Object Shape Error Response Using Bayesian 3-D Convolutional Neural Networks for Assembly Systems With Compliant Parts

Sumit Sinha, Pasquale Franciosa, and Dariusz Ceglarek

***Abstract*—This article proposes a novel object shape error response (OSER) approach to estimate the dimensional and geometric variation of assembled products and then, relate these to process parameters, which can be interpreted as root causes (RC) of the object shape defects. The OSER approach leverages Bayesian 3-D convolutional neural networks integrated with computer-aided engineering simulations for RC isolation. Compared with the existing methods, the proposed approach: 1) addresses a novel problem of applying deep learning for object shape error identification instead of object detection; 2) overcomes fundamental performance limitations of current linear approaches for RC analysis (RCA) of assembly systems that cannot be used on point cloud data; and 3) provides capabilities for unsolved challenges such as ill-conditioning, fault-multiplicity, RC prediction with uncertainty quantification, and learning at design phase when no measurement data are available. Comprehensive benchmarking with existing machine learning models demonstrates superior performance with $\mathrm{R}^2 = 0.98$ and $\mathrm{MAE} = 0.05$ mm, thus improving RCA capabilities by 29%.**

***Index Terms*—Assembly, Bayesian deep learning, manufacturing, 3-D convolutional neural networks (CNNs).**

I. Introduction

OBJECT shape error modeling and diagnosis are important enablers of Industry 4.0 and provide a transformative framework integrating facilitators such as big data, in-line 3-D scanners, robotics, and AI algorithms toward achieving near-zero-defect manufacturing. In this article, the proposed 3-D object shape error response (OSER) approach translates into estimating and discriminating between shape error patterns and linking them to manufacturing process parameters. Estimating at first and then reducing or eliminating these error patterns ensures dimensional product quality (as defined by GD&T), which is a major challenge for industries such as automotive, aerospace, and shipbuilding. Two-thirds of the quality issues in the automotive and aerospace sectors are caused by dimensional variations [1]. The key goal is developing a root cause (RC) analysis (RCA) model that can identify the relationship between shape errors and manufacturing process parameters.

Past methods used to diagnose manufacturing dimensional quality faults are based on: 1) statistical estimation and 2) pattern matching-based approaches. These approaches have been shown to have limitations in their applicability to complex, high-dimensional, and nonlinear systems [2] as these used linear models between process parameters and measurements of product dimensional quality for both systems with rigid [3] and complaint parts [4]. Ceglarek *et al.* [5] used computer-aided design (CAD)-based variation patterns and a fault matching technique which combined principal component analysis (PCA) and pattern similarity for fault diagnosis. This work was later extended to include the effect of measurement noise [6] and then generalized for multistage assembly process using state-space model, stream-of-variation [7]. Jin *et al.* [8] used a Bayesian network approach for estimating fixture faults using all measured points. Bastani *et al.* [9] used a spatially correlated Bayesian learning algorithm for an underdetermined system by exploiting the spatial correlation of dimensional variation from various error sources. In summary, the aforementioned approaches are linear and are designed to work for relatively small number of measurement points on each manufactured part. This significantly limits the application of the methods for 3-D object shape error modeling and diagnosis in manufacturing. The 3-D shape error modeling and diagnosis used in manufacturing must have the capability to satisfy a number of requirements with respect to the following.

1) *High data dimensionality* of a batch of 3-D objects [10], which are defined by CAD (ideal parts) and point-clouds (nonideal parts) with millions of points for each part or subassembly.
2) *Nonlinearity* due to compliant parts being constrained by assembly fixtures and part-to-part interactions [11].
3) *Collinearities* due to many manufacturing systems being ill-conditioned [12] with error patterns of key process

Manuscript received July 7, 2020; revised October 28, 2020; accepted November 24, 2020. Date of publication December 8, 2020; date of current version June 30, 2021. This work was supported in part by the U.K. EPSRC under Project EP/K019368/1: "Self-Resilient Reconfigurable Assembly Systems With In-Process Quality Improvement," in part by the UKRI Open Access Block Grant, and in part by the WMG-IIT Scholarship. Paper no. TII-20-3286. *(Corresponding author: Sumit Sinha.)*

The authors are with the Digital Lifecycle Management Group, Warwick Manufacturing Group, University of Warwick, CV4 7AL Coventry, U.K. (e-mail: sumit.sinha.1@warwick.ac.uk; p.franciosa@warwick.ac.uk; d.j.ceglarek@warwick.ac.uk).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TII.2020.3043226.

Digital Object Identifier 10.1109/TII.2020.3043226

parameters being near parallel, thus, yielding widely discrepant results.

