Process Monitoring and Control of Advanced Manufacturing based on Physics-Assisted Machine Learning

Jihoon Chung

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Industrial and Systems Engineering

Zhenyu (James) Kong, Chair
Blake Johnson
Haibo Zeng
Xiaowei Yue

May 9, 2023
Blacksburg, Virginia

Keywords: Quality Assurance, Reinforcement Learning, Sparse Bayesian Learning,
Generative Adversarial Network, Additive Manufacturing.

Process Monitoring and Control of Advanced Manufacturing based on Physics-Assisted Machine Learning

Jihoon Chung

(ABSTRACT)

With the advancement of equipment and the development of technology, the manufacturing process is becoming more and more advanced. This appears as an advanced manufacturing process that uses innovative technology, including robotics, artificial intelligence, and autonomous systems. Additive manufacturing (AM), also known as 3D printing, is the representative advanced manufacturing technology that creates 3D geometries in a layer-by-layer fashion with various types of materials. However, quality assurance in the manufacturing process requires high expectations as the process develops. Therefore, the objective of this dissertation is to propose innovative methodologies for process monitoring and control to achieve quality assurance in advanced manufacturing.

The development of sensor technologies and computational power offer process data, providing opportunities to achieve effective quality assurance through a machine learning approach. Hence, exploring the connections between sensor data and process quality using machine learning methodologies would be advantageous. Although this direction is promising, some constraints and complex process dynamics in the actual process hinder achieving quality assurance from the existing machine learning methods.

To address these challenges, several machine learning approaches assisted by the physics knowledge obtained from the process have been proposed in this dissertation. These approaches are successfully validated by various manufacturing processes, including AM and

multistage assembly processes. Specifically, three new methodologies are proposed and developed, as listed below.

1. To detect the process anomalies with imbalanced process data due to different ratios of occurrence between process states, a new Generative Adversarial Network (GAN)-based method is proposed. The proposed method jointly optimizes the GAN and classifier to augment realistic and state-distinguishable images to provide balanced data. Specifically, the method utilizes the knowledge and features of normal process data to generate effective abnormal process data. The benefits of the proposed approach have been confirmed in both polymer AM and metal AM processes.

2. To diagnose process faults with a limited number of sensors caused by the physical constraints in the multistage assembly process, a novel sparse Bayesian learning is proposed. The method is based on a practical assumption that it will likely have a few process faults (sparse). In addition, the temporal correlation of process faults and the prior knowledge of process faults are considered through the Bayesian framework. Based on the proposed method, process faults can be accurately identified with limited sensors.

3. To achieve online defect mitigation of new defects that occurred during the printing due to the complex process dynamics of the AM process, a novel Reinforcement Learning (RL)-based algorithm is proposed. The proposed method is to learn the machine parameter adjustment to mitigate the new defects during the printing. The method transfers knowledge learned from various sources in the AM process to RL. Therefore, with a theoretical guarantee, the proposed method learns the mitigation strategy with fewer training samples than traditional RL.

By overcoming the challenges in the process, the above-proposed methodologies successfully

achieve quality assurance in the advanced manufacturing process. Furthermore, the methods are not designed for the typical processes. Therefore, they can easily be applied to other domains, such as healthcare systems.

Process Monitoring and Control of Advanced Manufacturing based on Physics-Assisted Machine Learning

Jihoon Chung

(GENERAL AUDIENCE ABSTRACT)

The development of equipment and technologies has led to advanced manufacturing processes, including additive manufacturing processes. The advanced manufacturing processes greatly impact various industries, such as the automotive, aerospace, and medical industries. However, one of the major challenges is how to ensure product quality by detecting and mitigating defects. For example, the voids and cracks of the aircraft materials cause severe problems in the aerospace industry. Therefore, quality assurance in the manufacturing processes has become a very important issue in providing consistently high-quality products. Considering the importance of this problem, this dissertation aims to accomplish quality assurance by developing advanced machine-learning approaches.

In this dissertation, several advanced machine learning methodologies using the physics knowledge from the process are proposed. These methods overcome some constraints and complex process dynamics of the actual process that degrade the performance of existing machine learning methodologies in achieving quality assurance. To validate the effectiveness of the proposed methodologies, various advanced manufacturing processes, including additive manufacturing and multistage assembly processes, are utilized. The performance of the proposed methodologies provides superior results for achieving quality assurance in various scenarios compared to existing state-of-the-art machine learning methods.

The applications of the achievements in this dissertation are not limited to the manufacturing process. Therefore, the proposed machine learning approaches can be further extended to other application areas, such as healthcare systems.

Dedication

I express my gratitude to all my family members for the unwavering love and support they have shown me throughout the years.

Acknowledgments

I express my sincere gratitude to my advisor, Dr. Zhenyu (James) Kong, for his guidance and supervision throughout my dissertation work. I feel incredibly fortunate to have had such a great professor as my advisor, and I have learned so much from him. Without his help and support, I would not have been able to produce a high-quality dissertation. In addition, I also thank my wonderful Ph.D. advisory committee members, Dr. Blake Johnson, Dr. Haibo Zeng, and Dr. Xiaowei Yue, for their valuable advice and constant support of my research. Moreover, I received encouragement and assistance from many other individuals during my graduate study. In particular, I am grateful to my lab members, Dr. Chenang Liu, Dr. Bo Shen, Dr. Andrew Chung Chee Law, Mr. Rongxuan Wang, Mrs. Maede Maftouni, Mr. Raghav Gnanasambandam, Mr. Benjamin N. Standfield, and Mr. Chaoran Dou, for their unconditional support, help, and selfless friendship. Furthermore, I want to appreciate the support of all my family members throughout these years, despite being physically distant from them most of the time. Lastly, I would like to express my appreciation to everyone in the Grado Department of Industrial and Systems Engineering for their support during my Ph.D. studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Recently, with the development of equipment and technology, the manufacturing process is becoming more and more advanced. This appears as an advanced manufacturing process, including additive manufacturing (AM), providing consumers with many benefits and convenience. Despite the significant development achieved in the advanced manufacturing processes, producing high-quality products from the processes is a challenging task. Therefore, it is necessary to develop innovative approaches for process monitoring and control to achieve quality assurance in the advanced manufacturing process.

The development of sensor technologies offers rich data that enables data-driven approaches, including machine learning-based methods to deal with quality assurance in the manufacturing process. Utilizing the rich information from sensor data, the existing work provides effective process monitoring and control by identifying the relationship between the sensor data and process quality. Despite the capabilities of this direction, some constraints and complex process dynamics in the process hinder achieving quality assurance through the existing machine learning approaches. For example, the low frequency of occurrence of abnormal states (with defects) compared to normal ones (without defects) in the process results in imbalanced data, which may limit the information required for machine learning methods to achieve quality assurance. In addition, the actual processes might have some physical

constraints that prevent the attachment of enough sensors, which reduces the process monitoring & diagnosis accuracy. Furthermore, the complex process dynamics of AM process provide new defects during the printing that cannot be mitigated by the existing methods for process control.

Hence, this dissertation aims to create advanced machine learning-based methodologies for process monitoring and control. Specifically, the methods effectively utilize the physics knowledge from the process to overcome the above challenges. Three different advanced manufacturing processes, namely, the actual autobody assembly process, Fused Filament Fabrication, and Electron Beam Melting processes, are utilized to validate the proposed methodologies in this dissertation.

1.2 Research Objectives

As mentioned in the previous section, quality assurance is a critical problem in advanced manufacturing processes. Although the sensor data provides rich information for the machine learning approach for process monitoring and control, some constraints and complexity of the process deteriorate the performance of existing machine learning methods to achieve quality assurance. Thus, the ultimate objective of this dissertation is to propose advanced machine learning approaches assisted by physics knowledge from the process to overcome these challenges. Specifically, the goal consists of three research objectives as follows:

- Imbalanced data occurs in process sensing: how to overcome the imbalanced training data for the supervised classification method to detect process anomalies?
- Insufficient number of sensors for process monitoring & diagnosis: how to effectively diagnose the process faults from the limited information?

- Mitigating new defects for process control: how to learn the machine parameter adjustment to mitigate the new defects that occurred during the printing?

Through accomplishing the above three objectives, the contributions of these studies are summarized as the following three aspects.

1. First, a Generative Adversarial Network-based data augmentation method is proposed. The method augments the data to address the imbalanced data that occurs in the process sensing stage. Specifically, the method uses the knowledge and features from the normal process state to effectively generate the abnormal process state. The generated data is used for the supervised classification method to detect process anomalies.

2. Second, sparse Bayesian learning for the accurate process faults diagnosis in the multistage assembly process is developed. The method effectively utilizes the prior knowledge and the correlation structure of the process faults in the process. Therefore, the method accurately diagnoses the process faults from the limited number of sensors.

3. Finally, a Reinforcement Learning-based method is proposed. The proposed method is to learn the machine parameter adjustment to mitigate the new defects that occurred during the printing in process control. By considering the prior knowledge of defect mitigation from various sources of the process, the method learns to mitigate the new defect with a few parameter adjustments.

All three research are validated from the actual manufacturing processes. In practice, the methods are not limited to the manufacturing processes but to the various domains, such as healthcare systems.

1.3 Dissertation Organization

The rest of this dissertation consists of several chapters, as described as follows. Chapter 2 provides the overview of the proposed research and related studies. Based on the related studies, the research gap analysis is described. In Chapter 3, a new Generative Adversarial Network-based data augmentation model is provided. The model augments the process anomalies in AM process to address the imbalanced data that occurs in the process sensing stage. In Chapter 4, a novel sparse Bayesian hierarchical model is proposed. The method is proposed to diagnose process faults in the multistage assembly process when there exists a small number of sensor measurements. In Chapter 5, a novel Reinforcement Learning-based algorithm is proposed to learn the decision rule to mitigate the new defect that occurred during printing. Finally, Chapter 6 summarizes this dissertation and explores the possible directions for future research.

Chapter 2

Review of Literature

2.1 Research Overview

As mentioned in Section 1.1, the research goal of this dissertation is to develop innovative machine learning methods assisted by physics knowledge for quality assurance in advanced manufacturing processes. Figure 2.1 shows the overview of the proposed research framework to achieve the goal. The framework consists of sensing, monitoring & diagnosis, and control stages of the process sequentially. First, a new data augmentation method is developed

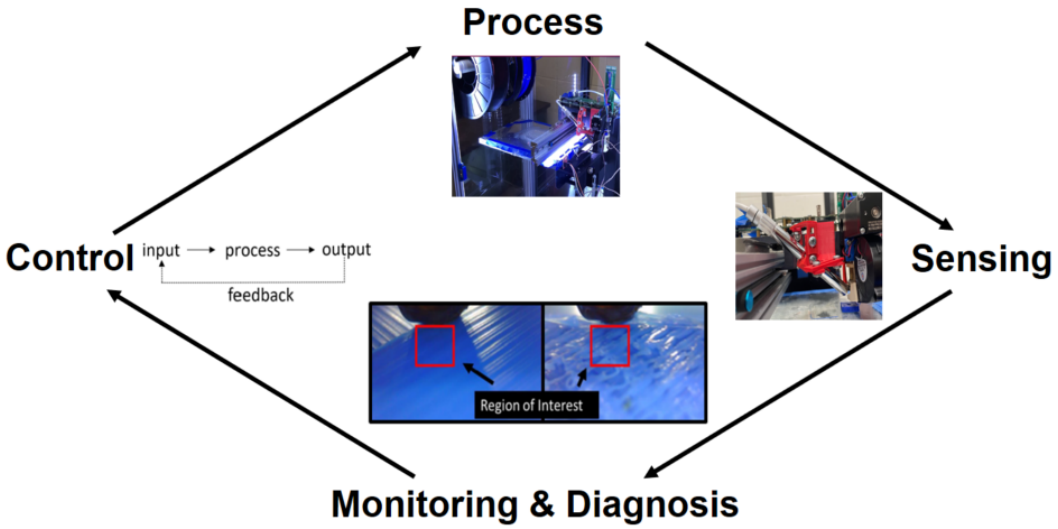


Figure 2.1: The overview of the proposed research framework.

to address the imbalanced process data issue that occurred in the process sensing stage.

Subsequently, a novel process diagnosis method is proposed to detect the process fault when a limited number of sensors exist in the process monitoring & diagnosis stage. Finally, the new control method is developed to learn the mitigation strategies for new defects that occurred during the process. According to this framework, the overall study of this dissertation consists of the following three tasks, which can be summarized as follows.

1. **Task 1**: To detect the process anomalies with imbalanced process data due to different ratios of occurrence between process states, a new Generative Adversarial Network-based method is proposed (Chapter 3).

2. **Task 2**: To diagnose process faults with a limited number of sensors caused by the physical constraints in the multistage assembly process, a novel sparse Bayesian learning is proposed (Chapter 4).

3. **Task 3**: To achieve online defect mitigation of new defects that occurred during the printing due to the complexity of the AM process, a novel Reinforcement Learning-based algorithm is proposed (Chapter 5).

By accomplishing the above three tasks, this dissertation provides innovative process monitoring and control methodologies to achieve quality assurance in advanced manufacturing processes.

2.2 Literature Review

This dissertation is focused on developing advanced machine learning approaches assisted by physics knowledge from the process to achieve quality assurance for advanced manufacturing. The existing studies are reviewed in this section. Section 2.2.1 provides the existing anomaly detection approaches for advanced manufacturing processes. Then, Section 2.2.2 present the existing work of fault diagnosis in the multistage assembly process. Subsequently, Section 2.2.3 describes the existing studies of defect mitigation in the Fused Filament Fabrication (FFF) process, which is the representative polymer AM process.

2.2.1 Anomaly Detection Approaches for the Advanced Manufacturing Processes

Based on the developments of sensor technologies and computational power, machine learning approaches, including deep learning methods, are widely utilized in anomaly detection in manufacturing processes to achieve quality assurance. Specifically, deep learning methods provide superior performance in anomaly detection by using the capability of the deep network structure. For example, [73] proposed using convolutional neural networks (CNNs) to classify mixed-type defect patterns in wafer bin maps (WBM) in the semiconductor manufacturing process. [97] developed a voting-based deep ensemble feature framework for efficiently classifying defective WBMs. [72] used deep neural networks to accurately classify melt pool images in the metal AM process concerning different laser powers, resulting in different porosity levels. [139] converted the vibration signal from the drive end bearing to the images. The images were used as training data for CNN for bearing fault diagnosis. In addition, [60] applied CNN to extract the features from the infrared thermal images for the

fault diagnosis of rotating machinery in the manufacturing process. [113] presented a novel CNN architecture that pixel-wise localization of layer-wise surface data from the powder bed metal AM process for anomaly detection.

In addition to the deep learning-based methods, various machine learning-based methods are utilized in anomaly detection in the manufacturing processes. For example, [98] proposed a graph-theoretic approach to differentiate the distinctive thermal signatures of melt pool images, leading to poor abnormal surface finish from the metal AM process. [89] also developed a novel anomaly detection framework in the metal AM process. The approach classified the process state by accounting for the spatial dependence among successive melt pools through the Gaussian process. For the polymer AM process, [84] proposed a feature extraction method based on manifold learning to diagnose surface defects such as under-fill. [114] also proposed a novel supervised feature extraction method to extract discriminant and informative features from the surface states in the FFF process. [107] developed a Bayesian nonparametric method for online quality monitoring for the surface defects in the FFF process using heterogeneous sensors. In addition to the several AM processes, [11] proposed a sparse estimation-based classification approach for process monitoring in the semiconductor chemical mechanical planarization process. [14] presented a classification method to diagnose the cutting tool condition in the manufacturing process by analyzing the machined surface. Specifically, a kernel-based support vector machine classifier was trained with the features extracted from the gray-level co-occurrence matrix of machined surface images. [129] proposed weight masks to extract rotation-invariant features for wafer map failure pattern detection in the semiconductor manufacturing process. Furthermore, [134] extracted geometry-based features from wafer maps and used support vector machines (SVM) to classify wafer map defects. [138] proposed the conditional two-dimensional principal component analysis (PCA) algorithm to extract features from wafer maps for accurate classification.

2.2.2 Fault Diagnosis Methodologies in Multistage Assembly Systems

Numerous machine learning and data-driven methods have focused on fault diagnosis methodologies for multistage assembly systems, utilizing the fault-quality model outlined in Eq. (4.1). [69] developed a PCA-based orthogonal diagonalization strategy to transform the measurement data. It enabled the estimation of the variance of process errors in a multistage assembly system. [144] utilized a mixed linear model to detect the process errors with both mean shift and variance change by utilizing a maximum likelihood estimator. [20] presented a fault diagnosis method by using the principal component analysis (PCA) and pattern recognition. [35] presented a fault diagnosis method in the multistage assembly systems integrating the state space model of the process and matrix perturbation theory. [126, 127] proposed a fault diagnosis method in the machining process, considering the process physics regarding how fixtures generate the patterns. Using the method, root cause identification was conducted sequentially. All these approaches assume the number of measurements is greater than that of the process errors, which may not be consistent with the industrial practice. However, if this assumption is violated, all of the approaches mentioned above are ineffective because the fault-quality linear model becomes an underdetermined system resulting in the non-existence of a unique solution. To overcome an underdetermined system in the multistage assembly system, sparse learning has been actively used recently. Specifically, sparse Bayesian learning has been actively utilized to incorporate the sparsity of process faults as the prior distribution. [9] proposed a fault diagnosis methodology by enhancing the relevant vector machine to detect process errors with the variance change. [12] proposed a spatially correlated sparse Bayesian learning to deal with the case when process errors have a spatial correlation. The premise of this study is that if one of the fixture locators varies

away from its intended design specifications, it is likely that the adjacent locators will also exhibit deviations. [75] presented a Bayesian approach for identifying variation sources in a multistage manufacturing process using the sparse variance component prior. The work focuses on the identification of process faults that have variance increases.

2.2.3 Defect Mitigation in the Fused Filament Fabrication Process

Fused Filament Fabrication (FFF) processes are vulnerable to surface defects because the thermoplastic properties of filaments that determine the ability to create a bond between layers and solidify the extruded filament are sensitive to the environment [71]. There were several proposed solutions using post-processing, DOE, mathematical optimization methods, and closed-loop control to deal with surface defects. [44] proved that significant improvements on the surface finish of acrylonitrile butadiene styrene (ABS) parts could be achieved using the chemical post-processing treatment. [74] used an acetone vapor bath for post-process smoothing to reduce the surface roughness and reach a maximum 95% reduction in surface roughness. Using the DOE method, [3] determined the effect of layer thickness and deposition speed on the surface roughness of the FFF process. [55] used a factorial design to improve the surface roughness of ABS 400 polymer materials in the FFF process. [108] studied the effect of process variables on surface texture parameters to predict surface roughness by the Taguchi method. In addition, there were several research efforts to deal with quality issues in the FFF process by optimization of mathematical models. [17] applied a genetic algorithm to determine the optimum part deposition orientation to improve the surface quality by measuring the arithmetic mean of surface roughness. [77] used a particle swarm optimization algorithm to obtain a target surface quality. Recently, [83] developed a closed-loop controller for the FFF process. It consists of the image-based process monitoring [114] and PID controller for defect mitigation.

2.3 Research Gaps Analysis

The related research work introduced in Section 2.2.1 summarizes the research efforts for the anomaly detection methods in the process monitoring stage. However, most of the existing work assumes balanced training data among the process states can be collected in the sensing stage, which is an unrealistic assumption in the actual manufacturing process. Section 2.2.2 is focused on the fault diagnosis method in the multistage assembly process. Most of the existing work assumes plenty of sensors in the process which may not be consistent with industrial practice. Recently several work have been published to address the issue by developing sparse Bayesian learning. However, the existing studies did not take into account the temporal correlation of sensor measurements and prior knowledge of process faults, so performance is limited when there are few sensors. The work summarized in Section 2.2.3 contributes to the defect mitigation for online process control in the representative polymer AM process. But they assumed the defects that occurred in the process had been identified in advance. Therefore, existing methods are not suitable for handling new defects during printing. In light of these gaps, the research methodologies proposed in this dissertation aim to overcome these challenges by creating novel and advanced machine learning methodologies assisted by physics knowledge from the process.

Chapter 3

Anomaly Detection in Additive Manufacturing Processes using Supervised Classification with Imbalanced Data based on Generative Adversarial Network

Supervised classification methods have been widely utilized for the quality assurance of the advanced manufacturing process, such as additive manufacturing (AM) for anomaly (defects) detection. However, since abnormal states (with defects) occur much less frequently than normal ones (without defects) in a manufacturing process, the number of sensor data samples collected from a normal state is usually much more than that from an abnormal state. This issue causes imbalanced training data for classification analysis, thus deteriorating the performance of detecting abnormal states in the process. It is beneficial to generate effective artificial sample data for the abnormal states to make a more balanced training set. To achieve this goal, this paper proposes a novel data augmentation method based on a generative adversarial network (GAN) using additive manufacturing process image sensor data. The novelty of our approach is that a standard GAN and classifier are jointly

optimized with techniques to stabilize the learning process of standard GAN. The diverse and high-quality generated samples provide balanced training data to the classifier. The iterative optimization between GAN and classifier provides the high-performance classifier. The effectiveness of the proposed method is validated by both open-source data and real-world case studies in polymer and metal AM processes.

3.1 Introduction

The utilization of advanced manufacturing techniques such as additive manufacturing (AM) is prevalent in diverse fields such as aerospace, healthcare, and the automotive industry [83]. However, a major barrier preventing broader industrial adoption of the processes is the quality assurance of products. For example, surface defects exist, such as under-fill from the Fused Filament Fabrication (FFF) process shown in Figure 3.1. It is due to highly complex interactions in consecutive layers during printing. The defect causes a deficiency in the mechanical properties of the final product, such as density, tensile strength, and compressive strength [51]. Therefore, timely detection of the anomaly in the process is necessary so that corrective actions can be taken to mitigate the defects and improve the quality of products [90].

In recent times, advancements in sensor technologies and computational capabilities have enabled the acquisition of online process data, creating opportunities for efficient quality assurance through a data-driven approach [84]. The sensor data, typically in high volume regarding dimensionality and sampling frequency, contains valuable insights into the manufacturing processes [143]. Hence, exploring the inherent connections between the sensor data and process quality state using a data-driven approach is highly advantageous [81]. Utilizing the sensor data, supervised classification methods have been widely applied to online

anomaly detection in the manufacturing process [8, 49, 63]. This is because these methods can fully utilize the label information of the process state, resulting in more accurate and reliable detection results than unsupervised methods. [8, 49, 63] utilized balanced training data collected from both normal and abnormal states in the manufacturing processes to achieve a high anomaly detection rate from classifiers [25].

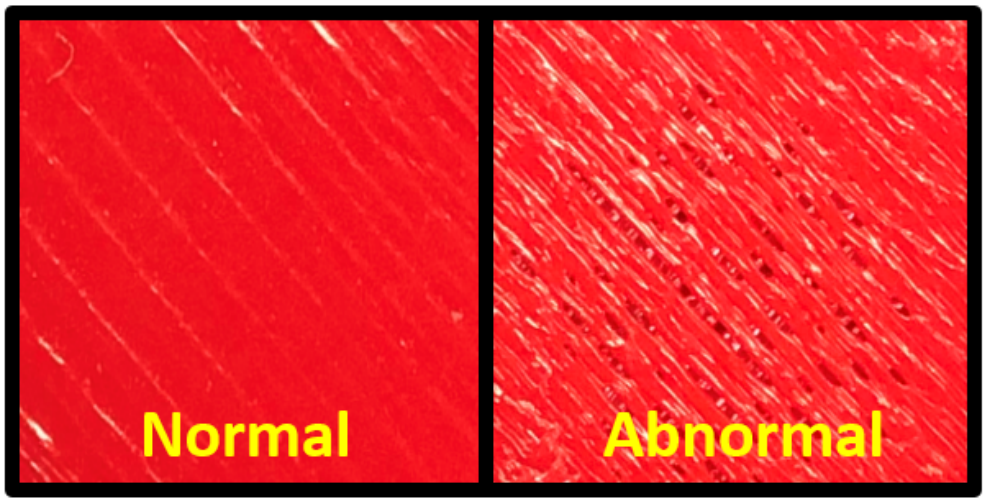


Figure 3.1: Normal surface and Abnormal defect in AM process.

However, the manufacturing process is usually in a normal state [81]. Therefore, balanced training data assumed by the existing work [8, 63] is expensive and not realistic. In reality, abnormal conditions, such as surface defects in AM process (Figure 3.1), may happen but rarely. Consequently, the sensor data collected under abnormal states are smaller than those collected in a normal state, and may not be sufficient for training supervised classification methods. It causes imbalanced training data among process states [22], leading to compromised anomaly detection performance in actual manufacturing processes [81]. Specifically, when the number of training data samples from a normal state is significantly more than that of abnormal states, the prediction in classification models tends to be biased towards the normal state (i.e., majority class) [25]. This leads to a high probability of misclassifying

samples from the abnormal states (i.e., minority classes).

To overcome this severe issue caused by imbalanced data problems in the manufacturing process, data augmentation can be applied to obtain balanced training data among process states.

Simple data augmentation techniques such as rotation, flipping, and synthetic minority oversampling technique (SMOTE) [21] are widely used for balancing training data in the supervised classification method for the manufacturing process because of their simplicity in application [28, 76, 99]. However, these methods reflect the local information; thus, they could not consider the global information, including the entire data distribution, and suffers from the problem of overfitting [37, 95]. As a result, these methods are unsuitable for generating realistic and diverse data on abnormal states in manufacturing applications [41, 106]. Recently, the generative adversarial network (GAN) [47] has been actively used for data augmentation in manufacturing. This is because GAN generates more representative data than the simple augmentation method by learning the entire distribution of actual data through two neural networks: discriminator and generator [4, 66]. For example, [46] used conditional GAN to generate layer images from the metal AM process. In addition, [81] developed a novel GAN method considering temporal orders of the sensor signal from the polymer AM process to generate the balanced training signal data.

Utilizing GAN, the existing studies achieved high classification performance by training the classifier using the balanced training samples which are generated from GAN. The generated samples of each process state are realistic by learning the distribution of actual data. However, features enabling differentiation between process states (i.e., state-distinguishable features) that can further improve the classification performance have not been considered in the generation process from GAN in the previous studies [46, 81].

This paper proposes a novel GAN-based data augmentation method in manufacturing to generate realistic and state-distinguishable generated data. Compared to a standard GAN consisting of two players, i.e., discriminator and generator, the proposed method comprises three players, including a classifier. All players are jointly optimized in the proposed approach to meet their respective goals. Specifically, the generator learns to generate data deceiving the discriminator. In contrast, a discriminator learns to distinguish whether data is from a generator or an actual process. This adversarial learning results in generating realistic samples from the generator. At the same time, the generator and classifier cooperate to generate distinctive samples among process states in the manufacturing process. Specifically, the classifier guides the generator to create samples that could benefit classification results. Then, the classifier is trained with balanced training data supplemented from generated samples of abnormal states. However, the joint optimization between GAN and classifier suffers from the explosion of the generator's gradient, resulting in unstable learning and low-quality generated samples [5, 123]. It limits the improvements of the classifier's performance by joint optimization.

Therefore, the proposed method provides two terms in the objective function of the discriminator to improve the training stability of standard GAN. First, the proposed method regularizes the gradient of the discriminator [42]. Second, the proposed approach provides an additional task to the discriminator compared to standard GAN, preventing the discriminator from discerning the origin of data very well [57]. Both terms improve the training stability by preventing gradient exploding in training the generator. Finally, the iterative learning among three-player with these stabilizing techniques finally provides the classifier with high performance. The contributions of this work are summarized as follows:

- From the methodological point of view, this paper proposes a novel data augmentation method that standard GAN and classifier are jointly optimized with techniques

to stabilize the learning process of standard GAN. The generated samples from the generator provide balanced training data to the classifier. The iterative optimization between GAN and classifier provides a classifier with superior performance.

- From the application perspective, the proposed method is applied to anomaly detection in actual additive manufacturing processes. The technique successfully detects process anomalies when highly imbalanced data sets exist. The effectiveness of the proposed method is validated by the statistical hypothesis test [109] from polymer and metal AM processes, namely, Fused Filament Fabrication (FFF) and Electron Beam Melting (EBM) processes.

- In terms of the broad applicability of the proposed technique in the manufacturing process, the usage of the method is not limited to surface image data but also to signal data from various sensors, including microphones and acceleration sensors, widely used for process monitoring. In addition, the method can be directly applied to other manufacturing processes, such as semiconductor manufacturing [130] and automobile industries [106] that suffer from imbalanced data for their process monitoring.

The subsequent sections of this paper are structured in the following manner. The proposed methodology is presented in Section 3.2, followed by case studies with open-source and actual AM data sets to validate its effectiveness in Section 3.3. In addition, the proposed method is applied to biomaterial data to show the generalizability of the method in Section 3.4. Finally, conclusions are summarized in Section 3.5.

3.2 Proposed Research Framework

This section proposes a novel GAN-based data augmentation method. The structure of the proposed method is illustrated in Section 3.2.1. Specifically, the objective function of the proposed method is described in Section 3.2.2, followed by the training procedure in Section 3.2.3.

3.2.1 Three-Player Structure for Imbalanced Data Learning

Figure 3.2 shows the structure of the proposed method, which consists of three players: a discriminator, a generator, and a classifier. The generator generates images of the

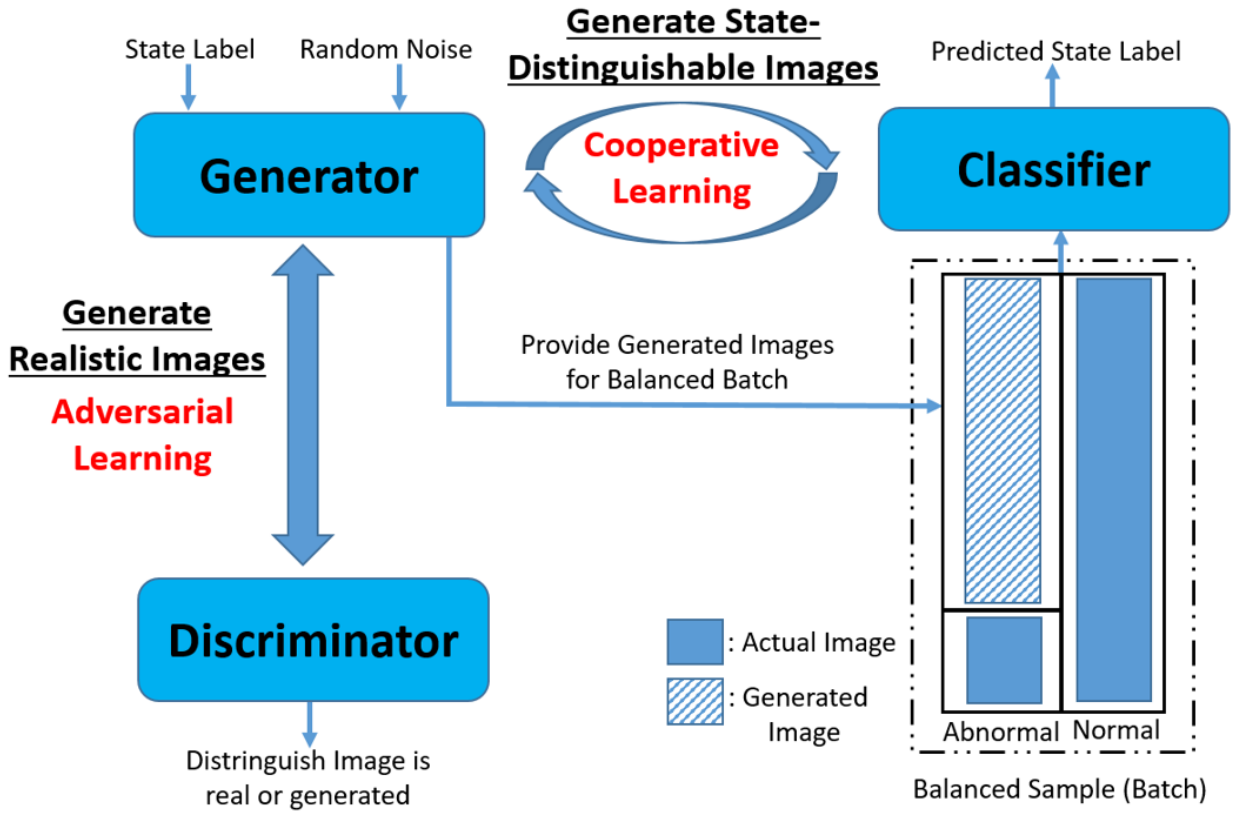


Figure 3.2: Structure of the proposed method.

manufacturing process from the random noise and corresponding state label. Among the generated images, the images of abnormal states are combined with the actual imbalanced manufacturing process images, providing a balanced training sample to the classifier. The proposed method offers adversarial and cooperative learning to make the generated images beneficial to the classifier's performance. The roles of these two are provided as follows.

- Adversarial learning: The relationship between discriminator and generator follows the adversarial relationship from the GAN structure. The relationship enables both networks to compete with each other, resulting in realistic generated samples of the manufacturing process from the generator.
- Cooperative learning: The cooperative relationship between the generator, and the classifier enables the generator to generate samples that are distinguishable among process states in the manufacturing process (i.e., state-distinguishable samples) from the classifier.

Based on these two relationships, the generator generates samples of abnormal states in the manufacturing process with both properties (i.e., realistic and state-distinguishable). The generated samples are added to actual ones and provided as a balanced training batch that the number of training samples passing through the network at one time of a classifier. The iterative learning process among these three players finally provides a classifier with high performance. The detailed objective function and training procedures are explained in the following sections.

3.2.2 Objective Functions for Three-Player

In this subsection, the review of the generative adversarial network in the manufacturing process is described in Section 3.2.2.1 initially. Then, the objective functions of the dis-

criminator, generator, and classifier are illustrated in Sections 3.2.2.1, 3.2.2.2, and 3.2.2.3, respectively.

3.2.2.1 Generative Adversarial Network in the Manufacturing Process

The idea of a Generative Adversarial Network (GAN) is to train two networks, namely, generator G and discriminator D, with a minimax game for $V(D,G)$ demonstrated in Eq. (3.1) [125].

$$\begin{aligned}\min_G \max_D V(D,G) = \mathbb{E}_{x_a \sim P(X_a)}[\log(D(x_a))] \\ + \mathbb{E}_{z \sim P(Z)}[\log(1 - D(G(z))],\end{aligned} \tag{3.1}$$

where z is the random noise, and x_a denotes actual samples from the manufacturing process. Specifically, the generator is to generate samples of the manufacturing process $G(z)$ from z, and the discriminator is to distinguish whether the origin of input samples is from actual (x_a) or generator $(G(z))$. In other words, the discriminator is to discern the input samples, while the generator synthesizes artificial samples to deceive the discriminator. This adversarial learning results in the distribution of newly generated samples, close to the underlying distribution of the actual samples in the manufacturing process, $P(X_a)$.

As shown in Figure 3.2, the proposed method needs to provide balanced training samples among the manufacturing process states in every batch of a classifier. To achieve this, conditional GAN [37] is applied in the proposed method making the process state label attached to the input of the generator and discriminator. It enables users to determine the number of generated samples from abnormal states of the manufacturing process to make a balanced batch. Therefore, the proposed method does not suffer from mode collapse among the process states that the generator only generates the specific states as output [117]. The

objective function of conditional GAN is demonstrated in Eq. (3.2) as follows.

$$\begin{aligned}\min_G \max_D V(D,G) = \mathbb{E}_{(x_a,y_a)\sim P(X_a,Y_a)}[\log(D(x_a,y_a))] \\ + \mathbb{E}_{(z,y_g)\sim P(Z,Y_g)}[\log(1-D(G(z,y_g),y_g)],\end{aligned} \tag{3.2}$$

where y_a and y_g denote the process state label of an actual and generated sample, respectively. subsubsectionObjective Function of Discriminator The objective of the discriminator in the proposed method is to maximize Eq. (3.2) through adversarial learning with the generator. Specifically, the discriminator learns to distinguish the input (x_a, y_a) and $(G(z, y_g), y_g)$ are actual and generated, respectively. In addition, the proposed method provides two additional terms for stabilizing the learning process, as the training of the GAN is unstable and challenging to converge due to the gradient exploding issue in adversarial learning [5, 120].

First, the proposed method regularizes the gradient of the discriminator by providing the gradient penalty in the form of Eq. (3.3) as follows.

$$\mathbb{E}_{(\hat{x},y_a)\sim P(\hat{X},Y_a)}[(\|\nabla_{(\hat{x},y_a)} D(\hat{x},y_a)\|_2 - 1)^2], \tag{3.3}$$

where $\hat{x} = \alpha x_a + (1-\alpha)G(z)$, and α is sampled uniformly between 0 and 1. It enforces 1-Lipschitz continuity to the discriminator [29]. The term is widely used in the previous work [42, 48, 57] to overcome the limited diversity and poor quality of generated samples from GAN caused by unstable adversarial learning.

Second, the proposed method provides an additional input consisting of the actual sample (x_a) and a mislabeled process state (y_m) to the discriminator. The discriminator learns to distinguish that the input (x_a, y_m) is not an actual but generated sample because of the

mislabeled process state by minimizing the following quantity.

$$-\mathbb{E}_{(x_a,y_m)\sim P(X_a,Y_m)}[\log(1-D(x_a,y_m)],$$

where y_m is randomly sampled from the remaining labels in the manufacturing process except for the actual state label. Since the input provides an additional task for the discriminator to learn, it prevents the discriminator distinguishes very well between actual and generated samples before the generator approximates the actual sample distribution of the manufacturing process. Otherwise, it causes unstable learning of GAN through exploding or vanishing the gradient of the generator [5, 123].

In summary, the objective function of the discriminator (L^D) in the proposed method is to minimize the following equation consisting of several losses.

$$\begin{aligned} L^D(Z,X_a,Y_a,Y_g,Y_m) = \\ &\underbrace{-\mathbb{E}_{(x_a,y_a)\sim P(X_a,Y_a)}[\log(D(x_a,y_a))]}_{\text{loss from actual sample in discriminator}} \\ &\underbrace{-\mathbb{E}_{(z,y_g)\sim P(Z,Y_g)}[\log(1-D(G(z,y_g),y_g)]}_{\text{loss from generated sample in discriminator}} \\ &\underbrace{-\mathbb{E}_{(x_a,y_m)\sim P(X_a,Y_m)}[\log(1-D(x_a,y_m)]}_{\text{loss from mislabeled sample in discriminator}} \\ &+\lambda\underbrace{\mathbb{E}_{(\hat{x},y_a)\sim P(\hat{X},Y_a)}[(\|\nabla_{(\hat{x},y_a)}D(\hat{x},y_a)\|_2-1)^2]}_{\text{loss from gradient penalty}}, \end{aligned} \tag{3.4}$$

where λ is the coefficient of the gradient penalty term. The first three losses in Eq. (3.4) are related to losses when the discriminator misclassified the origin of the actual, generated, and mislabeled sample. The last loss represents the loss related to the gradient of the discriminator.

