Variational Progressive-Transfer Network for Soft Sensing of Multirate Industrial Processes

Zheng Chai, Chunhui Zhao, *Senior Member, IEEE*, and Biao Huang, *Fellow, IEEE*

***Abstract*—Deep-learning-based soft sensors have been extensively developed for predicting key quality or performance variables in industrial processes. However, most approaches assume that data are uniformly sampled while the multiple variables are often acquired at different rates in practical processes. This article designed a progressive transfer strategy, based on which a variational progressive-transfer network (VPTN) method is proposed for the soft sensor development of industrial multirate processes. In VPTN, the multirate data are first separated into multiple data chunks where the variables within each chunk are acquired at a uniform rate. Then, a variational multichunk data modeling framework is developed to model the multiple chunks in a unified fashion through deep variational structures. The base models, including the unsupervised ones with only partial process variables and the supervised soft sensor model share a similar network structure, such that the subsequent transfer strategy can be readily implemented. Finally, a progressive transfer learning strategy is designed to transfer the model parameters from the fastest sampled data chunk to the slowest one in a progressive manner. Thus, the knowledge from various data chunks can be sequentially explored and transferred to enhance the performance of the terminal soft sensor model. Case studies on both a debutanizer column dataset and a real coal mill dataset in a thermal power plant validate the performance of the proposed method.**

***Index Terms*—Deep learning, multirate industrial processes, progressive transfer learning, soft sensor.**

I. Introduction

TIMELY and accurate measurements of key process variables are of great importance to effective control and monitoring for modern industrial processes [1]. In practice, however, the hardware costs, inadequacy of measurement techniques, or possible analysis delays have become constraints in achieving reliable measurements in industrial processes. Such restrictions to some extent hamper the improvement of product quality and operational safety.

To estimate the key variables in industrial processes, data-driven soft sensors provide a feasible way by building predictive models between the easy-to-measure process variables and the hard-to-measure quality or performance variable. In the past decades, machine learning algorithms, such as partial least squares (PLS) [2], support vector regression (SVR) [3], and slow feature analysis [4], [5], have been widely studied and extensively applied to data-driven process modeling. Due to the efficacy in nonlinear information processing, deep-learning-based methods have gained popularity in recent years in the process industry [6]–[8]. Shang *et al.* [9] proposed a deep belief network-based soft sensor for the crude-oil distillation unit. In [10], multiple supervised autoencoders (AEs) are stacked to extract the quality-relevant features and the method is validated through an industrial process of debutanizer column. As the AE structures learn deterministic feature representations and pay less attention to characterize the uncertainty distribution, the probabilistic counterpart of the deterministic AE, that is, the variational AE (VAE), has been studied for modeling the soft sensors in industrial processes [11]. Xie *et al.* [12] designed a supervised VAE by integrating the supervision information into the plain VAE for the soft sensor development of the polymerization process. In [13], a variable-reweighed VAE is designed to extract output-relevant feature representations, and the modified VAE is combined with the just-in-time learning framework to build more accurate local models. In these works, the characterization of uncertainty data distribution and capability of nonlinear information processing of VAE have improved the robustness of the soft sensors, yielding better prediction results for industrial processes.

Despite the popularity of the deep learning approaches, most of them assume that the multiple variables are sampled at a uniform rate. Practically, however, the variables in industrial processes are generally sampled at different rates [14], [15]. For example, several process variables (secondary variables) measured by the hardware sensors can be recorded per seconds while some are recorded per minutes, and the quality or performance variable (primary variable) is often obtained through hours of offline laboratory analysis. Under this case, the complete training samples with corresponding labels that can be used for soft sensor modeling are quite limited. This phenomenon poses a challenge for conventional deep-learning-based soft sensors. Specifically, industrial processes containing

Manuscript received December 3, 2020; revised April 10, 2021; accepted June 8, 2021. This work was supported in part by the NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization under Grant U1709211; in part by the Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China, under Grant ICT2021A15; and in part by the State Key Laboratory of Synthetical Automation for Process Industries under Grant 2020-KF-21-07. This paper was recommended by Associate Editor Q.-L. Han. *(Corresponding author: Chunhui Zhao.)*

Zheng Chai and Chunhui Zhao are with the State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: chaizheng@zju.edu.cn; chhzhao@zju.edu.cn).

Biao Huang is with the Department of Chemical and Materials Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada (e-mail: biao.huang@ualberta.ca).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2021.3090996.

Digital Object Identifier 10.1109/TCYB.2021.3090996

variables sampled at two or more kinds of rates are called as dual rate processes (two rates) or multirate processes (three or more rates). For the easier dual rate systems, it is considered that the process variables are sampled with an identical rate, which is different from that of the quality variable. As the collected data in dual-rate systems can be divided into samples with quality variable and without quality variable, a feasible solution is to use semisupervised learning to learn from both the labeled and unlabeled data [16], [17]. Gopakumar *et al.* [18] developed a semisupervised deep neural network for nonlinear bioprocesses. In [19], a just-in-time semisupervised extreme learning machine is developed to predict the Mooney viscosity in an industrial rubber mixer. Despite the efficacy, the semisupervised learning-based methods can be difficult to generalize to the multirate industrial processes. For the multirate systems, generally the secondary variables are sampled with two or more rates, which are different from that of the quality variable, yielding three or more sampling rates in the process. To deal with this problem, existing methods have made efforts by transforming the multirate-sampled data into a uniformly sampled version. For example, the data lifting technique rearranges the original dataset by stacking the variables with the fast sampling rates, such that the lifted variables can be downsampled and have the same frequency as the slowly sampled quality variable [14]. Besides, the probabilistic principal component analysis was proposed which can be used to deal with the missing variables and thus upsample the multirate data [20]. Also, some interpolation methods are developed to insert unavailable values for the missed variables to make the dataset complete [21]. However, for the data lifting methods, the dimensionality of the lifted dataset is generally much higher than the original dataset, especially when a large difference exists between the sampling rates. The upsampling methods would rely heavily on the estimated values, which may accumulate the error of the soft sensor built on the transformed data. In comparison with these methods, the multirate data are used without the need of conventional upsampling or downsampling in this article, which mines the data more directly and avoids the increasing dimension or the error accumulation problems.

Essentially, the multirate data consist of multiple data chunks in which each chunk has a uniform sampling rate, and the quality or performance variable is generally measured with the slowest rate. Thus, to handle the problem where the data collected under multiple rates are available, deep transfer learning (DTL) [22], [23], as a popular learning paradigm, provides a potential to fully mine the knowledge from multiple data. In recent years, DTL approaches have been widely developed and applied in manufacturing processes [24]–[26]. For example, Shao *et al.* [27] proposed a network transfer-based approach for fault diagnosis by transferring the source model parameter to the target model. To take the advantage of DTL for multirate data modeling, Chen *et al.* [28] built the pretrained model for the fault diagnosis task of rotating machinery vibration systems. However, only one data chunk is used to initialize the model, leaving the information in the other chunks not fully explored in the terminal model. Besides, most of the methods [24]–[28] are developed in a deterministic fashion, which may lead to weak robustness of the model in real applications [12]. Also, these approaches are designed for industrial fault diagnosis tasks, while researches on soft sensor problems are rarely considered.

To address the above problems, in this article, a variational progressive-transfer network (VPTN)-based soft sensor is developed for multirate industrial processes. For the multirate-structured data, the collected samples can be first divided into multiple data chunks and the variables within each chunk have a uniform rate, which has become a common practice in multirate data preprocessing [28], [29]. Note that a conventional neural network trained with only the slowest data chunk is insufficient due to the limited samples. On the contrary, the progressively increasing process variables across different chunks add more knowledge for better modeling the terminal soft sensor, and this motivates us to develop the VPTN approach which transfers the knowledge from the fastest to the slowest chunk in a sequential manner. The knowledge of the previous models can thus be fully explored and adapted to enhance the soft sensing performance of the terminal slowest model. Besides, the VPTN models the fast chunks with partial process variables and the slowest chunk with both the process and the quality variables in a unified network structure under the stochastic gradient variational Bayes (SGVB) framework [11]. This provides two benefits to VPTN. On one hand, the uncertainty distribution information can be well characterized in each model. On the other hand, different base models share a very similar structure such that the progressive transfer strategy can be readily implemented. The primary contribution of this article lies in that, the network-based transfer learning is introduced to soft sensor modeling of multirate industrial processes, and a new progressive transfer framework is designed. Different from the independent modeling for the slowest data chunk which is insufficient due to the limited training data, this article developed a VPTN method that propagates the knowledge from the fastest to the slowest chunk, which better captures the deep feature representations and more thoroughly mines the multirate data.

The remainder of this article starts with a brief introduction to the VAE and DTL in Section II. Section III gives the details of the proposed VPTN. In Section IV, illustrations on both a debutanizer column dataset and a coal mill dataset are used to verify the efficacy of the proposed method. Finally, Section V concludes this article.

