# Multi-task Learning with Latent Variation Decomposition for Multivariate Responses in a Manufacturing Network

Yifu Li, *Member, IEEE*, Hao Yan, *Member, IEEE*, and Ran Jin, *Member, IEEE*

***Abstract*— Modeling the relationships between the quality response variables and process settings or *in situ* sensing variables is a fundamental problem in quality engineering. Such relationships are important for product quality prediction, process monitoring, optimization. Data collected from a single system often only carry limited information, hence making modeling one system at a time challenging. Multi-task learning (MTL) jointly models multiple similar-but-non-identical systems and utilize the similarities among systems for better performance. However, existing MTL becomes much less effective if important variables are missing or unmeasurable in the underlying process (latent variables). More importantly, commonly shared latent variables across systems often reflect important process patterns/behaviors. We proposed an MTL framework for multivariate or profile responses by explicitly decomposing the variation among systems into explainable variation and latent variation. Specifically, the explainable variation is from variables observed in data, while the latent variation is from the latent basis functions automatically generated from model residuals. The proposed method improves the prediction accuracy and interpretability of modeling. The simulation and a case study in a silicon ingot manufacturing network demonstrate that the proposed method can improve the quality modeling performance and recovers critical process knowledge for silicon ingot manufacturing based on Czochralski (CZ) process.**

***Note to Practitioners*— This research is motivated by quality modeling of a semiconductor manufacturing network consist of multiple furnaces (systems) producing silicon ingots. An accurate quality model in manufacturing is essential for downstream tasks such as process monitoring and optimization. As data collected from a single system often only carry limited information, aggregating data from multiple systems in quality modeling can significantly improve the performance. However, different systems in a network are often similar-but-non-identical to each other due to different degradation status, usage history, product receipts. As a result, data from different systems are heterogeneous from each other, hence making combining all data for modeling inappropriate. Multi-task learning (MTL) solves this problem and recovers the similar-but-non-identical nature of systems. However, existing MTL is much less effective if important latent factors/variables are missing or unmeasurable in the underlying process. More importantly, such latent factors can reflect important process patterns/behaviors. In this work, we improve the MTL framework by explicitly explaining the unexplained variations using the latent factors automatically generated from model residuals. The simulation and a case study in a silicon ingot manufacturing network demonstrate that the proposed method improve the quality modeling performance and recovers critical process knowledge for silicon ingot manufacturing based on Czochralski (CZ) process.**

***Index Terms*— Czochralski (CZ) process, decomposition modeling, manufacturing network, multi-task learning, semiconductor manufacturing**

## I. Introduction

A manufacturing network connects manufacturing systems via sensing, computation, and actuation networks. For example, in the silicon ingot production of wafer manufacturing, shown in Fig. 1, each silicon ingot grows from a furnace/crucible using the same Czochralski (CZ) process [2]. However, an ingot typically requires over 50 hours to produce. Multiple furnaces are typically used as a connected manufacturing network to increase throughput. Different ingot growth furnaces are subject to different degradation conditions and operating environments, which lead to a different variable relationship among the *in situ* process variables and product quality variables [3]. Consequently, the data are collected from similar-but-not-identical furnaces. Another challenge is that important factors/variables are constantly missing or unmeasurable in various manufacturing processes (referred as latent factors). For examples, in silicon ingot production, critical variables such as deposition thickness can not be measured while the ingots are under formation [4]. More importantly, such latent factors, which assume to be shared across systems, can reflect important process

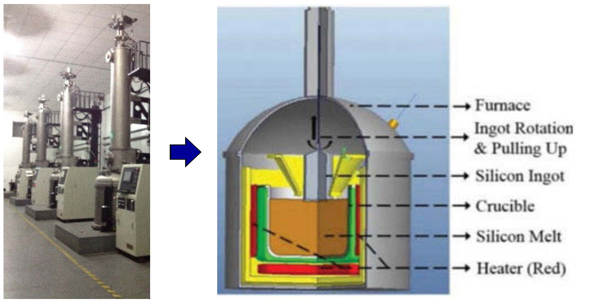


Fig. 1. A schematic of a crystal growth furnace network: the furnaces (left) and its internal structure (redrawn with authors' permission) [1]

This paper is submitted on 12/07/2020. This work was supported by the National Science Foundation under Grant CMMI-1634867

Yifu Li is with Department of Industrial and Systems Engineering at University of Oklahoma, Norman, OK, 73069, USA (e-mail. liyifu@ou.edu)

Hao Yan is with School of Computing, Informatics, and Decision Systems Engineering at Arizona State University, Phoenix, AZ, 85281, USA (e-mail. haoyan@asu.edu)

Ran Jin is with Grado Department of Industrial and Systems Engineering at Virginia Tech, Blacksburg, VA, 24060, USA (e-mail. jran5@vt.edu)

patterns/behaviors. We will demonstrate this observation in the case study. All above issues pose significant challenges for quality modeling in a manufacturing network.

In literature, there are two major strategies to model similar-but-not-identical systems (e.g., furnaces). One strategy is to model all systems with the same model structure and model parameters. It assumes that all systems are identical [1]. They do not consider the heterogeneities among the systems and cannot reflect the heterogeneities in the model structures or parameters.

The other strategy will use the same type of model with different model parameters for different machines. The model can be independently estimated for each machine based on its own data, but it may take a longer time to obtain sufficient samples. To model the data aggregated from different machines with heterogeneity, people proposed transferred learning and multi-task learning (MTL) framework. For transfer learning [5], it designates a source model, which is trusted to be accurate for the source system and transfers the model with updated parameters to the target system by incorporating the system similarities. However, it is sometimes difficult to find a reliable source model when all systems are experiencing a shortage of training samples. As a result, MTL, which simultaneously models the data from all systems, is a more suitable framework to adopt under this case scenario.

MTL treats modeling the data from each system/machine as a learning task. MTL jointly estimates the model parameters for multiple tasks and can adopt different regularizations to recover the similar-but-non-identical nature of model parameters among systems. In this work, we propose MTL with latent variation decomposition (LVD), called *MTL-LVD*, to integrate the parameter-based MTL framework with principal component analysis (PCA), which explains the latent variation in the model residual space for multivariate responses. The simulation study and the case study on the silicon ingot manufacturing data show that MTL-LVD discovers the latent variation patterns among similar-but-non-identical systems and achieve significantly better quality prediction accuracy.

The rest of the paper is organized as follows. Section II reviews the past literature on CZ process modeling and relevant technics. Section III introduces the proposed MTL-LVD framework; Section IV includes a simulation study, examining the performance of MTL-LVD on discovering the multi-dimensional latent variation terms and the model prediction accuracy; Section V performs a case study, illustrating the superior performance of MTL-LVD on product quality forecasting based on the ingot manufacturing data; Section VI draws the conclusion and discussed some future research directions.