3.2.2.2 Objective Function of Generator

The objective of the generator is to generate samples via learning the distribution of actual samples of the manufacturing process ($P(X_a)$) by minimizing Eq. (3.2). Alternatively, the proposed method trains the generator to maximize Eq. (3.5) to avoid the saturation problem occurring when minimizing Eq. (3.2) in practice [47].

$$\mathbb{E}_{(z,y_g)\sim P(Z,Y_g)}[\log(D(G(z,y_g),y_g)]. \tag{3.5}$$

Therefore, Eq. (3.5) and the second term in Eq. (3.4) provide the adversarial learning between the discriminator and generator. In addition to Eq. (3.5), the generator in the proposed method has an additional term in the objective function related to the classifier. In contrast to an adversarial relationship with a discriminator, the cooperative relationship is designed between the generator and classifier to generate well-distinguishable samples among process states in the manufacturing process. Specifically, the generator needs to generate the extra samples to make a balanced training data set for the classifier, as shown in Figure 3.2, and the data need to be helpful for the classifier. In other words, the generator learns to generate the data that is well-classified from the classifier. To achieve this, the classification loss from labels of generated samples is provided to the objective function of the generator (2^{nd} term in Eq. (3.6)). Finally, the generator in the proposed method is trained by minimizing its objective function (L^G), Eq. (3.6).

$$\begin{aligned} L^G(Z,Y_g) = & \underbrace{-\mathbb{E}_{(z,y_g)\sim P(Z,Y_g)}[\log(D(G(z,y_g),y_g))]}_{\text{loss from generated sample in discriminator}} \\ & \underbrace{-\alpha\mathbb{E}_{(z,y_g)\sim P(Z,Y_g)}[y_g\log(C(G(z,y_g)))]}_{\text{loss from generated sample in classifier}}, \end{aligned} \tag{3.6}$$

where $-\mathbb{E}_{(z,y_g)\sim P(Z,Y_g)}[y_g \log(C(G(z,y_g)))]$ (2^{nd} term in Eq. (3.6)) denotes the cross-entropy loss of generated samples from the classifier. α is the coefficient that needs to be optimized based on the performance of the validation set. The coefficient is optimized to be constant (e.g., 1) or increased as the classifier's epoch increases since the classifier is not trained enough in the initial part of the learning. To achieve this, the proposed method designs the rule of α as follows: $\alpha = (\gamma)^{\frac{100}{\text{epochs}}}$, where $\gamma \in (0, 1]$. γ is optimized by grid search.

3.2.2.3 Objective Function of Classifier

The objective function of the classifier (L^C) consists of the classification loss from both actual and generated samples of the manufacturing process as Eq. (3.7). As described in Figure 3.2, the samples from the generator are supplemented with actual samples from the manufacturing process to make balanced training data in every batch of the classifier. The classifier is optimized to minimize the classification loss from both actual and generated sample by minimizing Eq. (3.7).

$$L^C(Z, X_a, Y_a, Y_g) = \underbrace{-\mathbb{E}_{(x_a,y_a)\sim P(X_a,Y_a)}[y_a \log(C(x_a))]}_{\text{loss from actual sample in classifier}} \underbrace{-\mathbb{E}_{(z,y_g)\sim P(Z,Y_g)}[y_g \log(C(G(z,y_g))]}_{\text{loss from generated sample in classifier}}. \tag{3.7}$$

In particular, $-\mathbb{E}_{(z,y_g)\sim P(Z,Y_g)}[y_g \log(C(G(z,y_g))]$, a common term in both Eqs. (3.6) and (3.7) enables cooperative learning between the generator and classifier.

3.2.3 Training Procedure

To train the three players in the proposed method, the method adopts an alternating gradient descent method among the training of generator, discriminator, and classifier. Before

starting alternating optimization, the auto-encoder is pre-trained with the existing imbalanced samples from the manufacturing process. Auto-encoder is trained to minimize the reconstruction error of inputs. Auto-encoder is widely used for the initialization of the generator in GAN because it leads to stable learning and helps the generator learn common data set knowledge [92]. The pre-trained decoder from an auto-encoder is initialized as the generator in the proposed method. After the pre-training step, the three players are optimized alternatively. First, the discriminator is trained with a batch from actual and generated samples to minimize Eq. (3.4). Sequentially, a batch from generated samples is utilized to update the generator by minimizing Eq (3.6). Finally, the classifier is trained by minimizing Eq. (3.7) with balanced training data from all the process states in the manufacturing process. Specifically, a batch (m) from actual data is sampled first. Then, the remaining samples from abnormal states (m_g) are generated from the generator to make a balanced training set. The alternating training procedure is iterated until it reaches the pre-defined epochs. The overall training procedure of the proposed method is illustrated in Algorithm 3.1. The parameters of a discriminator, generator, and classifier are denoted as θ_d, θ_g, and θ_c, respectively.

3.3 Real-World Case Studies

Several real-world case studies are provided to show the effectiveness of the proposed method. Section 3.3.1 demonstrates the advantages of the proposed method based on the ablation study. The method is deeply self-analyzed in various aspects. Sections 3.3.2, 3.3.3, and 3.3.4 provide comparative case studies with benchmark methods across multiple data sets. Specifically, open-source data set is used in Section 3.3.2. In addition, the surface image data from two actual AM processes, namely, Fused Filament Fabrication (FFF) and Electron Beam

Algorithm 3.1 Training Procedures of the proposed method.

Require:
Initialize the parameters of three players (i.e., θ_d, θ_g, and θ_c).
P-Epoch: Number of epochs in the pre-training.
A-Epoch: Number of epochs in the alternating loop.
Batch size: Size of samples (m) in each batch.

Procedure:
[**Pre-Training Generator**]
Initialize: epoch=1
while epoch $\leq$ P-Epoch **do**
Sample a batch of m samples from X_r
Train auto-encoder (θ_g)
epoch + +
end while
[**Alternating Optimization**]
Initialize: epoch=1
while epoch $\leq$ A-Epoch **do**
Sample m samples from X_r, Z, Y_g, Y_m
Train a discriminator (θ_d) **by minimizing Eq.** (3.4)
Sample m samples from Z, Y_g
Train a generator (θ_g) **by minimizing Eq.** (3.6)
Sample m samples from X_r
Sample m_g from Z, Y_g of abnormal states to
balance the batch
Train a classifier (θ_c) **by minimizing Eq.** (3.7)
epoch + +
end while

Melting (EBM) processes, are utilized in Sections 3.3.3 and 3.3.4, respectively, to demonstrate the effectiveness of the proposed method in an actual AM process. The performance is evaluated by the classification results from the imbalanced training data set. The framework of all case studies is Keras with TensorFlow backend. The studies are performed by an NVIDIA Tesla P4 GPU with 8GB memory.

1) Benchmark Methods: For the benchmark methods, both the sampling-based and GAN-based approaches are used. In the sampling-based methods, SMOTE [21], B-SMOTE [52], and ADASYN [54], which are implemented in the imbalanced-learn library in python,

are adopted. For the GAN-based approaches, two state-of-the-art methods, CDRAGAN [57], and BAGAN-GP [57] are selected. In addition, Cooperative GAN [25], which jointly optimizes GAN and the classifier, is determined as one of the benchmark methods. Finally, classification performance without any data augmentation method is provided as the baseline.

2) Hyperparameters and Experimental Settings: The optimizer for the proposed method is the Adam algorithm [64] with a learning rate of 0.0002 and momentums of 0.5 and 0.9 [57]. To make the networks in the proposed method irrespective of the image size, all the image inputs are resized as 64×64×channels. The dimension of the random noise (z) is 128. The coefficient of cross-entropy loss in the generator (α in Eq. (3.6)) is set to one based on the grid search in the case studies. Since the proposed method deals with image data, the auto-encoder with convolution layers is used for pre-training [57]. The discriminator consists of the convolutional layers to judge whether the images are actual or generated. In contrast, the generator layer uses transposed convolution layers to generate the images from the random noise. For the classifier, Convolutional Neural Network (CNN) is utilized in the case studies since CNN extracts the features from the raw data directly, resulting in superior performance in image classification [32]. For a fair comparison, the CNN with the same hyperparameters is utilized to all the methods. In addition, there exist unique hyperparameters of each method, such as the scheduling parameter in the Cooperative GAN [25]. These hyperparameters are searched within a specific range following the guidelines provided in the literature and determined with the values that showed the best performance. The proposed method has a unique hyperparameter compared to the benchmark methods, which is the coefficient of cross-entropy loss in the generator (α in Eq. (3.6)). The coefficient is optimized based on the grid search in the case study. The detailed hyperparameters of the generator, discriminator, and classifier are provided in Appendix A.3.

3) Performance Evaluation Measure: The classification performance is evaluated by the value of the F-score, Precision, and Recall [104]. Precision and Recall are related to the level of type I and type II errors, respectively. F-score can be formulated by Eq. (3.8), which is the combination of Precision and Recall.

$$\text{F-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{3.8}$$

Since this paper aims to improve the classification accuracy with the imbalanced training data, case studies with various balanced ratios are provided. The balanced ratio refers to the proportion of the training data size for the minority and majority classes. All the case studies were repeated ten times, and the average of all classes' performances from ten repetitions was provided as the performance measure.

3.3.1 Ablation Studies

The ablation study is conducted with MNIST fashion data [135]. From the 1000 images of each of the three labels, namely, T-shirt, Pullover, and Dress, imbalanced training data is constructed as described in Table 3.1. To provide the challenging task, the balanced ratio between the majority and minority classes is 0.10. The remaining images are used as testing data. The study is conducted through a step-by-step process of integrating each

Table 3.1: Imbalanced training data samples in ablation studies.

Data Set	Majority Class	Minority Class	Balanced Ratio	Majority Class Training Samples	Minority Class Training Samples
MNIST fashion	T-shirt	Pullover, Dress	0.10	800	80

ablation component since each component is dependent on the previous ones and cannot be implemented independently. The importance of each component is validated by the

ablation of a baseline and three variants, as illustrated in Table 3.2. Baseline trains classifier with imbalanced training data without any data augmentation. Instead, Variant 1 uses a conditional version of standard GAN (i.e., CGAN [96]) to provide balanced training data of the classifier by generating additional minority class samples ($G(z, y_g)$ in Section 3.2). Furthermore, Variant 2 jointly trains the classifier and CGAN by providing cross-entropy losses ($y_g \log(C(G(z, y_g)))$, and $y_a \log(C(x_a))$ in Section 3.2) so that the generator in the CGAN produces distinctive samples among the process states. Finally, Variant 3, which is the proposed method, adds the terms ($\|\nabla D(\hat{x}, y_a)\|_2$, and $D(x_a, y_m)$ in Section 3.2) that are related to stabilizing the learning process in the objective function of the discriminator in Variant 2. In addition, the coefficient of cross-entropy loss in the generator ($\alpha = (\gamma)^{\frac{100}{\text{epochs}}}$, where $\gamma \in (0, 1]$ in Eq. (3.6)) is optimized in Variant 3. γ is searched by the grid size with intervals of 0.1 in the range. The validation data consists of 100 and 10 samples from majority and minority classes from Table 3.1. Finally, γ is optimized to one by the classification performance (F-score) of the validation data set.

Table 3.2 shows the results of the ablation study. In addition to F-score, Precision, and Recall, Frechet Inception Distance [38] (FID) is used as an additional performance measure in the ablation study. FID is a metric used to evaluate the quality of images generated by GAN. Specifically, it is a metric between two multidimensional Gaussian distributions that are the distribution of neural network features from the actual and generated images, respectively. FID is computed from the mean and the covariance of the activation function of the network as follows:

$$\text{FID} = \|\mu_a - \mu_g\|^2 + \text{Tr}(\Sigma_a + \Sigma_g - 2(\Sigma_a \Sigma_g)^{1/2}),$$

where μ_a and Σ_a are the mean and standard deviation from actual images, respectively; likewise, μ_g and Σ_g are those from generated images, respectively. The smaller FID represents that the generated images follow the distribution of actual images resulting in better quality

and diversity [23]. In Table 3.2, the FID scores of each label are presented. 1500 images of each label from actual and generated images, respectively, are used to calculate the FID.

Table 3.2: Performance evaluation in ablation studies.

	$G(z, y_g)$	$y_a \log(C(x_a))$, $y_g \log(C(G(z, y_g)))$	$\|\nabla D(\hat{x}, y_a)\|_2$, $D(x_a, y_m)$	**Precision**	**Recall**	**F-score**	**FID T-shirt**	**FID Pullover**	**FID Dress**
Baseline	✗	✗	✗	0.784	0.872	0.782	NA	NA	NA
Variant1	✓	✗	✗	0.789	0.878	0.791	190.7	222.9	232.6
Variant2	✓	✓	✗	0.804	0.892	0.816	185.9	238.1	233.2
Variant3 (Proposed)	✓	✓	✓	**0.818**	**0.904**	**0.836**	**108.5**	**134.6**	**176.7**

The performance of all the measures of Variant 1 improves compared to the baseline. It shows the effectiveness of GAN for supplementing imbalanced training data. In Variant 2, an improvement in the performance of most of the measures is achieved compared to the first ablation. These results show that joint training of GAN and classifier guides the generator to generate samples beneficial to the classifier, that is, both realistic and state-distinguishable samples. It consequently improves the performance of the classifier. Finally, when both terms stabilizing the learning process of GAN are added, which is the proposed method (Variant 3), the performances are significantly improved since stable training enables the generator to generate more realistic and state-distinctive samples. The classification results of each label in Table 3.2 are provided in Appendix A.1. In addition, Table 3.3 shows the p-value of the paired t-test [109] of each measure, where the alternative hypothesis is that the average results in Table 3.2 are better than the previous ablations. The results indicate that joint optimization of GAN and classifier and terms stabilizing the learning process of GAN significantly improve the classification performance at a 95% significance level.

Figure 3.3 compares the log scale norm for the gradient of the generator ($\log(\nabla \| L^G(Z, Y_g) \|)$) between Variant 2 and the proposed method. The results show the jointly optimization between GAN and classifier (Variant 2) suffers from the explosion of the generator's gradient, resulting in unstable learning and low-quality generated samples [5]. However, the gradient

Table 3.3: P-value of paired t-test in the ablation studies.

	Precision	Recall	F-score
Variant1 $\geq$ Baseline	0.067	0.047	0.025
Variant2 $\geq$ Variant1	0.020	0.019	0.014
Variant3 $\geq$ Variant2	0.012	0.012	0.014

of the proposed method decreases with the small variance.

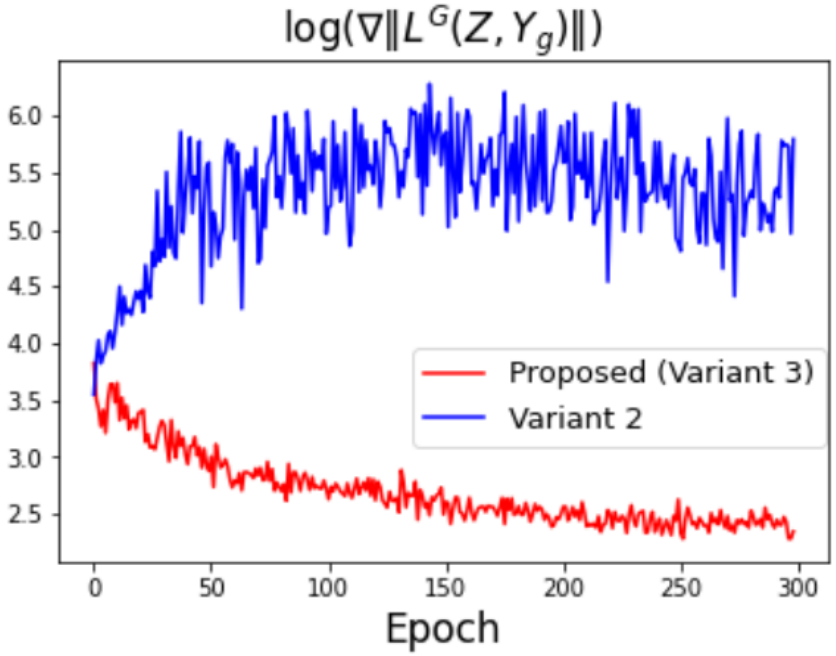


Figure 3.3: Comparison of the gradient of the generator.

Figure 3.4 shows the quality of generated images in each step of the ablation study. Figure 3.4 (a) represents the actual images from each label. Figures 3.4 (b), (c), and (d) illustrate the generated images of each label from Variant 1, 2, and the proposed method, respectively. Like the results of the FID score in Table 3.2, the quality of generated samples significantly improves in the proposed method.

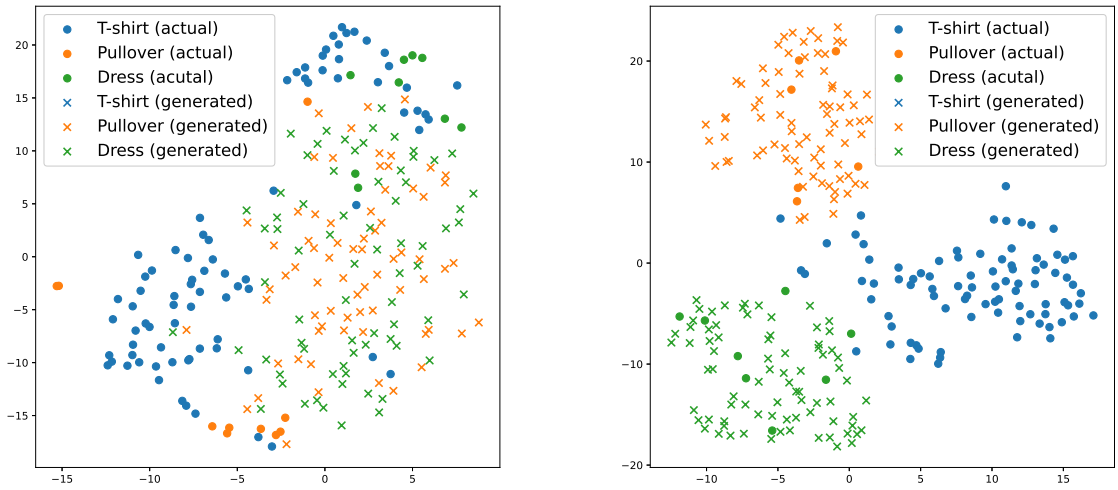
Beyond the image quality and FID score, Figure 3.5 shows the effectiveness of the generated samples from the proposed method based on the comparison between the feature of gen-

(a) (b)

(c) (d)

Figure 3.4: Samples of T-shirt, Pullover, and Dress from (a) actual images, (b) generated images from Variant 1, (c) generated images from Variant 2, and (d) generated images from Variant 3 (Proposed method).

erated and actual samples. Specifically, Figure 3.5 represents the t-distributed Stochastic Neighbourhood Embedding (t-SNE) of the feature from the intermediate layer of the classifier in the proposed method. t-SNE is a tool for visualizing high-dimensional data [33]. It is



(a) (b)

Figure 3.5: t-SNE of the feature from the intermediate layer of the classifier from the proposed method in epochs (a) 0 and (b) 300.

a nonlinear dimensionality reduction technique suitable for incorporating high-dimensional

data into lower-dimensional data (2-D or 3-D) for visualization. '•' and '×' in Figure 3.5 denote the feature in the intermediate layer of classifiers from actual and the generated samples in the balanced training batch, respectively. For a balanced training batch, the minority class has many '×' in each batch. Figure 3.5 (a) shows the distribution alignments of actual and generated samples are different at epoch 0. Since the proposed method learns to generate realistic and distinctive samples for the classifier, Figure 3.5 (b) shows that the features of generated samples ('×') correctly follow those of actual samples ('•') according to their respective labels at epoch 300. In addition, features from each label are clearly separated. It validates the authentic and state-distinguishable properties of the generated samples of the proposed method. The balanced training data with these properties let the proposed method achieve high classification performance. t-SNE results from the several epochs in Figure 3.5 are provided in Appendix A.2 to show that the generated samples from the proposed method follow the features of actual data in the classifier as the epoch increases.

3.3.2 Open-Source Data Case Study

The open-source image data set is used for the comparative study in this section. MNIST fashion [135], which is widely used for the evaluation of image-based classifiers, is selected. In MNIST fashion data, three labels related to upper clothes are selected for analysis. This step provides high similarity among the classes to make a challenging problem [25]. In each label of MNIST fashion, 1000 images are collected, respectively. Then, imbalanced training data is provided in Table 3.4. The balanced ratios are determined as 0.10. This is because the balanced ratios with less than these values provide poor performance that is meaningless for the comparison. The remaining data sets are used as testing data.

Table 3.5 shows the performance evaluation of the proposed and benchmark methods. The

Table 3.4: Imbalanced training data samples in open-source data case study.

Data Set	Majority Class	Minority Class	Balanced Ratio	Majority Class Training Samples	Minority Class Training Samples
MNIST fashion	T-shirt	Pullover, Dress	0.10	800	80

proposed method achieves the best performance in most measures in both data sets. Compared to a baseline using imbalanced training data to train the classifier, the sampling-based method such as SMOTE [21], B-SMOTE [52], and ADASYN [54] generally achieves similar or worse performance. Since this case study has a small number of minority data, the

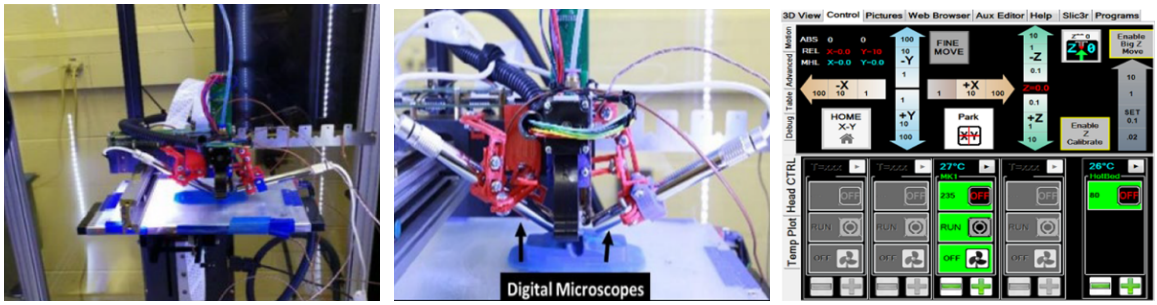
Table 3.5: Performance evaluation in open-source data case study.

	MNIST fashion		
	Precision	**Recall**	**F-score**
Baseline	0.784	0.872	0.782
SMOTE	0.780	0.867	0.774
B-SMOTE	0.778	0.864	0.771
ADASYN	0.780	0.866	0.774
CDRAGAN	0.800	0.891	0.809
BAGAN-GP	0.797	0.888	0.805
Cooperative GAN	0.800	0.890	0.811
Proposed	**0.818**	**0.904**	**0.836**

sampling-based method has limitations in generating various data that can cover the testing data. In contrast, the GAN-based techniques usually achieve better performance than sampling-based methods since their generators learn the actual distribution and produce various training data for the classifier. Especially, the generator from the proposed method provides more diverse and better quality images by jointly optimizing the classifier with stabilizing techniques, resulting in improvements in classification results.

3.3.3 Polymer Additive Manufacturing Process Data Case Studies

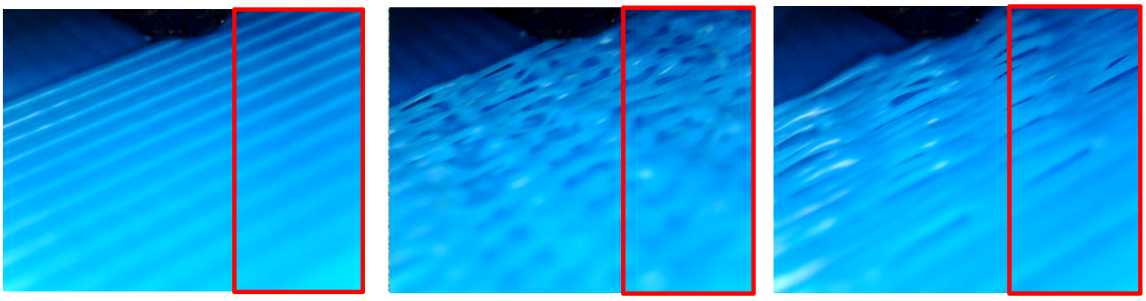
A Hyrel System 30M 3-D printer equipped with a 0.5 mm extruder nozzle is used for this case study. Figure 3.6 (a) shows a front view of the printer. Acrylonitrile butadiene styrene (ABS) is used as material (filament) in this case study. Specifically, the diameter of ABS is 1.75 mm. This study uses a digital microscope camera for high-quality image acquisition at a sampling frequency of 1 Hz. The camera is mounted near the extruder to collect images of the surface that is being printed (Figure 3.6 (b)). Based on the design of experiments



(a) (b) (c)

Figure 3.6: (a) Front view of Hyrel system 30M; (b) Digital Microscope Camera; (c) Software Controller.

in [83], the surface images for normal state, under-fill caused by feed rate and under-fill caused by the cooling fan, as described in Figure 3.7, can be achieved by configuring the software controller with particular machine parameters (Figure 3.6 (c)). Two process states which are the under-fill caused by feed rate and under-fill caused by the cooling fan, are the abnormal states that cause quality deterioration in the FFF process [83, 114]. To increase the size of image samples, the region of interest is one-third of each image in Figure 3.7 with the size of $460 \times 213 \times 3$. Therefore, each image collected from a microscope provides three image samples. In total, there are 915 samples from the normal process state and 591 and 459 samples from under-fill caused by feed rate and cooling fan, respectively. This section provides case studies with various balanced ratios between normal and abnormal states of



(a) (b) (c)

Figure 3.7: (a) Normal; (b) Under-fill caused by feed rate; (c) Under-fill caused by a cooling fan. The red rectangle represents the regions of interest in each image.

the FFF process to see the performance change according to ratio changes. The balanced ratios of training data are provided in Table 3.6, where the minority classes are denoted as the cause of abnormal states in the process. Specifically, the balanced ratios provide the performances of F-score, Precision, and Recall that are applicable in practice are utilized. The remaining images in each process state are used as testing data.

Table 3.6: Imbalanced training data samples in Polymer AM case studies.

Majority Class	**Minority Class**	**Balanced Ratio**	**Majority Class Training Samples**	**Minority Class Training Samples**
Normal	Under-feed, Under-fan	0.10	800	80
Normal	Under-feed, Under-fan	0.15	800	120
Normal	Under-feed, Under-fan	0.20	800	160

Figure 3.8 shows the performance evaluation of the proposed and benchmark methods in various balanced ratios. The performances of all the methods are improved when the number of training samples in minority classes increases (i.e., the balanced ratio increases). This is because a large number of samples provides more information for the generator to learn the actual distribution. In every balanced ratio, the proposed method achieves the best performance in all the measures, which shows the effectiveness of the realistic and state-distinguishable generated samples in the AM process. Specifically, the proposed method

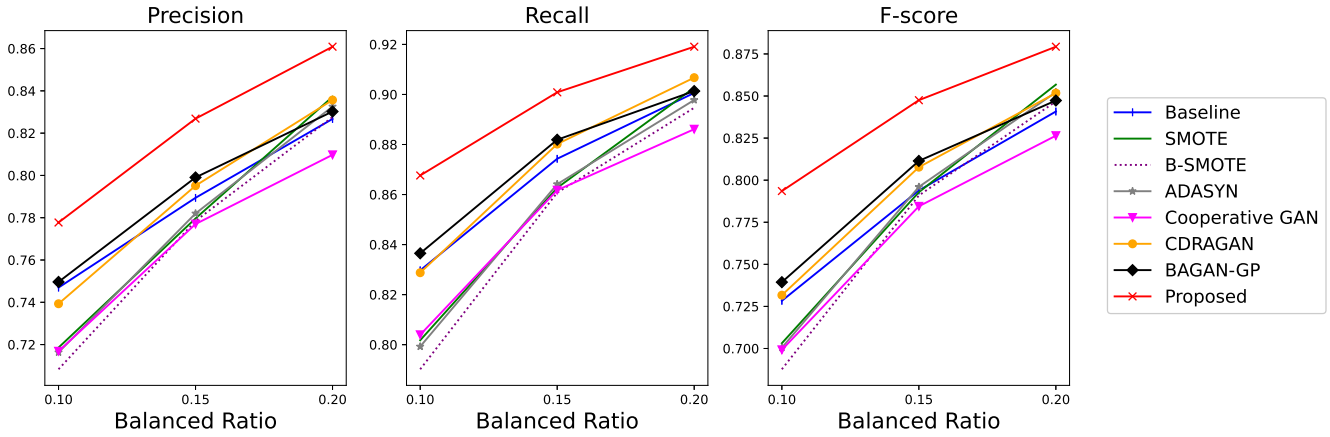


Figure 3.8: F-scores, precisions, and Recalls in FFF process case studies under different balanced ratios.

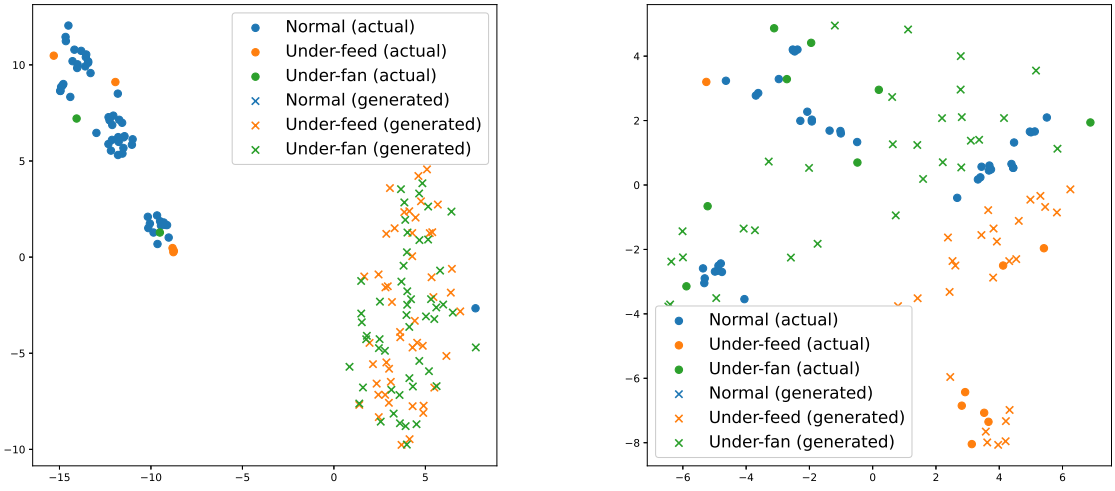
improved 3%∼13%, 3%∼10%, and 2%∼10% of the performance of the benchmark methods regarding their F-score, Precision, and Recall, respectively. In addition, the proposed method performs better than cooperative GAN, which also jointly optimizes GAN and classifier but follows the basic objective function of the discriminator in GAN. This demonstrates the effectiveness of stable learning through regularizing the gradient of the discriminator and an additional task provided to the discriminator in the proposed method. Sampling-based methods show worse performance than GAN-based methods in general since the methods only consider the local information resulting in limited diverse generated images [37]. Table 3.7 represents the p-value of the relative t-test, where the alternative hypothesis is that the performance of the proposed method in Figure 3.8 is better than the method that achieves the best performance among the benchmark methods in each balanced ratio. The results indicate the proposed method is better than the benchmark method at a 95% significance level in all cases.

Figure 3.9 shows the t-SNE of the feature from the intermediate layer of classifiers from the proposed method in epochs 0 and 300 when the balanced ratio is 0.20. Like Figure 3.5, '•' and '×' represent features of actual and generated samples, respectively. The colors differentiate

Table 3.7: P-value of paired t-test in Polymer AM case studies.

	Precision	Recall	F-score
Balanced Ratio: 0.10	0.000	0.000	0.000
Balanced Ratio: 0.15	0.001	0.002	0.000
Balanced Ratio: 0.20	0.008	0.029	0.007

each process state in the FFF process. To make a balanced training data, two abnormal

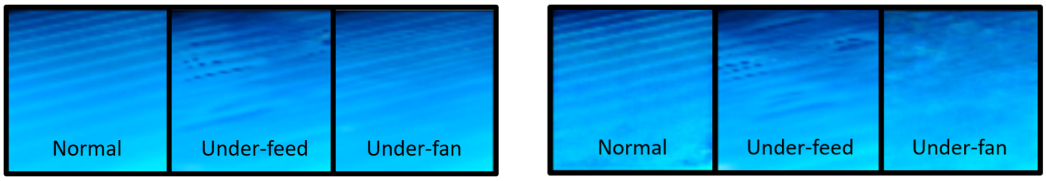


(a) (b)

Figure 3.9: t-SNE of the feature from the intermediate layer of classifiers from the proposed method in epochs (a) 0 and (b) 300 in the FFF process when the balanced ratio is 0.10.

states in the FFF process have adequate generated samples ('×') than actual samples ('•') in each batch. Compared to epoch 0 (Figure 3.9 (a)), features in epoch 300 (Figure 3.9 (b)) show that the features from generated samples ('×') of abnormal states of the FFF process follow those of actual samples ('•') correctly to each process state. Based on these balanced training data in the FFF process, the classifier achieves the best prediction results compared to benchmark methods. Figure 3.10 shows the samples of actual and generated images denoted as '•' and '×' in Figure 3.9 (b), respectively. The generated images (Figure 3.10

(b)) are realistic and distinguishable among process states based on both adversarial and cooperative learning in the proposed method.

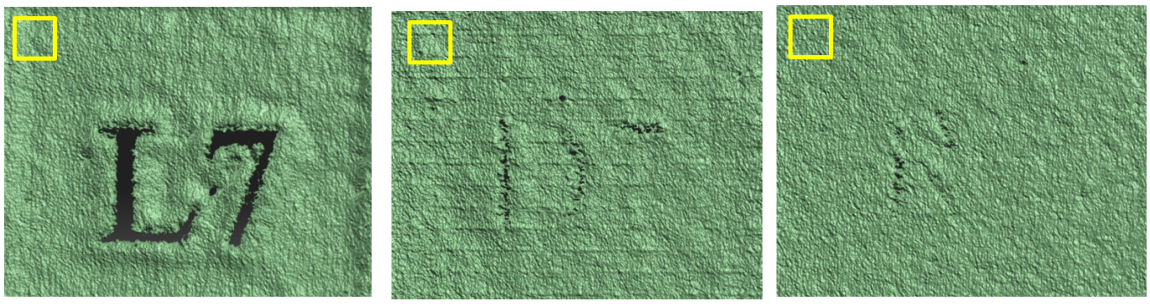


(a) (b)

Figure 3.10: Samples of each process state in the FFF process from (a) actual; (b) generator when the balanced ratio is 0.2.

3.3.4 Metal Additive Manufacturing Process Data Case Studies

The machine ARCAMQ10 plus is utilized to print samples from the EBM process using Ti-6A1-4V powder. The dimensions of the printed sample are 15 mm × 15 mm × 25 mm. In the EBM process, there exist three different scan strategies as shown in Figure 3.11, including Random, Dehoff, and Raster [65]. Distinct scanning techniques produce varied surface patterns on the printed samples. After printing three AM samples with each scan strategy, a 3-D scanner captures detailed 3-D information about the top surface (15 mm × 15 mm) quality [128].



(a) (b) (c)

Figure 3.11: (a) Raster. (b) Dehoff. (c) Random; The yellow square indicates the region of interest.

In this case study, the aim of classification is to recognize the scanning strategies used based on the surface images. The size of each image in Figure 3.11 is $824 \times 1118 \times 3$. For training samples, 322 images with sizes of 120 by 120 are collected from the upper part of each image ($300 \times 1118 \times 3$) since the bottom parts of the surface image with letters and numbers have many defects, such as porosity. Specifically, the collected images are highly overlapped in the horizontal directions (114 pixels) to have plenty of samples. Since the Raster scan strategy is commonly used and similar to the common AM bi-directional path, the strategy is considered a majority class [100, 112]. Therefore, the scan strategies with Dehoff and Random are determined as minority classes in these case studies. From the 322 images from each scan strategy, the various balanced ratios of training data are designed as in Table 3.8. As with polymer AM case studies in Section 3.3.3, the balanced ratios that provide the reasonable F-score, Precision, and Recall applicable in practice are utilized in case studies. The remaining images in each scan strategy are used as testing data.

Table 3.8: Imbalanced training data samples in Metal AM case studies.

Majority Class	**Minority Class**	**Balanced Ratio**	**Majority Class Training Samples**	**Minority Class Training Samples**
Raster	Dehoff, Random	0.3	150	45
Raster	Dehoff, Random	0.4	150	60
Raster	Dehoff, Random	0.5	150	75

Figure 3.12 shows the performance evaluation of the proposed and benchmark methods in the EBM process. In this case study, most sampling-based methods perform better than the baseline, but the GAN-based methods usually represent worse results than the baseline. This might be caused by the highly overlapped actual samples. Compared to the polymer AM case study, the number of actual images is small and highly overlapped since we only have a single image of the top surface. Therefore, it does not provide enough information

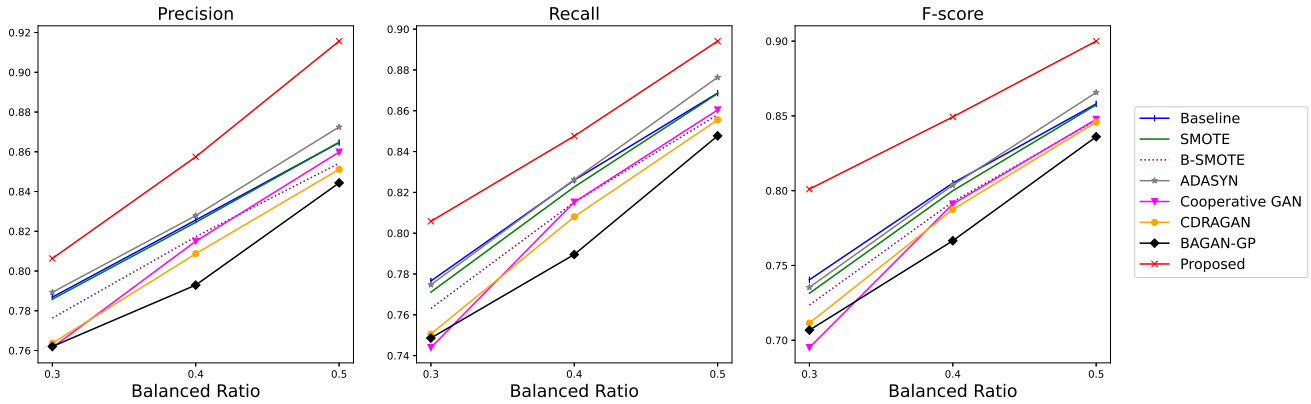


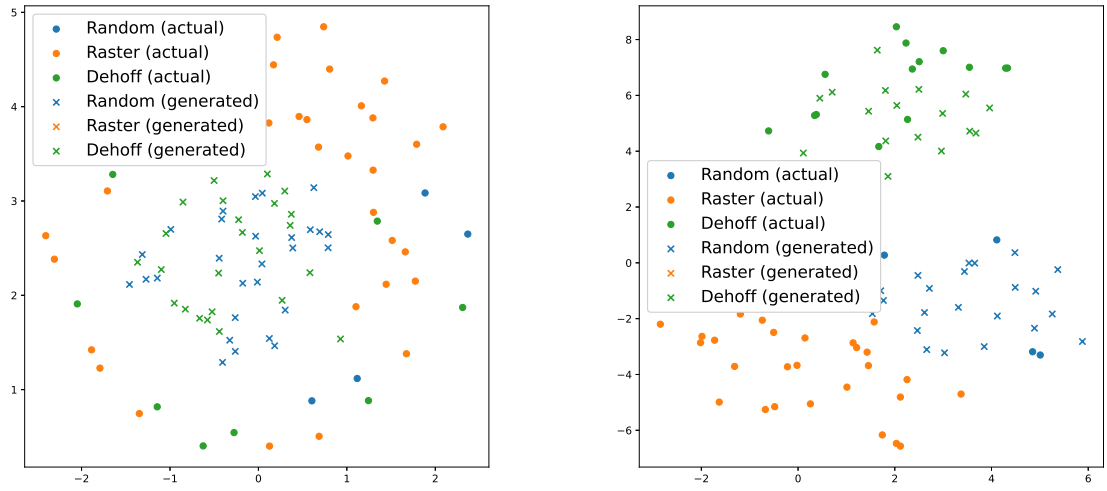
Figure 3.12: F-scores, Precisions, and Recalls in EBM process case studies under different balanced ratios.

for the GAN-based methods to learn the actual distribution. Relatively, the sampling-based methods show better performance because the actual images are highly overlapped with each other. Still, the proposed method achieves the best results compared to benchmark methods by generating balanced training samples that are both realistic and scan strategy-distinguishable. Specifically, the F-score, Precision, and Recall of the proposed method were improved by 4%∼15%, 2%∼8%, and 3%∼8% over those of benchmark methods, respectively. Table 3.9 represents the p-value of the relative t-test, where the alternative hypothesis is that the performance of the proposed method in Figure 3.12 is better than the method that achieves the best performance among the benchmark methods in each balanced ratio. The results indicate the proposed method is better than the benchmark method at a 95% significance level in most of the cases.