II. Preliminaries

A. Variational Autoencoder

The VAE is a deep generative model capable of learning complex data distribution via probabilistic latent feature representations [11]. In VAE, a recognition model $q_\phi(z|x)$, which is also called as a probabilistic encoder, is designed to approximate the intractable posterior $p_\theta(z|x)$ and output the distribution of the latent variable z based on the input x. Then, a generation model $p_\theta(x|z)$, which is also called as a probabilistic decoder, translates the sampled z into the reconstructed x. Based on the description regarding the generative procedure,

the goal of the VAE model is to maximize the variational evidence lower bound (ELBO) on the data loglikelihood

$$\log p_\theta(x) \geq -D_{KL}\big[q_\phi(z|x)||p_\theta(z)\big] + \mathbb{E}_{q_\phi(z|x)}\big[\log p_\theta(x|z)\big] \tag{1}$$

where $D_{KL}(\cdot||\cdot)$ is the Kullback–Leibler (KL) divergence between two densities and $p_\theta(z)$ signifies the prior over the latent variables z.

Note that the indeterministic sampling operation from the recognition distribution, that is, $z \sim q_\phi(z|x)$, is nondifferentiable in the neural network training phase. Thus, the reparameterization trick in VAE restricts the posterior $q_\phi(z|x)$ as some kinds of parametric distributions, for example, Gaussian. Then the deterministic z can be reparameterized as $z = \mu + \sigma \odot \varepsilon$ where $\varepsilon \sim \mathcal{N}(0, I)$ and $\odot$ denotes the element-wise multiplication. This allows the VAE to be trained efficiently using gradient descent through the Gaussian latent variables.

B. Deep Transfer Learning

The DTL is a popular learning paradigm that integrates the deep neural networks into the transfer learning framework to utilize the strong nonlinear information processing capability of deep structures [23]. Specifically, the DTL aims to discover and transfer the knowledge existed in the source domain data $\mathcal{D}_\mathcal{S}$ with the source task $t_\mathcal{S}$, such that the performance of the predictive function $f_\mathcal{T}(\cdot)$ formulated by a deep neural network on the target task $t_\mathcal{T}$ in $\mathcal{D}_\mathcal{T}$ can be improved.

Generally, the DTL approaches can be classified into four categories: 1) mapping-based methods; 2) adversarial-based methods; 3) instances-based methods; and 4) network-based methods [23]. The mapping- and adversarial-based approaches share a similar motivation which aims to find a transferrable feature space in which the distribution discrepancy between the latent representations of different datasets is minimized. The difference lies in the way that the discrepancy minimization is realized. The mapping-based methods use quantitative metrics, for example, the maximum mean discrepancy function to measure the difference between two datasets [24]. For the adversarial-based method, the domain adversarial training motivated by generative adversarial networks is utilized to make the feature representation from different datasets unrecognized. The instances-based methods select and reuse part of the training instances in the source domain by assigning proper weights [22]. Differently, the network-based methods reuse the partial network that pretrained on the source domain data and transfer it to the target task [23]. For example, Shao *et al.* [27] used the structure and parameters of pretrained VGG-16 network and transfer them into an induction motor dataset and a bearing dataset to diagnose different categories of occurred faults. In [28], the parameters of the source network with the highest accuracy are transferred to the model in the target task, and the performance is validated through a machinery vibration fault diagnosis dataset. Most of the existing methods are developed for mechanical fault diagnosis, while research on DTL-based industrial soft sensors is rarely considered. Besides, most of these methods are designed based on the single transfer, which means that the knowledge transfer is implemented just once, while for the multirate process, a potential solution is to derive proper base models and transfer them in a progressive manner to sequentially adapt the knowledge across multiple data chunks.

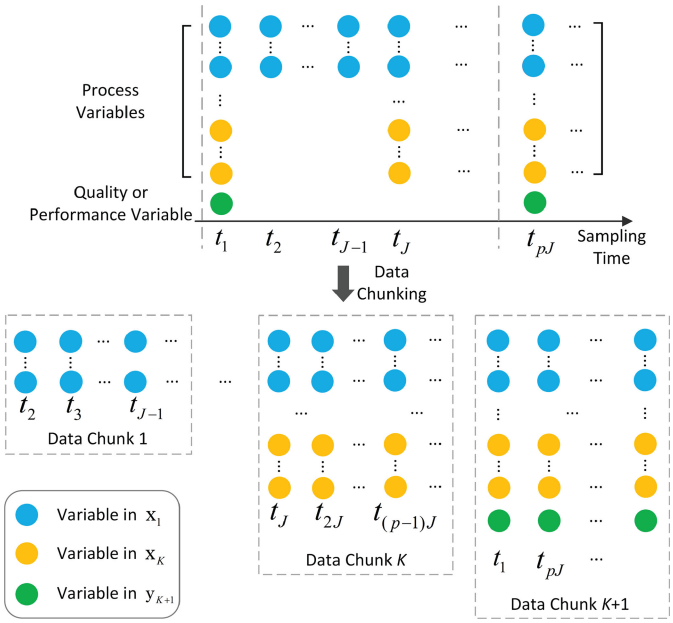

Fig. 1. Illustrations of the multirate sampled data in industrial processes and the data chunking procedure.

III. Methodology

In this section, the proposed VPTN is presented. First, the motivation and the overall progressive transfer framework are introduced. Then the variational multichunk data modeling and the progressive transfer strategy are described. Finally, the VPTN-based soft sensor development is presented.

A. Motivation and the Progressive Transfer Framework

To clearly describe the multirate phenomena we are concerned about, some necessary notations are given first. Denote the training dataset as $\{X^{m\times n}, y^m\}$ consisting of m samples measured on n process variables and a quality-relevant variable. Assume that there are $K+1$ kinds of sampling rates in the process, in which the process variables are sampled with K kinds of rates which are different from that of the quality or performance variable. Among the n variables, the n_1 variables with the highest frequency $1/T$ are signified by $x_1(x_1 = [x_1^1, x_1^2, \ldots, x_1^{n_1}])$. In a similar fashion, the variable group with the K-th highest frequency $1/JT$ is denoted by $x_K(x_K = [x_K^1, x_K^2, \ldots, x_K^{n_K}])$, where $\sum_{k=1}^{K} n_k = n$. Generally, the frequency of the quality or performance variable $y_{K+1}(y_{K+1} = [y])$ is the lowest. The illustration of the multirate-sampled dataset is shown in the top subfigure in Fig. 1.

The soft sensor modeling for multirate industrial processes aims to accurately predict the values of the slowest quality or performance variable using the multirate-sampled faster process variables. Although deep learning methods have been

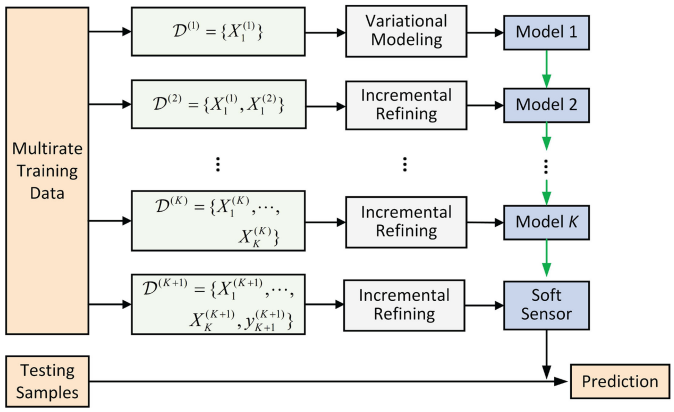

Fig. 2. Overall framework of the proposed VPTN. $\mathcal{D}^{(1)}$ to $\mathcal{D}^{(K+1)}$ in the green blocks indicate the multiple data chunks which are sorted from the fastest to the slowest according to the sampling rates in the multirate process. The green arrows indicate the progressive transfer strategy.

widely researched in soft sensing and achieve remarkable performance, most of them require that all the variables are uniformly sampled in the dataset. To tackle the problem that the variables are generally acquired at multiple different rates, this article developed a VPTN approach based on DTL. Two major questions need to be answered first: 1) for the multirate process data, how to specify proper source and target datasets/domains, and the corresponding learning tasks in different domains? and 2) considering the varying dimensionality of variables in different chunks, how to build appropriate models to learn the multiple datasets in a unified framework such that the network transfer can be readily achieved?