## II. Literature Review

Data-driven analysis and modeling help CZ process to explore its properties, optimize its productivity, have been extensive studied in the past. For example, researchers studied the dynamic properties of ingot striations using functional data analysis [6], defect concentration using neural network models [7], quality modeling using logistic regression [1], model-based CZ process control. The simplest approach to model all systems is to assume that all systems are identical with the same model structure and model parameters [1]. In the literature, various predictive regression models, such as Ridge Regression [8], LASSO [9], Elastic Net [10], can be estimated by using the identical system assumption. Since they do not consider the heterogeneities among the systems so they cannot reflect the heterogeneities in the model structures or parameters.

To handle the heterogeneities, people proposed MTL to jointly estimate the model parameters for multiple systems using various regularizations to enforce the similarity. Commonly adopted regularization includes low-rank regularization [11, 12], cluster-wise parameter regularization [13, 14], task-relation learning [15, 16]. Other MTL works include recovering variable relationships among tasks through multi-task sparse structure learning [17], improved clustering using multi-task nonnegative matrix factorization [18], and improved principal component analysis through Multitask PCA [19]. A major limitation of the MTL methods is the assumption that the observed variables can fully explain the variation of data. When important factors/variables are missing or unmeasurable, failing to estimate the variation pattern in the residual space can significantly impact the modeling performance.

For recovering the latent variation that can not be explained by the observed variables from residual space, we can apply principal component analysis (PCA). In the literature, there have been various PCA works handling various types of data such as kernel PCA [20], sparse PCA [21]. To handle the functional or tensor-structured data, there have been various functional PCA, such as smoothed FPCA [22], sparse FPCA [23], sparse longitudinal FPCA [24], multi-channel FPCA (MFPCA) [25], sparse multi-channel FPCA (SMFPCA) [26]. The past methods can effectively discover the latent variation patterns from data in an unsupervised fashion. However, extending the knowledge frontier on applying unsupervised variation decomposition for a predictive MTL model remains a challenge. In this work, we propose to integrate PCA for latent variation decomposition (LVD) to recover the latent variation in the residual space. We refer to the proposed method as *MTL-LVD*.

## III. The Proposed Data Filtering Method

### A. MTL-LVD

Without the loss of generality, we can assume that there are $m$ similar-but-non-identical systems and MTL-LVD is build based on the following assumption: 1) We assume that there are $m$ similar-but-non-identical systems/machines in a manufacturing network and the $j^{th}$ machine is producing $n_j$ parts (e.g., ingots). 2) When building a linear model for each system, model parameters for multiple systems are similar-but-non-identical (e.g., following low-rank structure). 3) There can be latent variation among systems lying in a low-dimensional linear space, which can be represented by a set of bases.

A linear regression model for the observable predictors, such as process setting variables or scalar features of *in situ* variables, $\boldsymbol{x}_j = \left(x_j{}^{(1)}, \cdots, x_j{}^{(p)}\right)' \in \mathbb{R}^{p\times 1}$ and a single response $y_j$ of the machine/system $j$, where $j = 1, \dots, m$, can be written as

$$y_j = \boldsymbol{x}_j{}'\boldsymbol{\beta}_j + \epsilon_j\,,\ j = 1, \cdots, m, \tag{1}$$

where $\boldsymbol{\beta}_j \in \mathbb{R}^{p\times 1}$ and the residual $\epsilon$ follows the normal i.i.d. distribution $N(0, \sigma^2)$. Multi-task learning (MTL) aims to jointly improve the estimation accuracy of $\boldsymbol{\beta}_j$ for multiple similar-but-non-identical systems by enforcing the similarity of $\boldsymbol{\beta}_j$. One of the most commonly adopted regularizations for MTL is nuclear norm $||\cdot||_*$ [27], which ensures the estimated $\boldsymbol{\beta}_j$ form low-rank structure matrix as $B = \begin{pmatrix} \boldsymbol{\beta}_1{}^{(1)} & \cdots & \boldsymbol{\beta}_m{}^{(1)} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\beta}_1{}^{(d)} & \cdots & \boldsymbol{\beta}_m{}^{(d)} \end{pmatrix}$ across systems

$$min \sum_{j=1}^{m} \left\| y_j - \boldsymbol{x}_j{}'\boldsymbol{\beta}_j{}^{(r)} \right\|_2^2 + \lambda_1 \|B\|_* \tag{2}$$

where the model hyper-parameter $\lambda_1$ of the nuclear-norm can be selected via five-fold cross-validation (CV).

In this work, to model multivariate responses for systems with $d$ responses, the observed data for each machine $j$ become $(\boldsymbol{x}_j,\ \boldsymbol{y}_j)$, where $\boldsymbol{y}_j = \left(y_j{}^{(1)}, \cdots, y_j{}^{(d)}\right)' \in \mathbb{R}^{d\times 1}$, $j = 1, \dots, m$. Furthermore, we propose to decompose the latent variation patterns into a low-dimensional subspace with latent parameters for each response $r$ as $\boldsymbol{v}^{(r)} = (v^{(r)}{}_1, \dots, v^{(r)}{}_q)'$, $r = 1, \dots, d$. The corresponding latent variables for each system $j$ are denoted as $\boldsymbol{u}_j = (u_{1,j}, \dots, u_{q,j})'$. Collectively, $y_j{}^{(r)}$ can be decoupled into the observable variations explained by the input variable $\boldsymbol{x}_j$ and the latent variables $\boldsymbol{u}_j$ as

$$y_j{}^{(r)} = \boldsymbol{x}_j{}'\boldsymbol{\beta}_j{}^{(r)} + \boldsymbol{u}_j{}'\boldsymbol{v}^{(r)} + \epsilon_j^{(r)}, j = 1, \cdots, m \tag{3}$$

where $\boldsymbol{\beta}_j{}^{(r)} \in \mathbb{R}^{p\times 1}$ is the model coefficient machine $j$ and response $r$, and the residual $\epsilon_j^{(r)}$ follows the normal i.i.d distribution $N(0, \sigma^2)$. It worth noting that the assumption would not hurt the generalization of the proposed method since we can always use more sets of latent parameters $\boldsymbol{v}^{(r)}$ to model the non-i.i.d. variation patterns.

Since the machines are similar-but-non-identical, we assume that the model coefficients among machines $\boldsymbol{\beta}_j = 1, \cdots, m$ form low-rank structures, which can be recovered by nuclear-norm regularization. Without loss of generality, we can assume that the latent parameters $\boldsymbol{v}^{(r)}$ are shared among all machines $j$. Collectively, we propose to borrow the framework from the parameter-based MTL and estimate the model parameters $\boldsymbol{\beta}_j{}^{(r)}$ via

$$\begin{gathered} min \sum_{j=1}^{m} \sum_{r=1}^{d} \left\| y_j{}^{(r)} - \boldsymbol{x}_j{}'\boldsymbol{\beta}_j{}^{(r)} - \boldsymbol{u}_j{}'\boldsymbol{v}^{(r)} \right\|_2^2 \\ + \lambda_1 \|B\|_* \\ s.t. V'V = I \end{gathered} \tag{4}$$

where $B = \begin{pmatrix} \boldsymbol{\beta}_1{}^{(1)} & \cdots & \boldsymbol{\beta}_m{}^{(1)} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\beta}_1{}^{(d)} & \cdots & \boldsymbol{\beta}_m{}^{(d)} \end{pmatrix}$ and $V' = (\boldsymbol{v}^{(1)} \ \ \cdots \ \ \boldsymbol{v}^{(d)})$. Here, $B$ represents model coefficient matrix corresponding to the predictors for $d$ responses and $j$ systems. Furthermore, $V$ is the matrix realized from $\boldsymbol{v}^{(r)}$ for $d$ responses. To ensure that $V$ supports the latent space, we assume that $V$ is a semi-orthogonal matrix, which is guaranteed by the constraint $V'V = I$. It is worth mentioning that we do not enforce the orthogonality between $B$ and $V$ to ensure the fidelity of parameters estimated for the observable predictors.