Table 3.9: P-value of paired t-test in Metal AM case studies.

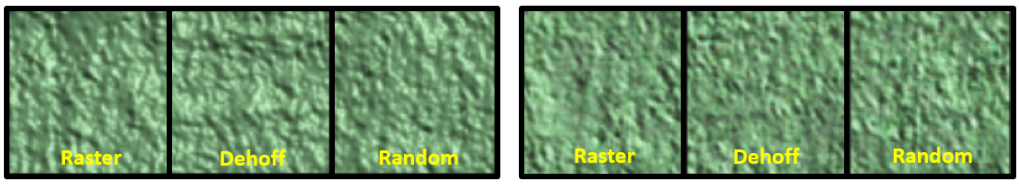
	Precision	Recall	F-score
Balanced Ratio: 0.3	0.044	0.003	0.000
Balanced Ratio: 0.4	0.041	0.109	0.009
Balanced Ratio: 0.5	0.002	0.132	0.028

t-SNE results in Figure 3.13 show similar results to the FFF process in Figure 3.9. In epoch 300, the features from generated samples follow those of corresponding scan strategies from actual samples. Figure 3.14 shows the actual and generated samples of all the scan strategies from the proposed method denoted as '•' and '×' in Figure 3.13 (b), respectively. Compared to the actual images in Figure 3.14 (a), the generated images in the EBM process (Figure 3.14 (b)) are realistic. In addition, the generated images are distinguishable according to each scan strategy, which is validated by the superior classification performance of the proposed method.



(a) (b)

Figure 3.13: t-SNE of the feature from the intermediate layer of classifiers from the proposed method in epochs (a) 0 and (b) 300 in the EBM process when the balanced ratio is 0.3.



(a) (b)

Figure 3.14: Samples of each scan strategy in the EBM process from (a) actual; (b) generator when balanced ratio is 0.3.

3.4 Biomaterial Application

In this section, we addressed the supervised classification problem using the imbalanced biomaterial data to show the generalizability of the proposed method. Spectral data associated with the 'solution (water)-gel (solid)' phase classification of Pluronic F127-water mixtures are utilized [141]. Specifically, spectral data of the Pluronic F127-water mixture are collected by varying the concentration ratio of the mixture. Figure 3.15 shows the phase angle spectral data of Pluronic F127-water mixtures by varying their concentration ratio [141]. Each spectral data is labeled 'solution' or 'gel.' The spectral data convert to the image data for usage in the case study. Among 288 spectral data, 181 data are labeled as gel, and the remaining 107 data are labeled as the solution. Imbalanced training data for binary supervised classification problem is provided in Table 3.10. The balanced ratios are determined as 0.016. This is because baseline achieves the satisfactory classification performance when the balanced ratios are higher than 0.016.

Table 3.10: Imbalanced training data samples in the biomaterial case study.

Data Set	Majority Class	Minority Class	Balanced Ratio	Majority Class Training Samples	Minority Class Training Samples
Pluronic F127-water mixtures	gel	solution	0.016	120	2

Table 3.11 shows the test performance. Compared to the baseline that only uses extremely imbalanced training data, all methods achieve better performance by augmenting the data

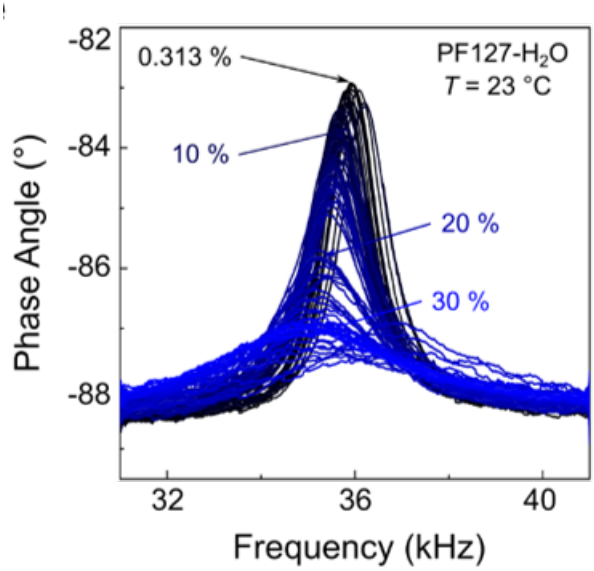


Figure 3.15: Phase angle spectral data of Pluronic F127-water mixtures by varying the concentration ratio [141].

from the minority class (i.e., solution). Specifically, the proposed method provides the best performance in all measures. The results indicate that the proposed method can effectively apply to biomaterial data to address the imbalanced supervised classification problem.

Table 3.11: Performance evaluation in the biomaterial case study.

	Pluronic F127-water mixtures		
	Precision	**Recall**	**F-score**
Baseline	0.845	0.861	0.824
SMOTE	0.937	0.953	0.938
B-SMOTE	0.923	0.941	0.924
ADASYN	0.926	0.943	0.927
CDRAGAN	0.933	0.951	0.936
BAGAN-GP	0.890	0.914	0.889
Cooperative GAN	0.910	0.924	0.905
Proposed	**0.948**	**0.961**	**0.951**

3.5 Conclusions

This paper proposes a novel GAN-based data augmentation method to deal with the imbalanced training data issue in manufacturing processes. The method consists of three-player, namely, generator, discriminator, and classifier, jointly optimized. Through the adversarial learning between the generator and the discriminator, the generator generates realistic samples of abnormal states in the manufacturing process. At the same time, the cooperative learning between the generator and classifier enables the generator to generate the state-distinguishable samples of the manufacturing process. The generated samples are added to actual samples and provided as a balanced training batch for a classifier. In addition, the method regularizes the gradient of the discriminator and provides an additional task to the discriminator compared to standard GAN. It prevents the gradient exploding of the generator resulting in a better quality of generated samples. The iterative learning process among these three players finally provides a classifier with high performance in classification results. The effectiveness of the proposed method is validated in both open-source and actual AM process data. Specifically, the method achieves the best performance compared to benchmark methods in the various balanced ratio of training data between normal and abnormal states in both the FFF and EBM processes. The outstanding results of the proposed method validated from the statistical hypothesis tests would contribute to mitigating the process defects timely, providing high reliability and yields of products in the processes.

The usage of the proposed method is not limited to image data. The method can be applied to signal data from various sensors, including microphone and acceleration sensors, widely used for process monitoring in manufacturing. Specifically, the signal data can be converted into image data by considering the temporal correlation of the signal [45] and can be applied to the proposed method. In addition, the proposed method can be directly applied to

any manufacturing processes that suffer from imbalanced data for process monitoring. For example, the number of defective wafers in semiconductor manufacturing is much smaller than that of non-defective wafers. The reason for this is that wafer fabrication is carried out within an environment that is both highly automated and free of dust[130]. In addition, the subtractive manufacturing [56] and automobile industries [106] also suffers from the imbalanced training data to achieve the effective process prognostic methods. Therefore, the proposed method can contribute to other advanced manufacturing processes in their process monitoring. Finally, the proposed method can be applied to various domains, including biomaterial data, as demonstrated in Section 3.4.

Chapter 4

A Novel Sparse Bayesian Learning and Its Application to Fault Diagnosis for Multistation Assembly Systems

The focus of this paper is on addressing the challenge of detecting process faults in multistation assembly systems. Fault diagnosis is to identify process faults that cause the excessive dimensional variation of the product using dimensional measurements. For such problems, the challenge is solving an underdetermined system caused by a common phenomenon in practice; namely, the number of measurements is less than that of the process errors. To address this challenge, this paper attempts to solve the following two problems: (1) how to utilize the temporal correlation of measurement samples and (2) how to apply prior knowledge regarding which process errors are more likely to be process faults. A novel sparse Bayesian learning method is proposed to achieve the above objectives. The method consists of three hierarchical layers. The first layer has parameterized prior distribution that exploits the temporal correlation of multiple measurement samples. Furthermore, the second and third layers achieve the prior distribution representing the prior knowledge of process faults. Then, these prior distributions are updated with the likelihood function of the measurement samples from the process, resulting in the accurate posterior distribution of process faults from an underdetermined system. Since posterior distributions of process faults are

intractable, this paper derives approximate posterior distributions via Variational Bayes inference. Numerical and simulation case studies using an actual autobody assembly process are performed to demonstrate the effectiveness of the proposed method.

4.1 Introduction

Multi-station assemblies refer to the systems that perform assembly operations to assemble a final product. The final product's quality is reliant on several components referred to as key control characteristics (KCCs) or process errors [12]. The positioning accuracy of fixture locators is KCC in the multistation assembly [9]. Fixture locators carry out the clamping of parts during the assembly process. Therefore, any deviation from their intended position can lead to dimensional quality issues in the final product. Thus, it is imperative to identify process errors that have mean shifts and/or large variance increases from their design specifications, namely, process faults [10]. Hence, fault diagnosis in multistation assemblies estimates the mean and variance of process errors, namely, the variations of fixture locators. This paper focuses on the process fault of the mean shift.

Because of the unduly cost and physical constraints, the dimensional variation of KCCs, namely, fixture locators, cannot be directly monitored using sensors in multistation assemblies [75]. Instead, key product characteristics (KPCs), namely, the key measurements from the final product, can be used to estimate the KCCs and, consequently, identify process faults among process errors. A fault-quality linear model of multistation process represents the relationship between KPCs and process errors as follows [58, 59]:

$$\mathbf{y} = \Phi\mathbf{x} + \mathbf{v}, \tag{4.1}$$

where $\mathbf{y} \in \mathbb{R}^{M\times 1}$represents M dimensional measurements (i.e., KPCs), $\mathbf{x} \in \mathbb{R}^{N\times 1}$denotes N process errors (i.e., KCCs), $\Phi \in \mathbb{R}^{M\times N}$ is a fault pattern matrix. The matrix contains all the process information of multistage process [34], and $\mathbf{v} \in \mathbb{R}^{M\times 1}$ denotes the noise. Although the relationship between KPCs and KCCs is nonlinear in many manufacturing processes, a fault-quality linear model is used in general as the nonlinear relationship could be approximated to a linear model utilizing Taylor series expansion because the variations of KCCs are small, and the relationship between KCCs and KPCs is smooth and without sharp changes [35, 75, 115]. Since process errors indicate the mean deviations of the fixture locators, process faults refer to nonzero elements in $\mathbf{x}$, namely, process errors with nonzero mean shifts [12].

Following the fault-quality model in Eq. (4.1), some research on fault diagnosis for manufacturing systems has been investigated [35, 144]. All these methods assume that the number of measurements (M) is greater than that of process errors (M) (i.e., $N < M$). However, this assumption may not always hold in actual manufacturing applications, as using an excessive number of sensors (measurements) will result in undue costs [12]. These approaches are unsuitable if this assumption is violated. This is because Eq. (4.1) becomes an underdetermined system that results in the non-existence of a unique solution. To overcome the challenge, the sparse solution assumption [36] that $\mathbf{x}$ in Eq. (4.1) has a minimal number of nonzero elements is required. In the context of the fault diagnosis problem, it denotes the sparsity of process faults in the fault-quality linear model. This is reasonable since it is likely to have a few process faults in practice [9]. Among the several sparse estimation methods, the Bayesian method called sparse Bayesian learning has received much attention recently because of its superior estimation performance guaranteed from the several theoretical properties [19, 121, 142].

Several studies have used sparse Bayesian learning for fault diagnosis in manufacturing sys-

tems [9, 12, 75, 80]. These studies successfully identified process faults by providing prior distribution of process errors (i.e., $\mathbf{x}$ in Eq. (4.1)) to promote the sparsity of process faults. Especially, the work in [12] applied Bayesian learning to diagnose mean shift fault, which is the most relevant work to our study. In [12], multiple KPCs samples have been used to estimate the mean deviation of fixture locators. Specifically, given the average of multiple KPCs samples, which is $\mathbf{y}$ in Eq. (4.1), the mean deviation of the fixture locators (namely, $\mathbf{x}$ in Eq. (4.1)) can be estimated. However, this work does not consider the characteristics of multiple KPCs samples where they are collected sequentially. In practice, time series data of each process error from the multiple KPCs samples may have a strong temporal correlation in the multistation assembly process. For example, the locator position of the fixture system could have a drifting due to wear [144], causing dimensional quality issues in the samples. Specifically, [24, 61] investigated the effect of pin wear over time and dimensional variation in the multistage assembly process. If the process faults are not mitigated immediately, the effect of fixture deviation is auto-correlated in terms of time due to the degradation of wear of production tooling over time [115]. Similarly, the dimensional variability caused by machine-tool thermal distortions in the multistation processes is highly correlated between the samples assembled in a certain period [1]. Therefore, the temporal correlation of each process error causes product samples manufactured over a period of time to exhibit the same patterns of faults by a specific source of variation [85]. The relationship between the sequentially collected multiple KPCs samples and their process errors can be formulated as the following multiple measurements vectors (MMV) model [27] that extends from Eq. (4.1):

$$\mathbf{Y} = \Phi\mathbf{X} + \mathbf{V}, \tag{4.2}$$

where $\mathbf{Y} = [\mathbf{Y}_{\cdot 1}, ..., \mathbf{Y}_{\cdot L}] \in \mathbb{R}^{M\times L}$ is a measurement matrix consisting of L KPCs samples, and $\mathbf{Y}_{\cdot i} \in \mathbb{R}^{M\times 1}$ is a vector that denotes the i^{th} KPCs sample. $\mathbf{X} = [\mathbf{X}_{\cdot 1}, ..., \mathbf{X}_{\cdot L}] \in \mathbb{R}^{N\times L}$ is

a process error matrix, where $\mathbf{X}_{\cdot i} \in \mathbb{R}^{N\times 1}$ is a vector that represents process errors (KCCs) of the i^{th} KPCs sample. $\mathbf{V} \in \mathbb{R}^{M\times L}$ is a noise matrix. Since L KPCs samples have the same process faults, all columns of $\mathbf{X}$ share the index of nonzero rows called support. It is called a common support assumption in the MMV model [27]. In addition, L elements in the j^{th} row of $\mathbf{X}$, representing the time series data of the j^{th} process error, are highly correlated as nonzero values if the j^{th} process error is process fault. However, the dynamic changes of process faults due to the complexity of the manufacturing process can easily violate the common support assumption in a large number of KPCs samples. Therefore, this paper focuses on the small number of KPCs samples (L) to satisfy the common support assumption. Utilizing a small number of KPCs samples is also efficient in fault diagnosis of process faults in the multistation assembly process for time and cost reduction.

Beyond the common support assumption and temporal correlation in the time series data of each process error, utilizing prior knowledge of process faults is an additional way to improve the identification of sparse process faults [75]. Specifically, the prior knowledge regarding which process errors are more likely to be process faults than others. This knowledge can be obtained from domain-specific knowledge from practitioners or collected based on the fault diagnosis at the past time stamps. For example, the manufacturing engineers in an assembly line usually know that some fixture locators may malfunction more frequently than others based on their experiences. In practice, the prior knowledge provides only part of the actual process faults. In addition, the knowledge may contain some erroneous information as to process faults, which are actually not. Therefore, the prior knowledge provides partial and even erroneous information about process faults. However, utilizing the prior knowledge is expected to improve the identification of process faults if the correct and erroneous information can be properly distinguished.

In the sparse Bayesian learning literature, there exist studies that consider the temporal

correlation of rows in matrix $\mathbf{X}$ in Eq. (4.2) based on the common support assumption [53, 86], and the work that utilizes the partial with some erroneous prior knowledge of support to improve the performance of sparse estimation [40, 50], separately. However, these studies did not integrate these aspects to improve the sparse estimation.

To address this challenge, this paper aims to develop a novel sparse Bayesian hierarchical learning method that simultaneously utilizes the temporal correlation in the time series data of each process error, as well as prior knowledge, which may contain erroneous information. The new method, namely, support knowledge aided with temporally correlated process error sparse Bayesian learning (SA-TSBL), is proposed to achieve this objective. The contributions of this work are summarized as follows:

- From the methodological point of view, this paper proposes a novel sparse Bayesian learning that considers both temporal correlation in the time series data of each process error and prior knowledge of process faults to improve the sparse estimation. This method also derives an approximate posterior distribution of the sparse solution via Variational Bayes inference [103] to address the intractable computational challenge.
- From the application perspective, the proposed method is applied to fault diagnosis in the multistation assembly systems. The method mitigates the dimensional quality issues in the assembly operation by effectively identifying the fixture locators with excessive mean shifts. The effectiveness of the proposed method is validated in real-world simulation case studies that use an actual auto body assembly process.

The subsequent sections of this paper are structured in the following manner. The proposed methodology is presented in Section 4.2, followed by numerical case studies to validate its effectiveness in Section 4.3. Section 4.4 offers real-world case studies on fault diagnosis problems in the multistation assembly process. Finally, conclusions and future work are

discussed in Section 4.5.

4.2 Proposed Research Methodology

This section proposes a novel sparse Bayesian hierarchical method: support knowledge aided temporally correlated process error SBL (SA-TSBL). The proposed SA-TSBL is described in Section 4.2.1, followed by Bayesian inference of the proposed method in Section 4.2.2.

4.2.1 Proposed Methodology

The proposed methodology is a sparse Bayesian hierarchical model using multiple KPCs samples that have the same process faults. The method considers the correlation in the time series data of each process error, and utilizes prior knowledge of process faults to improve the sparse estimation. To exploit the temporal correlation in the time series data of each process error, the proposed method transforms the MMV model in Eq. (4.2) to the following block single measurement vector model [142], where Vec(A) is a vectorization of the matrix A, $\mathbf{I}_L$ is an Identity matrix with the size $L \times L$, and A $\otimes$ B is Kronecker product of the matrices A and B.

$$\mathbf{y} = \mathbf{D}\mathbf{x} + \mathbf{v}, \tag{4.3}$$

where $\mathbf{y}$ =Vec$(\mathbf{Y}^\top) \in \mathbb{R}^{ML\times 1}, \mathbf{D} = \Phi \otimes \mathbf{I}_L, \mathbf{x}$ =Vec$(\mathbf{X}^\top) \in \mathbb{R}^{NL\times 1}$. Assume noise vector $\mathbf{v}$ follows Gaussian distribution with zero mean and variance λ. Eq. (4.3) can be rewritten as

$$\mathbf{y} = [\phi_1 \otimes \mathbf{I}_L, ..., \phi_N \otimes \mathbf{I}_L][\mathrm{x}_1^\top, ..., \mathrm{x}_N^\top]^\top + \mathbf{v}, \tag{4.4}$$

where ϕ_i is the i^{th} column in Φ, and x_i consists of the i^{th} process errors of L KPCs samples (i.e., $\mathrm{x}_i^\top = (\tilde{x}_{i1}, \tilde{x}_{i2}, ..., \tilde{x}_{iL})$, where $\tilde{x}_{ij}$ denotes the element in the i^{th} row and the j^{th} column of matrix $\mathbf{X}$ in Eq. (4.2)). In other words, $\mathrm{x}_i \in \mathbb{R}^{L\times 1}$ is the i^{th} block of $\mathbf{x}$ in Eq. (4.2), as illustrated in Figure 4.1. Therefore, K nonzero rows in $\mathbf{X}$ in Eq. (4.2) are represented as

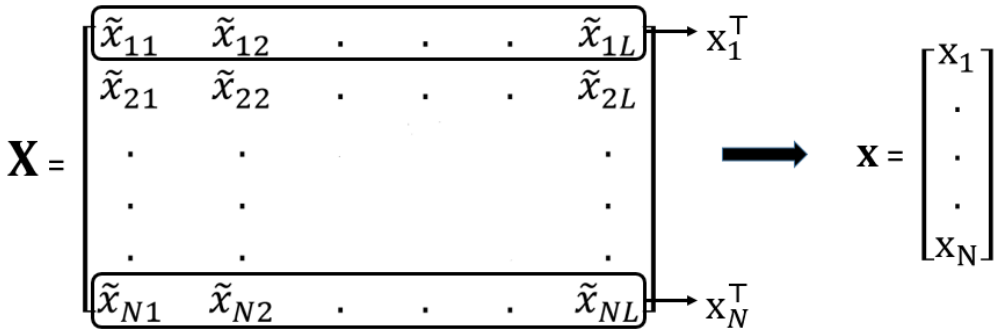


Figure 4.1: Block structure of $\mathbf{x}$.

K nonzero blocks in $\mathbf{x}$ in the proposed method. It implies that there exist K process errors that shifted from the design specification, namely, process faults. Since x_i consists of the time series data of i^{th} process error from L KPCs samples, the proposed method exploits the correlation between the L elements of x_i. Since noise vector $\mathbf{v}$ follows Gaussian distribution with zero mean and variance λ, the Gaussian likelihood is provided for the block model in Eq. (4.3) as follows:

$$p(\mathbf{y}|\mathbf{x};\lambda) \sim N(\mathbf{D}\mathbf{x}, \lambda\mathbf{I}_{ML}).$$

The proposed method consists of the following three layers. The prior distribution in the first layer is provided to exploit the correlation of the time series data of each KCC. The second and third layers consist of prior distribution representing the prior knowledge of process faults among process errors. A graphical representation of the proposed method is shown in Figure 4.2.

In the first layer of the hierarchical model in Figure 4.2, the Gaussian distribution is provided

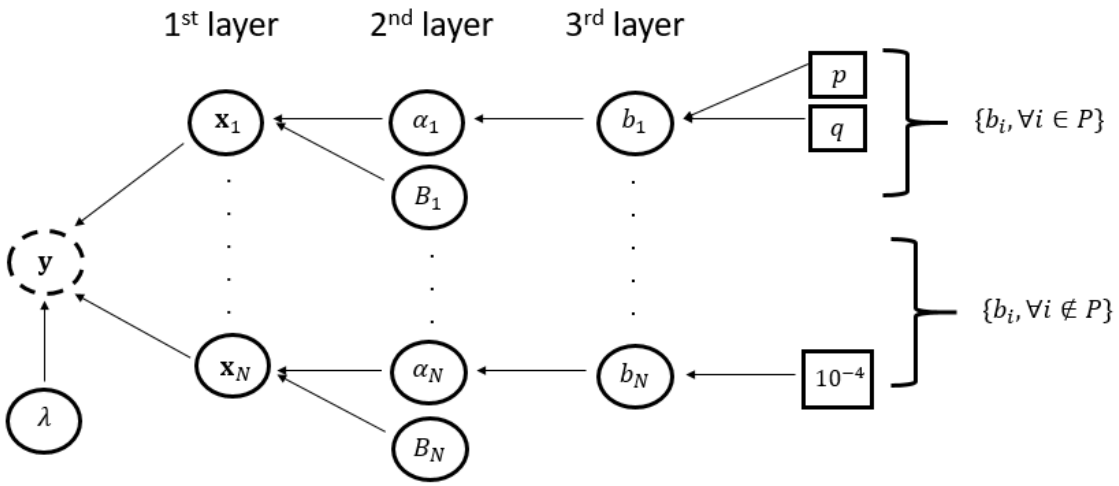


Figure 4.2: Graphical representation of the proposed method. A circle indicates a random variable or hyperparameter that needs to be estimated. A dashed circle and square represent an observation and a constant, respectively.

as the prior distribution for process error ($\mathbf{x}$). This is because the Gaussian distribution is widely used as prior distribution for process errors in the literature and practice [9, 75]. The prior distribution is given by

$$p(\mathbf{x}|\boldsymbol{\alpha}; \mathbf{B}) \sim N(\mathbf{0}, \Sigma_0),$$

where Σ_0 is

$$\Sigma_0 = \begin{bmatrix} \alpha_1^{-1}\mathrm{B}_1^{-1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \alpha_N^{-1}\mathrm{B}_N^{-1} \end{bmatrix},$$

$\boldsymbol{\alpha} = \{\alpha_1, ..., \alpha_N\}$, and $\mathbf{B} = \{\mathrm{B}_1, ..., \mathrm{B}_N\}$. Since x_i consists of the time series data of i^{th} process error from L KPCs samples, α_i^{-1} controls the sparsity of the i^{th} process error (x_i). For example, when α_i^{-1} converges to zero, the associated block x_i will be driven to zero [122]. B_i is a positive definite matrix that captures the correlation of the time series data of i^{th} process error. The proposed method assumes the independence between process errors (e.g., different fixture locators), which is a common assumption in the literature [10, 75, 80]. It

results in the block diagonal matrix Σ_0.

The second layer in Figure 4.2 specifies Gamma prior distributions over $\boldsymbol{\alpha}$. The distributions have an individual rate parameter b_i for each parameter α_i, namely,

$$p(\boldsymbol{\alpha}|\boldsymbol{b}) = \prod_{i=1}^{N} \text{Gamma}(\alpha_i|a, b_i) = \prod_{i=1}^{N} \Gamma(a)^{-1} b_i^a \alpha_i^{a-1} e^{-b_i \alpha_i}, \tag{4.5}$$

where $\boldsymbol{b} = \{b_1, ..., b_N\}$. Eq. (4.5) effectively integrates the prior information of process faults into the SBL framework. Basically, a and b_i are set to be very small values (e.g., 10^{-4}) to provide a large variance of a prior distribution over α_i in SBL [122]. It encourages the large values of α_i, and promotes the i^{th} process error as non-process faults (i.e., $\mathrm{x}_i = \mathbf{0}$ in Eq. (4.4)) [40]. Specifically, the marginal distribution of the time series data of i^{th} process error, namely, $p(\mathrm{x}_i) = \int p(\mathrm{x}_i|\alpha_i, \mathrm{B}) p(\alpha_i|a, b_i) \mathrm{d}\alpha_i$, follows multivariate student-t distribution which the probability is concentrated at zero [122]. Suppose set P consists of the indexes of process errors that are likely to be process faults based on prior knowledge. Then, the corresponding rate parameters $\{b_i, i \in P\}$ are set as a relatively large value (e.g., 1) to provide a small variance of the prior distribution over α_i [40]. The prior distribution allows the small value of α_i, and encourages the i^{th} process error as process faults (i.e., $\mathrm{x}_i \neq 0$ in Eq. (4.4)). However, assigning a fixed value to b_i in Eq. (4.5) has limitations to deal with the situation when the set P has some erroneous information of process faults.

To address this issue, the third layer in Figure 4.2 assigns a prior distribution over $\{b_i, i \in P\}$ [79]. Gamma distribution is assigned because of the support of the variable. In addition, the distribution enables users to derive the closed form of the approximate posterior distribution of $\{b_i, i \in P\}$, which will be described in the following section. The prior distribution over $\{b_i, i \in P\}$ is provided as follows:

$$\text{Gamma}(b_i|p,q) = \Gamma(p)^{-1} q^p b_i^{p-1} e^{-qb_i}, i \in P. \tag{4.6}$$

p and q in Eq. (4.6) are specified to characterize prior distribution. These two values in the proposed method can be adjusted by the users to satisfy the following two conditions as follows.

- First, the values are set to provide a large mean value of the prior distribution of $\{b_i, i \in P\}$.
- Second, the values are determined to have a large variance in the prior distribution of $\{b_i, i \in P\}$.

The first condition encourages the i^{th} process error as process faults based on prior knowledge. Compared to a small fixed value of $b_i = 10^{-4}$ that promotes the large α_i in previous research [40, 122], the large mean value of the prior distribution of $\{b_i, i \in P\}$ lets the small α_i. The small α_i provides a prior distribution that i^{th} process error is likely to be process faults of the mean shift.

The second condition is designed to deal with the case when erroneous prior knowledge is provided. Assume incorrect prior knowledge that the i^{th} process error may be a process fault is provided. If the variance of the prior distribution of $\{b_i, i \in P\}$ is small, the i^{th} process error is likely to be misdiagnosed as process faults because of the large mean value of the prior distribution of $\{b_i, i \in P\}$ from the first condition. However, if the variance of the prior distribution of $\{b_i, i \in P\}$ is large, $\{b_i, i \in P\}$ is not significantly affected by the prior distribution. Instead, $\{b_i, i \in P\}$ is highly affected by the data itself, where the i^{th} process error is not the process fault. This allows $\{b_i, i \in P\}$ to be learned as a small value from data and provides the sparsity prior to the i^{th} process error [40].

For $\{b_i, i \in P^c\}$ in the proposed method, the parameters are set to a fixed small value of

10^{-4} as in the previous study [122].

4.2.2 Bayesian Inference of the Proposed Methodology

The proposed method in Section 4.2.1 has several hidden variables that need to be estimated, that is, the process errors ($\mathbf{x}$), the variable controlling the sparsity of process errors ($\boldsymbol{\alpha}$), and the variable used to provide prior knowledge of process faults (b_i, $\forall i \in P$). In addition, there are hyperparameters related to temporal correlations in the time series data of i^{th} KCCs (B_i) and noise (λ), which also require estimation. To avoid too many parameters to being estimated, causing a challenging task in sparse estimation, the proposed method sets $\mathrm{B}_i = \mathrm{B}$ ($\forall i$) [142].

However, the posterior distribution of hidden variables in the proposed method (Eq. (4.7)) does not have a closed form because of the complexity of the proposed hierarchical model in Section 4.2.1. Specifically, the denominator in Eq. (4.7) cannot be calculated as closed form. Let b_i ($i \in P$) as $\bar{b}$ for convenience.

$$P(\mathbf{x}, \boldsymbol{\alpha}, \bar{b}|\mathbf{y}) = \frac{P(\mathbf{y}, \mathbf{x}, \boldsymbol{\alpha}, \bar{b})}{\int_{\mathbf{x},\boldsymbol{\alpha},\bar{b}} \cdots \int P(\mathbf{y}, \mathbf{x}, \boldsymbol{\alpha}, \bar{b})\, d\mathbf{x}\, d\boldsymbol{\alpha}\, d\bar{b}} \tag{4.7}$$

To overcome this challenge, this paper derives approximate posterior distributions of hidden variables via Variational Bayes inference (VBI). Specifically, Variational Bayes Expectation Maximization (VBEM) [103] is utilized to estimate hidden variables and hyperparameters in the proposed method to identify mean shifts process faults. VBEM consists of (1) E-step: Variational Bayesian expectation step to estimate hidden variables $\mathbf{x}, \boldsymbol{\alpha}, \bar{b}$ by approximating the posterior distribution of hidden variables; and (2) M-step: Variational Bayesian maximization step to update hyperparameters B and λ by maximizing the expected value of the

logarithm of the complete likelihood.

Let $\boldsymbol{\theta}$ be a vector with all hidden variables in the proposed method (i.e., $\boldsymbol{\theta} = (\mathbf{x}, \boldsymbol{\alpha}, \bar{b})$). VBI approximates the posterior distribution of $\boldsymbol{\theta}$, denoted as $q(\boldsymbol{\theta})$, by minimizing Kullback-Leibler (KL) divergence between $q(\boldsymbol{\theta})$ and the true posterior distribution, namely, $p(\boldsymbol{\theta}|\boldsymbol{y})$ (i.e., $D_{KL}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\boldsymbol{y}))$). $q(\boldsymbol{\theta})$ is factorized as

$$q(\boldsymbol{\theta}) = q(\mathbf{x})q(\boldsymbol{\alpha})q(\bar{b})$$

by the mean-field approximation [26], that all the variables are independent. The approximate posterior distribution $q(\theta_i)$, where θ_i is the i^{th} element in the set $\boldsymbol{\theta}$ is derived as follows by minimizing the $D_{KL}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\boldsymbol{y})$ under the mean-field approximation.

$$\ln q(\theta_i) = \mathbb{E}[\ln p(\boldsymbol{\theta}, \boldsymbol{y})]_{\boldsymbol{\theta}\setminus\theta_i} + const, \tag{4.8}$$

where $\mathbb{E}_{\boldsymbol{\theta}\setminus\theta_i}$ denotes the expectation taken with the set $\boldsymbol{\theta}$ without θ_i. const can be obtained through normalization. The detailed derivation of Eq. (4.8) is provided in Appendix B.1. Eq. (4.8) is used in the following E-step of VBEM to approximate the posterior distributions of hidden variables.

E-step of VBEM: The posterior distributions of hidden variables that are related to process errors ($\mathbf{x}$), sparsity of process errors ($\boldsymbol{\alpha}$), and prior knowledge of process faults ($\bar{b}$) are approximated by Eq. (4.8), respectively, as follows.

$$\begin{aligned}\ln q(\mathbf{x}) &= \mathbb{E}[\ln p(\mathbf{y}, \mathbf{x}, \boldsymbol{\alpha}, \bar{b})]_{q(\boldsymbol{\alpha})q(\bar{b})} + const \\ &= \mathbb{E}[\ln p(\mathbf{y}|\mathbf{x}; \lambda)p(\mathbf{x}|\boldsymbol{\alpha}; \mathbf{B})]_{q(\boldsymbol{\alpha})} + const,\end{aligned} \tag{4.9}$$

$$\begin{aligned}\ln q(\boldsymbol{\alpha}) &= \mathbb{E}[\ln p(\mathbf{y}, \mathbf{x}, \boldsymbol{\alpha}, \bar{b})]_{q(\mathbf{x})q(\bar{b})} + const \\ &= \mathbb{E}[\ln p(\mathbf{x}|\boldsymbol{\alpha}; \mathbf{B}) p(\boldsymbol{\alpha}|\bar{b})]_{q(\boldsymbol{\alpha})} + const, \end{aligned} \tag{4.10}$$

$$\begin{aligned}\ln q(\bar{b}) &= \mathbb{E}[\ln p(\mathbf{y}, \mathbf{x}, \boldsymbol{\alpha}, \bar{b})]_{q(\mathbf{x})q(\boldsymbol{\alpha})} + const \\ &= \mathbb{E}[\ln p(\boldsymbol{\alpha}|\bar{b}) p(\bar{b})]_{q(\boldsymbol{\alpha})} + const. \end{aligned} \tag{4.11}$$

Based on the statistical inference, the posterior distributions of hidden variables can be derived as

$$q(\mathbf{x}) = N(\mathbf{x}|\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}}), \tag{4.12}$$

$$q(\boldsymbol{\alpha}) = \prod_{i=1}^{N} \text{Gamma}(\alpha_i|\tilde{a}, \tilde{b}_i), \tag{4.13}$$

$$q(\bar{b}) = \prod_{i \in P} \text{Gamma}(b_i|p + a, \mathbb{E}[\alpha_i] + q). \tag{4.14}$$

Based on the mean-field approximation, $q(\boldsymbol{\theta})$ is the multiplication of Gaussian and Gamma distributions from Eqs. (4.12), (4.13), and (4.14). Let $\text{diag}\{\alpha_1, ..., \alpha_m\}$ is a diagonal matrix with principal diagonal elements being $\alpha_1, ..., \alpha_m$ and Tr(A) is a trace of matrix A. Then, the expectations and moments of distributions in Eqs. (4.12), (4.13), and (4.14) are

$$\mu_{\mathbf{x}} = \frac{1}{\lambda} \Sigma_{\mathbf{x}} \mathbf{D}^{\top} \mathbf{y}, \tag{4.15}$$

$$\mathbb{E}[\alpha_i] = \frac{a + \frac{L}{2}}{\frac{\text{Tr}[(\Sigma_{\mathbf{x}_i} + \mathbb{E}[\mathbf{x}_i]\mathbb{E}[\mathbf{x}_i]^{\top})\text{B}]}{2} + \mathbb{E}[b_i]}, \ (\forall i), \tag{4.16}$$

$$\mathbb{E}[b_i] = \frac{p + a}{q + \mathbb{E}[a_i]}, \ (i \in P), \tag{4.17}$$

$$\Sigma_{\mathbf{x}} = (\frac{1}{\lambda} \mathbf{D}^{\top} \mathbf{D} + \mathbb{E}[\text{AB}])^{-1}, \tag{4.18}$$

where $\mathbb{E}[\mathrm{AB}] = \mathrm{diag}[\mathbb{E}[\alpha_1]\mathrm{B}, ..., \mathbb{E}[\alpha_N]\mathrm{B}]$. The estimator of the time series data of i^{th} process error ($\mathbb{E}[\mathrm{x}_i]$) can be obtained from $\mu_{\mathbf{x}}$ in Eq. (4.15) as follows.

$$\mathbb{E}[\mathrm{x}_i] = \mu_{\mathbf{x}}((i-1)L + 1 : iL), i = 1, ..., N. \tag{4.19}$$

Detailed derivations of Eqs. (4.12), (4.13), and (4.14) are provided in the Appendices B.2, B.3, and B.4.