As shown in Fig. 1, it is feasible to separate the data into different data chunks according to the variables that the samples are available. For example, in Fig. 1, the data chunk $\mathcal{D}^{(1)}=\{X_1^{(1)}\}$ consists of samples that include the variable group x_1 only, where the superscript "(1)" denotes the first data chunk and the subscript "1" in $X_1^{(1)}$ denotes the first variable group x_1. Similarly, for $\mathcal{D}^{(K)}=\{X_1^{(K)},X_2^{(K)},\ldots,X_K^{(K)}\}$, it includes training samples measured on variables x_1 through x_K. Finally, $\mathcal{D}^{(K+1)}=\{X_1^{(K+1)},X_2^{(K+1)},\ldots,X_K^{(K+1)},y_{K+1}^{(K+1)}\}$ consists of samples with all the process variables and the quality or performance variable. After the data chunking, multiple chunks can be established and the sample length within each chunk is consistent. Thus, a single-transfer task can be designed by discovering the knowledge in data chunk $\mathcal{D}^{(i)}$, and transferring it to the model in the adjacent data chunk $\mathcal{D}^{(i+1)}$. Here, the data chunk $\mathcal{D}^{(i)}$ denotes the source domain and its adjacent data chunk $\mathcal{D}^{(i+1)}$ denotes the target domain. Thus, the soft sensor modeling of the multirate data can be fulfilled by sequentially transferring the knowledge from data chunk $\mathcal{D}^{(1)}$ to $\mathcal{D}^{(2)}$, and finally to $\mathcal{D}^{(K+1)}$. The overall framework of the designed VPTN is shown in Fig. 2. As shown in the figure, the training data are separated into $K+1$ data chunks first. For data chunk $\mathcal{D}^{(1)}$ in which the minimum number of process variables are measured, a variational deep neural network is first trained as Model 1. Then, the model is transferred to initialize the parameter of Model 2, and data chunk $\mathcal{D}^{(2)}$ is used to incrementally refine the model. Using a similar scheme, finally, the Model K is transferred to the Model $K+1$, that is, the soft sensor, and the soft sensor model is refined using the $K+1$-th data chunk. The terminal soft sensor can thus be obtained for online testing.

B. Variational Multichunk Data Modeling

The primary modeling steps of the VPTN contain two parts: 1) variational multichunk data modeling which builds proper base models for each data chunk and 2) complete multirate data modeling with the progressive knowledge transfer which sequentially propagates the knowledge across the models. In this section, the variational multichunk data modeling would be introduced first.

The data modeling of $\mathcal{D}^{(1)}$ through $\mathcal{D}^{(K+1)}$ can be divided into two categories: 1) the unsupervised modeling of $\mathcal{D}^{(k)}|_{k=1}^{K}$ and 2) the supervised modeling of $\mathcal{D}^{(K+1)}$. Due to the advantages in modeling uncertainty distributions and extracting nonlinear probabilistic representations through flexible and unified neural network structures, the VAEs can be explored and adopted as the base structures to model the different data chunks, which have been widely researched and applied to industrial soft sensors [11]–[13], [16], [17]. Specifically, the data likelihoods $p([x_1^{(k)};x_2^{(k)};\cdots;x_k^{(k)}])|_{k=1}^{K}$ and $p([x_1^{(K+1)};x_2^{(K+1)};\cdots;x_K^{(K+1)};y_{K+1}^{(K+1)}])$ are first calculated, where $[x_1^{(k)};x_2^{(k)};\cdots;x_k^{(k)}]|_{k=1}^{K}$ and $[x_1^{(K+1)};x_2^{(K+1)};\cdots;x_K^{(K+1)};y_{K+1}^{(K+1)}]$ are training samples from $\mathcal{D}^{(k)}|_{k=1}^{K}$, and $\mathcal{D}^{(K+1)}$, respectively.

1) Modeling Data Chunks With Partial Process Variables: To model the data chunks $\mathcal{D}^{(k)}|_{k=1}^{K}$ in which only partial process variables are measured, following the plain VAE [11], the marginal log-likelihood of a training sample $[x_1^{(k)};\cdots;x_k^{(k)}]$ in $\mathcal{D}^{(k)}$ can be given as follows:

$$
\begin{aligned}
&\log p_{\theta_k}\left(\left[x_1^{(k)};\cdots;x_k^{(k)}\right]\right)\\
&=\mathbb{E}_{q_{\phi_k}\left(z^{(k)}|\left[x_1^{(k)};\cdots;x_k^{(k)}\right]\right)}\\
&\quad\left[\log\frac{p_{\theta_k}\left(\left[x_1^{(k)};\cdots;x_k^{(k)}\right]|z^{(k)}\right)p_{\theta_k}\left(z^{(k)}\right)}{p_{\theta_k}\left(z^{(k)}|\left[x_1^{(k)};\cdots;x_k^{(k)}\right]\right)}\right]\\
&=\mathbb{E}_{q_{\phi_k}\left(z^{(k)}|\left[x_1^{(k)};\cdots;x_k^{(k)}\right]\right)}\\
&\quad\Bigg[\log p_{\theta_k}\left(\left[x_1^{(k)};\cdots;x_k^{(k)}\right]|z^{(k)}\right)\\
&\qquad-\log\frac{q_{\phi_k}\left(z^{(k)}|\left[x_1^{(k)};\cdots;x_k^{(k)}\right]\right)}{p_{\theta_k}\left(z^{(k)}\right)}\\
&\qquad+\log\frac{q_{\phi_k}\left(z^{(k)}|\left[x_1^{(k)};\cdots;x_k^{(k)}\right]\right)}{p_{\theta_k}\left(z^{(k)}|\left[x_1^{(k)};\cdots;x_k^{(k)}\right]\right)}\Bigg]\\
&\geq\mathbb{E}_{q_{\phi_k}\left(z^{(k)}|\left[x_1^{(k)};\cdots;x_k^{(k)}\right]\right)}\left[\log p_{\theta_k}\left(\left[x_1^{(k)};\cdots;x_k^{(k)}\right]|z^{(k)}\right)\right]\\
&\quad-D_{KL}\left(q_{\phi_k}\left(z^{(k)}|\left[x_1^{(k)};\cdots;x_k^{(k)}\right]\right)||p_{\theta_k}\left(z^{(k)}\right)\right)\\
&=\mathrm{ELBO}_k\left(\theta_k,\phi_k;\left[x_1^{(k)};\cdots;x_k^{(k)}\right]\right).
\end{aligned}
\tag{2}
$$

The ELBO of the log-likelihood of the k-th data chunk contains two parts. The $\mathbb{E}_{q_{\phi_k}(z^{(k)}|[x_1^{(k)};\cdots;x_k^{(k)}])}$ [$\log p_{\theta_k}$ ($[x_1^{(k)};\cdots;$

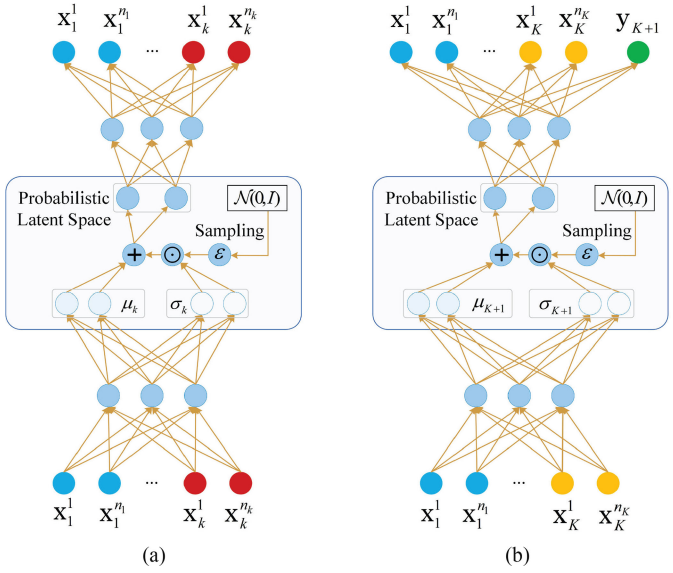

Fig. 3. Schematic of the structures of (a) model of the k-th data chunk with only partial process variables and (b) model of the $K+1$-th data chunk with both process variables and quality or performance variable. In (a), the blue neurons in the input and output layers indicate the variable group x_1 and the red ones indicate x_k. Similarly, the yellow and green neurons in (b) indicate x_K and y_{K+1}, respectively.

$x_k^{(k)}]|z^{(k)})]$ is an expected reconstruction error (RE) of the original input $[x_1^{(k)};\cdots;x_k^{(k)}]$ and the $D_{KL}\ (q_{\phi_k}\ (z^{(k)}|[x_1^{(k)};\cdots;x_k^{(k)}])||p_{\theta_k}(z^{(k)}))$ acts as a regularizer that calculates the KL divergence between the approximated posterior $q_{\phi_k}(z^{(k)}|[x_1^{(k)};\cdots;x_k^{(k)}])$ and the prior $p_{\theta_k}(z^{(k)})$.To optimize the ELBO for the first data chunk through the K-th data chunk, the deep neural networks are developed as base models whose schematic is depicted in Fig. 3(a).