### *B. Parameter estimation and hyper-parameter tuning of MTL-LVD*

Estimating the parameters of MTL-LVD in Equation (4) is not trivial due to the combination of the nuclear norm, orthogonality constraint, and decomposed model coefficients (i.e. $\boldsymbol{\beta}_j{}^{(r)}$ and $\boldsymbol{v}^{(r)}$). We proposed the regularized regression approach together with the block coordinate descent algorithm to iteratively update $\boldsymbol{\beta}_j{}^{(r)}$, $\boldsymbol{u}_j$, and $\boldsymbol{v}^{(r)}$. The model hyper-parameters in addition to $\lambda_1$, such as $q$, the dimension of the latent variation term $V$ in Equation (4), can be tuned via the five-fold cross-validation (CV).

Realizing the Equation (4) considering the observational data, we denote $X_j = (\boldsymbol{x}_{1,j}, \dots, \boldsymbol{x}_{n_j,j})'$ as the design matrix, $U_j = (\boldsymbol{u}_{1,j}, \dots, \boldsymbol{u}_{n_j,j})'$ as the generated PCA scores, and $\boldsymbol{y}_j{}^{(r)} = (y_{1,j}{}^{(r)}, \dots, y_{n_j,j}{}^{(r)})'$ to be the $r$th response vector for the $j$th system, where $n_j$ is the number of samples for the $j$th system. The following constrained quadratic optimization problem can be used to estimate $B, U, V$.

$$\begin{gathered} min \sum_{j=1}^{m} \sum_{r=1}^{d} \left\| y_j{}^{(r)} - X_j\boldsymbol{\beta}_j{}^{(r)} - U_j\boldsymbol{v}^{(r)} \right\|^2 \\ + \lambda_1 \|B\|_*, \\ s.t. V'V = I \end{gathered} \tag{5}$$

Equation (5) can be further written by organizing the multivariate responses in a matrix format as

$$\begin{gathered} min \sum_{j=1}^{m} \left\| Y_j - X_jB_j - U_jV' \right\|^2 + \lambda_1 \|B\|_*, \\ s.t. V'V = I \end{gathered} \tag{6}$$

where $Y_j = (\boldsymbol{y}_j{}^{(1)}, \dots, \boldsymbol{y}_j{}^{(d)})$ and $B_j = \left(\boldsymbol{\beta}_j{}^{(1)} \ \ \cdots \ \ \boldsymbol{\beta}_j{}^{(d)}\right)$. To optimize, we propose to follow the block coordinate type of strategy to update $B, U, V$ iteratively until convergence. To achieve this, we first prove the following propositions:

**Proposition 1**: Given $B$, $U_j$ in Equation (6), $V$ can be solved in a closed form via the singular value decomposition (SVD) as follows:

$$\begin{pmatrix} Y_1 - X_1{B_1}^{k-1} \\ \vdots \\ Y_m - X_m{B_m}^{k-1} \end{pmatrix}' \begin{pmatrix} {U_1}^{k-1} \\ \vdots \\ {U_m}^{k-1} \end{pmatrix} = RDW', \tag{7}$$

$$V = RW',$$

where the first $q$ columns in $R$ are used for $V = RW'$ if $q < d$. This proposition is also named as the Procrustes rotation procedure, and the detailed proof is shown in [21].

**Proposition 2**: Given $B_j$ and V in Equation (6), $U_j$ can be solved via

$$U_j = (Y_j - X_jB_j)V. \tag{8}$$

The proof of Proposition 2 is shown in the Appendix.

**Proposition 3**: Given $U_j$ and V in Equation (6), $B_j$ can be solved via the proximal gradient descent as follows:

$$B = P\ \mathrm{diag}(\hat{\sigma}_1 \ldots \hat{\sigma}_m)Q', \tag{9}$$

where $\hat{\sigma}_i = \begin{cases} \sigma_i - \lambda & \sigma_i \geq \lambda_1 \\ 0 & -\lambda_1 \leq \sigma_i \leq \lambda_1, \\ \sigma_i + \lambda & \sigma_i \leq -\lambda_1 \end{cases}$ and $P, \sigma_i, Q$ can be obtained by SVD decomposition on $\tilde{B} = \begin{pmatrix} \tilde{\boldsymbol{\beta}}_1^{(1)} & \cdots & \tilde{\boldsymbol{\beta}}_m^{(1)} \\ \vdots & \ddots & \vdots \\ \tilde{\boldsymbol{\beta}}_1^{(d)} & \cdots & \tilde{\boldsymbol{\beta}}_m^{(d)} \end{pmatrix}$ as $\tilde{B} = Pdiag(\sigma_1 \ldots \sigma_m)Q'$ and

$$\tilde{\boldsymbol{\beta}}_j^{(r)} = \boldsymbol{\beta}_j^{(r)} - t_1\left(\nabla\left\|y_j^{(r)} - X_j\boldsymbol{\beta}_j^{(r)} - U_j\boldsymbol{v}^{(r)}\right\|^2 / \nabla\boldsymbol{\beta}_j^{(r)}\right).$$

The derivation of $\nabla\left\|y_j^{(r)} - X_j\boldsymbol{\beta}_j^{(r)} + U_j\boldsymbol{v}^{(r)}\right\|^2/\nabla\boldsymbol{\beta}_j^{(r)}$ is shown in the Appendix.

By decomposing the optimization problem into three sub-problems via the block coordinate descent approach, the iterative algorithm is summarized in Algorithm 1.

ALGORITHM 1
PSEUDOCODE FOR OPTIMIZATION ALGORITHM FOR SOLVING MTL-LVD

```
Initialize $\boldsymbol{\beta}'$:

  Initialize
    $B_j, U_j, V$ as $B^0, {U_j}^0, V^0$.
  end

  for $j = 1,2,3, \ldots$ do
    Update $V$ given $B$ and $U_j$ according to (6).
    Update $U_j$ given $B$ and $V$ according to (7).
    Update $B$ given $U_j$ and $V$ according to (8).
      If $\frac{\|L^k - L^{k-1}\|_F^2}{L^{k-1}} \leq \epsilon$, then
        Stop
      end
end
```

**Proposition 4:** The proposed algorithm for solving Equation (6) can converge to a stationary point.

The convergence of the block coordinate descent algorithm (BCD) can be proved by the monotonic decrease of Equation (6) since each BCD update would only decrease the objective function. Furthermore, the objective function in Equation (6) is lower bounded by zero. Therefore, the BCD algorithm must converge to a stationary point.