M-step of VBEM: Temporal correlations in the time series data of each KCC (B) and noise (λ) are estimated in this step. Let $\tilde{\boldsymbol{\theta}} = \{\mathrm{B}, \lambda\}$. Posterior distributions of $\mathbf{x}, \boldsymbol{\alpha}, \bar{b}$ obtained in Eqs. (4.12), (4.13), and (4.14) are denoted as $q(\mathbf{x}; \tilde{\boldsymbol{\theta}}^{OLD}), q(\boldsymbol{\alpha}; \tilde{\boldsymbol{\theta}}^{OLD})$, and $q(\bar{b}; \tilde{\boldsymbol{\theta}}^{OLD})$ respectively. $\tilde{\boldsymbol{\theta}}$ can be updated by maximizing the complete likelihood as follows:

$$\begin{aligned}\tilde{\boldsymbol{\theta}}^{NEW} &= \underset{\tilde{\boldsymbol{\theta}}}{\mathrm{argmax}}\, \mathbb{E}[\ln p(\mathbf{y}, \mathbf{x}, \boldsymbol{\alpha}, \bar{b}; \tilde{\boldsymbol{\theta}})]_{q(\mathbf{x};\tilde{\boldsymbol{\theta}}^{OLD})q(\boldsymbol{\alpha};\tilde{\boldsymbol{\theta}}^{OLD})q(\bar{b};\tilde{\boldsymbol{\theta}}^{OLD})} \\ &= \underset{\tilde{\boldsymbol{\theta}}}{\mathrm{argmax}}\, \mathbb{E}[\ln p(\mathbf{y}|\mathbf{x}; \tilde{\boldsymbol{\theta}}) p(\mathbf{x}|\boldsymbol{\alpha}; \tilde{\boldsymbol{\theta}})]_{q(\mathbf{x};\tilde{\boldsymbol{\theta}}^{OLD})q(\boldsymbol{\alpha};\tilde{\boldsymbol{\theta}}^{OLD})}.\end{aligned}$$

Let $Q(\tilde{\boldsymbol{\theta}}) = \mathbb{E}[\ln p(\mathbf{y}|\mathbf{x}; \tilde{\boldsymbol{\theta}}) p(\mathbf{x}|\boldsymbol{\alpha}; \tilde{\boldsymbol{\theta}})]_{q(\mathbf{x};\tilde{\boldsymbol{\theta}}^{OLD})q(\boldsymbol{\alpha};\tilde{\boldsymbol{\theta}}^{OLD})}$, which results in

$$\begin{aligned}Q(B, \lambda) &= \mathbb{E}[\ln p(\mathbf{y}|\mathbf{x}; \lambda)]_{q(\mathbf{x};\tilde{\boldsymbol{\theta}}^{OLD})q(\boldsymbol{\alpha};\tilde{\boldsymbol{\theta}}^{OLD})} + \mathbb{E}[\ln p(\mathbf{x}|\boldsymbol{\alpha}; \mathrm{B})]_{q(\mathbf{x};\tilde{\boldsymbol{\theta}}^{OLD})q(\boldsymbol{\alpha};\tilde{\boldsymbol{\theta}}^{OLD})} \\ &= \mathbb{E}[\ln p(\mathbf{y}|\mathbf{x}; \lambda)]_{q(\mathbf{x};\tilde{\boldsymbol{\theta}}^{OLD})} + \mathbb{E}[\ln p(\mathbf{x}|\boldsymbol{\alpha}; \mathrm{B})]_{q(\mathbf{x};\tilde{\boldsymbol{\theta}}^{OLD})q(\boldsymbol{\alpha};\tilde{\boldsymbol{\theta}}^{OLD})}.\end{aligned} \tag{4.20}$$

B and λ are estimated as Eqs. (4.21) and (4.22), respectively, by maximizing the Eq. (4.20).

$$\mathrm{B} = [\frac{1}{N}\sum_{i=1}^{N} \mathbb{E}[\alpha_i](\Sigma_{\mathrm{x}_i} + \mathbb{E}[\mathrm{x}_i]\mathbb{E}[\mathrm{x}_i]^\top)]^{-1}. \tag{4.21}$$

$$\lambda = \frac{[\|\mathbf{y} - \mathbf{D}\mu_{\mathbf{x}}\|_2^2 + \hat{\lambda}[NL - \mathrm{Tr}(\Sigma_{\mathbf{x}}\mathbb{E}_{q(\boldsymbol{\alpha};\tilde{\boldsymbol{\theta}}^{OLD})}(\Sigma_0^{-1}))]]}{ML}, \tag{4.22}$$

where $\|\mathbf{x}\|_p$ denotes ℓ_p norm of the vector $\mathbf{x}$. Detailed derivations of Eqs. (4.21) and (4.22) are described in the Appendices B.5 and B.6.

Algorithm 4.1 shows the procedure of the proposed SA-TSBL method. Given the multiple KPCs samples ($\mathbf{y}$) and fault pattern matrix (Φ), the proposed method estimates the following variables and parameters in E and M steps, respectively.

- E-step: Process errors ($\mu_{\mathbf{x}}$), the variable related to the sparsity of process errors ($\boldsymbol{\alpha}$), and the variable that is used to provide prior knowledge of process faults ($\bar{b}$).

- M-step: Temporal correlations in the time series data of each KCC (B) and noise (λ).

These steps iterate until the estimator of process errors ($\mu_{\mathbf{x}}$) is rarely updated, namely, $\|\mu_{\mathbf{x}}^{t-1} - \mu_{\mathbf{x}}^{t}\|_\infty < \gamma$, where $\|\cdot\|_\infty$ indicates infinity norm and γ is a user-defined threshold (e.g., $\gamma = 10^{-6}$) or it reaches the maximum number of iterations ($\mathbf{T}$). Then, the mean deviation of process errors, which is the output of Algorithm 4.1, are calculated by the following procedure.

- Step 1: The matrix $\tilde{\mu}_{\mathbf{x}}$ is defined in which the i^{th} row represents Eq. (4.19) derived from $\mu_{\mathbf{x}}$.

- Step 2: The average of each row of matrix $\tilde{\mu}_{\mathbf{x}}$, that is, a vector $\bar{\tilde{\mu}}_{\mathbf{x}}$ is provided as the output of Algorithm 4.1, indicating the mean deviation of process errors.

Therefore, nonzero values in a vector $\bar{\tilde{\mu}}_{\mathbf{x}}$ are process faults of the mean shifts.

Algorithm 4.1 Proposed SA-TSBL method

Input: Multiple KPCs samples ($\mathbf{y}$), Fault pattern matrix (Φ).
Set $a = b_i\ (i \in P^c) = 10^{-4}$. p and q are set based on two conditions in Section 4.2.1.
Initialize $\mathrm{B} = \mathbf{I}_L, \lambda = 1, b_i\ (i \in P) = 1, , t = 1$.
While $\|\mu_{\mathbf{x}}^{t-1} - \mu_{\mathbf{x}}^{t}\|_\infty \geq \gamma$ **or** $t \leq \mathbf{T}$ **do**
    E-step of VBEM:
        Update $\mu_{\mathbf{x}}$ using Eq. (4.15)
        Update $\boldsymbol{\alpha}$ using Eq. (4.16)
        Update $\bar{b}$ using Eq. (4.17)
    M-step of VBEM:
        Update B using Eq. (4.21)
        Update λ using Eq. (4.22)
    $t = t + 1$
End
Output: Mean deviations of process errors $\bar{\tilde{\mu}}_{\mathbf{x}}$.

4.3 Numerical Case Studies

This section provides three scenarios to compare the performance between the proposed method and benchmark methods.

- Section 4.3.1 shows the performance evaluation by varying the temporal correlation. This study is to validate the effectiveness of strong temporal correlation in sparse estimation.
- Section 4.3.2 provides the numerical study to investigate the impact of the number of KPCs samples (i.e., measurement samples) on the performance of process faults identification.
- Section 4.3.3 illustrates the sparse estimation performance by varying the ratio between the number of measurements and process errors (i.e., the severity of the underdetermined systems).

All the numerical case studies consist of 100 independent trials. p and q in Eq. (4.7) are de-

termined as 1 and 0.1, respectively, in the case studies to satisfy the two conditions provided in Section 4.2.1. The code of the proposed SA-TSBL algorithm is implemented in Matlab 2021a. The CPU processor employed in this research is Intel® Xeon® E-2286M 2.40 GHz Processor.

The benchmark methods selected in this study are as follows, which are widely used in sparse Bayesian learning.

- MSBL proposed in [133] is a basic SBL method for the MMV model that assumes independence in the time series data of each KCC.
- T-MSBL proposed in [142] is a typical SBL method for the MMV model that considers temporal correlation in the time series data of each KCC.
- SA-MSBL proposed in [137] is the SBL method that considers prior knowledge of support in the MMV model. It assumes independence in the time series data of each KCC.
- SCBL proposed in [12] is the SBL method that considers the spatial correlation of process error (i.e., fixture locators) in the single measurement vector (SMV) model. The average of multiple KPCs samples is used to estimate the mean deviation of process errors.
- SA-SBL proposed in [40] and [79] is the SBL method that considers prior knowledge of process faults in the SMV model.
- [75] proposed the SBL method that considers prior knowledge of the support in the MMV model. It assumes independence in the time series data of each KCC and uses Gibbs sampling for the Bayesian inference. The total number of iterations and burn-in period are referred from [75].

Data Generations: The data generation process is summarized in Figure 4.3. The solution matrix $\mathbf{X}_{true} \in \mathbb{R}^{N\times L}$ is randomly generated with K nonzero rows. Indexes of the nonzero rows are randomly chosen in each trial. The nonzero rows in $\mathbf{X}_{true}$ are generated as AR(1) process that initiates from the standard Gaussian distribution since AR(1) processes are sufficient to represent the temporal structure of the small number of measurement samples (L) [142]. The AR coefficient, defined as β, represents the temporal correlation. A dictionary matrix $\Phi \in \mathbb{R}^{M\times N}$ is constructed with columns drawn from the surface of a unit hyper-sphere uniformly [36]. Finally, the measurements matrix is built by $\mathbf{Y} = \Phi\mathbf{X}_{true} + \mathbf{V}$ in the final step of Figure 4.3, where $\mathbf{V}$ is a Gaussian noise matrix with zero-mean [142]. The variance of the noise matrix is chosen to meet the determined value of the signal-to-noise ratio (SNR). SNR is defined by SNR(dB):$= 20(\log_{10}(\|\Phi\mathbf{X}_{true}\|_F/\|\mathbf{V}\|_F))$ [142].

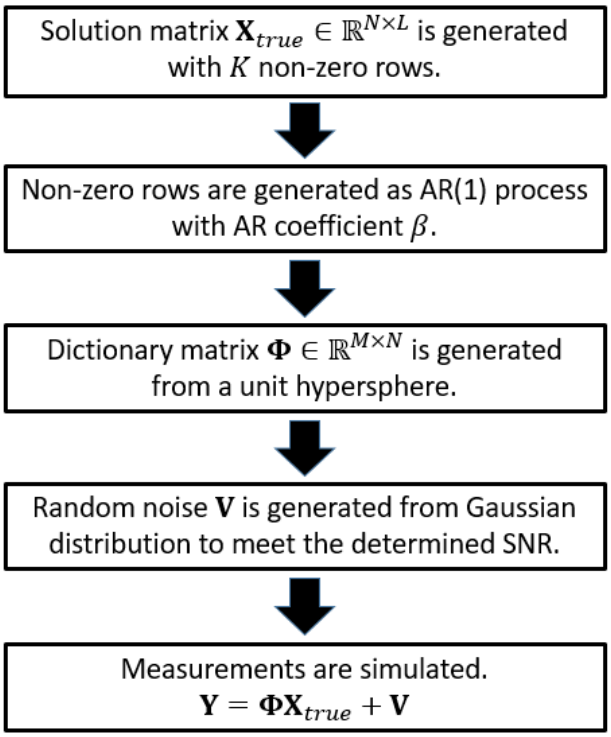


Figure 4.3: Flowchart of the data generation process.

Performance Evaluation: Since the objective of this paper is to diagnose the mean devi-

ation of process errors, the target of the proposed method is to accurately estimate $\bar{\mathbf{X}}_{true}$, which is the row-wise mean of $\mathbf{X}_{true}$.

Two performance measures are used in this paper. One is the failure rate defined in [79]. It measures the accuracy of detecting process fault, which are the nonzero rows in $\bar{\mathbf{X}}_{true}$. Assume that the number of process faults K is given. Then, the row indexes of the K largest ℓ_2-norms from $\bar{\tilde{\mu}}_{\mathbf{x}}$ in Algorithm 4.1 are identified. If the indexes are different from the indexes of nonzero rows in $\bar{\mathbf{X}}_{true}$, it is considered a failed trial. The failure rate is the percentage of failed trials in the total trials. The other performance measure is a normalized mean squared error (NMSE) that is defined by $\|\bar{\tilde{\mu}}_{\mathbf{x}} - \bar{\mathbf{X}}_{true}\|_F^2 / \|\bar{\mathbf{X}}_{true}\|_F^2$, where $\|\mathrm{A}\|_F$ is a Frobenius norm of the matrix A. Averages of failure rate and NMSE from 100 trials are used as performance measures. In addition, the standard deviations and quantile statistics of the two measures are provided in Appendix B.7.

Remark 4.1. (Performance Evaluations of SA-TSBL, SA-MSBL, SA-SBL, and [75])

The SA-MSBL, SA-SBL, [75], and the proposed SA-TSBL have several cases based on the set of prior knowledge (i.e., P in Eq. (4.6)) even in the same problem. Both correct and erroneous information of support exists in the set P as prior knowledge. To differentiate between the correct and erroneous information of support, two subsets, namely, P_C and P_E are defined. P_C consists of the indexes of nonzero rows obtained from prior knowledge among true nonzero rows in $\bar{\mathbf{X}}_{true}$. In contrast, P_E consists of the indexes of nonzero rows in prior knowledge but are actually zero in $\bar{\mathbf{X}}_{true}$. The cardinalities of P_C and P_E in this paper are assumed to satisfy two conditions. The first condition is the cardinality of set P_C and P_E should be less than or equal to 75% and 50% of the number of nonzero rows in $\bar{\mathbf{X}}_{true}$ (i.e., K), respectively. For example, if K is 4, the cardinality of P_C and P_E are less than or equal to 4 and 3, respectively. The first condition illustrates the cardinality of prior knowledge is

similar to K. The condition enables the performance evaluation of the proposed method in comprehensive situations. Since the cardinality of P_C is less than or equal to 4, the prior knowledge is partial (i.e., P_C misses some rows that are actually nonzero in $\bar{\mathbf{X}}_{true}$) in all cases. In addition, the cases with a cardinality of P_E greater than zero show the situations when the erroneous prior knowledge (i.e., P_E contains the rows that are actually zero in $\bar{\mathbf{X}}_{true}$) exist. The second condition is the cardinality of P_C is greater than or equal to that of P_E. The second condition prevents erroneous prior knowledge from being dominant prior knowledge. Therefore, performances of these three methods are evaluated as the average of all cases with different cardinalities of P_C and P_E, where each case consists of 100 trials.

4.3.1 Performance Evaluation in Various Temporal Correlations

This case study shows the performance of all methods in various temporal correlations when SNR is 100dB. The size of the dictionary matrix Φ is 5×40. The number of nonzero rows of $\bar{\mathbf{X}}_{true}$ (K), and measurement samples (L) are 4 and 4, respectively. Temporal correlation (β) varies among 0.0, 0.3, 0.6, 0.9, and 0.95. The proposed method shows the best performance in all temporal correlations, as shown in Table 4.1.

Table 4.1: Performance comparison with various temporal correlations (β).

	Failure Rate					NMSE				
β	0.0	0.3	0.6	0.9	0.95	0.0	0.3	0.6	0.9	0.95
T-MSBL	0.38	0.34	0.36	0.25	0.20	0.81	0.73	0.70	0.52	0.32
MSBL	0.64	0.68	0.74	0.95	0.96	0.80	0.86	0.92	1.27	1.32
SA-MSBL	0.96	0.96	0.96	0.97	0.96	0.71	0.71	0.75	0.76	0.75
SA-SBL	0.97	0.97	0.97	0.97	0.96	0.73	0.73	0.76	0.77	0.76
[75]	0.93	0.93	0.95	0.99	0.99	0.70	0.69	0.74	0.87	0.95
SCBL	0.94	0.96	0.96	0.99	1.0	1.24	1.31	1.35	1.42	1.45
SA-TSBL (Proposed)	**0.23**	**0.24**	**0.23**	**0.18**	**0.14**	**0.46**	**0.48**	**0.41**	**0.34**	**0.24**

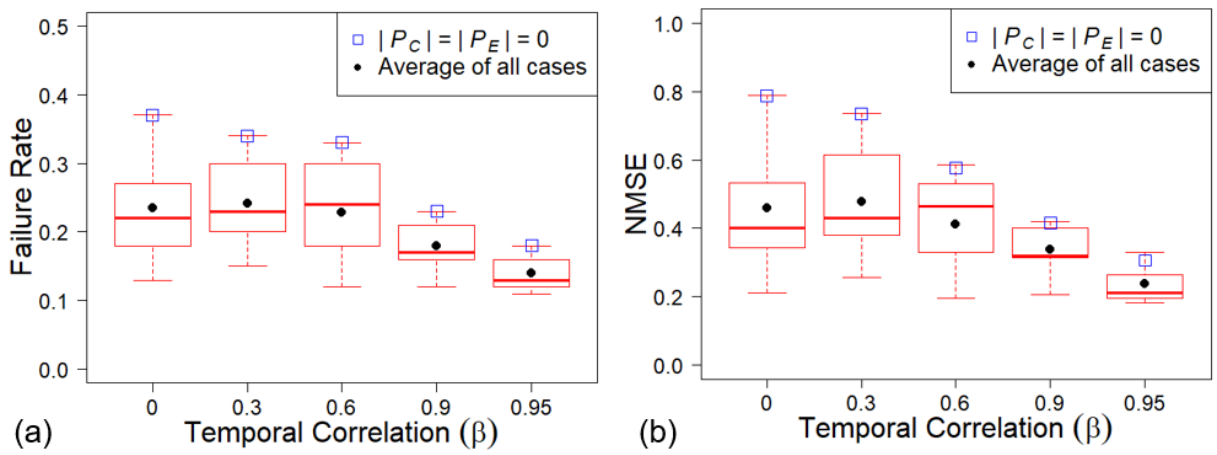


Figure 4.4: The boxplot for (a) failure rate and (b) NMSE in all cases of SA-TSBL in various temporal correlations.

Specifically, the failure rate and NMSE of the proposed method and T-MSBL tend to decrease as β increases since these two methods capture the temporal correlation. In contrast, both measures of MSBL, SA-MSBL, and [75] show the opposite trend as the three methods assume independence among samples.

The performance evaluations of the proposed method with different cardinalities of P_C and P_E are represented as boxplots in Figure 4.4. Since the number of nonzero rows is 4, each correlation has 9 cases designed under the two conditions of Remark 4.1. In each boxplot, the horizontal line represents the minimum, the first quartile, median, the third quartile, and maximum value sequentially from the bottom, respectively. The minimum value denotes the smallest value within the 1.5 $\times$ interquartile range (IQR) below the first quartile. Similarly, the maximum value is defined as the largest value that is within the 1.5 $\times$ IQR above the third quartile. A blue square in Figure 4.4 shows the performance of the proposed method without any prior knowledge (i.e., $|P_C| = |P_E| = 0$), and the black dot represents the average of all cases used as a performance evaluation measure of the proposed method. The trend of black dots shows the performance improvements of the proposed method as temporal

correlation (β) increases. In addition, boxplots in Figure 4.4 represent most cases with partial and some erroneous prior knowledge achieve better performance in both measures than those without prior knowledge. The results show the effectiveness of prior knowledge in the proposed method, which successfully distinguishes the correct and incorrect prior knowledge in various temporal correlations.

4.3.2 Performance Evaluation in the Various Numbers of Measurement Samples

This case study's aim is to compare the performance of all methods using the various numbers of measurement samples (L). The size of the dictionary matrix Φ is 7×55, and the number of nonzero rows in $\bar{\mathbf{X}}_{true}$ (K) is 4. SNR and temporal correlation (β) are set as 35dB, and 0.95, respectively. The number of measurement samples varies from 2 to 4. Table 4.2 shows that the proposed method achieves the best performance in all cases. In addition, the results

Table 4.2: Performance comparison with various number of measurement samples (L).

	Failure Rate			NMSE		
L	2	3	4	2	3	4
T-MSBL	0.59	0.17	0.13	0.76	0.19	0.14
MSBL	0.83	0.77	0.52	1.10	1.04	0.64
SA-MSBL	0.88	0.90	0.90	0.68	0.67	0.66
SA-SBL	0.89	0.90	0.90	0.70	0.67	0.70
[75]	0.71	0.50	0.49	0.63	0.52	0.56
SCBL	0.97	0.92	0.93	1.25	1.29	1.17
SA-TSBL (Proposed)	**0.45**	**0.12**	**0.11**	**0.52**	**0.13**	**0.09**

in Table 4.2 illustrate that most of the MMV models achieve performance improvements as the number of measurement samples increases because of common support assumption

[27]. Figure 4.5 shows the performance evaluation of the proposed method with various sizes

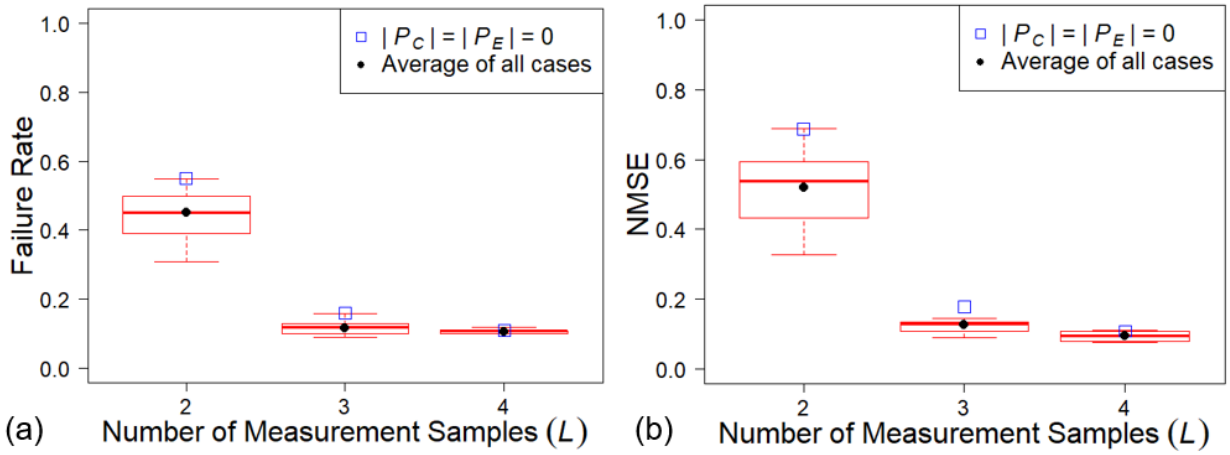


Figure 4.5: The boxplot for (a) failure rate and (b) NMSE in all cases of SA-TSBL in various numbers of measurement samples.

of P_C and P_E. Especially, the performance of T-MSBL and the proposed method without prior knowledge (blue square in Figure 4.5) is similar. However, the proposed achieves better performance than T-MSBL by utilizing partial and some erroneous prior knowledge. Specifically, the performance of the proposed method improves from 3% to 32% of those without any prior knowledge (an average of relative improvement from square to dots in Figure 4.5) in the various numbers of measurement samples.

4.3.3 Performance Evaluation in Various Ratios between the Number of Measurements and Process Errors

Results in Table 4.3 illustrate the performance of all methods by varying the ratio between the number of measurements and process errors (i.e., underdetermined ratio). In this study, the number of measurements (M) is fixed at 10, and the underdetermined ratio (N/M) is selected from 3, 5, 7, and 9 with 25dB for SNR, respectively. The number of nonzero rows in $\bar{\mathbf{X}}_{true}$ (K) and measurement samples (L) are set as 4, 3, respectively. The temporal

Table 4.3: Performance comparison with various underdetermined ratio (N/M).

	Failure Rate				NMSE			
N/M	3	5	7	9	3	5	7	9
T-MSBL	0.09	0.28	0.26	0.31	0.05	0.32	0.33	0.42
MSBL	0.29	0.56	0.64	0.69	0.30	0.61	0.77	0.86
SA-MSBL	0.31	0.55	0.64	0.74	0.24	0.43	0.49	0.56
SA-SBL	0.59	0.78	0.82	0.85	0.37	0.53	0.57	0.60
[75]	0.11	0.22	0.43	0.52	0.10	0.24	0.55	0.68
SCBL	0.45	0.79	0.83	0.92	0.48	0.88	1.01	1.15
SA-TSBL (Proposed)	**0.06**	**0.17**	**0.19**	**0.20**	**0.02**	**0.17**	**0.23**	**0.23**

correlation (β) is 0.99. As the underdetermined ratio increases, it becomes more challenging to identify the sparse process faults. However, capturing temporal correlation and utilizing the partial and some erroneous prior knowledge enable the proposed method to achieve the best performance even in a high underdetermined ratio. Figure 4.6 shows prior knowledge of support is still valuable in various underdetermined ratios. This study shows that the proposed algorithm can be applied to applications such as neuroimaging in that highly underdetermined systems exist.

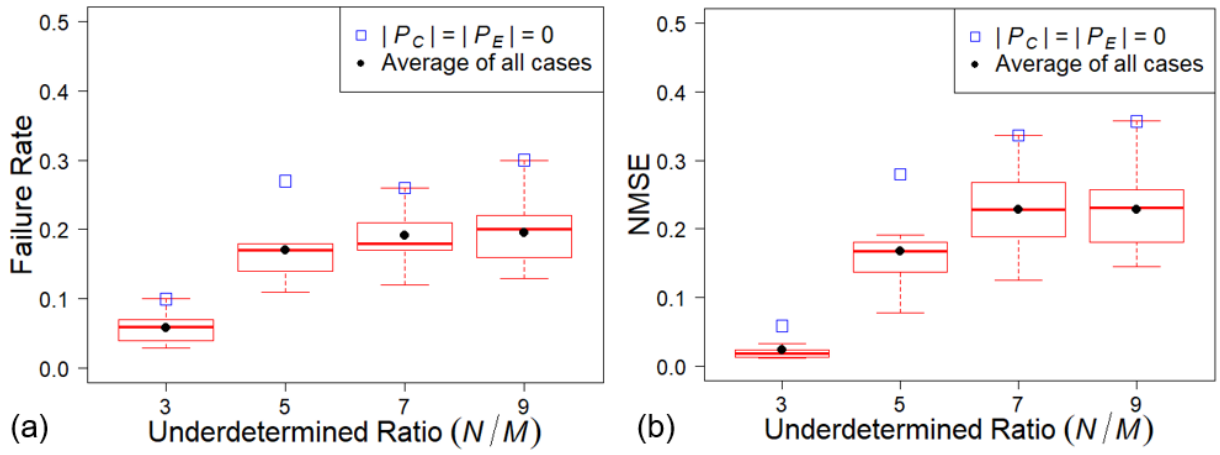


Figure 4.6: The boxplot for (a) failure rate and (b) NMSE in all cases of SA-TSBL in various underdetermined ratio.

4.4 Real-World Simulation Case Studies

An assembly operation from an actual auto-body assembly process is used as a real-world case study. Figure 4.7 describes the floor pan of a car which is the assembled product from this process. The assembled product consists of four parts, including the right bracket, left bracket, right floor pan, and left floor pan. Figure 4.8 shows the process assembly procedure

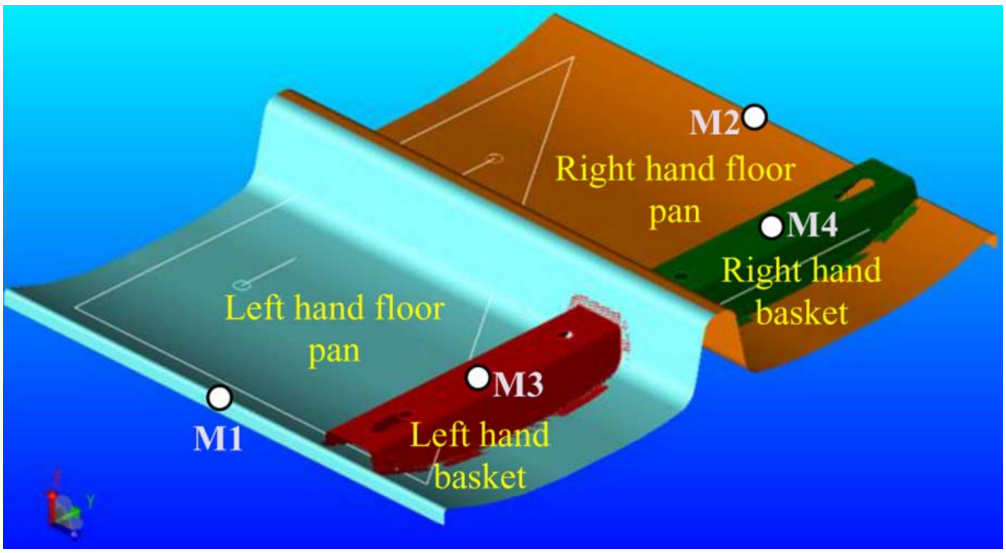


Figure 4.7: Floor-pan assembly model [9].

consisting of three stations. During the assembly process, the parts are held by fixtures, which are the KCCs in this process [10]. KPCs are measured from four points, namely, M1, M2, M3, and M4, respectively, as shown in Figure 4.7. Every part has a designated location for measuring. For instance, part 1 has M1, part 2 has M2, part 3 has M3, and part 4 has M4. These measurements can be taken at each station once the relevant part has been assembled in the preceding stations. For example, M4 on part 4 cannot be measured in station 1 because part 4 has not yet been assembled at station 1 [10]. In addition, KPCs are measured in three directions (X, Y, and Z) at each point. In this assembly process, thirty-three process errors, which are fixture locator dimensional errors, exist [12]. The

dimension of the fault pattern matrix is 12×33, and is established based on the literature [10, 58, 70]. Since the variations of KCCs can be propagated to deviations of other KCCs in the subsequent stations, the transfer and accumulation of KCCs deviations between multiple stations are considered in the formulation of the fault pattern matrix Φ [10, 34]. The matrix is provided in Appendix B.8. Since the number of measurements (12) is less than the number of process errors (33), it causes an underdetermined system in the fault quality linear model. It requires sparse estimation to identify process faults.

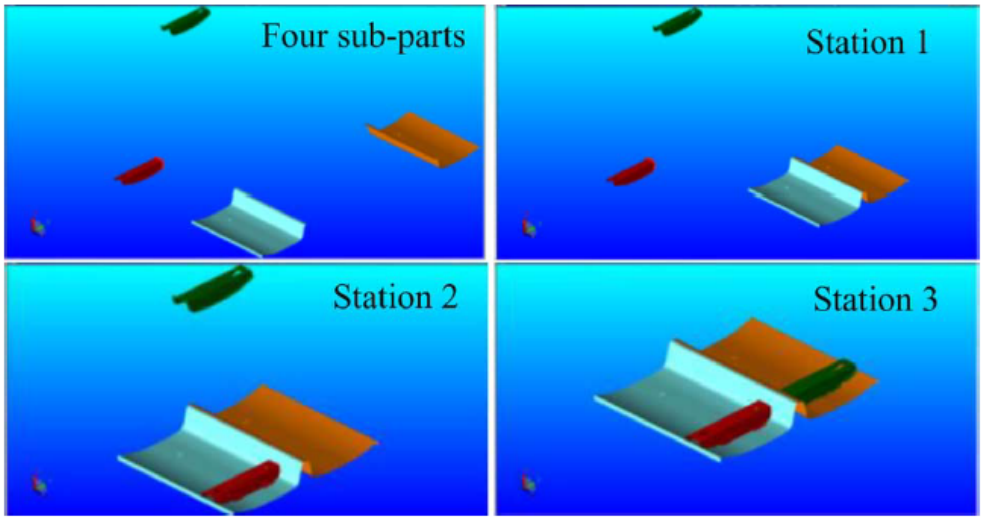


Figure 4.8: Floor-pan assembly procedure from three assembly stations [9].

To generate the multiple KPCs samples from the auto body assembly process, the time series data of each process fault is generated from AR (1) process initiated from the standard Gaussian distribution as in Section 4.3. The generated temporal correlated process faults ($\mathbf{X}_{true}$) and fault pattern matrix provide the multiple KPCs samples ($\mathbf{Y}$), as in Figure 4.3. In addition, performance evaluation measures, benchmark methods, and the values of p and q in Eq. (4.6) used in Section 4.3 are still utilized in Section 4.4. Prior knowledge of process faults is also provided in the same way as Section 4.3. Sections 4.4.1 and 4.4.2 show the performance evaluation by varying the number of KPCs samples, and temporal correlation

β, respectively.

4.4.1 Performance Evaluation in the Various Number of KPCs Samples

This case study aims to demonstrate the effectiveness of sparse estimation of process faults by varying the number of KPCs samples (L) when a strong correlation exists (β=0.99). The number of KPCs samples varies from 2 to 4, and SNR is 50dB. Three process faults (K) are determined among 33 process errors randomly. The proposed method shows the best performance in all various numbers of KPCs samples, as shown in Table 4.4. All methods

Table 4.4: Performance comparison by varying the number of KPCs samples.

	Failure Rate			NMSE		
KPCs samples	2	3	4	2	3	4
T-MSBL	0.60	0.64	0.47	0.34	0.32	0.25
MSBL	0.66	0.62	0.49	0.36	0.33	0.24
SA-MSBL	0.50	0.56	0.48	0.35	0.36	0.31
SA-SBL	0.54	0.60	0.54	0.37	0.38	0.36
[75]	0.92	0.94	0.96	3.80	3.09	3.72
SCBL	0.65	0.71	0.49	0.42	0.40	0.30
SA-TSBL (Proposed)	**0.42**	**0.49**	**0.35**	**0.28**	**0.29**	**0.20**

except SA-SBL generally tend to improve performance in both measures as the number of KPCs samples increases. However, the performances of all methods do not improve significantly as the number of KPCs samples increases, compared to Section 4.3.2. The reason could be the fault pattern matrix in this multistation assembly process, which is very structured compared to the random design matrix Φ in Section 4.3. The random matrix achieves an accurate sparse estimation based on some theoretical properties of the random

matrix, such as low mutual coherence that measures the highest correlation between columns of Φ [19]. In contrast, if the mutual coherence is high, it causes inaccurate sparse estimation. As shown in Appendix B.8, the columns of fault pattern matrix Φ in the assembly process are highly correlated, and mutual coherence is 1, causing the challenging sparse estimation task than Section 4.3.2. Therefore, the results in this section are not significantly improved compared to Section 4.3.2, which uses a random matrix with low mutual coherence, even if the number of KPCs samples is increased. Figure 4.9 shows that utilizing prior knowledge of process faults is still effective in the proposed method for failure rate and NMSE, even in the structured design matrix Φ. The property lets the proposed method obtain a more accurate sparse estimation than T-MSBL, MSBL, and SCBL that cannot incorporate the prior knowledge of process faults.

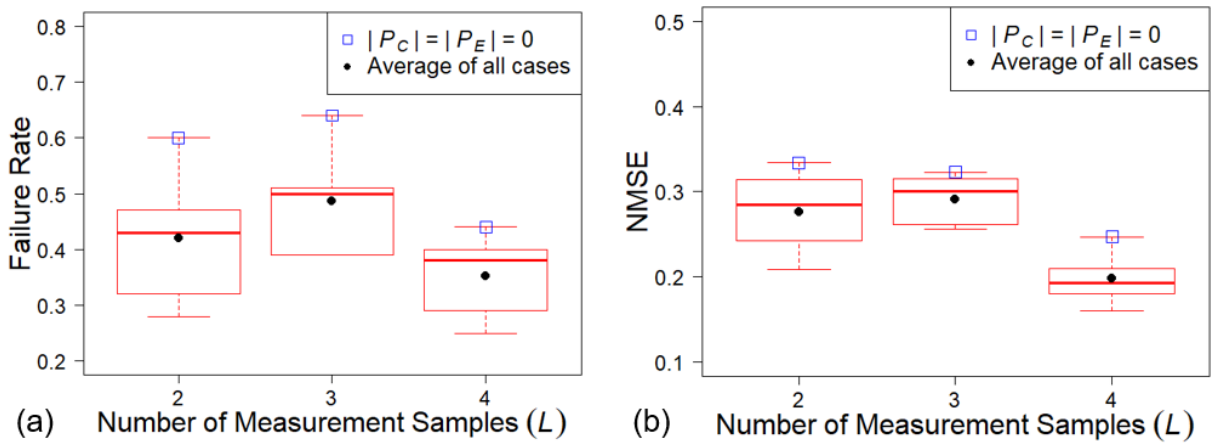


Figure 4.9: The boxplot for (a) failure rate and (b) NMSE in all cases of SA-TSBL in the various number of KPCs samples.

4.4.2 Performance Evaluation in Various Temporal Correlations

This case study presents the performance of all methods in various temporal correlations β, when there exist five KPCs samples (L). Temporal correlation varies among 0.0, 0.3, 0.6, 0.9, and 0.99. The study has three process faults (K) with noise level 80dB. Table 4.5 shows

the proposed method achieves the best performance, and the performance of other methods exhibit similar trends to the case in Section. 4.3.1. The performance of the proposed

Table 4.5: Performance comparison by varying temporal correlations (β).

	Failure Rate					NMSE				
β	0.0	0.3	0.6	0.9	0.99	0.0	0.3	0.6	0.9	0.99
T-MSBL	0.60	0.61	0.53	0.54	0.54	0.19	0.21	0.21	0.19	0.20
MSBL	0.60	0.55	0.58	0.61	0.63	0.18	0.20	0.21	0.24	0.30
SA-MSBL	0.57	0.58	0.55	0.54	0.54	0.29	0.30	0.29	0.31	0.35
SA-SBL	0.78	0.76	0.74	0.68	0.62	0.46	0.46	0.42	0.38	0.40
[75]	1.0	0.99	0.99	0.99	0.98	30.53	18.19	8.79	4.69	3.50
SCBL	0.71	0.73	0.68	0.69	0.66	0.27	0.32	0.29	0.32	0.32
SA-TSBL (Proposed)	**0.52**	**0.51**	**0.48**	**0.46**	**0.38**	**0.17**	**0.19**	**0.19**	**0.16**	**0.17**

method and T-MSBL improves in general as β increases, and MSBL and SA-MSBL show the opposite trends since they assume the independence in the time series data of each process error. [75] shows poor performance, which might be caused by the high mutual coherence of the fault pattern matrix. Figure 4.10 illustrates the prior knowledge of process faults is still effective in various temporal correlations even with the structured matrix Φ.

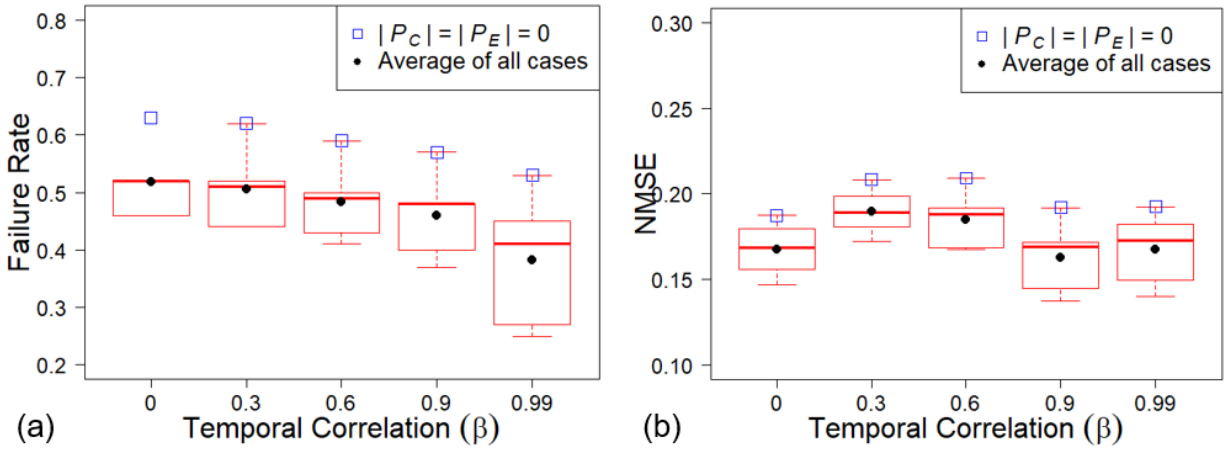


Figure 4.10: The boxplot for (a) failure rate and (b) NMSE in all cases of SA-TSBL in various temporal correlations.