2) Modeling Data Chunks With Both Process Variables and Quality or Performance Variable: For the modeling of the data chunk $\mathcal{D}^{(K+1)}$ in which both the process variables $[x_1,\ldots,x_K]$ and the quality or performance variable y are measured, note that it is a supervised task different from the chunk modeling with only partial process variables. Under this condition, using a generation model $p_\theta(x,y|z)$ instead of $p_\theta(x|z)$ in plain VAE has thus become a common choice in modeling the supervised data in industrial soft sensors [12], [17]. Inspired by this, the logarithm of the joint density of a training sample from the data chunk $\mathcal{D}^{(K+1)}$, that is, $\log p_{\theta_{K+1}}([x_1^{(K+1)};x_2^{(K+1)};\cdots;x_K^{(K+1)};y_{K+1}^{(K+1)}])$ can be given as follows by constructing the probabilistic decoder as $p_{\theta_{K+1}}([x_1^{(K+1)};\cdots;x_K^{(K+1)};y_{K+1}^{(K+1)}]|z^{(K+1)})$:

$$\begin{aligned}
&\log p_{\theta_{K+1}}\left(\left[x_1^{(K+1)};x_2^{(K+1)};\cdots;x_K^{(K+1)};y_{K+1}^{(K+1)}\right]\right)\\
&=\mathbb{E}_{q_{\phi_{K+1}}\left(z^{(K+1)}|\left[x_1^{(K+1)};\cdots;x_K^{(K+1)}\right]\right)}\\
&\quad\left[\log\left(\frac{p_{\theta_{K+1}}\left(\left[x_1^{(K+1)};\cdots;x_K^{(K+1)};y_{K+1}^{(K+1)}\right]|z^{(K+1)}\right)}{p_{\theta_{K+1}}\left(z^{(K+1)}|\left[x_1^{(K+1)};\cdots;x_K^{(K+1)};y_{K+1}^{(K+1)}\right]\right)}\right.\right.\\
&\qquad\left.\left.\times\ p_{\theta_{K+1}}\left(z^{(K+1)}\right)\right)\right]\\
&=\mathbb{E}_{q_{\phi_{K+1}}\left(z^{(K+1)}|\left[x_1^{(K+1)};\cdots;x_K^{(K+1)}\right]\right)}\\
&\quad\left[\log p_{\theta_{K+1}}\left(\left[x_1^{(K+1)};\cdots;x_K^{(K+1)};y_{K+1}^{(K+1)}\right]|z^{(K+1)}\right)\right.\\
&\qquad-\log\frac{q_{\phi_{K+1}}\left(z^{(K+1)}|\left[x_1^{(K+1)};\cdots;x_K^{(K+1)}\right]\right)}{p_{\theta_{K+1}}\left(z^{(K+1)}\right)}\\
&\qquad\left.+\log\frac{q_{\phi_{K+1}}\left(z^{(K+1)}|\left[x_1^{(K+1)};\cdots;x_K^{(K+1)}\right]\right)}{p_{\theta_{K+1}}\left(z^{(K+1)}|\left[x_1^{(K+1)};\cdots;x_K^{(K+1)};y_{K+1}^{(K+1)}\right]\right)}\right]\\
&\geq\mathbb{E}_{q_{\phi_{K+1}}\left(z^{(K+1)}|\left[x_1^{(K+1)};\cdots;x_K^{(K+1)}\right]\right)}\\
&\quad\left[\log p_{\theta_{K+1}}\left(\left[x_1^{(K+1)};\cdots;x_K^{(K+1)};y_{K+1}^{(K+1)}\right]|z^{(K+1)}\right)\right]\\
&\quad-D_{KL}\left(q_{\phi_{K+1}}\left(z^{(K+1)}|\left[x_1^{(K+1)};\cdots;x_K^{(K+1)}\right]\right)||\right.\\
&\quad\left.p_{\theta_{K+1}}\left(z^{(K+1)}\right)\right)
\end{aligned}\tag{3}$$

where the ELBO of $\log p_{\theta_{K+1}}([x_1^{(K+1)};x_2^{(K+1)};\cdots;x_K^{(K+1)};y_{K+1}^{(K+1)}])$ can be written as

$$\begin{aligned}
&\mathrm{ELBO}_{K+1}\left(\theta_{K+1},\phi_{K+1};\left[x_1^{(K+1)};\cdots;x_K^{(K+1)};y_{K+1}^{(K+1)}\right]\right)\\
&=\mathbb{E}_{q_{\phi_{K+1}}\left(z^{(K+1)}|\left[x_1^{(K+1)};\cdots;x_K^{(K+1)}\right]\right)}\\
&\quad\left[\log p_{\theta_{K+1}}\left(\left[x_1^{(K+1)};\cdots;x_K^{(K+1)};y_{K+1}^{(K+1)}\right]|z^{(K+1)}\right)\right]\\
&\quad-D_{KL}\left(q_{\phi_{K+1}}\left(z^{(K+1)}|\left[x_1^{(K+1)};\cdots;x_K^{(K+1)}\right]\right)||\right.\\
&\quad\left.p_{\theta_{K+1}}(z^{(K+1)})\right).
\end{aligned}\tag{4}$$

The ELBO on the $K+1$-th data chunk also contains two parts. Similar to (2), the last term in (4) is a regularizer that measures the KL divergence between the approximation $q_{\phi_{K+1}}(z^{(K+1)}|[x_1^{(K+1)};\cdots;x_K^{(K+1)}])$ and the prior $p_{\theta_{K+1}}(z^{(K+1)})$. Differently, as the modeling of $\mathcal{D}^{(K+1)}$ can be deemed as a supervised learning task, and thus the first term in (4) indicates the prediction error of the overall sample $[x_1^{(K+1)};\cdots;x_K^{(K+1)};y_{K+1}^{(K+1)}]$. A sketch map of the $K+1$-th base neural network in the VPTN is shown in Fig. 3(b). In the inference phase, a hidden variable $z^{(K+1)}$ can be first sampled from the probabilistic encoder $q_{\phi+1}(z^{(K+1)}|[x_1^{(K+1)},x_2^{(K+1)},\ldots,x_K^{(K+1)}])$. Then, the quality or performance variable $y_{K+1}^{(K+1)}$can be generated through the decoder network.

C. Complete Multirate Data Modeling With Progressive Knowledge Transfer

1) SGVB Lower Bound Estimation: It is noted that for the above two ELBOs in (2) and (4), the major difference lies in that there is an additional prediction error term for $y_{K+1}^{(K+1)}$ in (4), while the rest RE term and the KL divergence penalty are similar. The comparison of the two kinds of base model structures is depicted in Fig. 3.

Following VAE, assume that the prior of z in (2) and (4) are distributed as standard multivariate Gaussians, that is, $p_{\theta_k}(z)=\mathcal{N}(0,I)\ \forall k\in[1,K]$ and $p_{\theta_{K+1}}(z)=\mathcal{N}(0,I)$. Thus, the true

posterior of z is also a multivariate Gaussian. In this case, the approximated posterior of z is assumed to be a multivariate Gaussian with isotropic covariance

$$q_{\phi_k}\left(z|\left[x_1^{(k)};\cdots;x_k^{(k)}\right]\right)=\mathcal{N}\left(\mu_k,\sigma_k^2 I\right) \tag{5}$$

$$q_{\phi_{K+1}}\left(z|\left[x_1^{(K+1)};\cdots;x_K^{(K+1)}\right]\right)\Big)=\mathcal{N}\left(\mu_{K+1},\sigma_{K+1}^2 I\right) \tag{6}$$

where μ_k, σ_k, μ_{K+1}, and σ_{K+1} are estimated using the probabilistic encoder in the two models, as shown in Fig. 3.

Based on the parameterized distribution assumption of z, the KL divergence term in (2) can be analytically written as

$$\begin{aligned}&-D_{KL}\left(q_{\phi_k}\left(z|\left[x_1^{(k)};\cdots;x_k^{(k)}\right]\right)||p_{\theta_k}(z)\right)\\&=-\int\mathcal{N}\left(\mu_k,\sigma_k^2 I\right)\log\frac{\mathcal{N}\left(\mu_k,\sigma_k^2 I\right)}{\mathcal{N}(0,I)}\mathrm{d}z\\&=\frac{1}{2}\sum_{d=1}^{D}\left[\log\left(\left(\sigma_k^d\right)^2\right)-\left(\sigma_k^d\right)^2-\left(\mu_k^d\right)^2+1\right]\end{aligned} \tag{7}$$

where D is the dimensionality of z. Also, due to the merits of the parameterized distribution of z, the RE term in (2) can be calculated by sampling multiple z from $z_{k,l}=\mu_k+\sigma_k\odot\varepsilon_{k,l}$ where $\varepsilon_{k,l}\sim\mathcal{N}(0,I)$, $l\in[1,L]$, and L signifies the number of samplings. Then, the SGVB lower bound estimation of the $\mathrm{ELBO}_k(\theta_k,\phi_k;[x_1^{(k)};\cdots;x_k^{(k)}])$ in (2) can be formulated as

$$\begin{aligned}&\tilde{\mathcal{L}}_k\left(\theta_k,\phi_k;\left[x_1^{(k)};\cdots;x_k^{(k)}\right]\right)\\&=\frac{1}{L}\sum_{l=1}^{L}\mathbb{E}_{q_{\phi_k}\left(z|\left[x_1^{(k)};\cdots;x_k^{(k)}\right]\right)}\left[\log p_{\theta_k}\left(\left[x_1^{(k)};\cdots;x_k^{(k)}\right]|z\right)\right]\\&\quad+\frac{1}{2}\sum_{d=1}^{D}\left[\log\left(\left(\sigma_k^d\right)^2\right)-\left(\sigma_k^d\right)^2-\left(\mu_k^d\right)^2+1\right].\end{aligned} \tag{8}$$

