pp. 1–12, 2020.
- Y. Liu, C. Yang, Z. Gao, and Y. Yao, "Ensemble deep kernel learning with application to quality prediction in industrial polymerization processes," *Chemometrics Intell. Lab. Syst.*, vol. 174, pp. 15–21, Mar. 2018.
- J. C. B. Gonzaga, L. A. C. Meleiro, C. Kiang, and R. M. Filho, "ANN-based soft-sensor for real-time process monitoring and control of an industrial polymerization process," *Comput. Chem. Eng.*, vol. 33, no. 1, pp. 43–49, 2009.
- J. Lu, J. Ding, X. Dai, and T. Chai, "Ensemble stochastic configuration networks for estimating prediction intervals: A simultaneous robust training algorithm and its application," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 12, pp. 5426–5440, Dec. 2020.
- L. Yao and Z. Ge, "Nonlinear Gaussian mixture regression for multi-mode quality prediction with partially labeled data," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 4044–4053, Dec. 2018.
- B. Alakent, "Soft-sensor design via task transferred just-in-time-learning coupled transductive moving window learner," *J. Process Control*, vol. 101, pp. 52–67, May 2021.
- X. Jiang and Z. Ge, "Improving the performance of just-in-time learning-based soft sensor through data augmentation," *IEEE Trans. Ind. Electron.*, vol. 69, no. 12, pp. 13716–13726, Dec. 2022.
- S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Dec. 2009.
- L. Zhang and X. Gao, "Transfer adaptation learning: A decade survey," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 21, 2022, doi: [10.1109/TNNLS.2022.3183326](https://doi.org/10.1109/TNNLS.2022.3183326).
- 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, Jun. 2021, doi: [10.1109/TNNLS.2021.3085869](https://doi.org/10.1109/TNNLS.2021.3085869).
- G. Ma et al., "A transfer learning-based method for personalized state of health estimation of lithium-ion batteries," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 3, 2022, doi: [10.1109/TNNLS.2022.3176925](https://doi.org/10.1109/TNNLS.2022.3176925).
- 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.*, early access, Dec. 30, 2022, doi: [10.1109/TNNLS.2022.3231849](https://doi.org/10.1109/TNNLS.2022.3231849).
- J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018, pp. 1–8.
- Y. Zhang, T. Liu, M. Long, and M. Jordan, "Bridging theory and algorithm for domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7404–7413.
- Z.-G. Liu, L.-B. Ning, and Z.-W. Zhang, "A new progressive multi-source domain adaptation network with weighted decision fusion," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 8, 2022, doi: [10.1109/TNNLS.2022.3179805](https://doi.org/10.1109/TNNLS.2022.3179805).
- P. Wei, R. Sagarna, Y. Ke, and Y.-S. Ong, "Practical multisource transfer regression with source–target similarity captures," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3498–3509, Aug. 2021.
- X. Wang, T. K. Huang, and J. Schneider, "Active transfer learning under model shift," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1305–1313.
- R. Nikzad-Langerodi, W. Zellinger, E. Lughofer, and S. Saminger-Platz, "Domain-invariant PARTIAL-LEAST-SQUARES regression," *Anal. Chem.*, vol. 90, no. 11, pp. 6693–6701, Jun. 2018.
- R. Jia, S. Zhang, and F. You, "Transfer learning for end-product quality prediction of batch processes using domain-adaption joint-Y PLS," *Comput. Chem. Eng.*, vol. 140, Sep. 2020, Art. no. 106943.
- K. Wang, J. Chen, L. Xie, and H. Su, "Transfer learning based on incorporating source knowledge using Gaussian process models for quick modeling of dynamic target processes," *Chemometric Intell. Lab. Syst.*, vol. 198, Mar. 2020, Art. no. 103911.
- 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.

- [38] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 30, no. 1, 2016, pp. 1–8.
- [39] R. Jia, S. Zhang, and F. You, "Nonlinear soft sensor development for industrial thickeners using domain transfer functional-link neural network," *Control Eng. Pract.*, vol. 113, Aug. 2021, Art. no. 104853.
- [40] Y. Ganin et al., "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, 2017.
- [41] 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.
- [42] 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, Sep. 2019, Art. no. 103813.
- [43] 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, Sep. 2020.
- [44] 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," *Knowledge-Based Syst.*, vol. 248, Jul. 2022, Art. no. 108851.
- [45] J. Xie, B. Huang, and S. Djuljevic, "Transfer learning for dynamic feature extraction using variational Bayesian inference," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 11, pp. 5524–5535, Nov. 2022, doi: 10.1109/TKDE.2021.3054671.
- [46] 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, Jan. 2020.
- [47] 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.
- [48] H. Wold, *Estimation of Principal Components and Related Models by Iterative Least Squares, Multivariate Analysis*, P. K. Krishnaiah, Ed. New York, NY, USA: Academic, 1966, pp. 391–420.
- [49] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2960–2967.
- [50] S. Zhao et al., "A review of single-source deep unsupervised visual domain adaptation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 473–493, Feb. 2022.
- [51] X. Liu, Y. Li, and G. Chen, "Transfer learning for regression via latent variable represented conditional distribution alignment," *Knowledge-Based Syst.*, vol. 240, Mar. 2022, Art. no. 108110.
- [52] J. Liu, T. Liu, and J. Chen, "Quality prediction for multi-grade processes by just-in-time latent variable modeling with integration of common and special features," *Chem. Eng. Sci.*, vol. 191, pp. 31–41, Dec. 2018.
- [53] L. Fortuna, S. Graziani, and M. G. Xibilia, "Soft sensors for product quality monitoring in debutanizer distillation columns," *Control Eng. Pract.*, vol. 13, no. 4, pp. 499–508, 2005.



Chao Yang (Graduate Student Member, IEEE) received the M.Eng. degree from the Zhejiang University of Technology, Hangzhou, China, in 2019. He is currently pursuing the Ph.D. degree in control science and engineering with the State Key Laboratory of Synthetical Automation for Process Industry, Northeastern University, Shenyang, China. 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 was a Research Associate with the Department of Chemical Engineering, University of Southern California, Los Angeles, CA, USA from 2014 to 2016. He is currently a Full Professor with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang. 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.

Prof. Liu was a recipient of the Outstanding Young Scholar of the Liaoning Revitalization Talents Program, China. His article Perspectives on Big Data Modeling of Process Industries was selected as one of the F5000-Top academic papers in Chinese top-quality Science Citation Index (SCI) tech journals in 2019.



Yi Liu (Member, IEEE) received the Ph.D. degree in control theory and engineering from Zhejiang University, Hangzhou, China, in 2009.

He was an Associate Professor with the Institute of Process Equipment and Control Engineering, Zhejiang University of Technology, Hangzhou, from 2011 to 2020, where he has been a Full Professor since 2020. He has authored or coauthored over 50 research articles at the IEEE TRANSACTIONS and international journals. His research interests include data intelligence with applications for modeling, control, and optimization of industrial processes.



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

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

Dr. Cheung is a fellow of American Association for the Advancement of Science (AAAS), The Institution of Engineering and Technology (IET), and British Computer Society (BCS). He also serves as an Associate Editor for several prestigious journals, including the IEEE TRANSACTIONS ON CYBERNETICS, the IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS from 2014 to 2020, and *Pattern Recognition*. He has been the Editor-in-Chief of the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE since 2023. More details can be found at: <https://www.comp.hkbu.edu.hk/~ymc>.