Finally, we would like to analyze the complexity of the proposed algorithm. Recall that there are $m$ similar-but-non-identical systems, the dimensionality of the response is $d$, the number of samples in each system are $n_j$, the number of observed variables is $p$, the number of latent variables is $q$. The computational complexity of updating all parameters in each iteration becomes $\mathrm{O}(qpnd + p^2nd + p^2md^2 + pm^2d)$. The derivation can be seen in the Appendix.

## IV. SIMULATION

To evaluate the performance of MTL-LVD, we performed a simulation study to evaluate two performance metrics as follows: 1) MTL-LVD is able to recover the simulated orthogonal loadings of the decomposed latent variation. 2) The prediction error of the multivariate response variable in root mean squared prediction errors (RMSPEs) on the testing data set $(\boldsymbol{y}_j^{(r)}, X_j)$: $\sum_{j=1}^{m}\sum_{r=1}^{d}\left\|\boldsymbol{y}_j^{(r)} - X_j\boldsymbol{\beta}_j^{(r)} - U_j\boldsymbol{v}^{(r)}\right\|_F^2$ for $m$ systems and $d$ responses.

The settings to generate the simulation data set are summarized in TABLE I. Assuming that there are eight similar-but-non-identical systems ($m = 8$). Within each system, there are eight observable variables in $\boldsymbol{x}_j = (x_{1,j}, \ldots, x_{p,j})'$ ($p = 10$), and three variables in the latent variation term $\boldsymbol{u}_j = (u_{1,j}, \ldots, u_{q,j})'$ ($q = 3$). Each system has 15 data points generated from $\boldsymbol{x}_j' \sim N(\boldsymbol{\mu}_X, \Sigma_X)$, and $\boldsymbol{u}_j' \sim N(\boldsymbol{\mu}_U, \Sigma_U)$, where $\boldsymbol{\mu}_X$, $\boldsymbol{\mu}_U \sim N(\boldsymbol{0}, I)$, and $\Sigma_X$, $\Sigma_U$ are covariance matrix with diagonal elements as 1 and off-diagonal elements as 0.1. Additionally, each system has $d$ multivariate responses as $d = 50$. The underlying model used to generate the response is

$$y_j^{(r)} = \boldsymbol{x}_j'\boldsymbol{\beta}_j^{(r)} + \boldsymbol{u}_j'\boldsymbol{v}^{(r)} + \varepsilon, \tag{10}$$

where $j = 1, \ldots, 8$, $r = 1, \ldots, 50$. The similarity of the system is simulated through the low-rank structure of the model parameter matrix $B = \begin{pmatrix} \boldsymbol{\beta}_1^{(1)} & \cdots & \boldsymbol{\beta}_m^{(1)} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\beta}_1^{(d)} & \cdots & \boldsymbol{\beta}_m^{(d)} \end{pmatrix}$ of Equation (6). More specifically, the parameters for the observed predictors of

TABLE I
SETTINGS OF SIMULATION DATA SETS

| Simulation Settings | Values |
|---|---|
| Number of Machines Simulated | 8 |
| Length of Response | 50 |
| Number of Samples for Each Machine | 15 |
| Number of Observed Predictors | 10 |
| Number of Latent Predictors | 3 |

TABLE II
AVERAGES AND STANDARD ERRORS OF RMSPEs (OR MEs) FROM 50 SIMULATION RUNS

| Method | Testing RMSE |
|---|---|
| LASSO-S | 1.670<br>(0.264) |
| LASSO-A | 0.334<br>(0.003) |
| MTL-N | 0.348<br>(0.004) |
| MTL-LVD | **0.299**<br>**(0.004)** |

machine $j$, $\boldsymbol{\beta}_j = \left({\boldsymbol{\beta}_j}^{(1)'} \quad \cdots \quad {\boldsymbol{\beta}_j}^{(d)'}\right)'$, corresponds to the column $j$ of $B = \begin{pmatrix} {\boldsymbol{\beta}_1}^{(1)} & \cdots & {\boldsymbol{\beta}_m}^{(1)} \\ \vdots & \ddots & \vdots \\ {\boldsymbol{\beta}_1}^{(d)} & \cdots & {\boldsymbol{\beta}_m}^{(d)} \end{pmatrix}$, which was simulated with a random linear combination of $n$ basis vectors out of $m$ basis vectors [28], where $n < m$. To simulate the basis matrix $V' = (\boldsymbol{v}^{(1)} \quad \cdots \quad \boldsymbol{v}^{(d)})$, where $\boldsymbol{v}^{(r)} = ({v^{(r)}}_1, \dots, {v^{(r)}}_q)'$, which was randomly generated by pseudo-spline basis implemented in GAMSEL package [29] with an input vector of $cosine(\frac{\boldsymbol{z}}{10})$ and $\boldsymbol{z} = (1, \dots, 50)'$. The error term $\varepsilon$ follows $N(0, \Sigma)$, and we set the signal-to-noise ratio $(\frac{variance\left(\boldsymbol{x}_j{}'{\boldsymbol{\beta}_j}^{(r)} + \boldsymbol{u}_j{}'\boldsymbol{v}^{(r)}\right)}{variance(\varepsilon)})$ as 10.

To evaluate the accuracy of the estimated MTL-LVD models, we compare the proposed method with three benchmark methods: a linear regression models with $l_1$-norm (LASSO) [9] for every single system (LASSO-S), a linear regression model with $l_1$-norm (LASSO) for all systems (LASSO-A), and MTL with nuclear norm (MTL-N). MTL-N has the loss function from Equation (2), which models the data combined from all systems but does not consider the latent variation, represented by $\boldsymbol{u}_j{}'\boldsymbol{v}^{(r)}$ in Equation (10).

For all methods in the simulation and the case study in the next section, 5-fold CV was employed to tune the model hyper-parameters for all the models based on training data, including the penalty coefficient on the nuclear-norm ($\lambda_1$) of MTL-N and MTL-LVD. Furthermore, the dimension of the latent variation term $V'$ for MTL-LVD is also tuned via the 5-fold CV. To ensure the reproducibility of the results, 50 replications were performed in the simulation study. Within each replication, the data are randomly partitioned in a 70-30 fashion for the training and testing datasets. In each replication, all models were trained based on the same training data set and tested for prediction accuracy in root mean squared errors (RMSEs) using the testing data set.

TABLE II reports the average of RMSEs with standard errors (S.E.) shown in parenthesis based on 50 simulation replications. We can see that MTL-LVD offers a better prediction performance than all three benchmark models. Specifically, Linear-S has a significantly higher prediction error due to the limited amount of sample size for model training, since only the data from a single system are used. In the meantime, Linear-A has a much smaller prediction error by aggregating the samples from all similar-but-non-identical systems. However, neither of these two models consider the heterogeneity among systems when the data are aggregated. Using the MTL framework with nuclear norm (MTL-N) to fit different sets of model parameters for different systems, the prediction error can be further reduced. However, MTL-N was not capable of capturing the unexplained latent variation, which was simulated by the $\boldsymbol{u}_j{}'\boldsymbol{v}^{(r)}$ term in data generator of Equation (10). As a result, the MTL-N was outperformed by the proposed method MTL-LVD, which achieves the lowest prediction error among all methods tested.