Similarly, based on ELBO_{K+1} in (4), the SGVB estimation of the $K+1$-th model can be further written as

$$\begin{aligned}&\tilde{\mathcal{L}}_{K+1}\left(\theta_{K+1},\phi_{K+1};\left[x_1^{(K+1)};\cdots;x_K^{(K+1)};y_{K+1}^{(K+1)}\right]\right)\\&=\frac{1}{L}\sum_{l=1}^{L}\mathbb{E}_{q_{\phi_{K+1}}\left(z|\left[x_1^{(K+1)};\cdots;x_K^{(K+1)}\right]\right)}\\&\quad\left[\log p_{\theta_{K+1}}\left(\left[x_1^{(K+1)};\cdots;x_K^{(K+1)};y_{K+1}^{(K+1)}\right]|z\right)\right]\\&\quad+\frac{1}{2}\sum_{d=1}^{D}\left[\log\left(\left(\sigma_{K+1}^d\right)^2\right)-\left(\sigma_{K+1}^d\right)^2-\left(\mu_{K+1}^d\right)^2+1\right].\end{aligned} \tag{9}$$

2) Progressive Knowledge Transfer: According to the SGVB estimations in (8) and (9) and the model structures shown in Fig. 3, one can find that: 1) for different data chunks $\mathcal{D}^{(k)}|_{k=1}^{K}$ with process variables only, the difference lies in the dimensionalities of the input layer and output layer of different models, while the rest hidden layers and the bottleneck layer can remain the same and 2) compared with the SGVB estimations of data chunks with process variables only, the estimation of the $K+1$-th data chunk integrates an additional dimension corresponding to the label $y_{K+1}^{(K+1)}$ while the rest terms remain the same. These two properties motivate us to propagate the knowledge, that is, the parameters of the learned model, from the first model to the $K+1$-th model in a progressive fashion, in which each single network transfer belongs to a typical network-based DTL paradigm [23].

For the knowledge transfer from the k-th to the $k+1$-th model where $k\in[1,K-1]$, the $k+1$-th chunk integrates more process variables in comparison with the data chunk k, according to the discussion in Section III-A. Thus, the model parameters learned on the k-th data chunk are transferred to initialize part of the parameters of the next model. In comparison with the k-th model, the $k+1$-th model newly adds some first-layer and last-layer neurons which are integrated due to the observation of the new process variables, and the corresponding parameters are randomly initialized. On the contrary, the rest structures remain the same, including the hidden layers and the shared first-layer and last-layer neurons. The parameters connecting these layers and neurons are initialized using the parameters of the k-th model. Then, the model is incrementally refined using the $k+1$-th data chunk, and the knowledge transfer from the two models with only process variables is accomplished.

For the knowledge transfer from the K-th to the $K+1$-th model, the difference lies in the addition of the output label dimension, and the rest structure remains the same. Thus, the parameters of the K-th model are transferred to initialize the encoder and the decoder part corresponding to $[x_1^{(K+1)};x_2^{(K+1)};\cdots;x_K^{(K+1)}]$ in the $K+1$-th model first, and the parameters of the decoder corresponding to $y_{K+1}^{(K+1)}$ are initialized randomly. Then, the whole model is incrementally refined using the $K+1$-th data chunk, including both the process and the quality or performance variables with the loss function in (9).

D. VPTN-Based Soft Sensor Design

The procedure of the soft sensor development based on the proposed VPTN is described as follows.

1) Collect the multirate sampled training data.
2) Separate the multirate data into multiple data chunks and sort them from the fastest to the slowest, according to different sampling rates.
3) Build the VPTN model in the first data chunk with the learning objective (8). Then transfer the model to initialize the corresponding parameters in the second VPTN model. The rest parameters of the second model are randomly initialized.
4) Incrementally refine the second VPTN model in the second data chunk with the learning objective (8).
5) Following steps 3) and 4), progressively propagate the knowledge from the first data chunk to the $K+1$-th chunk, and incrementally refine the $K+1$-th VPTN model with the learning objective (9). Then the soft sensor model can be established.

After the VPTN-based soft sensor is developed, the online test samples with x_1 through x_K can be input to the model, and the values of the primary variable can be predicted as online estimations.

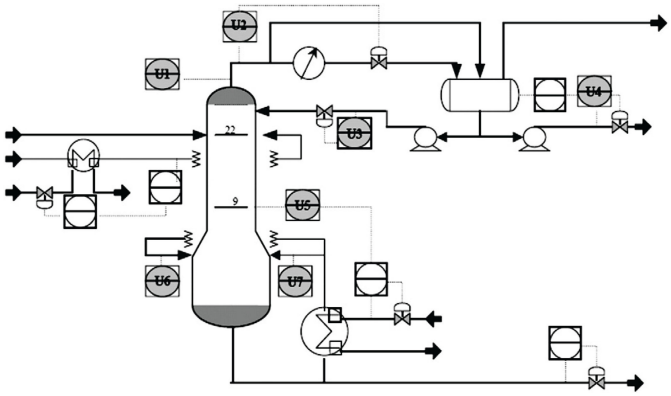

Fig. 4. Block scheme of the debutanizer column [30].

IV. Case Studies

In this section, case studies on both a debutanizer column dataset and a real thermal power plant dataset are used to verify the effectiveness of the proposed VPTN.

A. Debutanizer Column

The debutanizer column is an important component of the desulfurization and naphtha splitter plant in petroleum production processes, in which the propane and butane are removed as overheads from the naphtha stream [29]. As it is required that the butane content in the debutanizer bottom should be minimized and the direct measurement of the content is difficult, accurate soft sensor for bottom butane concentration is of great value to improve the control performance. The detailed block scheme of the column is depicted in Fig. 4, in which multiple hardware sensors are installed in the plant. In this case, seven typical variables are collected as the process variables, that is, the input of the soft sensor. The measured variables along with the corresponding names are shown in the gray circles in Fig. 4.

Among the measured variables in the debutanizer column process, five variables, including the temperature and pressure are sampled at a rate of 10 mins. The rest two variables regarding the flow measurements are collected at a sampling rate of 20 mins. The quality variable, that is, the concentration of the bottom butane is tested at lab and collected per 40 mins. The detailed description of the process variables is shown in Table I. In the debutanizer column, 2394 samples are collected. Among them, 80% are used for model training and the rest 20% are used for testing. Note that the sampling period of the soft sensor output variable is four times that of the base sampling period, and thus only 479 training samples and 120 testing samples are labeled.

To comprehensively evaluate the performance of the proposed VPTN, four methods are selected for comparison, including 1) the linear SVR; 2) lifted PLS (LPLS); 3) lifted AE (LAE); and 4) supervised VAE (SVAE), in which the progressive transfer in VPTN is ablated and only the learning objective in (9) is retained. Specifically, the SVR method is used as a baseline model, which is built using the 479 complete labeled data only. Then, to compare with the data lifting technique which is a popular solution for tackling the multirate problem, two models, including 1) a nondeep model, that is, the PLS and 2) a deep model, that is, the AE, are used for the soft sensor modeling of multirate sampled data. In particular, the process measurements are reorganized by stacking the fast-sampled variables in the original dataset, and thus the lifted dataset has 24 variables for each training and testing sample. Based on the lifted data, a PLS and an AE model are developed, respectively. For the AE method, a deep AE structure is first pretrained, and then the last layer of the decoder is replaced by a fully connected layer to predict the concentration variable. The overall network is finally finetuned by the labeled data. The encoder structure is {24, 5, 3, 2} and the decoder structure is {2, 3, 5, 24}. Besides, as the proposed VPTN is a network transfer-motivated method, an ablated version of VPTN is designed in which no transfer mechanism is conducted and only the labeled data are used to train a SVAE in (9) from scratch. For the SVAE and the proposed VPTN, both of them share the same network structure with the LAE method, except for the dimensionalities of the input and output layers. To remedy the rivalry of the items in both SVAE and VPTN and guarantee the prediction performance, a penalty

TABLE I
Detailed Description of the Input Variables for the Debutanizer Column

Variable	Variable description	Unit	Sampling Rates (mins)
U1	Top Temperature	℃	10
U2	Top pressure	kg/cm^2	10
U3	Reflux flow	m^3/h	20
U4	Flow to next process	m^3/h	20
U5	6th tray temperature	℃	10
U6	Bottom temperature A	℃	10
U7	Bottom temperature B	℃	10

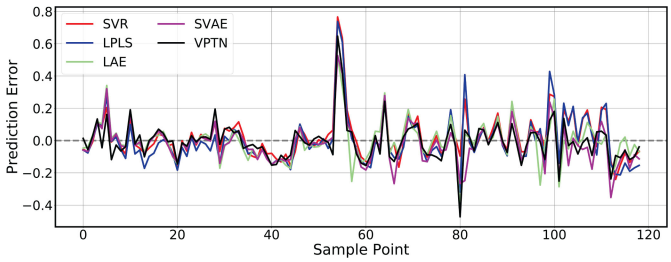

Fig. 5. Prediction error comparison for SVR, LPLS, LAE, SVAE, and VPTN.