4) *High faults multiplicity* [13], [14] due to current near-zero-defects strategies requiring to take into consideration 6-sigma defects that lead to redefining defects from binary {0, 1}, i.e., fault/no-fault, to continuous <0,1>, i.e., the fault being measured as a level of variation with dynamically changing threshold of acceptance, which significantly increases fault multiplicity.
5) *Uncertainty quantification* in the RCA output, as the identified RC frequently leads to costly corrective actions [15]; it is crucial therefore to enhance the RCA model by an uncertainty estimation of the predictions.
6) *Dual data generation capability* by using metrology gages and multiphysics-simulator needed for RCA model training. As the RCA model needs to be trained on a very large number of fault scenarios, which cannot be generated via real systems, and the training needs to be done before the real assembly systems are ready for production, there is a strong need to generate data via high-fidelity multiphysics simulator for training RCA model. Then, the RCA model will use point-cloud data of real free-form surfaces obtained via robotic 3-D scanner when implemented in a real system.

This article will address the above requirements as follows.

1) Requirements 1)–4) by developing a 3-D deep learning approach. As markets get competitive in terms of product quality, production volume, and costs, manufacturers aim to leverage developments in the field of artificial intelligence. Deep neural networks have revolutionized data-intensive tasks that involve generating insights from high-dimensional input data [16], [17]. 2-D/3-D convolutional neural networks (CNNs) are known to perform well when spatial data such as depth images, point-clouds, mesh, and medical scans have to be analyzed for tasks such as control systems, object detection, video analysis, and cancer detection. Manufacturing is one of the major domains that has benefited from this development [18]. This article proposes a 3-D CNN architecture that enables the extraction of spatial features from point clouds and hence, models nonlinear relationships between features and process parameters. This approach has high performance for nonlinear and ill-conditioned systems having high fault multiplicity.
2) Requirement 5) by leveraging a Bayesian 3-D CNN-based approach. Recent developments in artificial intelligence cautions making real-life decisions based on point-estimates. As compared to traditional CNNs with deterministic weights, Bayesian CNNs leverage probability distributions over model weights and model outputs and enable quantification of predictive uncertainty, prevent overfitting, and require comparatively lesser data to train [19]. Successful applications of the above have been done in healthcare [20] and load forecasting [21]. Using such models enables segregation of the uncertainties into aleatoric and epistemic, the former quantifying the uncertainty due to uncontrollable factors such as system noise, while the latter quantifies uncertainties due to model structure and insufficient training data [19]. The proposed approach estimates each model parameter as a distribution (epistemic uncertainty) while also modeling the output estimates as parameters of a multivariate distribution (aleatoric uncertainty). Such estimates involving different types of uncertainties in model predictions are crucial as these quantify when the model is "randomly guessing" as compared to making a confident prediction. Particularly within manufacturing environments, these uncertainty estimates integrate a degree of confidence within the estimates and hence, support the decision-maker in making cost-effective selection of corrective action(s) which can be quite costly.
3) Requirement 6) by making the developed Bayesian 3-D CNN from (2) compatible with point cloud data obtained via either multiphysics-simulator or 3-D scanners and leveraging the epistemic uncertainty estimates to perform intelligent *closed-loop training* and enable model convergence using a lesser number of training samples. In turn, this reduces total data generation and model training time. Since multiphysics-simulator is computationally expensive and generating each sample for assembly applications can be time intensive for high-fidelity simulations, it is crucial to reduce overall simulation time. The reduction in simulation time provided by leveraging the epistemic uncertainty estimates of Bayesian 3-D CNNs is significantly higher than the increased training time for Bayesian deep learning approaches. The approach developed in this article will utilize high-fidelity multiphysics-simulator of the assembly process, called variation response method (VRM) [22]. The VRM has the capability for, first, modeling and simulating the assembly process with compliant parts constrained by assembly fixtures and part-to-part interactions, and then, it enables high-fidelity point-cloud data generation of 3-D assembled products/subassemblies with error patterns as obtained under different sets of process parameters. The VRM model accuracy was verified and validated for various assembly processes [22]. Additionally, the approach is equipped to utilize data obtained from measurements using 3-D optical scanners. 3-D optical scanners enable real-time high-dimensional point cloud data extraction from manufacturing systems within short cycle times. This point cloud data can be postprocessed using alignment techniques to extract deviations for points, thus enabling dual data generation and integration.

In summary, the article develops a novel 3-D OSER approach in an effort to enable RCA within manufacturing systems using point cloud data. The proposed methodology integrates deep learning [which addresses requirements 1)–4)], Bayesian training enabled by Bayes-by-Backprop [23] and Flipout [24] [requirement 5)], and multiphysics-simulator to address requirement 6).