Another purpose of the simulation is to validate whether the MTL-LVD can discover the simulated latent variation, which can only be explained by the latent variation term. Specifically, we would like to validate if MTL-LVD recovers the true basis structure simulated in the latent variation term ($\boldsymbol{v}^{(r)}$ in Equation (5)), which has three simulated basis vectors aggregated as $V = (\boldsymbol{v}^{(1)} \quad \boldsymbol{v}^{(2)} \quad \boldsymbol{v}^{(3)})$. We generated Fig. 1 to compare the true values of three basis vectors in $V$ as solid lines and the recovered values by MTL-LVD as dotted lines in a simulaiton replication. We can observe that MTL-LVD recovers the all three basis vectors with minor discrepancies. The exact recovery of basis vectors is not always guaranteed since different sets of basis vectors can span the same latent space. As the result, the initialization of $V$ in Algorithm 1 impacts the final $V$ estimated. Although the basis recovered can be different each time, the improvement of prediction accuracy by introducing the latent variation decomposition is significant (TABLE II). In summary, the proposed MTL-LVD has the capability to achieve the best prediction accuracy while recovering the multi-dimensional latent variation.

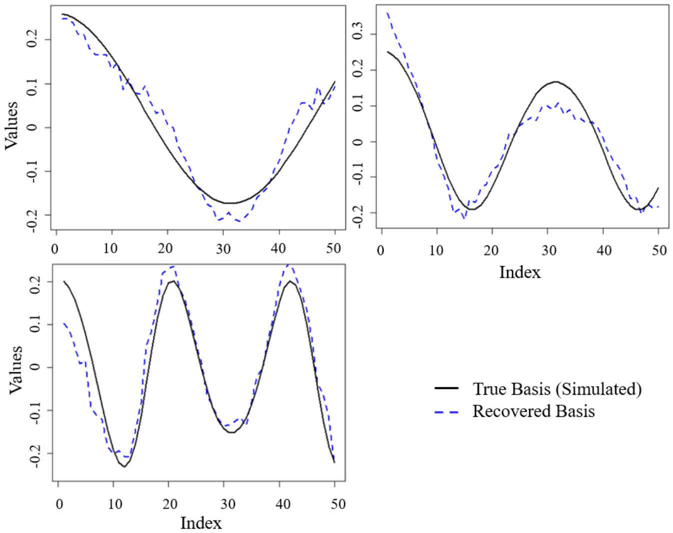


Fig. 2. The true values of the three basis vectors simulated (the solid line) and the correspondingly recovered basis vectors by MTL-LVD (the dotted line)

## V. Ingot Growth Manufacturing Case Study

The CZ ingot growth process produces monocrystalline silicon ingots is both costly and time-consuming [2]. There can be various types of defects occurred on the ingots during manufacturing, such as the growth of a polycrystalline ingot when the monocrystalline ingot is desired [1]. Therefore, it is important to model the relationship between the product quality variable and the process variables to understand how the process variables can impact the product quality. Furthermore, the numerical patterns recovered from the residual space represent the commonalities of systems for the CZ process, hence can contain important process information. In this case study, the response variable is the diameter of the continuously formed circular ingot surfaces, which are then moving-averaged into multivariate response vectors of length 50 for all ingot samples ($d = 50$ in Equation (4)). 10 process variables are generated as the summary statistics of the in situ process variables (${\boldsymbol{x}_j}' \in \mathbb{R}^{10}$ in Equation (4)), with the variable names omitted due to the confidential concerns. Among the five different manufacturing systems (furnaces), we had the data of 13, 19, 19, 21 and 25 ingots respectively. Each ingot is treated as a sample. TABLE IV summarizes the data setting of the crystal silicon growth process case study.

Similar to the simulation study, we compare the MTL-LVD with three representative benchmarks: LASSO-S, LASSO-A, and MTL-N. 50 replications were performed in the case study with the data from 5 furnaces randomly partitioned in a 70-30 fashion into the training and testing datasets within each replication with the same 5-fold CV to select tuning parameters as in the simulation studies.

Finally, we will still use RMSE to evaluate the prediction accuracy and the result of the prediction error is summarized in TABLE IV. The proposed MTL-LVD has a significantly better performance on predicting the diameter of the silicon ingot surface than the three benchmark models did over replications. LASSO-S has the largest prediction error mainly because separately modeling each system will suffer from the effect of a small training data set. LASSO-A has a much smaller prediction error comparing with LASSO-S, since LASSO-A aggregates the samples from five similar-but-non-identical furnaces to increase the training sample size of the model. However, neither of these two models consider the heterogeneities among different manufacturing systems. Using the MTL framework with nuclear norm (MTL-N), we can fit similar-but-non-identical models for multiple systems (furnaces) in the network. As a result, the prediction error was greatly reduced. Similar to the result of the simulation study, MTL-N could not capture the unexplained variation by the observed variables. As a result, introducing the latent variation term in MTL-LVD significantly reduced the prediction error on the testing data.

TABLE III
SETTINGS OF DATA FROM THE INGOT GROWTH MANUFACTURING

| Method | Testing RMSE |
|---|---|
| LASSO-S | 7.454<br>(1.478) |
| LASSO-A | 3.129<br>(0.143) |
| MTL-N | 1.075<br>(0.119) |
| MTL-LVD | **0.565**<br>**(0.008)** |

TABLE IV
SETTINGS OF DATA FROM THE INGOT GROWTH MANUFACTURING

| Settings | Values |
|---|---|
| Length of Response | 50 |
| Number of Observed Predictors | 10 |
| Sample Size for Furnace 1 | 13 |
| Sample Size for Furnace 2 | 19 |
| Sample Size for Furnace 3 | 19 |
| Sample Size for Furnace 4 | 21 |
| Sample Size for Furnace 5 | 25 |

Furthermore, similar to the simulation study, to demonstrate the usefulness of the latent variation term in MTL-LVD, we show the relationship between the parameters of the latent variation term ($\boldsymbol{v}^{(r)}$ in Equation (5)) and variation of the multivariate response (${y_j}^{(r)}$ in Equation (5)) in Fig. 3. Specifically, $\boldsymbol{v}^{(r)}$ is determined to be one dimensional by 5-fold CV and becomes a scalar value for each response, so that in total, we have $V' = (v^{(1)}, \dots, v^{(50)})$ for 50 responses. We expect that $v^{(r)}$should become a larger non-zero value as the variance of ${y_j}^{(r)}$increases to help to capture the unexplained variance by the observable variables.