TABLE II
Performance Comparison of the Five Methods on the Debutanizer Column Dataset

Methods	Concentration of the Bottom Butane	
	RMSE	MAE
SVR	0.1412	0.0963
LPLS	0.1541	0.1079
LAE	0.1252	0.0850
SVAE	0.1244	0.0879
VPTN	0.1207	0.0819

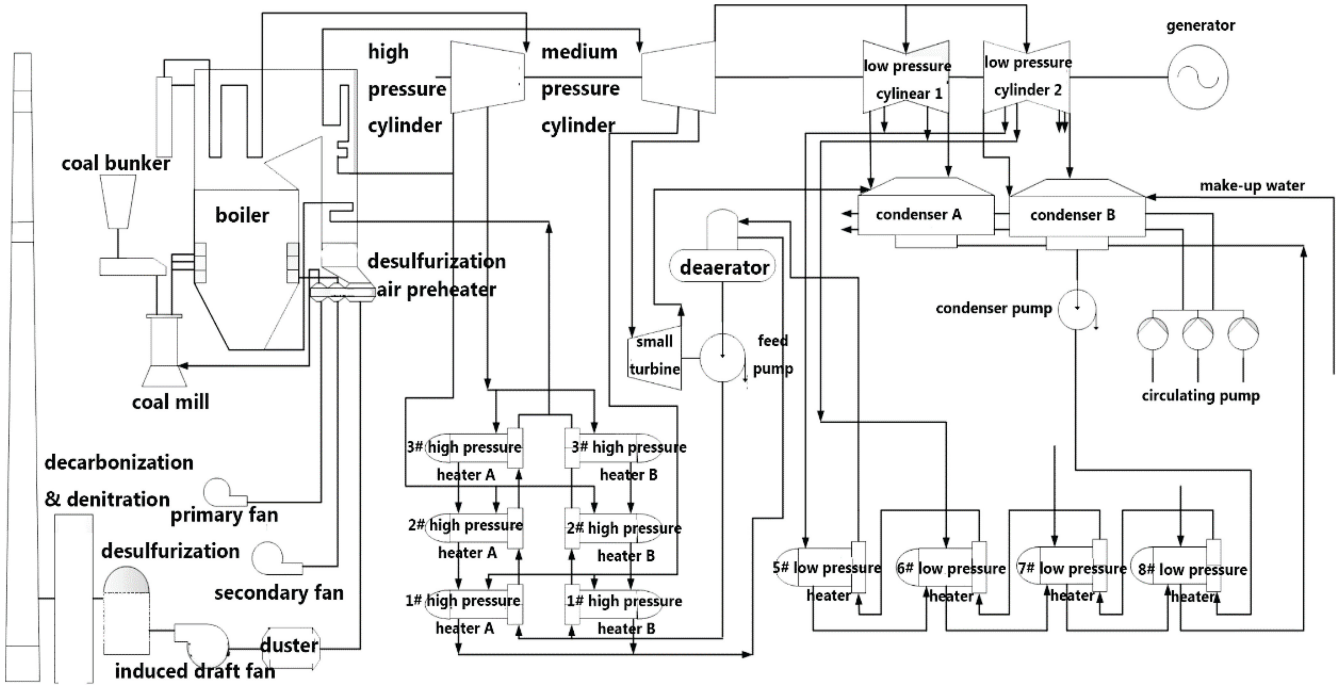

Fig. 6. Schematic of 1000-MW USC thermal power unit [31].

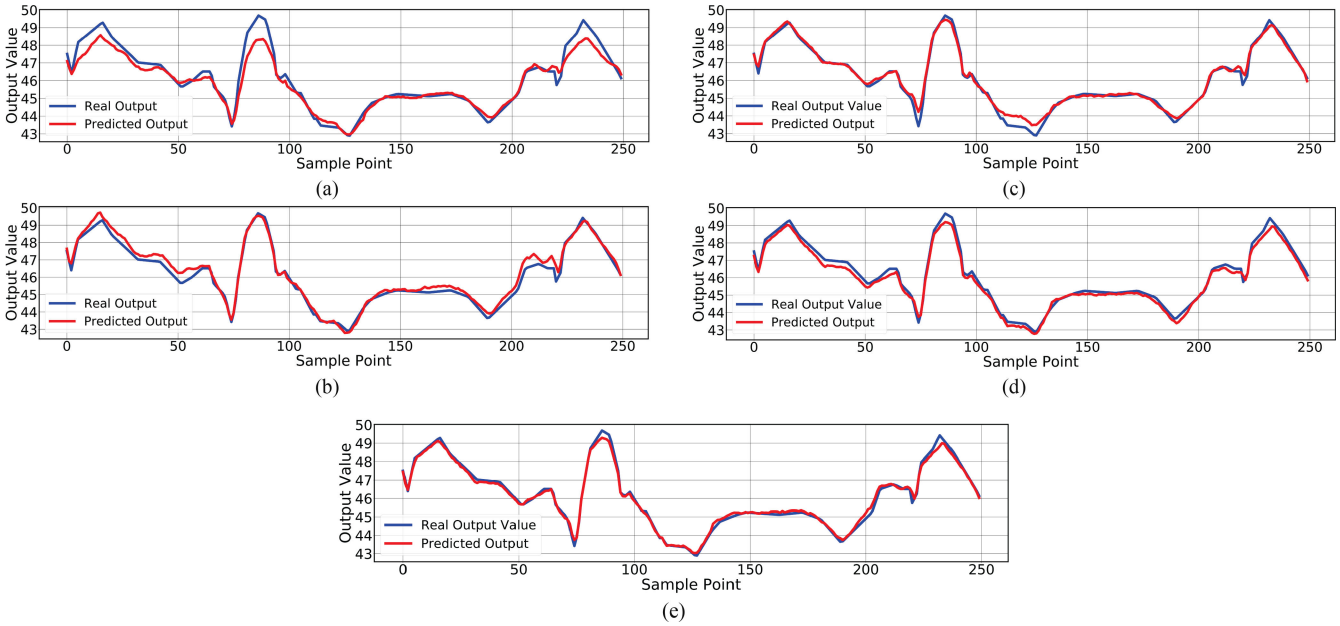

Fig. 7. Predictions and real values of (a) SVR, (b) LPLS, (c) LAE, (d) SVAE, and (e) VPTN for the motor coil temperature.

coefficient selected from {0.1, 0.01, 0.001} is adopted on the KL term and the RE term for x. The PReLU activation function and Adam optimizer is used in the deep neural networks. The learning rate is set as 0.01. Two metrics, including 1) the root mean-squared error (RMSE) and 2) the mean absolute error (MAE) are used to evaluate the prediction performance.

The comparison results on the testing data are shown in Table II. There are some observations from the table. On one hand, in comparison with the deep models, both the SVR and LPLS show the inferior performance. This can be attributed to the advantage of learning complex features of the deep structures. On the other hand, for the two compared deep methods LAE and SVAE, both of them are inferior to the designed VPTN method. The potential reason is twofold. First, LAE simply lifts the original data as a much higher dimensional dataset, and the uncertainty distribution information in the dataset is less considered in the deterministic feature learning of AE. Second, the SVAE method is trained based on the complete data only and is fully learned from scratch, and thus the random initialization leads to weak performance in comparison with a pretrained model [27]. To clearly show the comparison results, the visualization of the prediction error of

TABLE III
PERFORMANCE COMPARISON OF THE FIVE METHODS ON THE COAL MILL DATASET

Methods	Motor Coil Temperature		Planetary Gear Bearing Temperature		Rotary Separator Bearing Temperature		Mean RMSE	Mean MAE
	RMSE	MAE	RMSE	MAE	RMSE	MAE		
SVR	0.4563	0.3177	0.4237	0.3740	0.3971	0.3455	0.4257	0.3457
LPLS	0.2780	0.2203	0.2984	0.2826	0.4011	0.3141	0.3259	0.2723
LAE	0.2289	0.1244	0.1690	0.1333	0.3551	0.2854	0.2510	0.1810
SVAE	0.2329	0.2067	0.1650	0.1471	0.3333	0.2661	0.2437	0.2066
VPTN	0.1522	0.1158	0.1520	0.1300	0.3079	0.2438	0.2040	0.1632

different methods on the testing data is illustrated in Fig. 5. It can be observed that the proposed VPTN provides more accurate results in tracking real values.