The key contributions of the article are as follows.

1) Propose a *3-D OSER methodology* based on a novel Bayesian 3-D CNN architecture: It builds on current

work done in the area of 3-D object detection [16] by expanding it to manufacturing systems where the key goal is not to detect the object but to estimate various shape error patterns present on the final object/product and relate these variation patterns to manufacturing process parameter variations within the system. To the best of our knowledge, this is the first article to propose an uncertainty enabled 3-D CNN-based deep learning model for RCA of assembly systems.

2) Propose a *closed-loop framework for training and deployment* of the Bayesian 3-D CNN model that leverages a computer-aided engineering (CAE) simulator known as VRM [22] to emulate the multistage assembly system. The VRM performs sampling, which leverages the epistemic uncertainty estimates of the Bayesian 3-D CNN thereby, reducing overall simulation and training time. Given that data availability within manufacturing systems is costly, scarce, and the data can be highly skewed, the VRM functions as a physics-based digital twin for generating augmented data that are close to the real system and can therefore, be used to train the proposed model. The trained model can then be leveraged for applications such as RCA of assembly systems using point cloud scans obtained from 3-D scanners.

3) Verify and validate the methodology on an *industrial automotive door assembly process* made of compliant parts.

4) *Benchmark 3-D OSER methodology* against three categories of methods that can be leveraged to estimate the dimensional and geometric variation of assembled products namely: a) current linear state-of-the-art RCA models; b) machine learning models in a multi-output regression setting; and c) deep learning models such as various types of CNNs and fully connected networks to highlight the performance and the ability to fulfill the aforementioned six requirements.

The rest of the article is organized as follows. Section II formulates the object shape error estimation problem, presents the proposed Bayesian 3-D CNN architecture, the steps involved in architecture optimization, and the overall steps required to train and deploy the model. Section III presents the industrial case study. Finally, Section IV concludes this article.

II. Methodology

A. Object Shape Error Estimation in Manufacturing

Multistage assembly systems can be mathematically expressed as a state-space model where different states correspond to different stages of the manufacturing system [3]. The input is an object (set of parts to be assembled) entering the assembly process. Within the process, object shape errors can be introduced in any of the stages due to one or multiple variations in the process parameters and are further propagated through the stages (see Fig. 1). Any object o at its design nominal shape is characterized by a set of nominal points $\boldsymbol{P_o} = \{\boldsymbol{p_{ok}}\},\ k = 1,\ldots,\ n_o$, where $\boldsymbol{p_{0k}}$ is a vector consisting of the x-,y-, and z-coordinates of the kth input point and n_o represents the total number of points on object o. The object here represents a single subassembly which is assembled in a single station, which can be understood as a collective reference to all parts used in this assembly station. In practice, the points correspond to mesh nodes in the computer-aided design model of the object when considering CAE simulations and to actual points within the point cloud when considering the 3-D scan of the object. $\boldsymbol{d_o} = \{\boldsymbol{d_{ok}}\}$ denotes the deviation of each point k after the nominal object o has gone through different stages of the process, $\boldsymbol{d_{ok}}$ is a vector comprised of deviations of each point in x-,y-, and z-axes on object o. An assumption made in this article is that the assembly process has a single station which includes multiple stages $s = 1,\ \ldots, 4$ involving objects/parts—positioning (P), clamping (C), fastening (F), and release (R). Stage $s = 0$ is used to represent the incoming parts that includes deviations from the previous processes, such as part fabrication. As the object o goes through multiple stages, the set of points are represented as $\boldsymbol{P_o^s}$, while $\boldsymbol{d_o^s}$ represents the deviations.

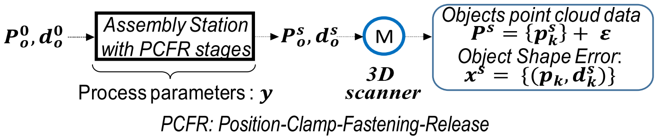
$\boldsymbol{P_o^0}, \boldsymbol{d_o^0}$
Assembly Station with PCFR stages
Process parameters : $\boldsymbol{y}$
$\boldsymbol{P_o^s}, \boldsymbol{d_o^s}$
M
3D scanner
Objects point cloud data
$\boldsymbol{P^s} = \{\boldsymbol{p_k^s}\} + \boldsymbol{\varepsilon}$
Object Shape Error:
$\boldsymbol{x^s} = \{(\boldsymbol{p_k}, \boldsymbol{d_k^s})\}$
PCFR: Position-Clamp-Fastening-Release

Fig. 1. Object shape error propagation in assembly systems.