In Fig. 3, the variances of the first few responses are large, which shows that the variations introduced by different furnaces are large at the initial manufacturing stages (i.e., for the first a few hours of the process). The variation gradually decreased after the initial stages and stayed at a low level towards the end of the process. Correspondingly, we also observed that the parameters of the latent variation term were large at the initial stages and decreased afterwards. The pattern of the parameter values ensembles the patterns of response variances. The instability at the beginning stages of ingot formation in the CZ process is perhaps due to an unstable polysilicon melting, which can lead to variations in oxygen concentration at the early stages of different ingot formation

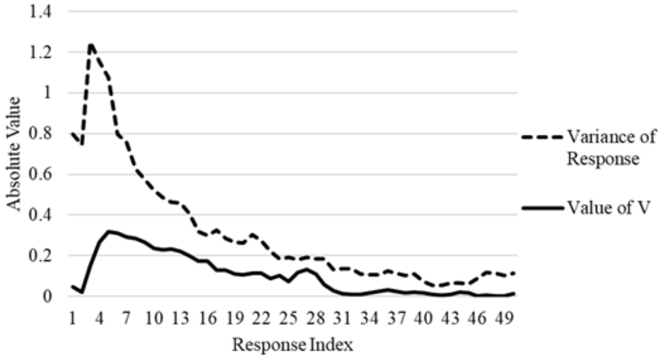


Fig. 3. The variance of responses data across five manufacturing systems for 50 responses (index 1-50) versus the value of V (the model parameters for latent variation term)

[30]. In summary, the latent variation term in MTL-LVD can help the observed variables to capture the variation introduced during the data aggregation of multiple similar-but-non-identical manufacturing systems. As a result, the prediction performance of MTL-LVD can be significantly improved.

## VI. Summary

Although machine learning methods require sufficient training data samples to generate a model with good prediction performance, such a sample size requirement can be hardly fulfilled in many manufacturing applications. In this research, we studied a motivating case of modeling the crystal silicon ingot manufacturing, which requires the furnace to be maintained at a temperature of approximately 1500 degrees Celsius over 50 hours. Due to the high cost of the production, the in situ manufacturing data and silicon ingot quality data can only be collected from a few silicon ingots from a furnace at a time. In this work, we incorporated a variation decomposition approach to model the latent variation, which cannot be captured by the observable variables, into the parameter-based MTL. Furthermore, we generalized the proposed methodology into a multivariate response and MTL model, which is capable of modeling multivariate responses monitored across multiple similar-but-non-identical systems (e.g., furnaces). The simulation study and the case study have shown that MTL-LVD can not only recover the latent variation among data of similar-but-non-identical systems but also offer significantly better accuracy on predicting the multivariate response.

There are several future research directions on MTL-LVD that deserve further investigations. The second one is incorporating different regularization terms into MTL-LVD based on different engineering perceptions/assumptions of the underlying problems. For example, $l_1$-norm can be applied to replace the nuclear norm in the MTL-LVD to perform variable selection and filter out the sensor signals, which are not strongly correlated with the responses [9]. The third research direction is how to incorporate the estimated $U$ in prediction. In this work, we adopted a straightforward approach by averaging $U$ in prediction, however, other approaches might further improve the performance. Another research direction can be using the model parameters of the latent variation term, which reflects the commonality of machines, to perform process monitoring. Such a learned commonality can also be continuously updated and become more accurate with incoming samples.

## Appendix

### A. Derivation of Proposition 2

Given $B$ and V, $U_j$ in re-written Equation (2)

$$L = \sum_{j=1}^{m} \parallel Y_j - X_j B_j - U_j V' \parallel^2 + \lambda_1 ||B||_*,$$

can be optimized at step $k$ via

$$U_j{}^k = \left(Y_j - X_j B_j{}^{k-1}\right) V^k.$$

Derivation:

For the ease of representation, let's denote $(Y_j - X_j B_j)$ above as $Y$ and $V'$ above as $B$. The proof of Proposition 2 is equivalent to solving matrix $U_j$ in the following matrix least square formulation:

$$\underset{A}{\operatorname{argmin}} \parallel Y - U_j B \parallel^2$$

where,

$$\begin{aligned} &\parallel Y - U_j B \parallel^2 \\ &= tr((Y - U_j B)^T (Y - U_j B)) \\ &= tr\left(Y^T Y + B^T U_j{}^T U_j B - Y^T U_j B - B^T U_j{}' Y\right) \\ &= tr(Y^T Y) + tr\left(B^T U_j{}^T U_j B\right) - 2tr\left(Y^T U_j B\right) \end{aligned}$$

By taking the first derivative of the above formulation and set it equal to zero, we have:

$$\begin{gathered} \nabla_{U_j} tr\left(U_j{}' U_j B B'\right) - 2\nabla_{U_j} tr\left(Y' U_j B\right) = 0, \\ 2U_j B B' - 2YB' = 0, \\ -2U_j B B^T = -2YB'. \end{gathered}$$

Therefore, the solution is

$$U_j = (YB^T)(BB^T)^{-1} = ((Y_j - X_j B_j)V)(V'V)^{-1},$$

and recall that $V'V = I$, then we have,

$$U_j = \left(Y_j - X_j B_j\right) V.$$

### B. Derivation of the gradient in Proposition 3

$$\begin{aligned} &\frac{\nabla\left(||\boldsymbol{y}_j{}^{(r)} - X_j \boldsymbol{\beta}_j{}^{(r)} - U_j{}^k \boldsymbol{v}^{(r)}||_F^2\right)}{\nabla \boldsymbol{\beta}_j{}^{(r)}} \\ &= \frac{\left(\boldsymbol{y}_j{}^{(r)} - X^j \boldsymbol{\beta}_j{}^{(r)} - U_j{}^k \boldsymbol{v}^{(r)}\right)' (\boldsymbol{y}_j{}^{(r)} - X^j \boldsymbol{\beta}_j{}^{(r)} - U_j \boldsymbol{v}^{(r)})}{\nabla \boldsymbol{\beta}_j{}^{(r)}} \\ &= -\left(\boldsymbol{y}_j{}^{(r)\prime} X_j\right)' - X_j{}' \boldsymbol{y}_j{}^{(r)} + 2X_j{}' X_j \boldsymbol{\beta}_j{}^{(r)} + X_j{}' U_j \boldsymbol{v}^{(r)} \\ &+ X_j{}' U_j \boldsymbol{v}^{(r)} = -2X_j{}' \boldsymbol{y}_j{}^{(r)} + 2X_j{}' X_j \boldsymbol{\beta}_j{}^{(r)} + 2X_j{}' U_j \boldsymbol{v}^{(r)} \end{aligned}$$

### C. The Computational Complexity of the Model Updating Algorithm

The model updating algorithm can be decomposed into three steps. At iteration $k$, we have:

11: Optimize $V$ given $B$ and $U_j$:

$$\begin{pmatrix} Y_1 - X_1 B_1{}^{k-1} \\ \vdots \\ Y_m - X_m B_m{}^{k-1} \end{pmatrix}' \begin{pmatrix} U_1{}^{k-1} \\ \vdots \\ U_m{}^{k-1} \end{pmatrix} = RDW',$$

$$V^k = RW'.$$

12: Optimize $U_j$ for given $B$ and $V$:

$$U_j{}^k = \left(Y_j - X_j B_j{}^{k-1}\right) V^k$$

13: Optimize $B$ given $U_j$ and $V$:

$$B^k = Pdiag(\hat{\sigma}_1 \ldots \hat{\sigma}_m) Q',$$

Recall that there are $m$ similar-but-non-identical systems, the dimensionality of the response is $d$ ($Y_j \in \mathbb{R}^{n_j \times d}$), the number of samples in each system are $n_j$, the number of observed variables is $p$ ($X_j \in \mathbb{R}^{n_j \times p}$), the number of latent variables is $q$ ($U_j \in \mathbb{R}^{n_j \times q}$). Furthermore, $B_j = \left(\boldsymbol{\beta}_j{}^{(1)} \cdots \boldsymbol{\beta}_j{}^{(d)}\right) \in \mathbb{R}^{p \times d}$, where $j = 1, \ldots, m$, $B = \begin{pmatrix} \boldsymbol{\beta}_1{}^{(1)} & \cdots & \boldsymbol{\beta}_m{}^{(1)} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\beta}_1{}^{(d)} & \cdots & \boldsymbol{\beta}_m{}^{(d)} \end{pmatrix} \in \mathbb{R}^{p \times d}$, $V \in \mathbb{R}^{d \times q}$and we assume that $q < d$.

Since $Y_j - X_j B_j$ has a complexity of O($n_j pd + n_j d$)=O(($p + 1)n_j d$), $\begin{pmatrix} Y_1 - X_1 B_1{}^{k-1} \\ \vdots \\ Y_m - X_m B_m{}^{k-1} \end{pmatrix}$ has a complexity of O(($p + 1)nd$), where $n = n_1 +, \ldots, +n_m$. $\begin{pmatrix} Y_1 - X_1 {B_1}^{k-1} \\ \vdots \\ Y_m - X_m {B_m}^{k-1} \end{pmatrix}' \begin{pmatrix} {U_1}^{k-1} \\ \vdots \\ {U_m}^{k-1} \end{pmatrix}$ has a complexity of O($(p+1)nd + ndq$)= O($(p+q+1)nd$). Generating $RDW'$ in Step 11 has a complexity of O($dq^2$) [31], and $V^k = RW'$ has a complexity of O($dq^2$). Collectively, Step 11 has a complexity of O($(p+q+1)nd + 2dq^2$)

Step 12 has a complexity of O($(p+1)n_j d + n_j dq$)= O($(p+q+1)n_j d$) for $U_j$, where $j = 1, \ldots, m$. To update all $U_j$, the complexity becomes O($(p+q+1)nd$).

In Step 13, we first use gradient to update un-thresholded $B_j$ as $\tilde{B} = \begin{pmatrix} \tilde{\boldsymbol{\beta}}_1^{(1)} & \cdots & \tilde{\boldsymbol{\beta}}_m^{(1)} \\ \vdots & \ddots & \vdots \\ \tilde{\boldsymbol{\beta}}_1^{(d)} & \cdots & \tilde{\boldsymbol{\beta}}_m^{(d)} \end{pmatrix} \in \mathbb{R}^{pd \times m}$, where $\tilde{\boldsymbol{\beta}}_j^{(r)} = \boldsymbol{\beta}_j^{(r)} - t_1 \left( \nabla \left\| \boldsymbol{y}_j^{(r)} - X_j \boldsymbol{\beta}_j^{(r)} - U_j \boldsymbol{v}^{(r)} \right\|_F^2 / \nabla \boldsymbol{\beta}_j^{(r)} \right)$.

The computation of the gradient

$$\nabla \left\| \boldsymbol{y}_j^{(r)} - X_j \boldsymbol{\beta}_j^{(r)} - U_j \boldsymbol{v}^{(r)} \right\|_F^2 / \nabla \boldsymbol{\beta}_j^{(r)} = -2X_j' \boldsymbol{y}_j^{(r)} + 2X_j' X_j \boldsymbol{\beta}_j^{(r)} + 2X_j' U_j \boldsymbol{v}^{(r)}$$

has a complexity O($n_j p + p + n_j p^2 + p^2 + p + n_j pq + pq + p$)= O($(n_j + 3 + n_j q + q)p + (n_j + 1)p^2$). Then, obtaining $\tilde{\boldsymbol{\beta}}_j^{(r)}$ has the complexity of O($\left(n_j + 5 + n_j q + q\right)p + \left(n_j + 1\right)p^2$) and obtaining $\tilde{B}$ has the complexity of O($(n + 5m + nq + qm)dp + (n+m)dp^2$).

We assume that $pd < m$ and performing $\tilde{B} = P diag(\sigma_1 \ldots \sigma_m) Q'$ has a complexity of O($pdm^2$). Performing thresholding and reconstruct $B$ as $B = P diag(\hat{\sigma}_1 \ldots \hat{\sigma}_m) Q'$ has a complexity of O($(3 + p^2 d^2)m + pdm^2$). Collectively, Step 13 has a time complexity as O($(n + 5m + nq + qm)dp + (n+m)dp^2 + pdm^2 + (3 + p^2 d^2)m + pdm^2$).

The overall computational complexity becomes O($2q^2 d + 2pnd + 2qnd + 2nd + qnd + 5pmd + qpnd + qpmd + p^2 nd + p^2 md + pm^2 d + 3m + p^2 md^2 + pm^2 d$).

$n$: the sample size

$p$: the number of observed variables

$q$: the number of latent variables

$d$: the dimensionality of the response

$m$: the number of similar-but-non-identical systems,

It is safe to assume $q < d$, $m < n$, We can simplify the complexity as O($qpnd + p^2 nd + p^2 md^2 + pm^2 d$).

## References

[1] H. Sun, X. Deng, K. Wang, and R. Jin, "Logistic regression for crystal growth process modeling through hierarchical nonnegative garrote-based variable selection," *IIE Transactions,* vol. 48, no. 8, pp. 787-796, 2016.

[2] G. Fisher, M. R. Seacrist, and R. W. Standley, "Silicon crystal growth and wafer technologies," *Proceedings of the IEEE,* vol. 100, no. 1, pp. 1454-1474, 2012.

[3] R. Jin, X. Deng, X. Chen, L. Zhu, and J. Zhang, "Dynamic quality-process model in consideration of equipment degradation," *Journal of Quality Technology,* vol. 51, no. 3, pp. 1-13, 2019.

[4] E. Zafiriou, H. Chiou, and R. Adomaitis, "Nonlinear model based run-to-run control for rapid thermal processing with unmeasured variable estimation," in *Symp. Control, Diagnosis and Modeling in Semiconductor Manufacturing, 187th Meeting Electrochem. Soc.*, 1995, vol. 95, p. 18.

[5] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering,* vol. 22, no. 10, pp. 1345-1359, 2009.

[6] A. Shintani, T. Miyano, and M. Hourai, "A novel apporach to the characterization of growth striations in Czochralski silicon crystals," *Journal of the Electrochemical Society,* vol. 142, no. 7, p. 2463, 1995.