4.5 Conclusions

This paper proposes a novel sparse hierarchical Bayesian method, SA-TSBL, to effectively identify the sparse process faults in multistation assembly systems. The method identifies process faults by considering the temporal correlation of each KCC and utilizing partial with some erroneous prior knowledge of process faults. Since posterior distributions of process errors in the proposed method are computationally intractable, this paper derives approximate posterior distributions of process errors via Variational Bayes inference. The effectiveness of the proposed method is validated by both numerical cases and real-world simulation application that uses an actual auto body assembly system. Based on these studies, it is evident that the direct use of multiple KPCs samples in the proposed SA-TSBL is more effective to process mean shift identification than in a previous study using the average of multiple KPCs samples in the single measurement vector model (Eq. (4.1)). This is because the proposed method can fully utilize multiple KPCs samples without any information loss. In addition, the results in case studies represent the proposed method achieves high performance in process faults estimation when the time series data of each KCC have a strong temporal correlation. Furthermore, utilizing the prior knowledge of process faults improves the performance of the proposed method in the case studies, even if the knowledge has some erroneous information. This is possible through the Bayesian framework in the proposed method that distinguishes between the correct and incorrect prior knowledge. Practically, the proposed method improves at least 27% and 28% of the failure rate and NMSE of [12, 75], the state-of-the-art sparse Bayesian diagnosis method for the process faults in the multistage assembly process in real-world simulation case studies.

In this work, all fixture locators are assumed to be independent, which is a common assumption in the literature. However, the fixture locators in the multistation assembly process have

some spatial correlation with each other if the locators are in the composite tolerance mode [68]. Therefore, considering the spatial correlation among the process errors is a promising direction for future work. In addition, instead of utilizing the Gamma distribution in Eq. (4.5) enforcing the sparsity of process error in the proposed method, other distributions such as Laplacian distribution can also be considered as prior distribution since the probability of these distributions is concentrated at zero.

Chapter 5

Reinforcement Learning-based Defect Mitigation for Quality Assurance of Additive Manufacturing

Additive Manufacturing (AM) is a powerful technology that produces complex 3D geometries using various materials in a layer-by-layer fashion. However, quality assurance is the main challenge in AM industry due to the possible time-varying processing conditions during AM process. Notably, new defects may occur during printing, which cannot be mitigated by offline analysis tools that focus on existing defects. This challenge motivates this work to develop online learning-based methods to deal with new defects during printing. Since AM typically fabricates a small number of customized products, this paper aims to create an online learning-based strategy to mitigate the new defects in AM process while minimizing the number of samples needed. The proposed method is based on model-free Reinforcement Learning (RL). It is called Continual G-learning since it transfers several sources of prior knowledge to reduce the needed training samples in the AM process. Offline knowledge is obtained from literature, while online knowledge is learned during printing. The proposed method develops a new algorithm for learning the optimal defect mitigation strategies proven the best performance when utilizing both knowledge sources. Numerical and real-world case studies in a fused filament fabrication (FFF) platform are performed and demonstrate the

effectiveness of the proposed method.

5.1 Introduction

Additive Manufacturing (AM), also known as "3D printing", makes a three-dimensional shape from a digital model. The Fused Filament Fabrication (FFF) process is one of the most widely used AM technologies attributed to its low cost and material flexibility [78, 101]. FFF uses a movable head that heats a thermoplastic filament to melting temperatures and then extrudes onto a substrate. This extruded material solidifies and subsequently bonds to the previous layers [15, 136]. During this repeated solidifying and bonding phase, some defects such as voids, over-fill, and under-fill may occur [2, 102]. These defects can cause a severe discrepancy in geometrical tolerance, loss of internal structure precision, and poor surface quality of AM products [62, 124]. Aiming to mitigate defects of AM products, many research efforts have been reported in the literature, such as post-processing [18, 74], design of experiments (DOE) [108], and mathematical optimization methods [77].

However, these methods mentioned above are primarily offline analysis tools, and they cannot identify and correct defects during printing. For example, Figure 5.1 shows the limitation of DOE in the AM process. Figure 5.1 (a) illustrates the CAD model for a printed part using the predetermined offline optimal process parameters based on the DOE. Figures 5.1 (b) and (c) show the surface quality of the 3^{rd} and 30^{th} layers of the part, respectively. The surface quality of the 30^{th} layer shows under-fill defects due to the accumulation of uncertainties from the complex process, such as machine vibration, ambient temperature, and humidity [83]. Therefore, online adjustments of the process parameters are necessary. To fulfill this need, [83] proposed a method based on closed-loop quality control for the FFF process using a PID controller to mitigate defects via online process parameter adjustments. However,

this method can only handle the defects identified and trained by the controller beforehand. It cannot deal with new defects during printing that was not recognized by the system.

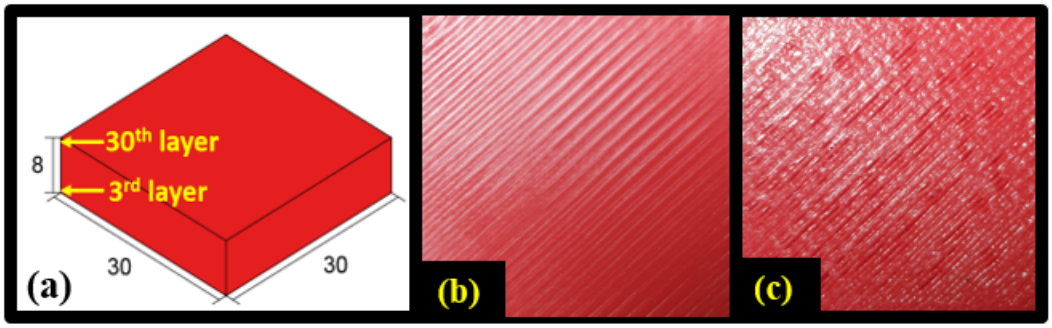


Figure 5.1: (a) CAD model for a printed part with units of mm; (b) Normal printing in 3rd layer based on predetermined optimal parameter based on the DOE; (c) Under-fill defects observed in 30th layer during printing.

This limitation motivates the online learning-based method to deal with new defects that did not occur during the previous printing. Specifically, the online learning-based method needs to learn the new process parameter adjustments to mitigate the new defects. Compared to conventional manufacturing that produces a large number of parts, AM processes usually fabricate a small number of customized products [132]. Thus, the training samples are relatively limited when used for modeling purposes. Several existing methods address this issue by transfer learning (TL) in shape deviation analysis [110, 111]. They built a statistical model first based on an AM process (e.g., shape deviation model) and then transferred it to a new process. However, they did not study the online defect mitigation problem.

This paper aims to develop an online learning-based method to mitigate the new defects in AM process with a limited number of samples. The baseline of the proposed method is a model-free Reinforcement Learning (RL) method, namely, G-learning [43], that does not require any model of AM process, which is challenging to develop due to its complexity and uncertainties. G-Learning can incorporate prior knowledge to reduce training samples of model-free RL [43]. However, it can transfer only one source of prior knowledge, but multiple

sources may exist in the AM process. For example, offline knowledge can be obtained from literature or previous experiments, while online knowledge can be learned during printing. The utilization of multiple sources of offline and online knowledge is beneficial for quickly learning how to mitigate new defects. To transfer both sources of prior knowledge to the current AM process, the proposed method, namely, Continual G-Learning, is developed in this paper. Specifically, the method provides an algorithm that learns the optimal defect mitigation strategy while utilizing both sources of prior knowledge. To demonstrate the effectiveness of the proposed method, a real-world case study in the FFF platform is conducted. To the best of our knowledge, this is the first work that uses a model-free RL-based method with various sources of knowledge for defect mitigation in AM processes.

The remainder of this paper is structured in the following manner. The overall research framework is introduced in Section 5.2. The proposed research methodology is presented in Section 5.3, followed by the case studies to validate the proposed method in Section 5.4 and Section 5.5. Finally, the conclusions of this research and future research directions are summarized in Section 5.6.

5.2 Research Framework

The overall research framework of online learning-based defect mitigation in AM process is provided in Figure 5.2. The framework iterates the following three steps:

- Step 1: Collect surface images in the FFF 3D printing process;
- Step 2: Detect surface defects using an image-based classifier (e.g., one-class support vector machine (SVM) [87]);
- Step 3: Mitigate the defects by learning how to adjust process parameters (e.g., printing

speed, layer height, flow rate multiplier, etc.).

This study uses a Hyrel System 30M 3D printer (Hyrel 3D, Norcross, GA, USA), an FFF machine, as shown in Figure 5.2. In Step 1, a digital microscope collects online surface images, and ABS is the printing material. An image-based classifier is implemented to classify the surface images and detect defects in Step 2. Once a defect is detected, it is mitigated in Step 3 through our proposed method, namely, Continual G-Learning, to adjust process parameters. Continual G-Learning is based on RL and utilizes several sources of prior knowledge to learn the optimal decisions to mitigate the defects accurately and quickly. Step 3 in the framework with the proposed method is described in detail in Section 5.3.

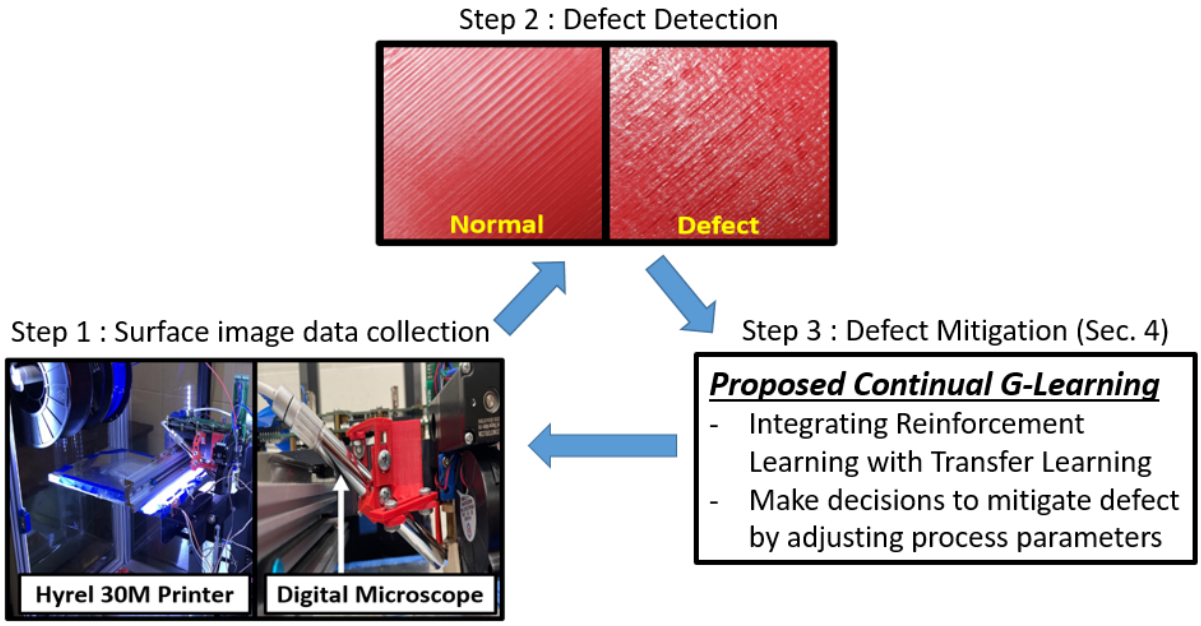


Figure 5.2: The proposed research framework.

5.3 Integration of RL and TL for AM Defect Mitigation

This section presents the proposed research methodology, which integrates RL and TL for AM defect mitigation. RL can be used as a tool for learning decisions to mitigate a new defect in AM process, described in Section 5.3.1. G-Learning [43] is a representative TL method in RL applied to reduce the number of training samples in Section 5.3.1. Finally, a new approach, namely, Continual G-learning, is proposed in Section 5.3.2 to integrate the two types of prior knowledge (i.e., offline and online) in the AM process to further reduce the needed training samples.

5.3.1 Application of RL and TL for Defect Mitigation in AM Processes

Once a defect is detected and identified as a new one during the printing process, there are no available solutions to mitigate it. The control system needs to identify a possible solution quickly to mitigate the new defect. The key here is the decision on the change of process parameters currently being used, by which the new defect will be mitigated.

RL is applied here for such purpose, based on the Markov decision process (MDP). MDP is a 4-tuple (S, A, P, R collection of states that describe the feasible processes parameter setting (e.g., printing speed, layer height, flow rate multiplier, etc.), and $s_t \in S$ is the state at time t; A is an action set that consists of increasing or decreasing the level of process parameters (i.e., parameter adjustments), and $a_t \in A$ is an action performs at time t; P is a transition probability between states, with $P(s_{t+1}|s_t, a_t)$ denoting the probability of transition to state

s_{t+1} from state s_t when action a_t is taken; $R(s_t, a_t, s_{t+1})$ represents reward function depending on the states and actions, which is determined based on the surface quality improvement or deterioration in the AM process, and r_t (i.e., $R(s_t, a_t, s_{t+1})$) is the reward incurred at time t in the process. The surface defects cause a deficiency in mechanical properties of the final product, such as density, tensile strength, and compressive strength [51]. Therefore, the defects need to be mitigated promptly to prevent quality losses in the AM process. To achieve this, RL learns the decisions that can mitigate the defect as soon as possible when it occurs. In other words, the goal of RL is to learn an optimal decision (i.e., policy used in the following context) that maximizes the total expectation of reward (r_t) incurred in the AM process based on the surface quality. The total expected reward in RL, namely, $V(s)$, is formulated as a state value function as follows [16]:

$$V(s) = \sum_{t \geq 0} \gamma^t \mathbb{E}_\pi \big[r_t | s_0 = s \big], \tag{5.1}$$

where γ and π denote discount factor and policy, respectively, and the discount factor is in the range of $0 \leq \gamma \leq 1$ which specifies the weights of future rewards. The policy, π, is a probability distribution of actions in each state. Specifically, a policy is a probability distribution of process parameter adjustments in the current parameter setting in AM process. r_t is provided as a positive value when the defect is successfully mitigated from the AM process; otherwise, r_t is zero. RL learns the policy that maximizes Eq. (5.1), by which the shortest sequence of process parameter adjustments will be generated to mitigate the defect. If the transition probability $P(s_{t+1}|s_t, a_t)$ and reward function $R(s_t, a_t, s_{t+1})$ is known, the optimal policy that maximizes Eq. (5.1) can be obtained by model-based RL [6]. However, it is challenging to estimate the accurate transition probability and reward function in AM process because of the high complexity and uncertainties of the process.

Model-free RL [131] learns optimal policy without estimating the transition probability and

the reward function of AM process. Instead, it learns an optimal policy directly based on the samples (s_t, a_t, r_t, s_{t+1}) that obtained from the interaction with AM process. Q-Learning is a representative model-free RL [131]. The objective of Q-Learning is to learn policy π at maximizes the total expected reward, which is quantified as state-action value function Q follows:

$$Q(s,a) = \sum_{t \geq 0} \gamma^t \mathbb{E}_\pi \big[r^t | s_0 = s, a_0 = a \big]. \tag{5.2}$$

Q-Learning updates Q value from time t to $t+1$ based on Eq. (5.3) as follows [13, 131]

$$Q_{t+1}(s_t, a_t) = (1 - \alpha_t(s_t, a_t)) Q_t(s_t, a_t) + \alpha_t(s_t, a_t)(r_t + \gamma \max_a Q_t(s_{t+1}, a_t)), \tag{5.3}$$

where $0 \leq \alpha_t(s_t, a_t) \leq 1$ is the learning rate in time t in state s_t with action a_t. Eq. (5.3) shows that the Q value is updated with the reward (r_t) measured by the surface quality and the maximum expected reward starting with subsequent process parameter settings $(\max_a Q_t(s_{t+1}, a))$. Since Q-Learning is a model-free RL method, the transition probability is not required in Eq. (5.3). In addition, r_t which is obtained from AM process is directly used for updating Q value, instead of using the estimated reward function. The optimal policy π in state s is the parameter adjustment (i.e., action) that results in the maximum Q value described as follows:

$$\pi(a|s) = \operatorname*{argmax}_a Q(s,a).$$

Q-Learning assumes there is no prior knowledge about defect mitigation, so it learns the policy from scratch. However, in actual AM processes, some general knowledge can be obtained from our previous experience or literature, such as the melting temperature range or printing speed of each material in AM process for the target surface quality. Utilizing this prior knowledge will improve the effectiveness and efficiency of the learning process for AM. As one of the most representative TL approaches in RL, G-Learning [43] can utilize

the prior knowledge, which is applied as the baseline for this study. The prior knowledge is used as a prior policy in G-Learning. Denote $\rho(a|s)$ and $\pi(a|s)$ as a prior policy and a policy to be learned, respectively. The divergence between $\rho(a|s)$ and $\pi(a|s)$ is defined as the information cost as follows [43]:

$$g^{\pi}(s,a) = \log\frac{\pi(a|s)}{\rho(a|s)}. \tag{5.4}$$

The expectation of the information cost represents the Kullback-Leibler divergence (KL-divergence) between both policies as follows:

$$\mathbb{E}_{\pi}[g^{\pi}(s,a)|s] = D_{KL}[\pi||\rho]. \tag{5.5}$$

Eq. (5.5) represents the divergence between the policy to be learned and prior knowledge in AM process. By considering both reward incurred in the AM process and the information cost, the total expected reward of G-Learning is represented as $GV(s)$ as follows [43]:

$$GV(s) = \sum_{t\geq 0}\gamma^{t}\mathbb{E}_{\pi}[r_t + \frac{1}{\beta}g^{\pi}(s_t,a_t)|s_0 = s], \tag{5.6}$$

where $\beta < 0$ a coefficient of information cost. By maximizing Eq. (5.6), G-Learning learns the optimal policy that maximizes the reward of the AM process while penalizing the policy that diverges from prior knowledge of the AM process.

5.3.2 The Proposed Continual G-learning using More Prior Information

In this section, a new approach named Continual G-learning that integrates both offline and online prior policies is developed. Compared to the G-Learning that only uses one source of prior knowledge, the proposed method aims to transfer prior knowledge from two sources. Specifically, it can transfer both offline knowledge and online knowledge simultaneously. Offline prior knowledge is the knowledge that can be acquired before printing, such as knowledge that can be obtained from literature or previous experiments. In contrast, online knowledge is the knowledge that is learned during printing. When G-Learning in Section 5.3.1 completes to learn the optimal policy by utilizing the offline prior policy in the AM process, an image-based classifier of the proposed framework in Section 5.2 provides a positive reward. However, when the properties of the process, such as the geometry of the part being printed changes at different layers, the classifier would provide a zero reward in the same process parameters since the parameters are not optimal in a new geometry. This provides a signal for the transition from G-Learning to Continual G-Learning in the proposed framework. Then, the optimal policy learned from G-Learning becomes the online prior knowledge in the proposed method, allowing the proposed method to utilize both prior knowledge sources.

Let $\rho_1(a|s), \rho_2(a|s)$ as offline and online prior policies in AM process, respectively. Instead of Eq. (5.4), the information cost of $\pi(a|s)$ is defined as

$$g_1^\pi(s,a) + g_2^\pi(s,a) = \log\frac{\pi(a|s)}{\rho_1(a|s)} + \log\frac{\pi(a|s)}{\rho_2(a|s)}, \tag{5.7}$$

where $g_1^\pi(s,a) = \log\frac{\pi(a|s)}{\rho_1(a|s)}$, $g_2^\pi(s,a) = \log\frac{\pi(a|s)}{\rho_2(a|s)}$. The expectation of Eq. (5.7) provides KL

divergence between prior policies and $\pi(a|s)$:

$$\mathbb{E}_\pi[g_1^\pi(s,a) + g_2^\pi(s,a)|s] = D_{KL}[\pi||\rho_1] + D_{KL}[\pi||\rho_2].$$

Considering both information cost from offline and online prior policies and the reward earned from the AM process, the total expected reward in Continual G-Learning is defined as its state value function as follows:

$$CGV(s) = \sum_{t\geq 0} \gamma^t \mathbb{E}_\pi[r_t + \frac{1}{\beta} g_1^\pi(s_t,a_t) + \frac{1}{\beta} g_2^\pi(s_t,a_t)|s_0 = s], \tag{5.8}$$

where $\beta_1 < 0$ and $\beta_2 < 0$ are the coefficients of information cost of each prior knowledge, respectively. To derive the optimal policy of the proposed Continual G-Learning, state-action value function of the proposed method is required and defined as,

$$CG(s,a) = \sum_{t\geq 0} \gamma^t \mathbb{E}_\pi[r_t + \frac{1}{\beta} g_1^\pi(s_{t+1},a_{t+1}) + \frac{1}{\beta} g_2^\pi(s_{t+1},a_{t+1})|s_0 = s, a_0 = a]. \tag{5.9}$$

By plugging Eq. (5.9) into Eq. (5.8), the state value function in Eq. (5.8) can be reformulated as follows:

$$CGV(s) = \sum_a \pi(a|s)[\frac{1}{\beta_1}\log\frac{\pi(a|s)}{\rho_1(a|s)} + \frac{1}{\beta_2}\log\frac{\pi(a|s)}{\rho_2(a|s)} + CG(s,a)]. \tag{5.10}$$

By maximizing Eq. (5.10) with constraint $\sum_a \pi(a|s) = 1$, Continual G-Learning learns the policy that maximizes the reward incurred from the AM process by penalizing the deviations from both prior policies (i.e., knowledge). It represents that both offline and online prior knowledge guide the learning direction of $\pi(a|s)$ in the learning procedure. Therefore, the knowledge aids the proposed method to learn how to mitigate defects quickly. Based on the

method of Largrange multipliers [116], the policy in Eq. (5.10) is derived as

$$\pi(a|s) = \frac{\rho_1(a|s)^{\frac{\beta_2}{\beta_1+\beta_2}}\rho_2(a|s)^{\frac{\beta_1}{\beta_1+\beta_2}}e^{-CG(s,a)\frac{\beta_1\beta_2}{\beta_1+\beta_2}}}{\sum_a \rho_1(a|s)^{\frac{\beta_2}{\beta_1+\beta_2}}\rho_2(a|s)^{\frac{\beta_1}{\beta_1+\beta_2}}e^{-CG(s,a)\frac{\beta_1\beta_2}{\beta_1+\beta_2}}}. \tag{5.11}$$

Compared to Q-Learning starts from the policy that selects the action randomly (i.e., random policy), $\pi(a|s)$ in Eq. (5.11) is initialized with various sources of prior knowledge about defect mitigation since state-action value, namely, $CG(s,a)$ is initialized as zero. The proposed Continual G-Learning provides an update rule of state-action value ($CG(s,a)$) in Eq. (5.12). Based on this rule, the value converges to the optimal state-action value proven theoretically by Theorem 5.2 that is described later in this section, and the optimal state-action value leads to optimal policy $\pi(a|s)$ in Eq. (5.11). The update rule of state-action value from time t to $t+1$ can be written as follows:

$$\begin{aligned} CG_{t+1}(s_t,a_t) = (1-\alpha_t(s_t,a_t))CG_t(s_t,a_t) + \alpha_t(s_t,a_t)(r_t - \gamma\frac{\beta_1+\beta_2}{\beta_1\beta_2}\times \\ \log(\sum_{a'}[\rho_1(a'|s_{t+1})^{\frac{\beta_2}{\beta_1+\beta_2}}\rho_2(a'|s_{t+1})^{\frac{\beta_1}{\beta_1+\beta_2}}e^{-CG(s_{t+1},a')\frac{\beta_1\beta_2}{\beta_1+\beta_2}}]), \end{aligned} \tag{5.12}$$

where the learning rate $\alpha_t(s_t,a_t)$ is defined as $n_t(s_t,a_t)^{-w}$. $n_t(s_t,a_t)$ is the number of times that the pair (s_t,a_t) is visited until time t, and $w \in (0.5,1]$ learning rate hyperparameter. Eq. (5.12) represents that the state-action value in the proposed method is updated by the reward (r_t) and the subsequent process parameter settings (s_{t+1}) that obtained by process parameter adjustment (a_t). Specifically, the value is updated by both the reward (r_t) in time t, and the maximum expected reward starting from s_{t+1} and follows policy $\pi(a|s)$ in Eq. (5.11) ($-\gamma\frac{\beta_1+\beta_2}{\beta_1\beta_2}\log(\sum_{a'}[\rho_1(a'|s_{t+1})^{\frac{\beta_2}{\beta_1+\beta_2}}\rho_2(a'|s_{t+1})^{\frac{\beta_1}{\beta_1+\beta_2}}e^{-CG(s_{t+1},a')\frac{\beta_1\beta_2}{\beta_1+\beta_2}}])$). The algorithm of the proposed Continual G-Learning is summarized in Algorithm 5.1. Starting from an initial process parameter settings in AM process, Continual G-Learning adjusts process parameters based on the policy in Eq. (5.11) (line 6 in Algorithm 5.1). Based on the parame-

ter adjustments, the method reaches subsequent process parameter settings and receives the reward based on the surface quality in the AM process (line 7 in Algorithm 5.1). Then, the state-action value in Continual G-learning is updated by Eq. (5.12) (line 10 in Algorithm 5.1). The value is used to update policy in Eq. (5.11). This algorithm iterates until the method achieves the optimal process parameter settings (i.e., terminal state) to mitigate the defects or reach the maximum number of iterations. The entire procedure is named an episode. The episode (from line 3 to line 13 in Algorithm 5.1) is repeated until the state-action value converges to optimal, eventually learning the shortest number of parameter adjustments to mitigate the defects.

Algorithm 5.1 Continual G-Learning

Continual G-Learning

Require: State S, Action A, Coefficient β_1, β_2, Discounting factor $\gamma \in [0, 1]$, Maximum number of iterations in an episode ($iter_{max}$), Learning rate hyperparameter (w)

1 Initialize $CG_0(s, a) = n_0(s, a)^{-w} = 0, \forall s, a.$
2 While $CG0_0(s, a) = n_0(s, a)^{-w} = 0, \forall s, a.$
3 Start from the initial state $s_t \in S, iter = 0.$
4 While state $s_t \in S$ is not terminal state or $iter \leq iter_{max}$
5 Calculate $\pi(a|s_t)$ as in Eq. (5.11)
6 Choose $a_t \in A$ using policy derived from $\pi(a|s_t)$
7 Obtain s_{t+1}, r_t
8 $n_t(s_t, a_t) = n_t(s_t, a_t) + 1$
9 $\alpha_t(s_t, a_t) = n_t(s_t, a_t)^{-w}$
10 Calculate $CG_{t+1}(s_t, a_t)$ based on Eq. (5.12)
11 $t \leftarrow t + 1$
12 $iter \leftarrow iter + 1$
13 Return $CG_t(s, a), \forall s, a.$

The theoretical convergence of state-action value to the optimal state-action value based on Algorithm 5.1 is provided as follows. To begin with, necessary definitions and results to build the convergence of Algorithm 5.1 are introduced. Let $CG^*(s_t, a_t)$ be the optimal state-action value of state s_t and action a_t. Subtracting the quantity $CG^*(s_t, a_t)$ from both

sides of Eq. (5.12) and letting

$$\Delta_t(s,a) = CG_t(s,a) - CG^*(s,a),$$

yields

$$\Delta_{t+1}(s_t,a_t) = (1-\alpha_t(s_t,a_t))\Delta_t(s_t,a_t) + \alpha_t(s_t,a_t)F_t(s_t,a_t),$$

where $F_t(s_t,a_t) = (r_t - \gamma\frac{\beta_1+\beta_2}{\beta_1\beta_2}\log(\sum_{a'}[\rho_1(a'|s_{t+1})^{\frac{\beta_2}{\beta_1+\beta_2}}\rho_2(a'|s_{t+1})^{\frac{\beta_1}{\beta_1+\beta_2}}e^{-CG(s_{t+1},a')\frac{\beta_1\beta_2}{\beta_1+\beta_2}}]) - CG^*(s_t,a_t))$ $\Delta_{t+1}(s,a)$ represents the difference between a state-action value in time $t+$ and optimal state-action value in state s and action a. To prove the convergence of state-action value in the proposed method, it is sufficient to prove Theorem 2 in [94] that a random iterative process $\Delta_{t+1}(s_t,a_t)$ converges to zero $w.p.$ 1 under the following assumptions:

1. $0 \leq \alpha_t(s,a) \leq 1, \sum_t \alpha_t(s,a) = \infty$ and $\sum_t \alpha_t(s,a)^2 < \infty, \forall s \in S, a \in A$;
2. $||\mathbb{E}[F_t(s,a)|U_t]||_\infty \leq \gamma||\Delta_t||_\infty$, with $\gamma < 1$;
3. $\text{var}[F_t(s,a)|U_t] \leq K(1+||\Delta_t||_\infty^2)$, for $K > 0$,

where $U_t = \{\Delta_t, \Delta_{t-1}, ..., \Delta_0, F_{t-1}, ..., F_0\}$.$||\cdot||_\infty$ refers to supremum norm, and, and K is a constant. For any policy π, operator $B^\pi[CG(s,a)]$ is defined as follows:

$$B^\pi[CG(s,a)] = k^\pi(s,a) + \gamma\sum_{s',a'} p(s'|s,a)\pi(a'|s')CG(s',a'), \tag{5.13}$$

where

$$k^\pi(s,a) = \mathbb{E}_p[r(s,a,s')] + \gamma\sum_{s',a'} p(s'|s,a)\pi(a'|s')[\frac{1}{\beta_1}\log\frac{\pi(a'|s')}{\rho_1(a'|s')} + \frac{1}{\beta_2}\log\frac{\pi(a'|s')}{\rho_2(a'|s')}].$$

To prove the convergence of our proposed algorithm, Lemma 5.1 is used to prove the second

assumption above.

Lemma 5.1. *For any policy π, the operator $B^{\pi}[CG(s,a)]$ is a contraction under the supremum norm over s, a. That is, for any $CG1(s,a)$ and $CG2(s,a)$, it follows*

$$||B^{\pi}[CG1(s,a)] - B^{\pi}[CG2(s,a)]||_{\infty} \leq \gamma ||CG1(s,a) - CG2(s,a)||_{\infty}.$$

proof. The proof is provided in Appendix C.1.

Based on Lemma 5.1, the main theorem about the convergence of proposed Continual G-Learning can be stated as follows.

Theorem 5.2. *Supposed that $0 < \rho_{min} \leq \rho_1(a|s), \rho_2(a|s) \leq \rho_{max} < 1$ for all (s,a) and $\alpha_t(s_t,a_t) = n_t(s_t,a_t)^{-w}$ for $w \in (0.5, 1]$. Then, three assumptions in Theorem 2 in [29] are satisfied where γ is the discount factor and $K = max\{K' + \max_{s\in S, a\in A} CG^*(s,a)^2, 64\gamma^2\}$. K' is defined as $2\mathbb{E}[R(s,a,s') - CG^*(s,a)]^2 + 4\gamma^2(\frac{\beta_1\beta_2}{\beta_1+\beta_2})^2[2(\log|A|)^2 + 4(\log\rho_{min})^2 + 4(\log\rho_{max})^2$. Therefore,*

$$\lim_{t\to\infty} \Delta_t(s,a) \xrightarrow{w.p.1} 0$$

Proof. The proof is provided in Appendix C.2.

Based on Theorem 5.2, Algorithm 1 is guaranteed to converge to optimal state-action value. Since optimal state-action value provides optimal policy from Eq. (5.11), the proposed Continual G-Learning learns the shortest parameter adjustments sequences to mitigate the defect by transferring offline and online prior knowledge in AM process.

5.4 Numerical Case Study

A numerical case study is performed in this section to evaluate the proposed Continual G-Learning performance. The Grid world-based simulation [119] is used for illustration of the effectiveness of the proposed Continual G-Learning. Random-Policy selecting the action randomly without any learning process and Q-Learning [131] and G-Learning [43] in Section 5.3.1 are selected as benchmark methods to compare with the proposed Continual G-Learning. To compare the performance of our proposed method with the benchmark methods, a total number of actions to complete a certain number of episodes is used as performance metrics, which is widely adopted in the RL algorithms as performance measures [31, 118]. The smaller number of actions to complete the episodes represents that method learns the optimal policy more quickly.

5.4.1 Description of Grid World

In the grid world, the unavailable squares are occupied by walls, shown in black in Figure 5.3. An episode starts from an initial state ($s_t \in S$ in Section 5.3) and terminates when reaching a goal state in Figure 5.3 or reaching the maximum number of iterations (i.e., $iter_{max}$). Each method repeats a number of episodes to learn the optimal policy. At each square, a method moves one square in any of the four directions, namely, left, right, up, and down ($a_t \in A$ in Section 5.3). If a move is blocked by the wall or the edge of the board, it stays in the same place.

Arrows in Figure 5.3 denote actions with the highest probability in a prior policy in each state. For example, in Figures 5.3 (a) and (b), blue arrows point the right direction in an initial state. It implies the prior policy of moving to the right has a higher probability

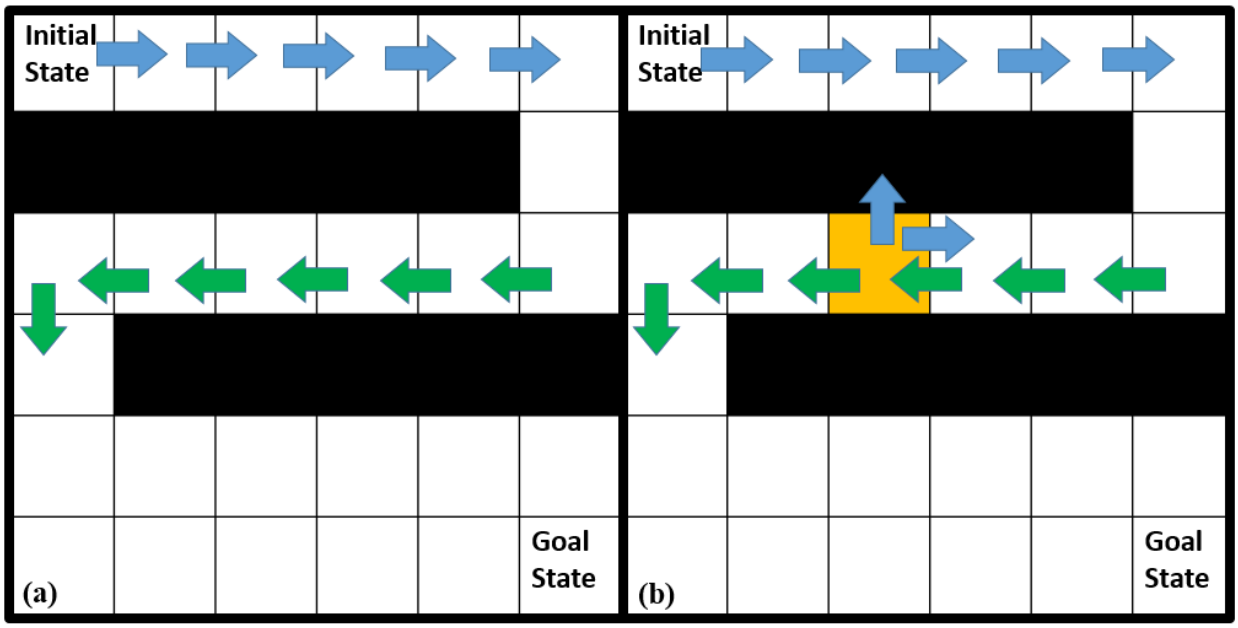


Figure 5.3: Grid world domain with prior policies. (a) Case with all the prior policies following the optimal policy; (b) Case with not all the prior policies following the optimal policy. Arrows represent actions with the highest probability in a prior policy in each state.

(i.e., 0.9) than the probability of the other three directions (i.e., 0.03 in each direction) in the initial state. The policy is called an informative prior policy. On the other hand, if all directions of prior policy have the same probability as 0.25, it is called random prior policy. The prior policy in G-Learning ($\rho(a|s)$ in Eq. (5.4)) uses the informative policy in states with blue arrows and random policy in the remaining states. The first prior policy in Continual G-Learning ($\rho_1(a|s)$ in Eq. (5.7)) is defined as the same policy as the prior policy in G-Learning. The second prior policy in the proposed method ($\rho_2(a|s)$ in Eq. (5.7)) consists of informative policy in states with green arrows and random policy in the remaining states. Detailed information of prior policy is illustrated in Appendix C.3. In this study, two cases are investigated, namely, (a) all the prior policies correspond to optimal policies (i.e., all the arrows in Figure 5.3 (a) have the same direction with optimal policy), (b) there exists a state with a prior policy that disagrees with the optimal policy that can hinder the

learning process (e.g., blue arrows in a state with orange color in Figure 5.3 (b)).

5.4.2 Performance Evaluation

The average number of actions to complete 100 episodes in 50 replications is used as a performance evaluation measure. Hyperparameters used in this study are provided in Table 5.1. The learning rate hyperparameter (w) is chosen as 0.6 to meet the first condition of Theorem 2 in [94]. The reward is provided as 1 when a method reaches the goal state. Otherwise, the reward is assigned as zero. The discount factor (γ) selected as 0.9 which is a typical value used in RL when the reward provided in the terminal state is larger than other states like in this simulation study [91]. β and β_1, β_2 are the negative values which are the coefficients of information cost of each prior policy in G-Learning and Continual G-Learning, respectively. When the coefficients are small, both methods learn the policy that approaches prior knowledge since information cost in Eqs. (5.6) and (5.8) are dominant. As shown in Figure 5.3, there exist prior policies that correspond to the optimal policies. However, many prior policies, such as random policies, disagree with the optimal policies. Therefore, the coefficients are determined by tuning, and grid search [82] is used for tuning in this case study. To reduce the computational burden in a grid search, β_1 and β_2 in the proposed method are assumed to be equal. The coefficients are searched at intervals of 100 in the range of -500 to -3000.

Hyperparameters	Value
w	0.6
r (reach to goal state / otherwise)	1 / 0
γ	0.9
$\beta = \beta_1 = \beta_2$	-2×10^3
$iter_{max}$	1000

Table 5.1: Hyperparameters in the numerical case study.