B. Thermal Power Plant

In this part, the proposed method is illustrated through a real-world thermal power case. The 1000-MW ultrasupercritical (USC) unit is a highly complex industrial process with the rated superheated steam pressure of 30 MPa and the temperature of 600 °C [31]. A schematic view of the USC thermal power unit is shown in Fig. 6. One of the important machines in the USC thermal power plant, that is, the coal mill, is selected as the testbed to verify the performance.

In this case, 36 variables are measured and there are 15 000 samples overall. The goal of the coal mill is to crush the mill for improving further power generation efficiency. A too low temperature of some key components in the coal mill will affect the crushing efficiency, while a very high temperature will create security risks. Thus, among the measured variables, three variables measured on key components of the coal mill, including 1) the motor coil temperature; 2) the planetary gear bearing temperature; and 3) the rotary separator bearing temperature, are selected as the soft sensor outputs, whose sampling rates are 20 mins. For the rest 33 variables, they can be divided into three groups according to different sampling rates. The fastest sampling variable group consists of three variables, including 1) the inlet coal mass; 2) the ambient temperature; and 3) the power signal value. The sampling period is 1 min. The second variable group consists of nine variables, including the current value, the wind pressure, and so on, and is sampled per 2 mins. Temperature variables measured on other components form the third process variable group, which are sampled per 10 mins. Among the collected 15 000 samples, 10 000 samples are used for model training, and the rest 5000 ones are used for evaluation. Thus, according to the description of the multiple sampling rates, 5000, 4000, and 500 samples in the training data are measured on 3, 12, and 33 variables, respectively. The rest 500 samples are complete with all the 36 variables measured. Similarly, only 250 samples out of the 5000 samples are labeled in the testing dataset. Thus, this is a more realistic and challenging task in comparison with the debutanizer column case. To compare the performance with the proposed VPTN, the SVR, LPLS, LAE, and SVAE are selected as the comparison methods, which have been introduced in the previous case study.

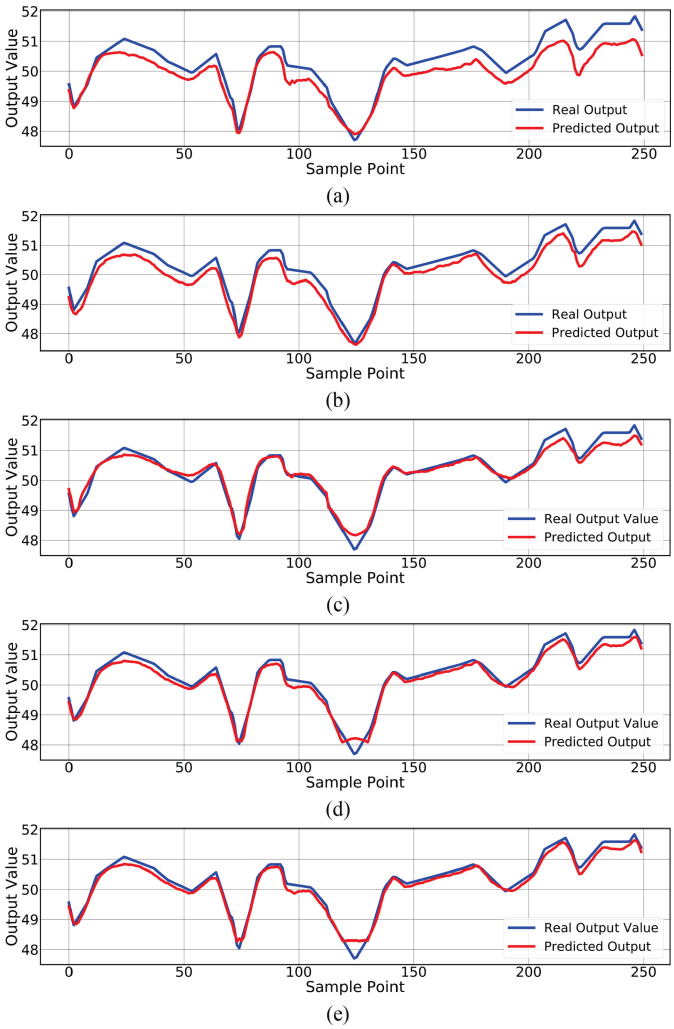

Fig. 8. Predictions and real values of (a) SVR, (b) LPLS, (c) LAE, (d) SVAE, and (e) VPTN for the planetary gear temperature.

The comparison results on the motor coil, planetary gear bearing, and rotary separator bearing temperatures on the testing dataset are shown in Table III. From the table, it can be seen that the proposed VPTN shows much better performance than the other four methods on both RMSE and MAE metrics, due to its strong capability of learning uncertainty, nonlinearity, and transferability. Besides, it is found that in this case where the sampling rate has considerable variation, the performance of the SVR is inferior to that of LPLS. The LAE

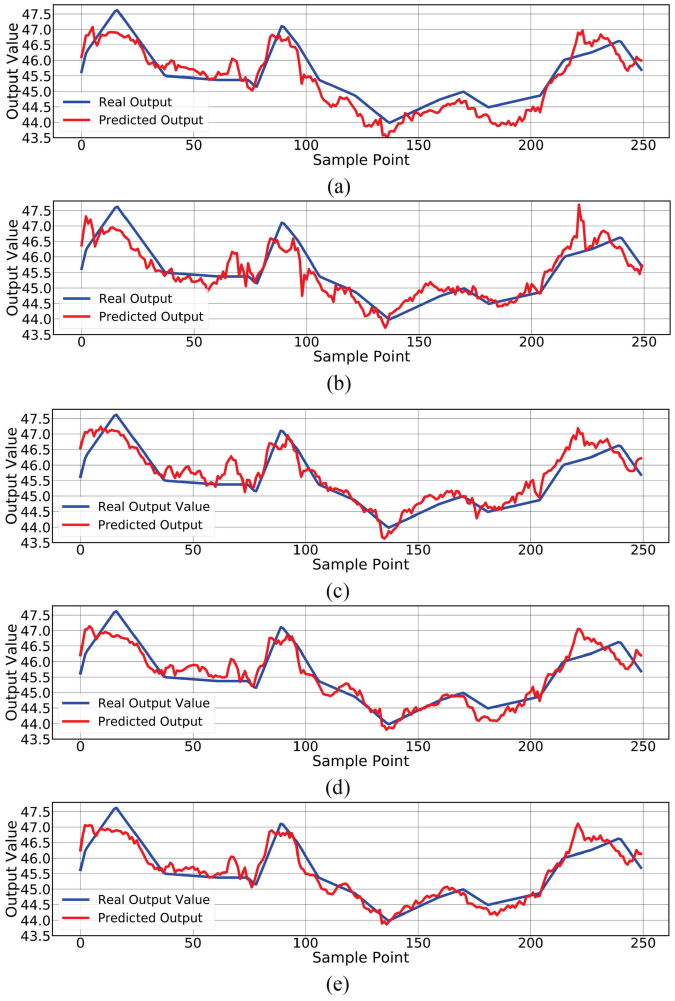

Fig. 9. Predictions and real values of (a) SVR, (b) LPLS, (c) LAE, (d) SVAE, and (e) VPTN for the rotary separator bearing temperature.

and SVAE show similar performance on the three cases, which are better than the nondeep models like SVR and LPLS.

Furthermore, the visualization of the predictions and real values of the five methods on the three output variables are shown in Figs. 7–9. First, it is observed that in Fig. 7, the prediction of the motor coil temperature is relatively easier than the other two variables, and the designed VPTN shows more accurate prediction ability on the testing data. The LAE method presents the second-best performance on this variable, and the RMSE and MAE are 0.0767 and 0.0086 higher than those of the designed VPTN, respectively. Then, in Fig. 8, the proposed VPTN shows better performance than SVAE on the testing samples (0.1520 versus 0.1650 on RMSE, 0.1300 versus 0.1471 on MAE), while their prediction performances on the 120–130th samples are weak than that of the SVR and LPLS. Finally, as shown in Fig. 9, the prediction of the rotary separator bearing temperature is rather difficult in comparison with the first two tasks. The SVR and LPLS show weak tracking ability in the rotary separator bearing temperature case. The tracking performances of the rest three methods are similar, and the proposed VPTN provides more accurate results. Particularly, the RMSE and MAE values of the proposed VPTN are reduced by 7.62% and 8.38% in comparison with the second-best method.

V. Conclusion

In this article, a VPTN is developed for the soft sensor modeling of multirate industrial processes. The designed VPTN consists of a variational data modeling part and a progressive transfer part. The variational data modeling part is driven by learning multiple base models for different data chunks in the multirate sampled dataset. The uncertainty distribution is characterized and different base models are unified into a transferrable structure. Then, the progressive strategy is designed to sequentially transfer the model from the fastest process data chunk to the slowest chunk, based on which the overall multirate data can be modeled within a unified framework. Experiments are conducted on both a debutanizer column case and a real coal mill case in a thermal power plant. The results illustrate that the designed method can significantly improve the performance of the soft sensor in multirate processes in comparison with the model trained from scratch and other existing methods. Future research topics include the development of soft sensors for industrial processes with measurement outliers [32], [33] or nonstationary characteristics [34].