[7] M. Avci and S. Yamacli, "Neural network reinforced point defect concentration estimation model for Czochralski-grown silicon crystals," *Mathematical and computer modelling,* vol. 51, no. 7-8, pp. 857-862, 2010.

[8] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics,* vol. 12, no. 1, pp. 55-67, 1970.

[9] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological),* vol. 58, no. 1, pp. 267-288, 1996.

[10] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology),* vol. 67, no. 2, pp. 301-320, 2005.

[11] R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *Journal of Machine Learning Research,* vol. 6, no. 1, pp. 1817-1853, 2005.

[12] J. Chen, L. Tang, J. Liu, and J. Ye, "A convex formulation for learning shared structures from multiple tasks," in *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, pp. 137-144: ACM.

[13] L. Jacob, P. Vert, and F. R. Bach, "Clustered multi-task learning: A convex formulation," in *Proceedings of 2009 Advances in Neural Information Processing Systems*, 2009, pp. 745-752.

[14] Z. Kang, K. Grauman, and F. Sha, "Learning with Whom to Share in Multi-task Feature Learning," in *Proceedings of the 28th International Conference on Machine Learning*, 2011, pp. 521-528.

[15] T. Evgeniou and M. Pontil, "Regularized multi-task learning," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004, pp. 109-117: ACM.

[16] S. Parameswaran and K. Q. Weinberger, "Large margin multi-task metric learning," in *Proceedings of 2010 Advances in Neural Information Processing Systems*, 2010, pp. 1867-1875.

[17] A. R. Gonçalves, P. Das, S. Chatterjee, V. Sivakumar, F. J. Von Zuben, and A. Banerjee, "Multi-task sparse structure learning," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, 2014, pp. 451-460.

[18] A. R. Goncalves, P. Das, S. Chatterjee, V. Sivakumar, F. J. V. Zuben, and A. Banerjee, "Multi-task Sparse Structure Learning," presented at the Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, Shanghai, China, 2014. Available: https://doi.org/10.1145/2661829.2662091

[19] I. Yamane, F. Yger, M. Berar, and M. Sugiyama, "Multitask principal component analysis," in *Asian Conference on Machine Learning*, 2016, pp. 302-317.

[20] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *International conference on artificial neural networks*, 1997, pp. 583-588: Springer.

[21] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *Journal of Computational and Graphical Statistics,* vol. 15, no. 2, pp. 265-286, 2006.

[22] B. W. Silverman, "Smoothed functional principal components analysis by choice of norm," *The Annals of Statistics,* vol. 24, no. 1, pp. 1-24, 1996.

[23] G. M. James, T. J. Hastie, and C. A. Sugar, "Principal component models for sparse functional data," *Biometrika,* vol. 87, no. 3, pp. 587-602, 2000.

[24] F. Yao, H. G. Müller, and J. L. Wang, "Functional data analysis for sparse longitudinal data," *Journal of the American Statistical Association,* vol. 100, no. 470, pp. 577-590, 2005.

[25] K. Paynabar, C. Zou, and P. Qiu, "A change-point approach for phase-I analysis in multivariate profile monitoring and diagnosis," *Technometrics,* vol. 58, no. 2, pp. 191-204, 2016.

[26] C. Zhang, H. Yan, S. Lee, and J. Shi, "Weakly correlated profile monitoring based on sparse multi-channel functional principal component analysis," *IISE Transactions,* vol. 50, no. 10, pp. 878-891, 2018.

[27] H. Zhou and L. Li, "Regularized matrix regression," *Journal of the Royal Statistical Society: Series B (Statistical Methodology),* vol. 76, no. 2, pp. 463-483, 2014.

[28] J. R. Rice, "Experiments on gram-schmidt orthogonalization," *Mathematics of Computation,* vol. 20, no. 94, pp. 325-328, 1966.

[29] A. Chouldechova and T. Hastie, "Generalized additive model selection," *arXiv preprint arXiv:1506.03850,* 2015.

[30] S. S. Pascoa, "Oxygen and related defects in Czochralski silicon crowns," MS Thesis, Department of Materials Science and Engineering, Norwegian University of Science and Technology, 2014.

[31] V. Vasudevan and M. Ramakrishna, "A hierarchical singular value decomposition algorithm for low rank matrices," *arXiv preprint arXiv:1710.02812,* 2017.

**Yifu Li** is an assistant professor of Industrial & Systems Engineering at the University of Oklahoma. He received his Ph.D. and B.S. degree in Industrial and Systems Engineering from Virginia Tech.

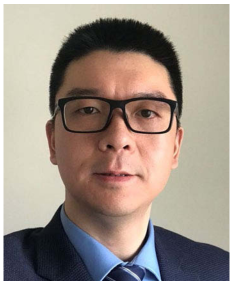
He previously worked as Research Assistant at Grado Department of Industrial and Systems Engineering of Virginia Tech and Feinberg School of Medicine of Northwestern University. His research interest is mainly on studying the interface between manufacturing data-driven modeling and data quality.

He is a member of IEEE, INFORMS and IISE. His awards and honors include Best Poster Presentation at Center for Excellence in Logistics and Distribution (CELDi), Fayetteville AR, 2017, and Young Author Award at International Workshop on Intelligentized Welding Manufacturing (IWIWM), Lexington KY, 2019

**Hao Yan** received the B.S. degree in Physics from the Peking University, Beijing, China, in 2011. He also received the M.S. degree in Statistics, the M.S. degree in Computational Science and Engineering, and the Ph.D. degree in Industrial Engineering from Georgia Institute of Technology, Atlanta, in 2015, 2016, 2017, respectively.

Currently, he is an assistant professor in the School of Computing, Informatics, and Decision Systems Engineering in Arizona State University. His research interests focus on developing scalable statistical learning algorithms for large-scale high-dimensional data with complex heterogeneous structures to extract useful information for the purpose of system performance assessment, anomaly detection, intelligent sampling and decision making.

He was also recipients of multiple awards including best paper award in IISE Transaction, IEEE TASE, and ASQ Brumbaugh Award. He is a member of IEEE, INFORMS and IISE.



**Dr. Ran Jin** is an Associate Professor and the Director of Laboratory of Data Science and Visualization at the Grado Department of Industrial and Systems Engineering at Virginia Tech. He received his Ph.D. degree in Industrial Engineering from Georgia Tech, Atlanta, his Master’s degrees in Industrial Engineering, and in Statistics, both from the University of Michigan, Ann Arbor, and his bachelor’s degree in Electronic Engineering from Tsinghua University, Beijing.

He has been working with leading manufacturing companies in aerospace, semiconductor, personal care, optical fiber industries. His research focuses on machine learning in manufacturing, manufacturing computation services and cognitive-based interactive visualization.

He is a member of IEEE and currently serving as an Associate Editor for IISE Transactions and an Associate Editor for ASME Transactions, Journal of Manufacturing Science and Engineering. For more information about Dr. Jin, please visit his faculty website at Virginia Tech: https://ise.vt.edu/ran-jin.