Figures 5.4 (a) and (b) show the performance evaluation of all methods in Cases (a) and (b), respectively. Continual G-Learning has the smallest number of actions to complete episodes in both Cases (a) and (b). After several episodes, the proposed method converges to the optimal policy, which is the validation of Theorem 5.2 in Section 5.3.2. Performance of G-Learning is significantly degraded in Case (b) compared to Case (a) since there exist prior policies that can hinder the learning process. The performance is similar to that of Q-Learning, which does not utilize any prior knowledge. However, Continual G-Learning overcomes this challenge by utilizing an additional source of prior policy that corresponds to optimal policy (i.e., green arrow in the state with orange color in Figure 5.3 (b)). Therefore, Continual G-Learning has a similar performance in both Cases (a) and (b). Random-Policy shows the worst performance among the benchmark methods. As shown in Figure 5.4, the performance of Random-Policy does not have improvements over episodes since this approach is not learning-based. To demonstrate the performance of the proposed method

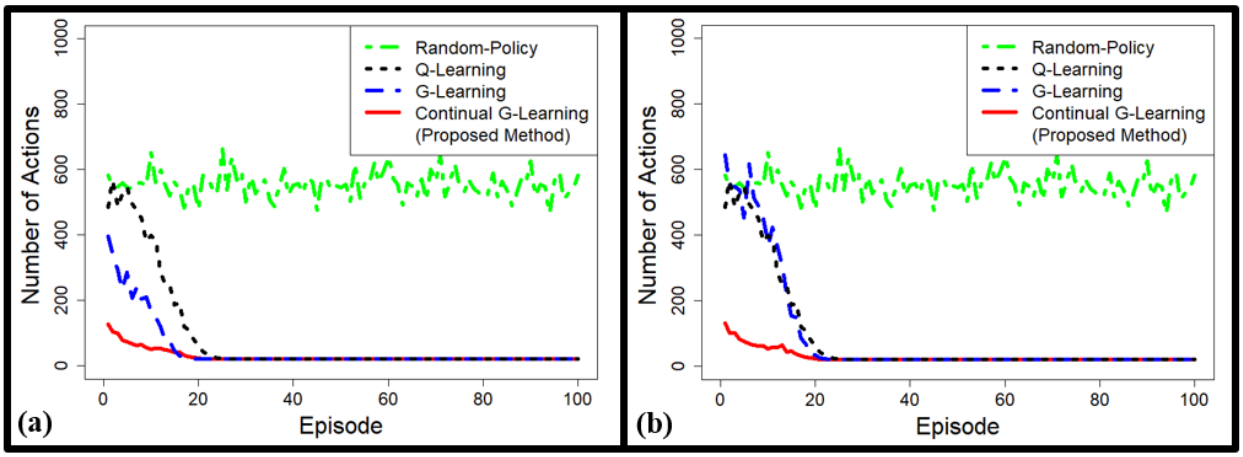


Figure 5.4: The number of actions to complete episodes when state size is 6×6 with the (a) prior policies corresponding to optimal policies and (b) prior policies do not follow optimal policies, respectively.

with different sizes of state space, the general rule is designed to provide blue and green arrows in Cases (a) and (b) when state size is $n \times n$. Let (i, j) as an index of the state. For

both Cases (a) and (b), an initial state is (1,1), and a goal state is (n, n). Unavailable states are $(2, j)$, where $j = 1, ..., n-1$, and $(4, j)$, where $j = 2, ..., n$. A blue rightward pointing arrow is provided to states $(i, 1)$, where $i = 1, ..., n-1$. A green leftward pointing arrow is used to states $(3, j)$, where $j = 2, ..., n$. A green downward pointing arrow is allowed to a state (3,1). In addition, blue upward and leftward pointing arrows are provided to a state (3,4) in Case (b). The arrows represent the prior policy that can hinder the learning process.

Table 5.2 summarizes the performance evaluations of all methods in different sizes of state. Since Random-Policy and Q-Learning do not utilize any prior policies, the results of Cases (a) and (b) are the same. Compared to the result in Case (a), the performance of G-Learning in Case (b) in all different state sizes is significantly deteriorated because of the prior policy, which hinders the learning process. However, the proposed Continual G-Learning shows similar results in both cases in every size of states, and it shows the best performance compared to benchmark methods.

	Case (a)				Case (b)			
State size	RP	QL	GL	CGL	RP	QL	GL	CGL
6×6	55323.8	8459.0	4766.7	**2786.4**	55323.8	8459.0	8633.5	**2832.8**
7×7	65441.6	14163.7	7128.5	**3693.8**	65441.6	14163.7	13607.2	**3750.4**
8×8	74680.9	22752.7	10162.1	**4686.3**	74680.9	22752.7	21952.9	**4690.6**
9×9	82648.3	34426.3	15018.4	**6104.7**	82648.3	34426.3	30385.1	**6193.8**
10×10	88494.9	50000.2	20524.1	**7823.5**	88494.9	50000.2	42515.1	**7829.6**

Table 5.2: The average number of actions to complete 100 episodes in 50 replications by varying the size of the state in Cases (a) and (b). RP, QL, GL, and CGL denote Random-Policy, Q-Learning, G-Learning, and Continual G-Learning, respectively.

5.5 Real-World Case Study

This section shows a real-world case study based on the FFF platform to test our proposed Continual G-Learning. The part has two different geometries that are printed sequentially (Figure 5.5 (a)):

- 1) Geometry 1: a cuboid with size 30 mm×30 mm×6 mm on the bottom, and
- 2) Geometry 2: a cuboid with size 15 mm×15 mm×18 mm on the top.

For clarity, the bottom and top parts of the print are denoted as Geometry 1 and Geometry 2, respectively. This case study aims to learn the optimal process parameter adjustments (i.e., policy) in both geometries to meet the target surface quality by mitigating defects, as shown in Figure 5.5 (b). Specifically, compared to the benchmark methods, the proposed method quickly learns the shortest sequence of decisions from current process parameters to the optimal process parameters. This real-world case study uses the same benchmark methods except for Random-Policy because of its poor performance that was validated in Section 5.4. The performance evaluation criterion used in Section 5.4 is utilized in this case study. Section 5.5.1 describes the AM experimental platform used in this case study. In Section 5.5.2, state ($s_t \in S$in Section 5.3), action ($a_t \in A$ in Section 5.3), and reward (r_t in Section 5.3) are defined for our AM application. Description of experiments and performance evaluation are provided in Sections 5.5.3 and 5.5.4, respectively. Finally, the printed part with optimal parameter setting is illustrated in Sec 5.5.5.

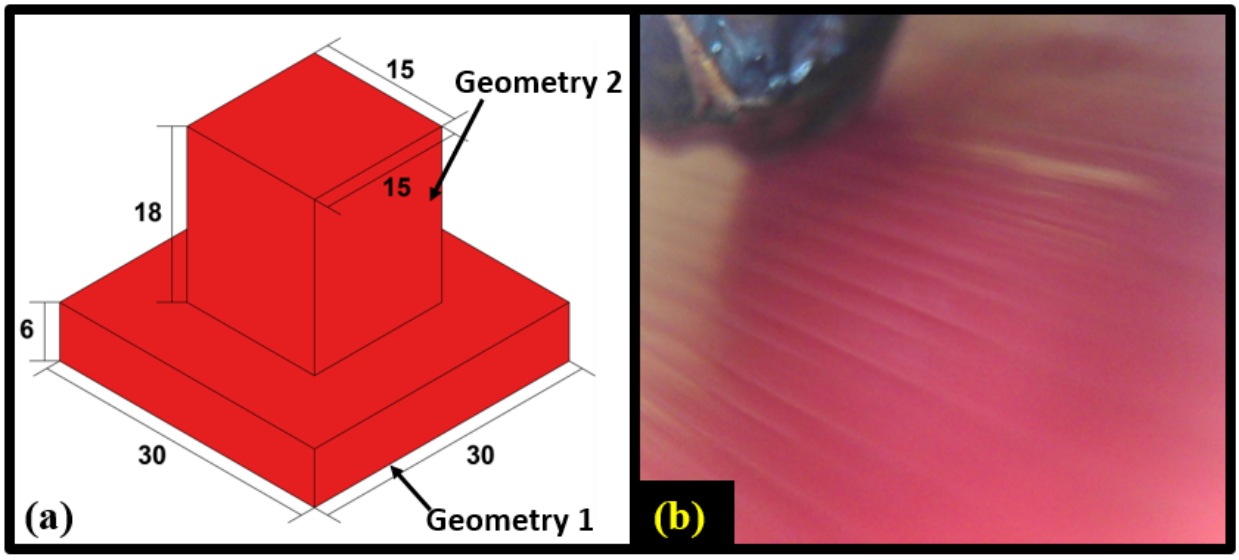


Figure 5.5: (a) CAD model for print in case studies with units of mm; (b) Target surface quality.

5.5.1 Experimental Platform

A Hyrel System 30M 3D printer (Hyrel 3D, Norcross, GA, USA) equipped with a 0.5mm extruder nozzle is used for this case study. Figure 5.6 (a) shows a front view of the printer. ABS (Hatchbox, Pomona, CA, USA) is used for printing with a diameter of 1.75mm. In every episode, the temperature of the extruder starts from 245°C which is in the printing temperature range for ABS [7]. An Opti-Tekscope Digital Microscope Camera (Opti-Tekscope, Chandler, AZ, USA) is utilized for online image acquisition (18 Hz) of surface quality, as shown in Figure 5.6 (b). The camera is mounted near the extruder to collect images of the surface that are being printed. A cooling fan is installed next to the extruder to cool down the surface of the printed part. Figure 5.6 (c) shows an open communication-based software controller. It allows the proposed Continual G-Learning to adjust the process parameters (in the form of G-code) during printing. A virtual serial port (RS-232) is used to communicate between the 3D printer controller and the external program that runs the proposed

Continual G-learning algorithm. Defect detection and mitigation are executed based on the surface images acquired from the camera and the proposed method.

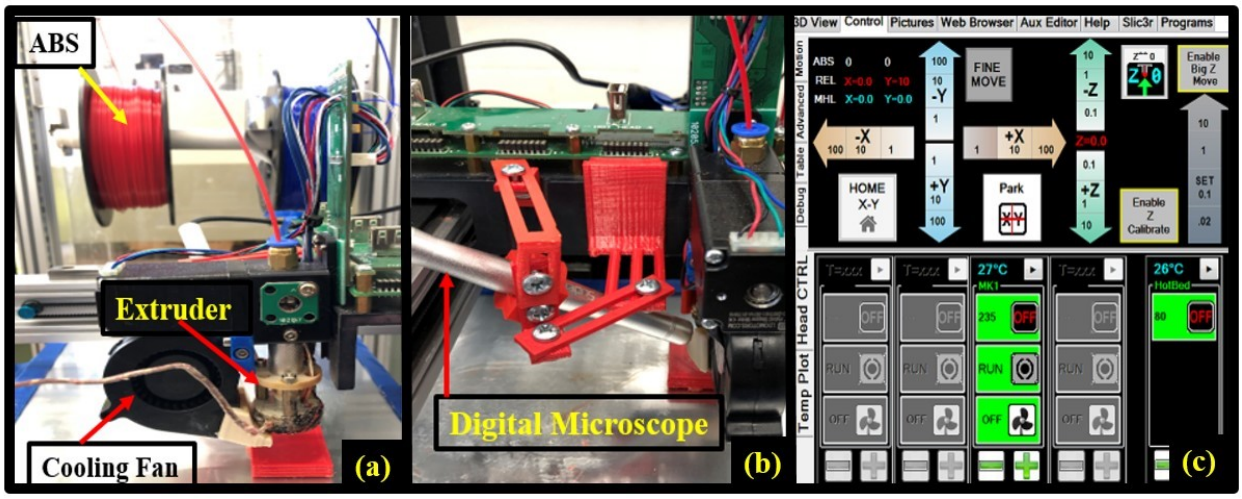


Figure 5.6: (a) Front view of Hyrel system 30M; (b) Digital Microscope Camera; (c) Software controller.

5.5.2 Description of State, Action, and Reward in the FFF Applications

Three process parameters, namely, flow rate multiplier, printing speed, and cooling fan, are adjusted in this case study. The printing speed denotes the speed of the extruder head in (mm/min). The flow rate multiplier indicates how much plastic the printer is to extrude. For example, a flow rate with a multiplier of 1.0 indicates the extruder would deposit material at normal flow (mm^3/s) while a multiplier of 0.8 or 1.2 indicates the extruder would deposit 20% less or 20% more material. The cooling fan can be controlled in terms of the operation of the fan. Each process parameter in this case study has two levels, as shown in Table 5.3. The action is defined as tuning the level of a single process parameter from the current setting. The state is defined as the combinations of levels of each parameter in Table 5.3.

Parameter \ Level	1	2
Flow rate multiplier	0.5	1.0
Printing speed (mm/min)	4500	2500
Cooling fan	Off	On

Table 5.3: Level representation of each process parameter.

The reward is provided as a positive value when the printed part achieves the target surface quality. Otherwise, the reward is assigned as zero. To measure surface quality, the offline trained one-class support vector machine (SVM) [87] is utilized as an image-based classifier. The classifier is trained with the features of target surface quality images (Figure 5.5 (b)) extracted by a pooling layer of pre-trained ResNet, which is a standard feature extraction method in many vision applications [88, 93]. This offline trained classifier, named a target classifier, predicts the quality of the surface image captured online by the digital microscope as target surface quality or an anomaly that is not. To collect the surface image data, a window-based approach is used. A window size of 21 is utilized. Namely, the 21 consecutive surface images are captured by a digital microscope with a sampling frequency of 5Hz. Surface quality is determined by voting from the classification results of 21 images from the target classifier. The window size and sampling frequency are determined to provide a robust classification result to unintended noise in the process, and they are validated from the previous printing. The reward is provided as a positive value if the surface quality is classified as the target surface, otherwise provides zero. Hyperparameters used in the real-world case study are presented in Table 5.4, and they are selected for the same reasons provided in Section 5.4.

Hyperparameters	Value
w	0.6
r (reach to goal state / otherwise)	1 / 0
γ	0.9
$\beta = \beta_1 = \beta_2$	-7×10^2
$iter_{max}$	50

Table 5.4: Hyperparameters in the real-world case study.

5.5.3 Description of Experiments

Three experiments are performed in this case study. Since the printed part consists of two different geometries (i.e., Geometry 1 and Geometry 2, respectively), each experiment consists of combinations of two methods summarized in Table 5.5. Additionally, the prior knowledge that transferred in each geometry is illustrated in Table 5.5. For offline knowledge, observation from [83] that surface defects are minimized when flow rate multiplier approaches one is used. The knowledge encourages methods with offline prior knowledge to select flow rate multiplier as one in high probability. Online knowledge is the optimal policy learned from Geometry 1. Detailed information of prior policy is illustrated in Appendix C.3. Experiment 1 uses G-Learning in Geometry 1 with offline knowledge. Then, it uses the proposed Continual G-Learning in Geometry 2 by transferring both offline and online knowledge. Experiments 2 and 3 start to print Geometry 1 without offline prior knowledge. Therefore, they use Q-Learning in Geometry 1. In Geometry 2, experiment 2 transfers offline prior knowledge by G-Learning, and experiment 3 uses Q-Learning.

	Geometry 1	**Geometry 2**
Experiment 1	G-Learning (Offline)	Continual G-Learning (Offline, Online)
Experiment 2	Q-Learning (None)	G-Learning (Offline)
Experiment 3	Q-Learning (None)	Q-Learning (None)

Table 5.5: Experiments description based on which prior knowledge (in the parenthesis) is used in each geometry.

5.5.4 Performance Evaluation

In Geometry 1, the cooling fan is excluded from the process parameters since the temperature of the extruder (245°C) is in the range of printing temperatures of ABS (220°C∼270°C) [7]. The initial state of Geometry 1 is the state with a flow rate multiplier of 0.4 and a printing speed of 7500 mm/min that causes surface defects, named as Defect 1. Since Defect 1 is classified as an anomaly from the target classifier, it is identified as a new defect. Several images of Defect 1 are collected to train the one-class SVM, denoted as the Defect 1 classifier. The classifier is used to identify new defects in further printing. After the training the classifier, the proposed Continual G-Learning starts to learn the parameter adjustments to mitigate the new defect. The episode in the proposed method starts from the initial state and terminates when it reaches the optimal parameter setting that produces the target surface quality. Figure 5.7 (a) shows the performance evaluations in Geometry 1. Based on offline knowledge, G-Learning learns the sequences of decisions from the initial state to the optimal parameter setting that flow rate multiplier of 1.0 and printing speed of 2500 mm/min in the 1^{st} episode. It performs the same parameter adjustments in the 2^{nd} and 3^{rd} episodes. Therefore, the number of layers in Geometry 1 for each episode is constant, as shown in Figure 5.7 (b). The Q-Learning that does not use prior knowledge needs several more actions in the 1st episode to learn the optimal policy.

Due to the printing sequence, the episode's initial state in Geometry 2 is the optimal process parameter setting learned from Geometry 1. However, the surface quality from Figure 5.8 (b) shows that the optimal setting in Geometry 1 is no longer optimal in Geometry 2 anymore, and the surface quality is classified as an anomaly from the target classifier representing it as the defect. In addition, the surface is classified as an anomaly in the Defect 1 classifier, indicating that the surface is a new defect. It implies the learned process parameter adjust-

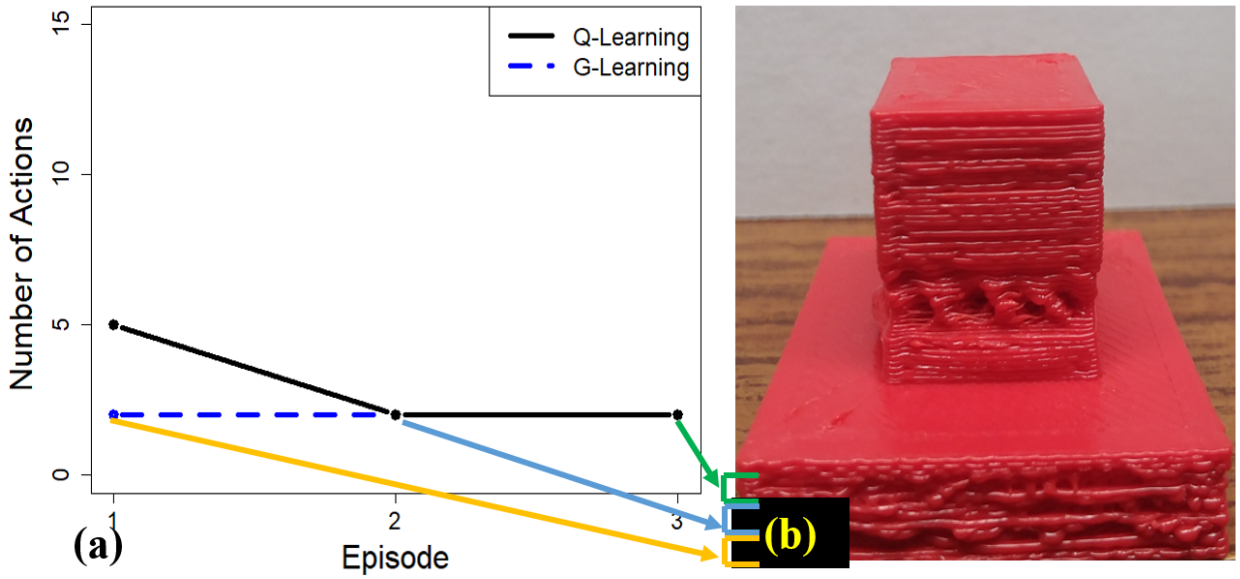


Figure 5.7: (a) Number of actions needed to reach the target surface quality in Geometry 1 using G-Learning; (b) Corresponding layers of Geometry 1 in each episode using G-Learning.

ments from Geometry 1 is not suitable to mitigate this defect, and the new optimal process parameters need to be learned. This new defect in Geometry 2 is caused by insufficient time for layers to solidify before reheating due to the small size of Geometry 2 [140]. Therefore, the cooling fan becomes one of the process parameters that need to be adjusted in Geometry 2. If the temperature of the extruder is below 220°C the fan is turned off irrespective of parameter setting to avoid the nozzle from being clogged [7].

Figure 5.9 (a) shows the performance evaluations in Geometry 2. Continual G-Learning needs fewer actions to learn the optimal parameter adjustments than other methods by using offline and online prior knowledge. The knowledge encourages the flow rate and printing speed to set 1.0 and 2500 mm/min, respectively. The proposed method learns the optimal parameter adjustments from the 3rd episode based on the knowledge. The optimal process parameter setting in Geometry 2 is a flow rate multiplier of 1.0, printing speed of 2500 mm/min, and turning on the cooling fan. Figure 5.8 (c) shows the surface quality in Geometry 2 with the optimal parameter setting. It offers a similar surface quality to Figure 5.8 (a), the

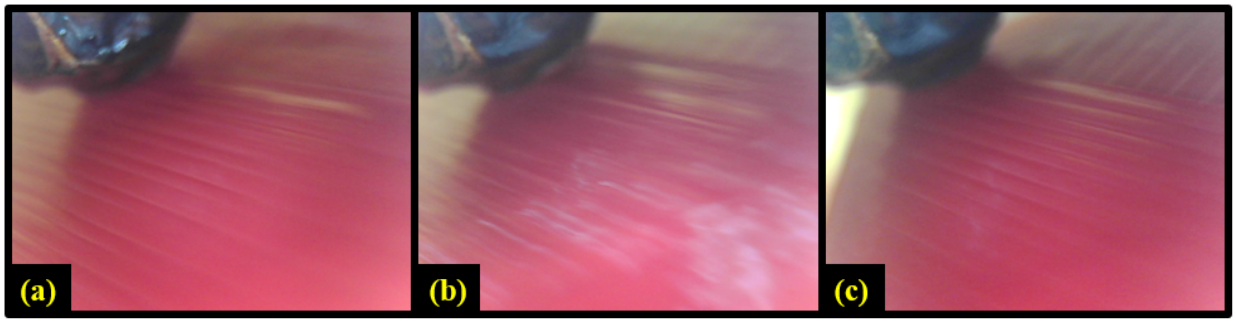


Figure 5.8: (a) Surface quality in Geometry 1 with optimal parameter setting for Geometry 1 (target surface quality); (b) Surface quality in Geometry 2 with optimal parameter setting for Geometry 1; (c) Surface quality Geometry 2 with optimal parameter setting for Geometry 2.

target surface quality collected in Geometry 1. Figure 5.9 (b) shows that the number of layers in Geometry 2 that need to be completed is reduced over episodes. It shows the proposed method learns the optimal policy as the episode increases. Experiment 1 with the proposed Continual G-Learning needs the least number of actions to learn the optimal parameter adjustments to meet the target surface quality in both geometries by transferring both sources of prior knowledge. Table 5.6 shows the number of actions needed in Geometries 1 and 2 to complete 3 and 6 episodes, respectively. Table 5.7 illustrates the optimal process parameters for defect mitigation learned from each geometry.

	Geometry 1	**Geometry 2**	**Total**
Experiment 1	G-Learning (**6**)	Continual G-Learning (**12**)	**18**
Experiment 2	Q-Learning (9)	G-Learning (16)	25
Experiment 3	Q-Learning (9)	Q-Learning (24)	33

Table 5.6: The number of actions (in the parenthesis) required to complete three episodes in Geometry 1 and six episodes in Geometry 2 for each experiment.

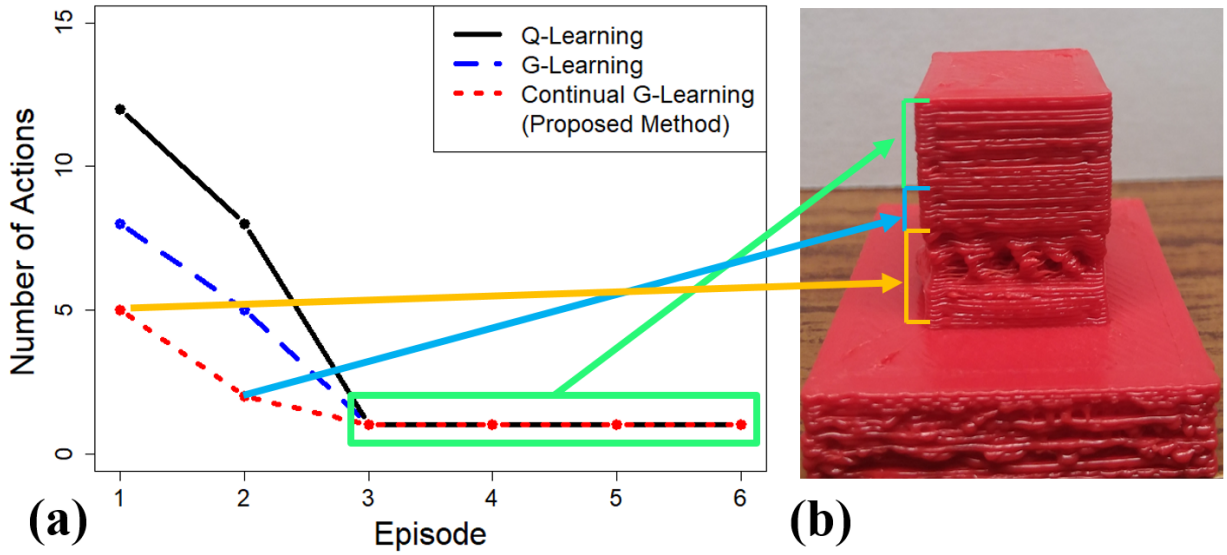


Figure 5.9: (a) Number of actions needed to reach the target surface quality in Geometry 2 using Continual G-Learning; (b) Corresponding layers of Geometry 2 in each episode using Continual G-Learning.

	Flow rate multiplier	Printing speed (mm/min)	Cooling fan
Geometry 1	1.0	2500	Off
Geometry 2	1.0	2500	On

Table 5.7: Optimal process parameters for defect mitigation in each geometry.

5.5.5 Verification of the Learned Optimal Parameter Setting

Figure 5.10 shows a newly printed part using the learned optimal parameter settings in Geometries 1 and 2, listed in Table 5.7. The optimal parameter setting of Geometry 1 is flow rate multiplier with one and printing speed with 2500 mm/min. The optimal parameter setting for Geometry 2 is the same as Geometry 1 while turning on the cooling fan. Figure 5.10 shows defect-free print by optimal parameter settings in both geometries.

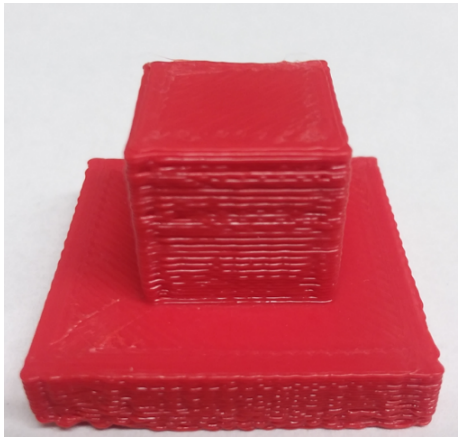


Figure 5.10: Printed part with learned optimal parameter settings in both geometries.

5.6 Conclusions

This paper proposed an online learning-based method, namely, Continual G-Learning, to mitigate the new defects in AM process with limited samples. The proposed method addresses the challenge of limited samples in AM process by transferring offline and online prior knowledge into the current AM process. The proposed method develops an algorithm for learning the optimal defect mitigation strategies when utilizing both knowledge sources. Both numerical and real-world case studies show the effectiveness of the proposed method. In the real-world case study, the proposed method learned optimal process parameter adjustments for a printed part with two different geometries in the fewest number of actions (18 actions) compared with two benchmark methods, which need 25 and 33 actions, respectively. The results demonstrate that this proposed method significantly improves online defect mitigation in the AM process.

The future work is focused on two directions. One direction is to investigate a decision rule to determine whether the transferred knowledge has positive or negative effects on the target process. It prevents negative knowledge transfer that can hinder the learning process.

The other direction is to apply the proposed method to the multi-material AM process. Multi-material AM has various processing conditions by varying the composition and types of materials used in printing [18, 105]. Therefore, it demands a learning-based method to deal with new defects in various process conditions.

Chapter 6

Conclusions and Future Work

The contribution of this dissertation is to develop three machine learning methodologies assisted by physics knowledge for process monitoring and control to achieve quality assurance. Specific contributions and conclusions of each chapter are summarized as follows:

1. In Chapter 3, a new Generative Adversarial Network (GAN)-based method is proposed to detect the process anomalies with imbalanced process data due to different ratios of occurrence between process states. Specifically, the proposed method jointly optimizes the GAN and classifier to augment the realistic and state-distinguishable images to provide balanced data. The method enables using the features of the normal process state to generate images of the abnormal process state effectively. In addition, the method stabilizes the training process by regularizing the gradient of the discriminator and providing additional tasks to the discriminator. The efficacy of the suggested approach has been confirmed in both polymer AM and metal AM processes.

2. In Chapter 4, a novel sparse Bayesian learning (SBL) is proposed to diagnose process faults with a limited number of sensors caused by the physical constraints in the multistage assembly process. The method is based on a practical assumption that it will likely have a few process faults (sparse). In addition, the temporal correlation of sensor data and the prior knowledge of process faults are considered through the Bayesian hierarchical model. Since the posterior distributions of process faults are

intractable, this work derives approximate posterior distributions using Variational Bayes inference [103]. Based on the proposed method, process faults can be accurately identified with limited sensors.

3. In Chapter 5, a novel Reinforcement Learning (RL)-based algorithm is proposed to achieve online defect mitigation of new defects that occurred during the printing due to the complexity of the AM process. The proposed method is to learn the machine parameter adjustment to mitigate the new defects during the printing. The method transfers knowledge learned from various sources in the AM process to RL in a probabilistic manner. Therefore, with a theoretical guarantee, the proposed method learns the mitigation strategy with fewer training samples than traditional RL. The effectiveness of the proposed method has been validated in a closed-loop control system implemented at an actual polymer AM process.

Based on the promising research outcomes in this dissertation, future research will focus on developing novel machine learning methods that can address practical issues in advanced manufacturing processes. I plan to pursue the following three research directions.

1. **Cybersecurity for digital manufacturing.** While advancement in sensing, artificial intelligence, and wireless technologies enables a paradigm shift into digital manufacturing, paradoxically, cyber-attacks caused by these technologies pose significant threats to the digital manufacturing process. To detect the process anomalies caused by cyber-attacks, building a monitoring system isolated from the network is necessary. Therefore, developing the unsupervised learning method utilizing air-gapped side-channel measurements would be a promising direction.

2. **Physics-informed machine learning for limited data.** The unprecedented amount of data plays a crucial role in the success of machine learning. However, the high cost

of the actual experiments and the simulations hinders providing extensive data in many advanced manufacturing processes. Therefore, a lack of data in advanced manufacturing may seriously deteriorate the performance of the pure data-driven machine learning approach. To address this issue, I will focus on developing physics-informed machine learning that utilizes physical principles to overcome the limited data issue in advanced manufacturing processes.

3. **Material synthesis using artificial intelligence.** In order to accelerate the development of advanced materials, material synthesis in the minimum number of trials is of paramount importance. The conventional synthesis methods based on empirical trial and error require numerous trials and ridiculous costs. To address these challenges, I plan to propose machine learning methods based on Bayesian optimization and Monte Carlo Tree Search. This research framework is expected to provide a wide range of opportunities for collaboration with researchers majoring in material science and chemical engineering.

Bibliography

[1] José V Abellan-Nebot, Jian Liu, Fernando Romero Subirón, and Jianjun Shi. State space modeling of variation propagation in multistation machining processes considering machining-induced variations. *Journal of Manufacturing Science and Engineering*, 134(2):021002, 2012.

[2] Mukesh K Agarwala, Vikram R Jamalabad, Noshir A Langrana, Ahmad Safari, Philip J Whalen, and Stephen C Danforth. Structural quality of parts processed by fused deposition. *Rapid prototyping journal*, 1996.

[3] R Anitha, S Arunachalam, and P Radhakrishnan. Critical parameters influencing the quality of prototypes in fused deposition modelling. *Journal of Materials Processing Technology*, 118(1-3):385–388, 2001.

[4] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

[5] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

[6] Christopher G Atkeson and Juan Carlos Santamaria. A comparison of direct and model-based reinforcement learning. In *Proceedings of international conference on robotics and automation*, volume 4, pages 3557–3564. IEEE, 1997.

[7] Michele Angelo Attolico, Caterina Casavola, Alberto Cazzato, Vincenzo Moramarco, and Gilda Renna. Effect of extrusion temperature on fused filament fabrication parts orthotropic behaviour. *Rapid Prototyping Journal*, 2020.

[8] Yaser Banadaki, Nariman Razaviarab, Hadi Fekrmandi, and Safura Sharifi. Toward enabling a reliable quality monitoring system for additive manufacturing process using deep convolutional neural networks. *arXiv preprint arXiv:2003.08749*, 2020.

[9] Kaveh Bastani, Zhenyu Kong, Wenzhen Huang, Xiaoming Huo, and Yingqing Zhou. Fault diagnosis using an enhanced relevance vector machine (rvm) for partially diagnosable multistation assembly processes. *IEEE Transactions on Automation Science and Engineering*, 10(1):124–136, 2012.

[10] Kaveh Bastani, Zhenyu Kong, Wenzhen Huang, and Yingqing Zhou. Compressive sensing–based optimal sensor placement and fault diagnosis for multi-station assembly processes. *IIE Transactions*, 48(5):462–474, 2016.

[11] Kaveh Bastani, Prahalad K Rao, and Zhenyu Kong. An online sparse estimation-based classification approach for real-time monitoring in advanced manufacturing processes from heterogeneous sensor data. *IIE Transactions*, 48(7):579–598, 2016.

[12] Kaveh Bastani, Babak Barazandeh, and Zhenyu James Kong. Fault diagnosis in multi-station assembly systems using spatially correlated bayesian learning algorithm. *Journal of Manufacturing Science and Engineering*, 140(3), 2018.

[13] Dimitri P Bertsekas et al. Dynamic programming and optimal control 3rd edition, volume ii. *Belmont, MA: Athena Scientific*, 2011.

[14] Nagaraj N Bhat, Samik Dutta, Tarun Vashisth, Srikanta Pal, Surjya K Pal, and Ranjan Sen. Tool condition monitoring by svm classification of machined surface images in turning. *The International Journal of Advanced Manufacturing Technology*, 83(9): 1487–1502, 2016.

[15] Harry Bikas, Panagiotis Stavropoulos, and George Chryssolouris. Additive manufacturing methods and modelling approaches: a critical review. *The International Journal of Advanced Manufacturing Technology*, 83(1):389–405, 2016.

[16] Lucian Busoniu, Robert Babuska, Bart De Schutter, and Damien Ernst. *Reinforcement learning and dynamic programming using function approximators*. CRC press, 2017.

[17] Hong-Seok Byun and Kwan H Lee*. Determination of the optimal part orientation in layered manufacturing using a genetic algorithm. *International journal of production research*, 43(13):2709–2724, 2005.

[18] Yi Cai, Yi Wang, and Morice Burnett. Using augmented reality to build digital twin for reconfigurable additive manufacturing system. *Journal of Manufacturing Systems*, 56:598–604, 2020.

[19] Emmanuel J Candès and Michael B Wakin. An introduction to compressive sampling. *IEEE signal processing magazine*, 25(2):21–30, 2008.

[20] Dariusz Ceglarek and Jianjun Shi. Fixture failure diagnosis for autobody assembly using pattern recognition. *Journal of manufacturing science and engineering*, 118(1), 1996.

[21] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[22] Nitesh V Chawla, Nathalie Japkowicz, and Aleksander Kotcz. Special issue on learning from imbalanced data sets. *ACM SIGKDD explorations newsletter*, 6(1):1–6, 2004.

[23] Tianlong Chen, Yu Cheng, Zhe Gan, Jingjing Liu, and Zhangyang Wang. Data-efficient

gan training beyond (just) augmentations: A lottery ticket perspective. *Advances in Neural Information Processing Systems*, 34, 2021.

[24] Yong Chen, Jionghua Jin, and Jianjun Shi. Integration of dimensional quality and locator reliability in design and evaluation of multi-station body-in-white assembly processes. *IIE transactions*, 36(9):827–839, 2004.

[25] Hyun-Soo Choi, Dahuin Jung, Siwon Kim, and Sungroh Yoon. Imbalanced data classification via cooperative interaction between classifier and generator. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[26] Ido Cohn, Tal El-Hay, Nir Friedman, and Raz Kupferman. Mean field variational approximation for continuous-time bayesian networks. *The Journal of Machine Learning Research*, 11:2745–2783, 2010.

[27] Shane F Cotter, Bhaskar D Rao, Kjersti Engan, and Kenneth Kreutz-Delgado. Sparse solutions to linear inverse problems with multiple measurement vectors. *IEEE Transactions on Signal Processing*, 53(7):2477–2488, 2005.

[28] Wenyuan Cui, Yunlu Zhang, Xinchang Zhang, Lan Li, and Frank Liou. Metal additive manufacturing parts inspection using convolutional neural network. *Applied Sciences*, 10(2):545, 2020.

[29] Harold Davenport. On a principle of lipschitz. *Journal of the London Mathematical Society*, 1(3):179–183, 1951.

[30] Eric V Denardo. Contraction mappings in the theory underlying dynamic programming. *Siam Review*, 9(2):165–177, 1967.

[31] Misha Denil, Pulkit Agrawal, Tejas D Kulkarni, Tom Erez, Peter Battaglia, and Nando

De Freitas. Learning to perform physics experiments via deep reinforcement learning. *arXiv preprint arXiv:1611.01843*, 2016.

[32] Anamika Dhillon and Gyanendra K Verma. Convolutional neural network: a review of models, methodologies and applications to object detection. *Progress in Artificial Intelligence*, 9(2):85–112, 2020.

[33] George Dimitriadis, Joana P Neto, and Adam R Kampff. t-sne visualization of large-scale neural recordings. *Neural computation*, 30(7):1750–1774, 2018.

[34] Yu Ding, Dariusz Ceglarek, Jianjun Shi, et al. Modeling and diagnosis of multistage manufacturing processes: part i state space model. In *Proceedings of the 2000 Japan/USA symposium on flexible automation*, pages 23–26. Ann Arbor, MI, 2000.

[35] Yu Ding, Dariusz Ceglarek, and Jianjun Shi. Fault diagnosis of multistage manufacturing processes by using state space approach. *J. Manuf. Sci. Eng.*, 124(2):313–322, 2002.

[36] David L Donoho. For most large underdetermined systems of linear equations the minimal ℓ_1 norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(6):797–829, 2006.

[37] Georgios Douzas and Fernando Bacao. Effective data generation for imbalanced learning using conditional generative adversarial networks. *Expert Systems with applications*, 91:464–471, 2018.

[38] DC Dowson and BV666017 Landau. The fréchet distance between multivariate normal distributions. *Journal of multivariate analysis*, 12(3):450–455, 1982.

[39] Eyal Even-Dar, Yishay Mansour, and Peter Bartlett. Learning rates for q-learning. *Journal of machine learning Research*, 5(1), 2003.

[40] Jun Fang, Yanning Shen, Fuwei Li, Hongbin Li, and Zhi Chen. Support knowledge-aided sparse bayesian learning for compressed sensing. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3786–3790. IEEE, 2015.

[41] Yasmin Fathy, Mona Jaber, and Alexandra Brintrup. Learning with imbalanced data in smart manufacturing: A comparative analysis. *IEEE Access*, 9:2734–2757, 2020.

[42] William Fedus, Mihaela Rosca, Balaji Lakshminarayanan, Andrew M Dai, Shakir Mohamed, and Ian Goodfellow. Many paths to equilibrium: Gans do not need to decrease a divergence at every step. *arXiv preprint arXiv:1710.08446*, 2017.

[43] Roy Fox, Ari Pakman, and Naftali Tishby. Taming the noise in reinforcement learning via soft updates. *arXiv preprint arXiv:1512.08562*, 2015.

[44] Luigi Maria Galantucci, Fulvio Lavecchia, and Gianluca Percoco. Experimental study aiming to enhance the surface finish of fused deposition modeled parts. *CIRP annals*, 58(1):189–192, 2009.

[45] Gabriel Rodriguez Garcia, Gabriel Michau, Mélanie Ducoffe, Jayant Sen Gupta, and Olga Fink. Temporal signals to images: Monitoring the condition of industrial assets with deep learning image processing algorithms. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 236(4):617–627, 2022.

[46] Christian Gobert, Edel Arrieta, Adrian Belmontes, Ryan B Wicker, Francisco Medina, and Brandon McWilliams. Conditional generative adversarial networks for in-situ lay-

erwise additive manufacturing data. In *2019 International Solid Freeform Fabrication Symposium*. University of Texas at Austin, 2019.

[47] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[48] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[49] Jianwen Guo, Jiapeng Wu, Zhengzhong Sun, Jianyu Long, and Shaohui Zhang. Fault diagnosis of delta 3d printers using transfer support vector machine with attitude signals. *IEEE Access*, 7:40359–40368, 2019.