References

[1] Y. Zhao, A. Fatehi, and B. Huang, "Robust estimation of ARX models with time varying time delays using variational Bayesian approach," *IEEE Trans. Cybern.*, vol. 48, no. 2, pp. 532–542, Feb. 2018.

[2] C. Zhao, F. Wang, Z. Mao, N. Lu, and M. Jia, "Quality prediction based on phase-specific average trajectory for batch processes," *AIChE J.*, vol. 54, no. 3, pp. 693–705, Mar. 2008.

[3] 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.

[4] J. Corrigan and J. Zhang, "Integrating dynamic slow feature analysis with neural networks for enhancing soft sensor performance," *Comput. Chem. Eng.*, vol. 139, Aug. 2020, Art. no. 106842.

[5] Z. Chai and C. Zhao, "Enhanced random forest with concurrent analysis of static and dynamic nodes for industrial fault classification," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 54–66, Jan. 2020.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[7] J. Yu and X. Yan, "Whole process monitoring based on unstable neuron output information in hidden layers of deep belief network," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3998–4007, Sep. 2020.

[8] 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.*, early access, Aug. 24, 2020, doi: 10.1109/TNNLS.2020.3015929.

[9] C. Shang, F. Yang, D. Huang, and W. Lyu, "Data-driven soft sensor development based on deep learning technique," *J. Process Control*, vol. 24, no. 3, pp. 223–233, Mar. 2014.

[10] X. Yuan, Y. Gu, Y. Wang, C. Yang, and W. Gui, "A deep supervised learning framework for data-driven soft sensor modeling of industrial processes," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4737–4746, Nov. 2020.

[11] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–14.

[12] R. Xie, N. M. Jan, K. Hao, L. Chen, and B. Huang, "Supervised variational autoencoders for soft sensor modeling with missing data," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2820–2828, Apr. 2020.

[13] F. Guo, W. Bai, and B. Huang, "Output-relevant variational autoencoder for just-in-time soft sensor modeling with missing data," *J. Process Control*, vol. 92, pp. 90–97, Aug. 2020.

[14] B. Lin, B. Recke, T. M. Schmidt, J. K. H. Knudsen, and S. B. Jorgensen, "Data-driven soft sensor design with multiple-rate sampled data: A comparative study," *Ind. Eng. Chem. Res.*, vol. 48, no. 11, pp. 5379–5387, May 2009.

[15] Y. Zhang, H. Fang, Y. Zheng, and X. Li, "Torus-event-based fault diagnosis for stochastic multirate time-varying systems with constrained fault," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2803–2813, Jun. 2020.

[16] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semisupervised learning with deep generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3581–3589.

[17] B. Shen, L. Yao, and Z. Ge, "Nonlinear probabilistic latent variable regression models for soft sensor application: From shallow to deep structure," *Control Eng. Pract.*, vol. 94, Jan. 2020, Art. no. 104198.

[18] V. Gopakumar, S. Tiwari, and I. Rahman, "A deep learning based data driven soft sensor for bioprocesses," *Biochem. Eng. J.*, vol. 136, no. 15, pp. 28–39, Aug. 2018.

[19] W. Zheng, Y. Liu, Z. Gao, and J. Yang, "Just-in-time semi-supervised soft sensor for quality prediction in industrial rubber mixers," *Chemometr. Intell. Lab. Syst.*, vol. 180, no. 15, pp. 36–41, Sep. 2018.

[20] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *J. Royal Stat. Soc. B, Stat. Methodol.*, vol. 61, no. 3, pp. 611–622, Jan. 2002.

[21] R. Amirthalingam and J. H. Lee, "Subspace identification based inferential control applied to a continuous pulp digester," *J. Process Control*, vol. 9, no. 5, pp. 397–406, Oct. 1999.

[22] 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.

[23] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Proc. Int. Conf. Artif. Neural Netw.*, 2018, pp. 270–279.

[24] W. Lu, B. Liang, Y. Cheng, D. Meng, J. Yang, and T. Zhang, "Deep model based domain adaptation for fault diagnosis," *IEEE Trans. Ind. Electron.*, vol. 64, no. 3, pp. 2296–2305, Mar. 2017.

[25] H. Chen, Z. Chai, B. Jiang, and B. Huang, "Data-driven fault detection for dynamic systems with performance degradation: A unified transfer learning framework," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2020, doi: 10.1109/TIM.2020.3033943.

[26] Z. Chai and C. Zhao, "A fine-grained adversarial network method for cross-domain industrial fault diagnosis," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 3, pp. 1432–1442, Jul. 2020.

[27] S. Shao, S. McAleer, R. Yan, and P. Baldi, "Highly-accurate machine fault diagnosis using deep transfer learning," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2446–2455, Apr. 2019.

[28] D. Chen, S. Yang, and F. Zhou, "Transfer learning based fault diagnosis with missing data due to multi-rate sampling," *Sensors*, vol. 19, no. 8, pp. 1826–1846, Apr. 2019.

[29] J. Feng and K. Li, "MRS-kNN fault detection method for multirate sampling process based variable grouping threshold," *J. Process Control*, vol. 85, pp. 149–158, Jan. 2020.

[30] L. Fortuna, S. Graziani, A. Rizzo, and M. G. Xibilia, *Soft Sensors for Monitoring and Control of Industrial Processes*. London, U.K.: Springer-Verlag, 2007.

[31] C. Zhao and H. Sun, "Dynamic distributed monitoring strategy for large-scale nonstationary processes subject to frequently varying conditions under closed-loop control," *IEEE Trans. Ind. Electron.*, vol. 66, no. 6, pp. 4749–4758, Jun. 2019.

[32] L. Zou, Z. Wang, H. Geng, and X. Liu, "Set-membership filtering subject to impulsive measurement outliers: A recursive algorithm," *IEEE/CAA J. Automatica Sinica*, vol. 8, no. 2, pp. 377–388, Feb. 2021.

[33] L. Zou, Z. Wang, H. Dong, and Q.-L. Han, "Energy-to-peak state estimation with intermittent measurement outliers: The single-output case," *IEEE Trans. Cybern.*, early access, Mar. 22, 2021, doi: 10.1109/TCYB.2021.3057545.

[34] C. Zhao, J. Chen, and H. Jing, "Condition-driven data analytics and monitoring for wide-range nonstationary and transient continuous processes," *IEEE Trans. Autom. Sci. Eng.*, early access, Aug. 4, 2020, doi: 10.1109/TASE.2020.3010536.

Zheng Chai received the B.Eng. degree in automation from the College of Automation, Harbin Engineering University, Harbin, China, in 2017. He is currently pursuing the Ph.D. degree in control science and engineering with the College of Control Science and Engineering, Zhejiang University, Hangzhou, China.

He was a Visiting Scholar with the Department of Chemical and Materials Engineering, University of Alberta, Edmonton, AB, Canada, from 2019 to 2020. His current research interests include deep learning and its industrial applications.

Chunhui Zhao (Senior Member, IEEE) received the Ph.D. degree from Northeastern University, Shenyang, China, in 2009.

From 2009 to 2012, she was a Postdoctoral Fellow with the Hong Kong University of Science and Technology, Hong Kong, and the University of California at Santa Barbara, Los Angeles, CA, USA. Since January 2012, she has been a Professor with the College of Control Science and Engineering, Zhejiang University, Hangzhou, China. She has authored or coauthored more than 140 papers in peer-reviewed international journals. Her research interests include statistical machine learning and data mining for industrial application.

Dr. Zhao has served a Senior Editor for *Journal of Process Control*, and an Associate Editor for two International Journals, including *Control Engineering Practice* and *Neurocomputing*.

Biao Huang (Fellow, IEEE) received the B.Sc. and M.Sc. degrees in automatic control from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 1983 and 1986, respectively, and the Ph.D. degree in process control from the University of Alberta, Edmonton, AB, Canada, in 1997.

He joined the University of Alberta in 1997, as an Assistant Professor with the Department of Chemical and Materials Engineering, where he is currently a Professor and the NSERC Industrial Research Chair of Control of Oil Sands Processes. He has applied his expertise extensively in industrial practice. His current research interests include process control, system identification, control performance assessment, Bayesian methods, and state estimation.

Dr. Huang was a recipient of the Germany's Alexander von Humboldt Research Fellowship, the Canadian Chemical Engineer Society's Syncrude Canada Innovation and D. G. Fisher Awards, the APEGA Summit Research Excellence Award, the University of Alberta McCalla and Killam Professorship Awards, the Petro-Canada Young Innovator Award, and the Best Paper Award from the *Journal of Process Control*. He is a Fellow of the Canadian Academy of Engineering and the Chemical Institute of Canada.