[50] Yan Guo, Baoming Sun, Ning Li, and Dagang Fang. Variational bayesian inference-based counting and localization for off-grid targets with faulty prior information in wireless sensor networks. *IEEE Transactions on Communications*, 66(3):1273–1283, 2017.

[51] Musab Hajalfadul and Martin Baumers. Building a quality cost model for additive manufacturing. *University Of Khartoum Engineering Journal*, 11(1), 2021.

[52] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pages 878–887. Springer, 2005.

[53] Ningning Han and Zhanjie Song. Bayesian multiple measurement vector problem with spatial structured sparsity patterns. *Digital Signal Processing*, 75:184–201, 2018.

[54] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pages 1322–1328. IEEE, 2008.

[55] Daniel Horvath, Rafiq Noorani, and Mel Mendelson. Improvement of surface roughness on abs 400 polymer using design of experiments (doe). In *Materials science forum*, volume 561, pages 2389–2392. Trans Tech Publ, 2007.

[56] Wen Hou, Hong Guo, Bingnan Yan, Zhuang Xu, Chao Yuan, and Yuan Mao. Tool wear state recognition under imbalanced data based on wgan-gp and lightweight neural network shufflenet. *Journal of Mechanical Science and Technology*, 36(10):4993–5009, 2022.

[57] Gaofeng Huang and Amir Hossein Jafari. Enhanced balancing gan: Minority-class image generation. *Neural Computing and Applications*, pages 1–10, 2021.

[58] Wenzhen Huang, Jijun Lin, Zhenyu Kong, and Dariusz Ceglarek. Stream-of-variation (sova) modeling ii: a generic 3d variation model for rigid body assembly in multistation assembly processes. *Journal of Manufacturing Science and Engineering*, 129(4), 2007.

[59] Wenzhen Huang, Jijun Lin, Michelle Bezdecny, Zhenyu Kong, and Dariusz Ceglarek. Stream-of-variation modeling—part i: A generic three-dimensional variation model for rigid-body assembly in single station assembly processes. *Journal of Manufacturing Science and Engineering*, 140(3), 2018.

[60] Zhen Jia, Zhenbao Liu, Chi-Man Vong, and Michael Pecht. A rotating machinery fault diagnosis method based on feature learning of thermal images. *Ieee Access*, 7: 12348–12359, 2019.

[61] Jionghua Jin and Yong Chen. Quality and reliability information integration for design evaluation of fixture system reliability. *Quality and Reliability Engineering International*, 17(5):355–372, 2001.

[62] Yuan Jin, Jianke Du, and Yong He. Optimization of process planning for reducing material consumption in additive manufacturing. *Journal of Manufacturing Systems*, 44:65–78, 2017.

[63] Zeqing Jin, Zhizhou Zhang, and Grace X Gu. Autonomous in-situ correction of fused deposition modeling printers using computer vision and deep learning. *Manufacturing Letters*, 22:11–15, 2019.

[64] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[65] Michael M Kirka, Yousub Lee, Duncan A Greeley, Alfred Okello, Michael J Goin, Michael T Pearce, and Ryan R Dehoff. Strategy for texture management in metals additive manufacturing. *Jom*, 69(3):523–531, 2017.

[66] Dani Kiyasseh, Girmaw Abebe Tadesse, Louise Thwaites, Tingting Zhu, David Clifton, et al. Plethaugment: Gan-based ppg augmentation for medical diagnosis in low-resource settings. *IEEE journal of biomedical and health informatics*, 24(11):3226–3235, 2020.

[67] Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*, 2017.

[68] Zhenyu Kong, Ramesh Kumar, Suren Gogineni, Yingqing Zhou, and Dariusz Ceglarek. Mode-based tolerance analysis in multi-station assembly using stream of variation model. *Trans. NAMRI/SME*, 34:469–476, 2006.

[69] Zhenyu Kong, Dariusz Ceglarek, and Wenzhen Huang. Multiple fault diagnosis method in multistation assembly processes using orthogonal diagonalization analysis. *Journal of manufacturing science and engineering*, 130(1), 2008.

[70] Zhenyu Kong, Wenzhen Huang, and Asil Oztekin. Variation propagation analysis for multistation assembly process with consideration of gd&t factors. *Journal of Manufacturing Science and Engineering*, 131(5), 2009.

[71] Ruben Bayu Kristiawan, Fitrian Imaduddin, Dody Ariawan, Zainal Arifin, et al. A review on the fused deposition modeling (fdm) 3d printing: Filament processing, materials, and printing parameters. *Open Engineering*, 11(1):639–649, 2021.

[72] Ohyung Kwon, Hyung Giun Kim, Min Ji Ham, Wonrae Kim, Gun-Hee Kim, Jae-Hyung Cho, Nam Il Kim, and Kangil Kim. A deep neural network for classification of melt-pool images in metal additive manufacturing. *Journal of Intelligent Manufacturing*, 31(2):375–386, 2020.

[73] Kiryong Kyeong and Heeyoung Kim. Classification of mixed-type defect patterns in wafer bin maps using convolutional neural networks. *IEEE Transactions on Semiconductor Manufacturing*, 31(3):395–402, 2018.

[74] Amirali Lalehpour and Ahmad Barari. Post processing for fused deposition modeling parts with acetone vapour bath. *IFAC-PapersOnLine*, 49(31):42–48, 2016.

[75] Jaesung Lee, Junbo Son, Shiyu Zhou, and Yong Chen. Variation source identification in manufacturing processes using bayesian approach with sparse variance components prior. *IEEE Transactions on Automation Science and Engineering*, 17(3):1469–1485, 2020.

[76] Xian Yeow Lee, Sourabh K Saha, Soumik Sarkar, and Brian Giera. Automated detection of part quality during two-photon lithography via deep learning. *Additive Manufacturing*, 36:101444, 2020.

[77] Aijun Li, Zhuohui Zhang, Daoming Wang, and Jinyong Yang. Optimization method to fabrication orientation of parts in fused deposition modeling rapid prototyping. In *2010 International conference on mechanic automation and control engineering*, pages 416–419. IEEE, 2010.

[78] Hao Li, Zhonghua Yu, Feng Li, Qingshun Kong, and Jie Tang. Real-time polymer flow state monitoring during fused filament fabrication based on acoustic emission. *Journal of Manufacturing Systems*, 62:628–635, 2022.

[79] Hongbin Li, Yuan Jiang, Jun Fang, and Muralidhar Rangaswamy. Adaptive subspace signal detection with uncertain partial prior knowledge. *IEEE Transactions on Signal Processing*, 65(16):4394–4405, 2017.

[80] Shan Li and Yong Chen. A bayesian variable selection method for joint diagnosis of manufacturing process and sensor faults. *IIE transactions*, 48(4):313–323, 2016.

[81] Yuxuan Li, Zhangyue Shi, Chenang Liu, Wenmeng Tian, Zhenyu Kong, and Christopher B Williams. Augmented time regularized generative adversarial network (atr-gan) for data augmentation in online process anomaly detection. *IEEE Transactions on Automation Science and Engineering*, 2021.

[82] Petro Liashchynskyi and Pavlo Liashchynskyi. Grid search, random search, genetic algorithm: A big comparison for nas. *arXiv preprint arXiv:1912.06059*, 2019.

[83] Chenang Liu, Andrew Chung Chee Law, David Roberson, and Zhenyu James Kong.

Image analysis-based closed loop quality control for additive manufacturing with fused filament fabrication. *Journal of Manufacturing Systems*, 51:75–86, 2019.

[84] Chenang Liu, Zhenyu Kong, Suresh Babu, Chase Joslin, and James Ferguson. An integrated manifold learning approach for high-dimensional data feature extractions and its applications to online process monitoring of additive manufacturing. *IISE Transactions*, 53(11):1215–1230, 2021.

[85] Jian Liu. Variation reduction for multistage manufacturing processes: a comparison survey of statistical-process-control vs stream-of-variation methodologies. *Quality and Reliability Engineering International*, 26(7):645–661, 2010.

[86] Martin Luessi, S Derin Babacan, Rafael Molina, and Aggelos K Katsaggelos. Bayesian simultaneous sparse approximation with smooth signals. *IEEE Transactions on Signal Processing*, 61(22):5716–5729, 2013.

[87] Hanna Lukashevich, Stefanie Nowak, and Peter Dunker. Using one-class svm outliers detection for verification of collaboratively tagged image training sets. In *2009 IEEE International Conference on Multimedia and Expo*, pages 682–685. IEEE, 2009.

[88] Arpana Mahajan and Sanjay Chaudhary. Categorical image classification based on representational deep network (resnet). In *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 327–330. IEEE, 2019.

[89] Mohamad Mahmoudi, Ahmed Aziz Ezzat, and Alaa Elwany. Layerwise anomaly detection in laser powder-bed fusion metal additive manufacturing. *Journal of Manufacturing Science and Engineering*, 141(3), 2019.

[90] America Makes and ANSI Additive Manufacturing Standardization Collaborative. Standardization roadmap for additive manufacturing. *February), Public Draft*, 2017.

[91] Patrick Mannion, Jim Duggan, and Enda Howley. An experimental review of reinforcement learning algorithms for adaptive traffic signal control. *Autonomic road transport support systems*, pages 47–66, 2016.

[92] Giovanni Mariani, Florian Scheidegger, Roxana Istrate, Costas Bekas, and Cristiano Malossi. Bagan: Data augmentation with balancing gan. *arXiv preprint arXiv:1803.09655*, 2018.

[93] David McNeely-White, J Ross Beveridge, and Bruce A Draper. Inception and resnet features are (almost) equivalent. *Cognitive Systems Research*, 59:312–318, 2020.

[94] Francisco S Melo. Convergence of q-learning: A simple proof. *Institute Of Systems and Robotics, Tech. Rep*, pages 1–4, 2001.

[95] Agnieszka Mikołajczyk and Michał Grochowski. Data augmentation for improving deep learning in image classification problem. In *2018 international interdisciplinary PhD workshop (IIPhDW)*, pages 117–122. IEEE, 2018.

[96] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[97] Sampa Misra, Donggyu Kim, Jongbeom Kim, Woncheol Shin, and Chulhong Kim. A voting-based ensemble feature network for semiconductor wafer defect classification. *Scientific Reports*, 12(1):16254, 2022.

[98] Mohammad Montazeri and Prahalada Rao. Sensor-based build condition monitoring in laser powder bed fusion additive manufacturing process using a spectral graph theoretic approach. *Journal of Manufacturing Science and Engineering*, 140(9), 2018.

[99] William Mycroft, Mordechai Katzman, Samuel Tammas-Williams, Everth Hernandez-Nava, George Panoutsos, Iain Todd, and Visakan Kadirkamanathan. A data-driven approach for predicting printability in metal additive manufacturing processes. *Journal of Intelligent Manufacturing*, 31(7):1769–1781, 2020.

[100] Peeyush Nandwana and Yousub Lee. Influence of scan strategy on porosity and microstructure of ti-6al-4v fabricated by electron beam powder bed fusion. *Materials Today Communications*, 24:100962, 2020.

[101] Fuda Ning, Weilong Cong, Jingjing Qiu, Junhua Wei, and Shiren Wang. Additive manufacturing of carbon fiber reinforced thermoplastic composites using fused deposition modeling. *Composites Part B: Engineering*, 80:369–378, 2015.

[102] AnHua Peng and XingMing Xiao. Investigation on reasons inducing error and measures improving accuracy in fused deposition modeling. *Advances in Information Sciences and Service Sciences*, 4(5), 2012.

[103] Kaare Brandt Petersen, Ole Winther, and Lars Kai Hansen. On the slow convergence of em and vbem in low-noise linear models. *Neural computation*, 17(9):1921–1926, 2005.

[104] David MW Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*, 2020.

[105] NE Putra, MJ Mirzaali, I Apachitei, Jie Zhou, and AA Zadpoor. Multi-material additive manufacturing technologies for ti-, mg-, and fe-based biomaterials for bone substitution. *Acta biomaterialia*, 109:1–20, 2020.

[106] Gishan Don Ranasinghe and Ajith Kumar Parlikad. Generating real-valued failure data for prognostics under the conditions of limited data availability. In *2019 IEEE*

International Conference on Prognostics and Health Management (ICPHM), pages 1–8. IEEE, 2019.

[107] Prahalad K Rao, Jia Peter Liu, David Roberson, Zhenyu James Kong, and Christopher Williams. Online real-time quality monitoring in additive manufacturing processes using heterogeneous sensors. *Journal of Manufacturing Science and Engineering*, 137 (6), 2015.

[108] Vijeth Reddy, Olena Flys, Anish Chaparala, Chihab E Berrimi, V Amogh, and Bengt Göran Rosen. Study on surface texture of fused deposition modeling. *Procedia Manufacturing*, 25:389–396, 2018.

[109] Amanda Ross, Victor L Willson, Amanda Ross, and Victor L Willson. Paired samples t-test. *Basic and Advanced Statistical Tests: Writing Results Sections and Creating Tables and Figures*, pages 17–19, 2017.

[110] Arman Sabbaghi and Qiang Huang. Predictive model building across different process conditions and shapes in 3d printing. In *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 774–779. IEEE, 2016.

[111] Arman Sabbaghi and Qiang Huang. Model transfer across additive manufacturing processes via mean effect equivalence of lurking variables. *The Annals of Applied Statistics*, 12(4):2409–2429, 2018.

[112] Alec I Saville, Sven C Vogel, Adam Creuziger, Jake T Benzing, Adam L Pilchak, Peeyush Nandwana, Jonah Klemm-Toole, Kester D Clarke, S Lee Semiatin, and Amy J Clarke. Texture evolution as a function of scan strategy and build height in electron beam melted ti-6al-4v. *Additive Manufacturing*, 46:102118, 2021.

[113] Luke Scime, Derek Siddel, Seth Baird, and Vincent Paquit. Layer-wise anomaly detection and classification for powder bed additive manufacturing processes: A machine-agnostic algorithm for real-time pixel-wise semantic segmentation. *Additive Manufacturing*, 36:101453, 2020.

[114] Bo Shen, Weijun Xie, and Zhenyu James Kong. Clustered discriminant regression for high-dimensional data feature extraction and its applications in healthcare and additive manufacturing. *IEEE Transactions on Automation Science and Engineering*, 18(4):1998–2010, 2020.

[115] Jianjun Shi. In-process quality improvement: Concepts, methodologies, and applications. *IISE transactions*, 55(1):2–21, 2023.

[116] Samuel D Silvey. The lagrangian multiplier test. *The Annals of Mathematical Statistics*, 30(2):389–407, 1959.

[117] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. *Advances in neural information processing systems*, 30, 2017.

[118] Peter Stone and Richard S Sutton. Scaling reinforcement learning toward robocup soccer. In *Icml*, volume 1, pages 537–544, 2001.

[119] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[120] Song Tao and Jia Wang. Alleviation of gradient exploding in gans: Fake can be real. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1191–1200, 2020.

[121] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[122] Michael E Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1(Jun):211–244, 2001.

[123] Ngoc-Trung Tran, Tuan-Anh Bui, and Ngai-Man Cheung. Dist-gan: An improved gan using distance constraints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 370–385, 2018.

[124] L Villalpando, Hasti Eiliat, and Ruth Jill Urbanic. An optimization approach for components built by fused deposition modeling with parametric internal structures. *Procedia Cirp*, 17:800–805, 2014.

[125] Chao Wang, Zhibin Yu, Haiyong Zheng, Nan Wang, and Bing Zheng. Cgan-plankton: Towards large-scale imbalanced class generation and fine-grained classification. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 855–859. IEEE, 2017.

[126] Hui Wang and Qiang Huang. Error cancellation modeling and its application to machining process control. *IIE transactions*, 38(4):355–364, 2006.

[127] Hui Wang, Qiang Huang, and Reuven Katz. Multi-operational machining processes modeling for sequential root cause identification and measurement reduction. *Journal of manufacturing science and engineering*, 127(3):512–521, 2005.

[128] Rongxuan Wang, Andrew C Law, David Garcia, Shuo Yang, and Zhenyu Kong. Development of structured light 3d-scanner with high spatial resolution and its applications for additive manufacturing quality assurance. *The International Journal of Advanced Manufacturing Technology*, 117(3):845–862, 2021.

[129] Rui Wang and Nan Chen. Wafer map defect pattern recognition using rotation-invariant features. *IEEE Transactions on Semiconductor Manufacturing*, 32(4):596–604, 2019.

[130] Yitian Wang, Yuxiang Wei, and Huan Wang. A class imbalanced wafer defect classification framework based on variational autoencoder generative adversarial network. *Measurement Science and Technology*, 34(2):024008, 2022.

[131] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3): 279–292, 1992.

[132] Christian Weller, Robin Kleer, and Frank T Piller. Economic implications of 3d printing: Market structure models in light of additive manufacturing revisited. *International Journal of Production Economics*, 164:43–56, 2015.

[133] David P Wipf and Bhaskar D Rao. An empirical bayesian strategy for solving the simultaneous sparse approximation problem. *IEEE Transactions on Signal Processing*, 55(7):3704–3716, 2007.

[134] Ming-Ju Wu, Jyh-Shing R Jang, and Jui-Long Chen. Wafer map failure pattern recognition and similarity ranking for large-scale data sets. *IEEE Transactions on Semiconductor Manufacturing*, 28(1):1–12, 2014.

[135] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[136] Zehao Ye, Chenang Liu, Wenmeng Tian, and Chen Kan. In-situ point cloud fusion for layer-wise monitoring of additive manufacturing. *Journal of Manufacturing Systems*, 61:210–222, 2021.

[137] Dongping Yu, Yan Guo, Ning Li, and Meng Wang. Sa-m-sbl: An algorithm for csi-based device-free localization with faulty prior information. *IEEE Access*, 7:61831–61839, 2019.

[138] Jianbo Yu and Jiatong Liu. Two-dimensional principal component analysis-based convolutional autoencoder for wafer map defect detection. *IEEE Transactions on Industrial Electronics*, 68(9):8789–8797, 2020.

[139] Jiangquan Zhang, Sun Yi, GUO Liang, GAO Hongli, HONG Xin, and SONG Hongliang. A new bearing fault diagnosis method based on modified convolutional neural networks. *Chinese Journal of Aeronautics*, 33(2):439–447, 2020.

[140] Jie Zhang, Xin Zhou Wang, Wang Wang Yu, and Yu He Deng. Numerical investigation of the influence of process conditions on the temperature variation in fused deposition modeling. *Materials & Design*, 130:59–68, 2017.

[141] Junru Zhang, Yang Liu, Manjot Singh, Yuxin Tong, Ezgi Kucukdeger, Hu Young Yoon, Alexander P Haring, Maren Roman, Zhenyu James Kong, Blake N Johnson, et al. Rapid, autonomous high-throughput characterization of hydrogel rheological properties via automated sensing and physics-guided machine learning. *Applied Materials Today*, 30:101720, 2023.

[142] Zhilin Zhang and Bhaskar D Rao. Sparse signal recovery with temporally correlated source vectors using sparse bayesian learning. *IEEE Journal of Selected Topics in Signal Processing*, 5(5):912–926, 2011.

[143] Longhui Zhou, Hongfeng Tao, Wojciech Paszke, Vladimir Stojanovic, and Huizhong Yang. Pd-type iterative learning control for uncertain spatially interconnected systems. *Mathematics*, 8(9):1528, 2020.

[144] Shiyu Zhou, Yong Chen, and Jianjun Shi. Statistical estimation and testing for variation root-cause identification of multistage manufacturing processes. *IEEE Transactions on Automation Science and Engineering*, 1(1):73–83, 2004.

Publication Statement

The major content of Chapter 4 was accepted by Taylor & Francis in IISE Transactions on 3/24/2023, available online https://www.tandfonline.com/doi/full/10.1080/24725854.2023.2199813. Similarly, the major content of Chapter 5 was published by Elsevier in the Journal of Manufacturing Systems on 11/13/2022.

Appendices

Appendix A

Appendix of Chapter 3

A.1 Classification Results of Each Label in the Ablation Study

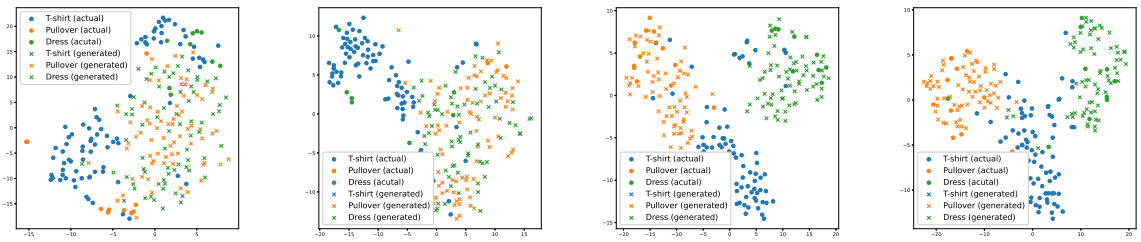
The classification results of each label in Table 3.2 are provided in Table A.1. The F-score of all labels in the proposed method achieves the best performance in the ablation studies. The proposed method significantly improves the precision of the majority class and recall of minority classes where the imbalanced training data in the baseline degrades the performances. Since all the remaining data except the training data are used as testing data, a relatively small number of T-shirt than Pullover and Dress exist in testing data. It causes the overall low results of T-shirts in Precision and F-score.

Table A.1: Performance evaluation of each label in the ablation studies.

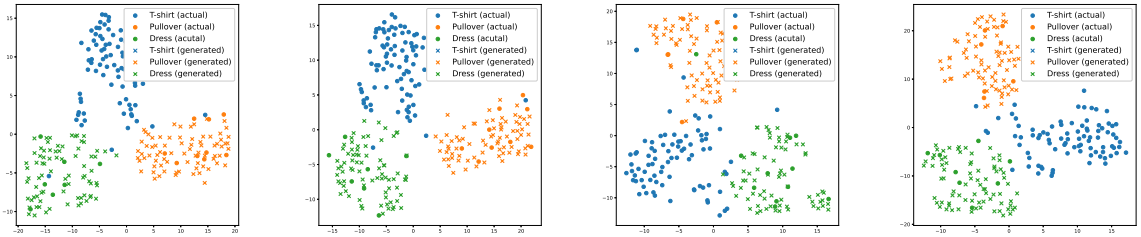
	MNIST fashion								
	Precision			**Recall**			**F-score**		
	T-shirt	Pullover	Dress	T-shirt	Pullover	Dress	T-shirt	Pullover	Dress
Baseline	0.393	0.978	0.981	0.972	0.875	0.769	0.560	0.923	0.862
Variant1	0.408	0.977	0.980	0.972	0.883	0.778	0.574	0.927	0.867
Variant2	0.465	0.974	0.973	0.963	0.887	0.827	0.625	0.928	0.893
Variant3 (Proposed)	**0.505**	**0.973**	**0.974**	**0.952**	**0.905**	**0.853**	**0.660**	**0.938**	**0.909**

A.2 T-SNE Results in the Ablation Studies

Figure A.1 shows the t-SNE of the feature from the intermediate layer of the classifier from the proposed method in several epochs in the ablation studies. The results represent that the generated samples from the proposed method follow the features of actual data in the classifier as the epoch increases. It validates the authentic and state-distinguishable properties of the generated samples of the proposed method.



(a) epoch 0 (b) epoch 10 (c) epoch 50 (d) epoch 100



(e) epoch 150 (f) epoch 200 (g) epoch 250 (h) epoch 300

Figure A.1: t-SNE of the feature from the intermediate layer of the classifier from the proposed method in several epochs.

A.3 Structure and Hyperparameters in the Methods

The detailed structure and hyperparameters of the proposed and benchmark methods are provided. Table A.2 provides the hyperparameters of each method.

Table A.2: Hyperparameters of each method.

Methods	Parameters	Value
SMOTE, ADASYN	Nearest K samples	5
B-SMOTE	Type	1
	Nearest K samples	5
CDRAGAN BAGAN-GP Cooperative GAN Proposed	Number of epochs	300
	Optimizer	Adam
	Learning rate	0.0002
	Momentum	$\beta_1 = 0.5, \beta_2 = 0.9$
	Hidden layers (Discriminator)	4 blocks of [Conv2D, LeakyRelu]
	Hidden layers (Generator)	4 blocks of [Conv2D-Transpose, LeakyRelu, BatchNormalization]
	Number of Kernels in each block (Discriminator)	(64,128,128,256)
	Number of Kernels in each block (Generator)	(128,128,64,Number of channel)
	Kernel sizes	(4,4)
	Strides	(2,2)
	Padding	Same
	Activation functions	LeakyRelu, Tanh
	Kernel initializer	Random Normal(sd=0.02)
	Slope of Leaky Relu	0.2
CDRAGAN, BAGAN-GP Proposed	Gradient Penalty Coefficient	10
Cooperative GAN	Range of scheduling parameter	[0.1,0.9]
BAGAN-GP, Proposed	Epochs in pre-training	300

The gradient penalty coefficient is determined as 10 as suggested by [57, 67]. The scheduling parameter in the Cooperative GAN, is searched within a specific range ([0.1, 0.9]) following the guidelines provided in the literature [25] and selected with the values that showed the best

validation performance. A batch size (m in Algorithm 4.1) varies along the number of actual samples in each case study to consider the computational time. Specifically, batch sizes are 100 in Section. 3.3.1, 100 in Section. 3.3.2, 60 in Section. 3.3.3, and 50 in Section. 3.3.4. Table A.3 shows the hyperparameters of the classifier in case studies. Convolutional Neural Network is used for the classifier. For a fair comparison, all the methods use the same classifier as described in Table A.3.

Table A.3: Hyperparameters of the classifier.

Parameters	Value
Number of epochs	300
Optimizer	Adam
Learning rate	0.0002
Momentum	$\beta_1 = 0.5, \beta_2 = 0.9$
Hidden Layers	4 blocks of [Conv2D, LeakyRelu]
Number kernels in each block	(32,32,128,256)
Kernel sizes	(4,4)
Strides	(2,2)
Padding	Same
Activation functions	Leaky Relu, Softmax
Kernel initializer	Random Normal (sd=0.02)
Slope of Leaky Relu	0.2

Appendix B

Appendix of Chapter 4

B.1 Derivation for Eq. (4.8)

Let $\boldsymbol{\theta}$ denote all hidden variables in the proposed method (i.e., $\boldsymbol{\theta} = (\mathbf{x}, \boldsymbol{\alpha}, \bar{b})$). Variational Bayes Inference minimizes the Kullback-Leibler (KL) divergence between $q(\theta)$ and true posterior distribution, $p(\boldsymbol{\theta}|\mathbf{y})$ given by

$$\min_{q(\boldsymbol{\theta})} KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathbf{y})) = \int q(\boldsymbol{\theta}) \ln \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\mathbf{y})} d\boldsymbol{\theta}.$$

Equivalently, it can be reformulated as follows.

$$\max_{q(\boldsymbol{\theta})} -KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathbf{y})) = \int q(\boldsymbol{\theta}) \ln \frac{p(\boldsymbol{\theta}, \mathbf{y})}{q(\boldsymbol{\theta})p(\mathbf{y})} d\boldsymbol{\theta}. \tag{B.1}$$

The right side of Eq. (B.1) can be reformulated as follows.

$$\begin{aligned}
\int q(\boldsymbol{\theta}) \ln \frac{p(\boldsymbol{\theta}, \mathbf{y})}{q(\boldsymbol{\theta})p(\mathbf{y})} d\boldsymbol{\theta} &= \int q(\boldsymbol{\theta})[\ln p(\boldsymbol{\theta}, \mathbf{y}) - \ln q(\boldsymbol{\theta}) - \ln p(\mathbf{y})] d\boldsymbol{\theta} \\
&= \int q(\boldsymbol{\theta})[\ln p(\boldsymbol{\theta}, \mathbf{y}) - \ln q(\boldsymbol{\theta})] d\boldsymbol{\theta} - \int q(\boldsymbol{\theta}) \ln p(\mathbf{y}) d\boldsymbol{\theta} \\
&= \int q(\boldsymbol{\theta})[\ln p(\boldsymbol{\theta}, \mathbf{y}) - \ln q(\boldsymbol{\theta})] d\boldsymbol{\theta} - \ln p(\mathbf{y}).
\end{aligned} \tag{B.2}$$

Since $\ln p(\mathbf{y})$ in Eq. (B.2) is irrelevant with $q(\boldsymbol{\theta})$, the problem is reformulated into maximizing the first two terms in Eq. (B.2) ($L(\boldsymbol{\theta})$) under the mean-field approximation as follows.

$$\begin{aligned} L(\boldsymbol{\theta}) &= \int \prod_k q(\theta_k)[\ln p(\boldsymbol{\theta}, \mathbf{y}) - \sum_k \ln q(\theta_k)]d\boldsymbol{\theta} \\ &= \int \prod_k q(\theta_k) \ln p(\boldsymbol{\theta}, \mathbf{y}) \prod_k d\theta_k - \sum_k \int \prod_j q(\theta_j) \ln q(\theta_k) \prod_j d\theta_j \qquad \text{(B.3)} \\ &= \int q(\theta_k)[\ln p(\boldsymbol{\theta}, \mathbf{y}) \prod_{j \neq k} (q(\theta_j)d\theta_j)]d\theta_k - \sum_k \int q(\theta_k) \ln q(\theta_k)d\theta_k \qquad \text{(B.4)} \\ &= \int q(\theta_k)[\ln p(\boldsymbol{\theta}, \mathbf{y}) \prod_{j \neq k} (q(\theta_j)d\theta_j)]d\theta_k - \int q(\theta_k) \ln q(\theta_k)d\theta_k - \sum_{j \neq k} \int q(\theta_j) \ln q(\theta_j)d\theta_j. \\ & \qquad \text{(B.5)} \end{aligned}$$

Eq. (B.4) is derived from Eq. (B.3) since $\int q(\theta_j)d\theta_j = 1, \forall j$. Define a new distribution as follows.

$$\ln \tilde{p}(\theta_k, \mathbf{y}) = \mathbb{E}[\ln p(\boldsymbol{\theta}, \mathbf{y})]_{\boldsymbol{\theta} \setminus \theta_k} + C,$$

where the expectation $\mathbb{E}_{\boldsymbol{\theta} \setminus \theta_k}$ is taken about the set $\boldsymbol{\theta}$ with θ_k removed. C is constant and can be obtained through normalization. Then, Eq. (B.5) can be reformulated as follows.

$$\begin{aligned} L(\boldsymbol{\theta}) &= \int q(\theta_k) \ln \tilde{p}(\theta_k, \mathbf{y})d\theta_k - \int q(\theta_k) \ln q(\theta_k)d\theta_k - \sum_{j \neq k} \int q(\theta_j) \ln q(\theta_j)d\theta_j + C \\ &= -\mathrm{KL}(q(\theta_k)||\tilde{p}(\theta_k, \mathbf{y})) - \sum_{j \neq k} \int q(\theta_j) \ln q(\theta_j)d\theta_j + C. \qquad \text{(B.6)} \end{aligned}$$

Clearly, Eq. (B.6) is maximized with respect to $q(\theta_k)$ when the KL distance is zero, which is the case for $q(\theta_k) = \tilde{p}(\theta_k, \mathbf{y})$. Consequently, the expression of the optimal posterior approximation $q_k(\theta_k)$ with other variables fixed is

$$\ln q(\theta_k) = \mathbb{E}[\ln p(\boldsymbol{\theta}, \mathbf{y})]_{\boldsymbol{\theta} \setminus \theta_k} + C.$$

B.2 Inference for Eq. (4.12)

Based on Eq. (4.9), $\ln q(\mathbf{x}) = \mathbb{E}[\ln p(\mathbf{y}|\mathbf{x};\lambda)p(\mathbf{x}|\boldsymbol{\alpha};\mathrm{B})]_{q(\boldsymbol{\alpha})} + const.$ Therefore,

$$\begin{aligned}\ln q(\mathbf{x}) &\propto \mathbb{E}[\ln p(\mathbf{y}|\mathbf{x};\lambda)p(\mathbf{x}|\boldsymbol{\alpha};\mathrm{B})]_{q(\boldsymbol{\alpha})} \\ &= \mathbb{E}[\ln\left(N(\mathbf{y}|\mathbf{D}\mathbf{x},\lambda\mathbf{I}_{ML})\prod_{i=1}^{N}N(\mathrm{x}_i|\mathbf{0},(\alpha_i\mathrm{B})^{-1})\right]_{q(\boldsymbol{\alpha})} \\ &\propto \mathbb{E}[-\frac{1}{2\lambda}(\mathbf{y}-\mathbf{D}\mathbf{x})(\mathbf{y}-\mathbf{D}\mathbf{x})^{\top}-\frac{1}{2}\mathbf{x}[\mathrm{AB}]\mathbf{x}]_{q(\boldsymbol{\alpha})}\end{aligned}$$

where $\mathrm{AB} = \mathrm{diag}[\alpha_1\mathrm{B},...,\alpha_N\mathrm{B}]$. Therefore, Eq. (4.12) is derived, and $q(\mathbf{x})$ follows Gaussian distribution as follows:

$$q(\mathbf{x}) \sim N(\mathbf{x}|\mu_{\mathbf{x}},\Sigma_{\mathbf{x}}),$$

where $\mu_{\mathbf{x}} = \frac{1}{\lambda}\Sigma_{\mathbf{x}}\mathbf{D}^{\top}\mathbf{y}$, $\Sigma_{\mathbf{x}} = (\frac{1}{\lambda}\mathbf{D}^{\top}\mathbf{D}+\mathbb{E}[\mathrm{AB}])^{-1}$

B.3 Inference for Eq. (4.13)

Based on Eq. (4.10), $\ln q(\boldsymbol{\alpha}) = \mathbb{E}[\ln p(\mathbf{x}|\boldsymbol{\alpha};\mathrm{B})p(\boldsymbol{\alpha}|\bar{b})]_{q(\mathbf{x})q(\bar{b})} + const.$ Therefore,

$$\begin{aligned}\ln q(\boldsymbol{\alpha}) &\propto \mathbb{E}[\ln p(\mathbf{x}|\boldsymbol{\alpha};\mathrm{B})p(\boldsymbol{\alpha}|\bar{b})]_{q(\mathbf{x})q(\bar{b})} \\ &= \mathbb{E}[\ln\left(\prod_{i=1}^{N}N(x_i|0,(\alpha_i\mathrm{B})^{-1})\prod_{i=1}^{N}\mathrm{Gamma}(\alpha_i|a,b_i)\right]_{q(\mathbf{x})q(\bar{b})} \\ &\propto \mathbb{E}[\Sigma_{i=1}^{N}\ln{\alpha_i}^{\frac{L}{2}}-(\frac{\alpha}{2}\mathrm{x}_i^{\top}\mathrm{B}\mathrm{x}_i)+\ln\alpha_i^{a-1}-b_i\alpha_i]_{q(\mathbf{x})q(\bar{b})} \\ &\propto \ln\alpha_i^{\frac{L}{2}+a-1}-\frac{\alpha_i}{2}\mathbb{E}[\mathrm{x}_i^{\top}\mathrm{B}\mathrm{x}_i]-\mathbb{E}[b_i]\alpha_i,\end{aligned}$$

where

$$\begin{aligned}\mathbb{E}[\mathrm{x}_i^\top \mathrm{B}\mathrm{x}_i] &= \mathbb{E}[\mathrm{Tr}(\mathrm{x}_i^\top \mathrm{B}\mathrm{x}_i)] \\ &= \mathbb{E}[\mathrm{Tr}(\mathrm{x}_i^\top \mathrm{x}_i \mathrm{B})] \\ &= \mathrm{Tr}(\mathbb{E}[\mathrm{x}_i^\top \mathrm{x}_i]\mathrm{B}) \\ &= \mathrm{Tr}[(\Sigma_{\mathrm{x}_i} + \mathbb{E}[\mathrm{x}_i]\mathbb{E}[\mathrm{x}_i]^\top)\mathrm{B}].\end{aligned}$$

Therefore, Eq. (4.13) is derived, and $q(\alpha_i), i \in P$ follows Gamma distribution as follows:

$$q(\alpha_i) = \mathrm{Gamma}(\alpha_i|\frac{L}{2} + a, \frac{\mathrm{Tr}[(\Sigma_{\mathrm{x}_i} + \mathbb{E}[\mathrm{x}_i]\mathbb{E}[\mathrm{x}_i]^\top)\mathrm{B}]}{2} + \mathbb{E}[b_i]).$$

For $q(\alpha_i), i \in P^c$ follows Gamma distribution as follows:

$$q(\alpha_i) = \mathrm{Gamma}(\alpha_i|\frac{L}{2} + a, \frac{\mathrm{Tr}[(\Sigma_{\mathrm{x}_i} + \mathbb{E}[\mathrm{x}_i]\mathbb{E}[\mathrm{x}_i]^\top)\mathrm{B}]}{2} + b_i).$$

B.4 Inference for Eq. (4.14)

Based on Eq. (4.11), $\ln q(\bar{b}) = \mathbb{E}[\ln p(\boldsymbol{\alpha}|\bar{b})p(\bar{b})]_{q(\mathbf{x})q(\boldsymbol{\alpha})} + const$. Therefore,

$$\begin{aligned}\ln q(\bar{b}) &\propto \mathbb{E}[\ln p(\boldsymbol{\alpha}|\bar{b})p(\bar{b})]_{q(\mathbf{x})q(\boldsymbol{\alpha})} \\ &= \mathbb{E}[\ln \prod_{i\in P} \mathrm{Gamma}(\alpha_i|a, b_i) \prod_{i\in P} \mathrm{Gamma}(b_i|p, q)]_{q(\mathbf{x})q(\boldsymbol{\alpha})} \\ &\propto \Sigma_{i\in P}(-b_i\mathbb{E}[\alpha_i] + a\ln b_i + (p-1)\ln b_i - qb_i) \\ &= \Sigma_{i\in P}((p+a-1)\ln b_i - (q + \mathbb{E}[\alpha_i])b_i).\end{aligned}$$

Therefore Eq. (4.14) is derived, and $q(b_i), \forall i \in P$ follows Gamma distribution as follows:

$$q(b_i) = \text{Gamma}(b_i|p + a, \mathbb{E}[\alpha_i] + q), \forall i \in P.$$

B.5 Inference for Eq. (4.21)

To estimate B, let the second term in Eq. (4.20) as follows:

$$Q(\mathrm{B}) = \mathbb{E}[\ln p(\mathbf{x}|\boldsymbol{\alpha}; \mathrm{B})]_{q(\mathbf{x};\tilde{\boldsymbol{\theta}}^{OLD})q(\boldsymbol{\alpha};\tilde{\boldsymbol{\theta}}^{OLD})}.$$

Let Γ=diag$(\alpha_1^{-1}, ..., \alpha_N^{-1})$. It can be shown that

$$\begin{aligned}\ln p(\mathbf{x}|\boldsymbol{\alpha}; \mathrm{B}) &= -\frac{1}{2}\ln\left(|\Gamma^{-1}|^L|\mathrm{B}^{-1}|^N\right) - \frac{1}{2}\mathbf{x}^\top(\Gamma \otimes \mathrm{B})\mathbf{x} \\ &= -\frac{1}{2}\ln|\Gamma^{-1}|^L - \frac{1}{2}\ln|\mathrm{B}^{-1}|^N - \frac{1}{2}\mathbf{x}^\top(\Gamma \otimes \mathrm{B})\mathbf{x}. \end{aligned} \tag{B.7}$$

$Q(\mathrm{B})$ can be calculated by taking the expectation to Eq. (B.7). Then, taking derivative $Q(\mathrm{B})$ with respect to B leads to

$$\frac{\partial Q(\mathrm{B})}{\partial \mathrm{B}} = \frac{N}{2}\mathrm{B}^{-1} - \frac{1}{2}\sum_{i=1}^{N}\mathbb{E}[\alpha_i](\Sigma_{\mathrm{x}_i} + \mathbb{E}[\mathrm{x}_i]\mathbb{E}[\mathrm{x}_i]^\top). \tag{B.8}$$

By letting Eq. (B.8) equals to zero, B is estimated as follows:

$$\mathrm{B} = [\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[\alpha_i](\Sigma_{\mathrm{x}_i} + \mathbb{E}[\mathrm{x}_i]\mathbb{E}[\mathrm{x}_i]^\top)]^{-1}$$

B.6 Inference for Eq. (4.22)

To estimate λ, let the first term in Eq. (4.20) as follows:

$$Q(\lambda) = \mathbb{E}[\ln p(\mathbf{y}|\mathbf{x};\lambda)]_{q(\mathbf{x};\tilde{\boldsymbol{\theta}}^{OLD})q(\boldsymbol{\alpha};\tilde{\boldsymbol{\theta}}^{OLD})}.$$

It can be shown that

$$\begin{aligned}
Q(\lambda) &\propto -\frac{ML}{2}\ln\lambda - \frac{1}{2\lambda}\mathbb{E}_{q(\mathbf{x};\tilde{\boldsymbol{\theta}}^{OLD})q(\boldsymbol{\alpha};\tilde{\boldsymbol{\theta}}^{OLD})}\|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2^2 \\
&= -\frac{ML}{2}\ln\lambda - \frac{1}{2\lambda}\mathbb{E}_{q(\mathbf{x};\tilde{\boldsymbol{\theta}}^{OLD})q(\boldsymbol{\alpha};\tilde{\boldsymbol{\theta}}^{OLD})}[\|\mathbf{y}-\mathbf{D}\mu_{\mathbf{x}}\|_2^2 + \mathrm{Tr}(\Sigma_{\mathbf{x}}\mathbf{D}^\top\mathbf{D})] \\
&= -\frac{ML}{2}\ln\lambda - \frac{1}{2\lambda}[\|\mathbf{y}-\mathbf{D}\mu_{\mathbf{x}}\|_2^2 + \hat{\lambda}\mathbb{E}_{q(\boldsymbol{\alpha};\tilde{\boldsymbol{\theta}}^{OLD})}\mathrm{Tr}(\Sigma_{\mathbf{x}}(\Sigma_{\mathbf{x}}^{-1}-\Sigma_0^{-1}))] \qquad \text{(B.9)} \\
&= -\frac{ML}{2}\ln\lambda - \frac{1}{2\lambda}[\|\mathbf{y}-\mathbf{D}\mu_{\mathbf{x}}\|_2^2 + \hat{\lambda}[NL - \mathbb{E}_{q(\boldsymbol{\alpha};\tilde{\boldsymbol{\theta}}^{OLD})}\mathrm{Tr}(\Sigma_{\mathbf{x}}(\Sigma_0^{-1}))] \\
&= -\frac{ML}{2}\ln\lambda - \frac{1}{2\lambda}[\|\mathbf{y}-\mathbf{D}\mu_{\mathbf{x}}\|_2^2 + \hat{\lambda}[NL - \mathrm{Tr}(\Sigma_{\mathbf{x}}\mathbb{E}_{q(\boldsymbol{\alpha};\tilde{\boldsymbol{\theta}}^{OLD})}(\Sigma_0^{-1}))], \qquad \text{(B.10)}
\end{aligned}$$

where Eq. (B.9) follows Eq. (4.18), and $\hat{\lambda}$ denotes the estimated λ in the previous iteration. By setting the derivative of Eq. (B.10) over λ to zero, λ can be estimated as follows:

$$\lambda = \frac{[\|\mathbf{y}-\mathbf{D}\mu_{\mathbf{x}}\|_2^2 + \hat{\lambda}[NL - \mathrm{Tr}(\Sigma_{\mathbf{x}}\mathbb{E}_{q(\boldsymbol{\alpha};\tilde{\boldsymbol{\theta}}^{OLD})}(\Sigma_0^{-1}))]]}{ML}.$$

B.7 Standard deviations and quantile statistics in each case study.

The standard deviations and quantile statistics of failure rate and NMSE in the case studies Sections 4.3 and 4.4 are provided. The failure rate is the percentage of failed trials with mis-

classified process faults from the total trials. Specifically, failed trials denote one; otherwise, zero. Tables B.1 to B.5 represent standard deviations and 0.25, 0.5, and 0.75 quantile results of each method of Tables 4.1 to 4.5. The proposed method provides a large standard deviation in most cases compared to the benchmark methods. This is because the distributions of both measures from the repetitive trials of the proposed method are highly skewed to zero. Since the zero FR and NMSE represent the perfect performance, these results represent the low failure rate and NMSE of the proposed method. Specifically, the quantile statistics of the proposed method achieve the lowest value in most case studies compared to benchmark methods. In addition, the proposed method provides the lowest median and interquartile range in most case studies.

Table B.1: Standard deviations and quantile statistics (in the parenthesis) of Table 4.1.

	β	0.0	0.3	0.6	0.9	0.95
TMSBL	FR	0.49, (0, 0, 1)	0.48, (0, 0, 1)	0.48, (0, 0, 1)	0.44, (0, 0, 0)	0.40, (0, 0, 0)
	NMSE	1.30, (0.0, 0.0, 1.63)	1.23, (0.0, 0.0, 1.57)	1.05, (0.0, 0.0, 1.60)	1.11, (0.0, 0.0, 0.26)	0.70, (0.0, 0.0, 0.0)
MSBL	FR	0.49, (0, 1, 1)	0.47, (0, 1, 1)	0.44, (0, 1, 1)	0.22, (1, 1, 1)	0.20, (1, 1, 1)
	NMSE	0.75, (0.0, 0.74, 1.5)	0.73, (0.0, 1.0, 1.44)	0.73, (0.0, 1.02, 1.47)	0.55, (0.89, 1.26, 1.68)	0.55, (0.97, 1.33, 1.62)
SA-MSBL	FR	0.20, (1, 1, 1)	0.21, (1, 1, 1)	0.20, (1, 1, 1)	0.18, (1, 1, 1)	0.19, (1, 1, 1)
	NMSE	0.37, (0.40, 0.72, 0.97)	0.37, (0.43, 0.70, 0.96)	0.38, (0.48, 0.71, 1.0)	0.33, (0.52, 0.74, 0.96)	0.34, (0.50, 0.73, 0.93)
SA-SBL	FR	0.17, (1, 1, 1)	0.18, (1, 1, 1)	0.16, (1, 1, 1)	0.18, (1, 1, 1)	0.20, (1, 1, 1)
	NMSE	0.40, (0.40, 0.73, 1.00)	0.40, (0.44, 0.72, 0.99)	0.39, (0.48, 0.71, 0.98)	0.34, (0.52, 0.74, 0.96)	0.34, (0.52, 0.73, 0.94)
[75]	FR	0.25, (1, 1, 1)	0.26, (1, 1, 1)	0.22, (1, 1, 1)	0.11, (1, 1, 1)	0.10, (1, 1, 1)
	NMSE	0.30, (0.51, 0.72, 0.89)	0.29, (0.49, 0.73, 0.89)	0.31, (0.54, 0.76, 0.93)	0.30, (0.69, 0.84, 1.0)	0.36, (0.74, 0.89, 1.06)
SCBL	FR	0.24, (1, 1, 1)	0.20, (1, 1, 1)	0.20, (1, 1, 1)	0.10, (1, 1, 1)	0.0, (1, 1, 1)
	NMSE	0.69, (0.82, 1.36, 1.67)	0.62, (0.99, 1.4, 1.69)	0.63, (1.09, 1.32, 1.69)	0.48, (1.12, 1.43, 1.64)	0.49, (1.13, 1.45, 1.76)
SA-TSBL (Proposed)	FR	0.42, (0, 0, 0)	0.43, (0, 0, 0)	0.42, (0, 0, 0)	0.38, (0, 0, 0)	0.35, (0, 0, 0)
	NMSE	0.99, (0.0, 0.0, 0.0)	0.98, (0.0, 0.0, 0.0)	0.84, (0.0, 0.0, 0.0)	0.80, (0.0, 0.0, 0.0)	0.64, (0.0, 0.0, 0.0)

Table B.2: Standard deviations and quantile statistics (in the parenthesis) of Table 4.2.

	L	2	3	4
TMSBL	FR	0.49, (0, 1, 1)	0.38, (0, 0, 0)	0.34, (0, 0, 0)
	NMSE	0.76, (0.0, 0.69, 1.35)	0.49, (0.0, 0.0, 0.01)	0.41, (0.0, 0.0, 0.01)
MSBL	FR	0.38, (1, 1, 1)	0.42, (1, 1, 1)	0.50, (0, 1, 1)
	NMSE	0.68, (0.73, 1.11, 1.56)	0.72, (0.51, 1.13, 1.52)	0.71, (0.0, 0.56, 1.22)
SA-MSBL	FR	0.33, (1, 1, 1)	0.30, (1, 1, 1)	0.30, (1, 1, 1)
	NMSE	0.37, (0.40, 0.62, 0.90)	0.35, (0.42, 0.61, 0.87)	0.32, (0.44, 0.64, 0.87)
SA-SBL	FR	0.31, (1, 1, 1)	0.30, (1, 1, 1)	0.29, (1, 1, 1)
	NMSE	0.36, (0.42, 0.64, 0.91)	0.33, (0.42, 0.62, 0.86)	0.37, (0.45, 0.65, 0.90)
[75]	FR	0.46, (0, 1, 1)	0.50, (0, 1, 1)	0.50, (0, 0, 1)
	NMSE	0.44, (0.12, 0.73, 0.95)	0.60, (0.0, 0.26, 0.96)	0.80, (0.0, 0.03, 0.99)
SCBL	FR	0.17, (1, 1, 1)	0.27, (1, 1, 1)	0.26, (1, 1, 1)
	NMSE	0.52, (0.91, 1.27, 1.56)	0.61, (0.98, 1.27, 1.62)	0.53, (0.89, 1.23, 1.53)
SA-TSBL (Proposed)	FR	0.50, (0, 0, 1)	0.32, (0, 0, 0)	0.31, (0, 0, 0)
	NMSE	0.66, (0.0, 0.02, 0.98)	0.44, (0.0, 0.0, 0.01)	0.31, (0.0, 0.0, 0.01)

Table B.3: Standard deviations and quantile statistics (in the parenthesis) of Table 4.3.

	N/M	3	5	7	9
TMSBL	FR	0.29, (0, 0, 0)	0.45, (0, 0, 1)	0.44, (0, 0, 1)	0.46, (0, 0, 1)
	NMSE	0.21, (0.0, 0.01, 0.01)	0.58, (0.0, 0.01, 0.57)	0.63, (0.0, 0.01, 0.71)	0.74, (0.0, 0.01, 0.89)
MSBL	FR	0.46, (0, 0, 1)	0.50, (0, 1, 1)	0.48, (0, 1, 1)	0.46, (0, 1, 1)
	NMSE	0.53, (0.0, 0.0, 0.34)	0.62, (0.0, 0.63, 1.14)	0.67, (0.0, 0.82, 1.26)	0.65, (0.01, 1.06, 1.33)
SA-MSBL	FR	0.46, (0, 0, 1)	0.50, (0, 1, 1)	0.48, (0, 1, 1)	0.44, (0, 1, 1)
	NMSE	0.36, (0.0, 0.02, 0.42)	0.40, (0.05, 0.38, 0.73)	0.40, (0.11, 0.47, 0.73)	0.36, (0.25, 0.54, 0.80)
SA-SBL	FR	0.49, (0, 1, 1)	0.39, (1, 1, 1)	0.39, (1, 1, 1)	0.36, (1, 1, 1)
	NMSE	0.38, (0.0, 0.25, 0.63)	0.34, (0.28, 0.51, 0.73)	0.32, (0.34, 0.53, 0.76)	0.28, (0.37, 0.55, 0.78)
[75]	FR	0.31, (0, 0, 0)	0.42, (0, 0, 0)	0.50, (0, 0, 1)	0.50, (0, 1, 1)
	NMSE	0.32, (0.0, 0.0, 0.0)	0.48, (0.0, 0.0, 0.0)	0.70, (0.0, 0.0, 1.10)	0.76, (0.0, 0.51, 1.24)
SCBL	FR	0.50, (0, 0, 1)	0.41, (1, 1, 1)	0.38, (1, 1, 1)	0.27, (1, 1, 1)
	NMSE	0.62, (0.0, 0.0, 1.01)	0.60, (0.57, 0.95, 1.3)	0.59, (0.76, 1.05, 1.38)	0.49, (0.91, 1.19, 1.44)
SA-TSBL (Proposed)	FR	0.24, (0, 0, 0)	0.38, (0, 0, 0)	0.39, (0, 0, 0)	0.40, (0, 0, 0)
	NMSE	0.11, (0.0, 0.01, 0.01)	0.42, (0.0, 0.01, 0.02)	0.51, (0.0, 0.01, 0.03)	0.50, (0.0, 0.01, 0.02)

Table B.4: Standard deviations and quantile statistics (in the parenthesis) of Table 4.4.

	L	2	3	4
TMSBL	FR	0.49, (0, 1, 1)	0.48, (0, 1, 1)	0.50, (0, 0, 1)
	NMSE	0.40, (0.07, 0.17, 0.38)	0.39, (0.12, 0.17, 0.34)	0.39, (0.0, 0.17, 0.32)
MSBL	FR	0.48, (0, 1, 1)	0.49, (0, 1, 1)	0.50, (0, 0, 1)
	NMSE	0.41, (0.16, 0.18, 0.5)	0.40, (0.12, 0.17, 0.35)	0.29, (0.0, 0.17, 0.33)
SA-MSBL	FR	0.50, (0, 1, 1)	0.50, (0, 1, 1)	0.50, (0, 0, 1)
	NMSE	0.39, (0.02, 0.19, 0.65)	0.37, (0.04, 0.19, 0.64)	0.35, (0.01, 0.18, 0.52)
SA-SBL	FR	0.50, (0, 1, 1)	0.49, (0, 1, 1)	0.50, (0, 1, 1)
	NMSE	0.37, (0.06, 0.23, 0.61)	0.35, (0.16, 0.25, 0.62)	0.34, (0.1, 0.24, 0.52)
[75]	FR	0.28, (1, 1, 1)	0.24, (1, 1, 1)	0.19, (1, 1, 1)
	NMSE	5.20, (1.05, 1.94, 4.42)	3.17, (1.07, 2.07, 3.91)	4.53, (1.18, 2.32, 4.44)
SCBL	FR	0.48, (0, 1, 1)	0.46, (0, 1, 1)	0.50, (0, 0, 1)
	NMSE	0.47, (0.16, 0.26, 0.63)	0.46, (0.16, 0.21, 0.50)	0.33, (0.02, 0.18, 0.36)
SA-TSBL (Proposed)	FR	0.49, (0, 0, 1)	0.50, (0, 0, 1)	0.48, (0, 0, 1)
	NMSE	0.41, (0.0, 0.17, 0.33)	0.40, (0.0, 0.17, 0.33)	0.39, (0.0, 0.07, 0.18)

Table B.5: Standard deviations and quantile statistics (in the parenthesis) of Table 4.5.

	β	0.0	0.3	0.6	0.9	0.99
TMSBL	FR	0.49, (0, 1, 1)	0.49, (0, 1, 1)	0.50, (0, 1, 1)	0.50, (0, 1, 1)	0.50, (0, 1, 1)
	NMSE	0.24, (0.0, 0.09, 0.32)	0.26, (0.0, 0.14, 0.32)	0.26, (0.0, 0.15, 0.33)	0.19, (0.0, 0.17, 0.31)	0.23, (0.0, 0.17, 0.32)
MSBL	FR	0.49, (0, 1, 1)	0.50, (0, 1, 1)	0.50, (0, 1, 1)	0.49, (0, 1, 1)	0.49, (0, 1, 1)
	NMSE	0.22, (0.0, 0.09, 0.32)	0.23, (0.0, 0.14, 0.32)	0.26, (0.0, 0.15, 0.33)	0.26, (0.0, 0.19, 0.33)	0.33, (0.05, 0.18, 0.34)
SA-MSBL	FR	0.50, (0, 1, 1)	0.49, (0, 1, 1)	0.50, (0, 1, 1)	0.50, (0, 1, 1)	0.50, (0, 1, 1)
	NMSE	0.43, (0.01, 0.12, 0.40)	0.40, (0.01, 0.18, 0.39)	0.37, (0.01, 0.19, 0.39)	0.38, (0.01, 0.19, 0.43)	0.41, (0.01, 0.19, 0.61)
SA-SBL	FR	0.42, (1, 1, 1)	0.43, (1, 1, 1)	0.44, (0, 1, 1)	0.47, (0, 1, 1)	0.49, (0, 1, 1)
	NMSE	0.42, (0.16, 0.4, 0.64)	0.39, (0.17, 0.38, 0.61)	0.40, (0.15, 0.33, 0.57)	0.37, (0.13, 0.29, 0.52)	0.37, (0.17, 0.28, 0.6)
[75]	FR	0.06, (1, 1, 1)	0.08, (1, 1, 1)	0.09, (1, 1, 1)	0.10, (1, 1, 1)	0.15, (1, 1, 1)
	NMSE	109.6, (3.76, 9.29, 20.97)	46.88, (2.89, 6.72, 14.57)	15.06, (2.17, 4.53, 9.39)	5.45, (1.65, 3.13, 5.89)	3.58, (1.38, 2.39, 4.37)
SCBL	FR	0.46, (0, 1, 1)	0.45, (0, 1, 1)	0.47, (0, 1, 1)	0.46, (0, 1, 1)	0.48, (0, 1, 1)
	NMSE	0.45, (0.0, 0.14, 0.38)	0.42, (0.01, 0.21, 0.41)	0.34, (0.01, 0.19, 0.42)	0.33, (0.07, 0.22, 0.39)	0.35, (0.08, 0.18, 0.38)
SA-TSBL (Proposed)	FR	0.50, (0, 1, 1)	0.50, (0, 1, 1)	0.50, (0, 0, 1)	0.50, (0, 0, 1)	0.49, (0, 0, 1)
	NMSE	0.23, (0.0, 0.06, 0.26)	0.26, (0.0, 0.1, 0.3)	0.25, (0.0, 0.1, 0.26)	0.18, (0.0, 0.14, 0.25)	0.22, (0.0, 0.16, 0.22)

B.8 Fault pattern matrix Φ in Section 4.4

Figure B.1 shows the fault pattern matrix Φ from previous researches [10, 59, 70] that used in Section 4.4.

$$
\begin{bmatrix}
0 & 0 & 0 & 0 & -0.030 & 0.030 & 0 & 0.420 & -0.420 & 0 & 0 \\
0.043 & 0 & -0.043 & 0 & -0.027 & 0.027 & 0 & 0.373 & -0.373 & 0 & 0 \\
0.051 & 0 & -0.051 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.422 & -0.422 \\
0.043 & 0 & -0.043 & 0 & -0.073 & -0.009 & 0.010 & 0.871 & 0.124 & 0.500 & 0.500 \\
-0.051 & 0 & 0.051 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.013 & -0.065 & -0.013 & 0 & 0 & 0 & 0 & -0.002 & 0.002 & 0 & 0 \\
0.001 & 0.023 & -0.001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.202 & 0.995 & 0.202 & 0 & -0.001 & 0.001 & 0 & 0.012 & -0.012 & 0 & 0 \\
-0.013 & 0 & 0.013 & -0.065 & 0 & 0 & 0 & 0.001 & 0 & -0.001 & 0.002 \\
0.001 & 0 & -0.001 & -0.023 & 0 & 0 & 0 & 0 & 0 & 0.001 & 0 \\
0.202 & 0 & -0.202 & 0.995 & 0.002 & 0 & 0 & 0.020 & -0.003 & -0.008 & -0.015
\end{bmatrix}
$$

$$
\begin{bmatrix}
-5.88 & 0.588 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.523 & 0.523 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.588 & -0.588 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.523 & 0.523 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.003 & -0.003 & 0.065 & -0.040 & -0.012 & -0.013 & -1.082 & 0.086 & -0.002 & 0.498 & 0.498 \\
0 & 0 & -0.023 & -0.068 & 0.019 & 0.071 & -1.000 & 0.999 & -0.999 & 0 & 0 \\
-0.017 & 0.017 & -0.995 & -0.001 & -0.001 & -0.003 & -0.048 & -0.017 & 0.023 & 0.033 & 0.033 \\
-0.003 & 0.003 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.017 & -0.017 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

$$
\begin{bmatrix}
0 & 0 & 0.060 & -0.060 & 0 & 1.588 & -0.588 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.043 & 0 & -0.043 & 0.523 & -0.523 & 1.000 & 0 & 0 & 0 \\
0 & 0 & 0.051 & 0.500 & -0.051 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.060 & -0.060 & 0 & -0.588 & 1.588 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.043 & 0 & -0.043 & 0.523 & -0.523 & 1.000 & 0 & 0 & 0 \\
0 & 0 & -0.551 & 0.500 & 1.051 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.002 & 0 & 0.102 & -0.102 & 0 & 1.083 & -0.083 & 0 & 0 & 0 & 0 \\
-0.999 & 0 & 0.072 & 0 & -0.072 & 0.999 & -0.999 & 1.000 & 0 & 0 & 0 \\
0.023 & 0 & -0.046 & 0.841 & 0.205 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.065 & 0.101 & -0.114 & -0.052 & 0.003 & 0.0001 & 0.002 & 0.498 & 0.498 & -0.002 \\
0 & 0.023 & 0.021 & -0.019 & -0.024 & 0 & 0.002 & 0.001 & 0 & 0 & 0.999 \\
0 & -0.995 & -0.638 & 0.840 & 0.793 & -0.017 & -0.048 & -0.023 & 0.033 & 0.033 & 0.023
\end{bmatrix}
$$

Figure B.1: Fault pattern matrix Φ in Section 4.4 [12].

Appendix C

Appendix of Chapter 5

C.1 Proof of Lemma 5.1

Lemma 5.1 *For any policy π, the operator $B^{\pi}[CG(s,a)]$ is a contraction under the supremum norm over s, a. That is, for any $CG1(s,a)$ and $CG2(s,a)$, it follows*

$$||B^{\pi}[CG1(s,a)] - B^{\pi}[CG2(s,a)]||_{\infty} \leq \gamma ||CG1(s,a) - CG2(s,a)||_{\infty}.$$

proof.

$$
\begin{aligned}
&||B^{\pi}[CG1(s,a)] - B^{\pi}[CG2(s,a)]||_{\infty} \\
&= \max_{(s,a)} |k^{\pi}(s,a) + \gamma \sum_{s',a'} p(s'|s,a)\pi(a'|s')CG1(s',a') \\
&- k^{\pi}(s,a) + \gamma \sum_{s',a'} p(s'|s,a)\pi(a'|s')CG2(s',a')| \\
&= \max_{(s,a)} |\gamma \sum_{s',a'} p(s'|s,a)\pi(a'|s')[CG1(s',a') - CG2(s',a')]| \\
&\leq \max_{(s,a)} \gamma \sum_{s',a'} p(s'|s,a)\pi(a'|s')|[CG1(s',a') - CG2(s',a')]| \\
&\leq \max_{(s,a)} \gamma \sum_{s',a'} p(s'|s,a)\pi(a'|s') \max_{(s',a')} |[CG1(s',a') - CG2(s',a')]| \\
&= \gamma \max_{(s',a')} |CG1(s',a') - CG2(s',a')| \\
&= \gamma ||CG1(s,a) - CG2(s,a)||_{\infty}
\end{aligned}
$$

where the first equality is based on the definition of the operator in Eq. (5.13), the first inequality is due to the triangle inequality, the second inequality is based on the property of the max operator, and the third equality is because of the property of p.m.f. function.

C.2 Proof of Theorem 5.2

Theorem 5.2 *Supposed that* $0 < \rho_{min} \leq \rho_1(a|s), \rho_2(a|s) \leq \rho_{max} < 1$ *for all* (s,a) *and* $\alpha_t(s_t,a_t) = n_t(s_t,a_t)^{-w}$ *for* $w \in (0.5, 1]$. *Then, three assumptions in Theorem 2 in [94] are satisfied where* γ *is the discount factor and* $K = max\{K' + \max_{s\in S, a\in A} CG^*(s,a)^2, 64\gamma^2\}$. K' *is defined as* $2\mathbb{E}[R(s,a,s') - CG^*(s,a)]^2 + 4\gamma^2(\frac{\beta_1\beta_2}{\beta_1+\beta_2})^2[2(\log|A|)^2 + 4(\log\rho_{min})^2 + 4(\log\rho_{max})^2$.

Therefore,

$$\lim_{t\to\infty} \Delta_t(s,a) \xrightarrow{w.p.1} 0$$

According to Theorem 2 in [94], it is sufficient to prove that a random iterative process convergence of a random iterative process $\Delta_{(t+1)}(s_t, a_t)$ converges to zero $w.p.1$ under the following assumptions:

1. $0 \leq \alpha_t(s,a) \leq 1, \sum_t \alpha_t(s,a) = \infty$ and $\sum_t \alpha_t(s,a)^2 < \infty, \forall s \in S, a \in A$;

2. $||\mathbb{E}[F_t(s,a)|U_t]||_\infty \leq \gamma ||\Delta_t||_\infty$, with $\gamma < 1$;

3. $\text{var}[F_t(s,a)|U_t] \leq K(1 + ||\Delta_t||_\infty^2)$, for $K > 0$,

where $U_t = \{\Delta_t, \Delta_{t-1}, ..., \Delta_0, F_{t-1}, ..., F_0\}$. $||\cdot||_\infty$ refers to supremum norm, and, and K is some constant.

Proof for Assumption 1:

Learning rate $\alpha_t(s_t, a_t)$ is defined as $n_t(s_t, a_t)^{-w}, w \in (0.5, 1]$. It satisfies the assumption 1 [39].

Proof for Assumption 2:

Let

$$B^*[CG(s,a)] = \min_\pi B^\pi[CG(s,a)]$$

where the optimum is achieved at

$$\pi(a|s) = \frac{\rho_1(a|s)^{\frac{\beta_2}{\beta_1+\beta_2}} \rho_2(a|s)^{\frac{\beta_1}{\beta_1+\beta_2}} e^{-CG(s,a)\frac{\beta_1\beta_2}{\beta_1+\beta_2}}}{\sum_a \rho_1(a|s)^{\frac{\beta_2}{\beta_1+\beta_2}} \rho_2(a|s)^{\frac{\beta_1}{\beta_1+\beta_2}} e^{-CG(s,a)\frac{\beta_1\beta_2}{\beta_1+\beta_2}}}. \tag{C.1}$$

Then,

$$\mathbb{E}[F_t(s,a)|U_t]$$
$$= \sum_{s' \in S} p(s'|s,a)[R(s,a,s') - CG^*(s,a) - \gamma \frac{\beta_1+\beta_2}{\beta_1\beta_2} \log(\sum_{a'} \rho_1(a'|s)^{\frac{\beta_2}{\beta_1+\beta_2}} \rho_2(a'|s)^{\frac{\beta_1}{\beta_1+\beta_2}} e^{-CG_t(s',a')\frac{\beta_1\beta_2}{\beta_1+\beta_2}}]$$
$$= B^*(CG_t(s,a)) - CG^*(s,a)$$

where the second equality is obtained by plugging Eq. (C.1) into Eq. (5.13). Therefore,

$$\begin{aligned} ||\mathbb{E}[F_t(s,a)|U_t]||_\infty &= ||B^*(CG_t(s,a)) - CG^*(s,a)||_\infty \\ &= ||B^*(CG_t(s,a)) - B^*(CG^*(s,a))||_\infty \\ &\leq \gamma ||CG_t(s,a) - CG^*(s,a)||_\infty \\ &= \gamma ||\Delta_t||_\infty, \end{aligned}$$

where the second equality comes from the fact that operator B^* has contraction property based on Lemma 5.1 and monotonicity property over $CG_t(s,a)$. Both properties guarantees that applying the operator B^* converges to unique optimal fixed point [30]. First inequality is based on Lemma 5.1.

Proof for Assumption 3:

Assuming that $0 < \rho_{min} \leq \rho_1(a|s), \rho_2(a|s) \leq \rho_{max} < 1, \forall(s,a)$.

$$\text{var}[F_t(s,a)|U_t]$$
$$\leq \mathbb{E}[F_t(s,a)]^2$$
$$= \mathbb{E}[((R(s,a,s') - CG^*(s,a) - \gamma \frac{\beta_1+\beta_2}{\beta_1\beta_2} \log(\sum_{a'} \rho_1(a'|s)^{\frac{\beta_2}{\beta_1+\beta_2}} \rho_2(a'|s)^{\frac{\beta_1}{\beta_1+\beta_2}} e^{-CG_t(s',a')\frac{\beta_1\beta_2}{\beta_1+\beta_2}}))^2]$$
$$\leq 2\mathbb{E}[R(s,a,s') - CG^*(s,a)]^2 + 2\gamma^2(\frac{\beta_1+\beta_2}{\beta_1\beta_2})^2 \mathbb{E}(\log(\sum_{a'} \rho_1(a'|s)^{\frac{\beta_2}{\beta_1+\beta_2}} \rho_2(a'|s)^{\frac{\beta_1}{\beta_1+\beta_2}} e^{-CG_t(s',a')\frac{\beta_1\beta_2}{\beta_1+\beta_2}}))^2$$

where the first inequality is due to the definition of variance and the second inequality is based on Cauchy–Schwarz inequality. Next, we will estimate the upper bound for the term in right hand side of above inequality.

$$\begin{aligned}
&\mathbb{E}(\log(\sum_{a'}\rho_1(a'|s')^{\frac{\beta_2}{\beta_1+\beta_2}}\rho_2(a'|s')^{\frac{\beta_1}{\beta_1+\beta_2}}e^{-CG_t(s',a')\frac{\beta_1\beta_2}{\beta_1+\beta_2}}))^2\\
&\leq \max_{s'\in S}(\log(\sum_{a'}\rho_1(a'|s')^{\frac{\beta_2}{\beta_1+\beta_2}}\rho_2(a'|s')^{\frac{\beta_1}{\beta_1+\beta_2}}e^{-CG_t(s',a')\frac{\beta_1\beta_2}{\beta_1+\beta_2}}))^2\\
&\leq \max_{s'\in S}(\log(|A|\min_{a'\in A}\rho_1(a'|s')^{\frac{\beta_2}{\beta_1+\beta_2}}\rho_2(a'|s')^{\frac{\beta_1}{\beta_1+\beta_2}}e^{-CG_t(s',a')\frac{\beta_1\beta_2}{\beta_1+\beta_2}}))^2\\
&+ \max_{s'\in S}(\log(|A|\max_{a'\in A}\rho_1(a'|s')^{\frac{\beta_2}{\beta_1+\beta_2}}\rho_2(a'|s')^{\frac{\beta_1}{\beta_1+\beta_2}}e^{-CG_t(s',a')\frac{\beta_1\beta_2}{\beta_1+\beta_2}}))^2
\end{aligned} \tag{C.2}$$

where the first inequality is based on the property of max operator and the second inequality is derived by considering the range of the value in the square sign. Next, we will try to bound the first term of the last inequality in Eq. (C.2).

$$\begin{aligned}
&(\log([|A|\min_{a'\in A}\rho_1(a'|s')^{\frac{\beta_2}{\beta_1+\beta_2}}\rho_2(a'|s')^{\frac{\beta_1}{\beta_1+\beta_2}}e^{-CG_t(s',a')\frac{\beta_1\beta_2}{\beta_1+\beta_2}}))^2\\
&\leq 2(\log|A|)^2+2(\log(\min_{a'\in A}\rho_1(a'|s')^{\frac{\beta_2}{\beta_1+\beta_2}}\rho_2(a'|s')^{\frac{\beta_1}{\beta_1+\beta_2}}e^{-CG_t(s',a')\frac{\beta_1\beta_2}{\beta_1+\beta_2}}))^2\\
&\leq 2(\log|A|)^2+2(\log(\min_{a'\in A}\rho_{min}(a'|s')^{\frac{\beta_2}{\beta_1+\beta_2}}\rho_{min}(a'|s')^{\frac{\beta_1}{\beta_1+\beta_2}}e^{-CG_t(s',a')\frac{\beta_1\beta_2}{\beta_1+\beta_2}}))^2\\
&+2(\log(\min_{a'\in A}\rho_{max}(a'|s')^{\frac{\beta_2}{\beta_1+\beta_2}}\rho_{max}(a'|s')^{\frac{\beta_1}{\beta_1+\beta_2}}e^{-CG_t(s',a')\frac{\beta_1\beta_2}{\beta_1+\beta_2}}))^2\\
&=2(\log|A|)^2+2(\log\rho_{min}+\log\min_{a'\in A}e^{-CG_t(s',a')\frac{\beta_1\beta_2}{\beta_1+\beta_2}})^2+2(\log\rho_{max}+\log\min_{a'\in A}e^{-CG_t(s',a')\frac{\beta_1\beta_2}{\beta_1+\beta_2}})^2\\
&\leq 2(\log|A|)^2+4(\log\rho_{min})^2+4(\log\rho_{max})^2+8(\frac{\beta_1\beta_2}{\beta_1+\beta_2})^2(\max_{a'\in A}CG_t(s',a'))^2\\
&\leq 2(\log|A|)^2+4(\log\rho_{min})^2+4(\log\rho_{max})^2+8(\frac{\beta_1\beta_2}{\beta_1+\beta_2})^2\max_{a'\in A}CG_t(s',a')^2
\end{aligned}$$

where the first inequality comes from Cauchy–Schwarz inequality, the second inequality is derived by considering the range of ρ_1, ρ_2 and the third inequality is due to the Cauchy–Schwarz inequality. Similarly, for the second term of the last inequality in Eq. (C.2), we

have

$$
\begin{aligned}
&(\log([|A|\max_{a'\in A}\rho_1(a'|s')^{\frac{\beta_2}{\beta_1+\beta_2}}\rho_2(a'|s')^{\frac{\beta_1}{\beta_1+\beta_2}}e^{-CG_t(s',a')\frac{\beta_1\beta_2}{\beta_1+\beta_2}}))^2\\
&\leq 2(\log|A|)^2+4(\log\rho_{min})^2+4(\log\rho_{max})^2+8(\frac{\beta_1\beta_2}{\beta_1+\beta_2})^2(\max_{a'\in A}CG_t(s',a'))^2\\
&\leq 2(\log|A|)^2+4(\log\rho_{min})^2+4(\log\rho_{max})^2+8(\frac{\beta_1\beta_2}{\beta_1+\beta_2})^2\max_{a'\in A}CG_t(s',a')^2
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
&\mathrm{var}[F_t(s,a)|U_t]\\
&\leq 2\mathbb{E}[R(s,a,s')-CG^*(s,a)]^2+4\gamma^2(\frac{\beta_1+\beta_2}{\beta_1\beta_2})^2\\
&\times\max_{s'\in S}[2(\log|A|)^2+4(\log\rho_{min})^2+4(\log\rho_{max})^2+8(\frac{\beta_1\beta_2}{\beta_1+\beta_2})^2\max_{a'\in A}CG(s',a')^2]\\
&=K'+32\gamma^2\max_{s'\in S,a'\in A}CG(s',a')^2\\
&\leq K'+64\gamma^2(\max_{s'\in S,a'\in A}(CG(s',a')-CG^*(s',a'))^2+CG^*(s',a')^2)\\
&\leq K'+64\gamma^2\max_{s\in S,a\in A}CG^*(s,a)^2+64\gamma^2\max_{s'\in S,a'\in A}(CG_t(s',a')-CG^*(s',a'))^2\\
&\leq K(1+||\Delta_t||_\infty^2)
\end{aligned}
$$

where $K'=2\mathbb{E}[R(s,a,s')-CG^*(s,a)]^2+4\gamma^2(\frac{\beta_1\beta_2}{\beta_1+\beta_2})^2[2(\log|A|)^2+4(\log\rho_{min})^2+4(\log\rho_{max})^2]$ and $K=\max\{K'+\max_{s\in S,a\in A}CG^*(s,a)^2,64\gamma^2\}$. The second inequality comes from Cauchy–Schwarz inequality.

Therefore, the three assumptions in Theorem 2 [94] are satisfied so that the proof of Theorem 5.2 is finished. For the generalization in terms of number of prior polices, we assume that there exist M prior polices. The optimal policy can be achieved as follows:

$$
\pi_{CG}=\frac{H}{\sum_a H}, \tag{C.3}
$$

where

$$H = \exp(\frac{1}{\sum_{i=1}^{M} \beta_i^{-1}}(\sum_{i=1}^{M} \frac{1}{\beta_i} \log \rho_i - CG(s,a)))$$

All the proof that we provided in the appendix can be directly applied to M prior polices case with the same conclusion by plugging Eq. (C.3) into $\pi(a|s)$ in Eq. (5.13).

C.3 Prior knowledge in Sections 5.4 and 5.5

Figure 5.3 in Section 5.4 shows the grid world with prior policies of Cases (a) and (b) in the numerical case study. Prior policies vary from state to state. When the state (s) has a single arrow, the prior policy in the direction of the arrow (e.g., right) is 0.9 and the remaining three directions have the same probability of prior policy (i.e., $\rho(right|s) = 0.9, \rho(left|s) = \rho(up|s) = \rho(down|s) = 0.03$. When the state ($s$) has two arrows, the prior policy in both directions (i.e., up, right) is 0.4 respectively and the remaining two directions have the equal probability (i.e., $\rho(up|s) = \rho(right|s)$=0.4, $\rho(down|s) = \rho(left|s)$=0.1) . If the state ($s$) has the random prior policy, it means the probability of selecting all directions are the same with 0.25 (i.e., $\rho(up|s) = \rho(down|s) = \rho(right|s) = \rho(left|s) = 0.25$)

Table 5.5 in Section 5.5 shows which prior knowledge is used in each geometry in the real-world case study. The total number of actions in each state is different between geometry 1 (4) and 2 (6) since the cooling fan is excluded from the process parameters in geometry 1. States in each geometry can be expressed as the level of process parameters in the order of the flow rate multiplier, printing speed, and cooling fan specified in Table 5.3.

In geometry1 in experiment 1, offline knowledge that encourages to adjust flow rate multiplier as 1.0 is used. Therefore, in the states with levels of (1,1) and (1,2), 0.9 is set as the prior probability for the corresponding action, and 0.03 is equally provided to the remaining three

actions. The states with the remaining levels in offline prior policy at geometry 1 have the random policy that has the same probability in four actions. In geometry 2 in experiment 1, online knowledge that is the optimal policy learned from geometry 1 is used. The optimal policy in geometry 1 is to adjust the flow rate multiplier as 1.0 at the state level (1,1,1) and printing speed to 2500 mm/min at the state level (2,1,1). The prior probability of the corresponding action is 0.9 and the probability of the remaining five actions is equal to 0.02. The states with the remaining levels in online prior policy at geometry 2 have the random policy that has the same probability in six actions. In geometry 2 in experiment 2, offline knowledge that recommends adjusting flow rate multiplier as 1.0 is utilized. Therefore, 0.9 is provided to the probability to set flow rate multiplier as 1.0, and 0.02 is provided to the each of remaining five actions in the states with levels of (1,1,1) and (1,2,1). The states with the remaining levels in offline knowledge at geometry 2 have the random policy that has the same probability in six actions.