As the main goal of this article is object shape error estimation, the article extends the problem formulation in object detection, which only considers the set of points $\{\boldsymbol{p_{ok}}\}$ [17], by including deviations for each point $\{\boldsymbol{d_{ok}}\}$ as additional features. This adds the required discriminative ability in the data, hence, enabling object shape error estimation. Thus, the object shape error for object o after stage s can be represented as

$$\boldsymbol{x_o^s} = (\boldsymbol{P_o^s}, \boldsymbol{d_o^s})\ . \tag{1}$$

On the other hand, the set of all process parameters across all stages are denoted by $\boldsymbol{y}$ where $\boldsymbol{y} = \{y_1, \ldots, y_h\},\ h$ denotes the total number of process parameters. The deviation of points at each stage s for object o can be expressed as the sum of all deviations accumulated in all stages from stage 0 up to stage s

$$\boldsymbol{d_o^s} = \sum_{j=0}^{s} \boldsymbol{d_o^j} \tag{2}$$

where $\boldsymbol{d_o^0}$ represents the shape error of incoming object o caused by upstream manufacturing processes. After each stage s, the actual points of the object o with error can be written as

$$\boldsymbol{P_o^s} = \boldsymbol{P_o}\ + \boldsymbol{d_o^s}. \tag{3}$$

At the end of the final assembly stage $s = 4$, the object shape error data for the assembly $\boldsymbol{P^{s=4}}$ is collected and decomposed into the nominal points $\boldsymbol{P}$ and their deviations $\boldsymbol{d^{s=4}}$ by using alignment techniques [10], where $\boldsymbol{P^{s=4}}, \boldsymbol{d^{s=4}}$ are now a collective reference to the set of all incoming objects that have been assembled. The measurement system error ε is considered to be negligible ($\varepsilon \approx 0$). The object with errors is represented as a point cloud of nonideal parts

$$\boldsymbol{x^s} = \{(\boldsymbol{p_k}, \boldsymbol{d_k^s})\} = (\boldsymbol{P}, \boldsymbol{d^s}) \tag{4}$$

where $\boldsymbol{d_k^s}$ can be considered as features at each point $\boldsymbol{p_k}$.

The aim of the Bayesian 3-D CNN model training is to learn assembly process transfer function $f(.)$ (equivalent to state transition matrix in [7]). The function $f(.)$ is parametrized by weights and biases of a CNN that can accurately estimate the process parameters $\boldsymbol{y}$ given the point cloud data of nonideal $\boldsymbol{x^s}$ parts collected from the system

$$\boldsymbol{y} = \ f\left(\boldsymbol{x^s}\right). \tag{5}$$

The high accuracy of the 3-D CNN in estimating all assembly process parameters $\boldsymbol{y}$ provides the underlined capability of the OSER approach for high RC isolability. Essentially, within assembly systems, RCs are estimated as a subset of the estimated process parameters

$$\boldsymbol{RC} \ \subseteq \ \boldsymbol{y}. \tag{6}$$

Based on the requirements and the production phase of the assembly system, the exact definition of an RC may differ but the key requirement to conduct RCA under any definition is to accurately estimate all process parameters $\boldsymbol{y}$. Hence, the proposed OSER approach aims to do the aforementioned by estimating $f(.)$ as specified in (5).

B. Three-Dimensional Object Shape Error Voxelization

In the presented OSER approach, the simulation output represented as mesh or point cloud data $\{(\boldsymbol{p_k}, \boldsymbol{d_k^s})\}$ (4) is transformed to voxel grids $\{\boldsymbol{V_{u,v,w}}\}$ with discrete voxel coordinates (u, v, w) in the following way—for all points $\boldsymbol{p_k} = (x_k, y_k, z_k)$ that fall within a voxel grid $\{\boldsymbol{V_{u,v,w}}\}$, the maximum value of $\boldsymbol{d_k} = (\tilde{x}_k, \tilde{y}_k, \tilde{z}_k)$ characterizes the features of the corresponding voxel grid and is represented as $\{\boldsymbol{V_{u,v,w,d}}\}$. The voxelization technique~~s~~ used in object detection [17] are applied to construct the initial voxel structure of the object and for each unique object, the voxel features are characterized by the shape error $\boldsymbol{d_k}$. The key difference is that in object detection, voxel grids are characterized by either binary voxels or voxels containing RGB values for each point instead of real values of shape error $\boldsymbol{d_k}$, as in the OSER approach. Although binary voxels, traditionally used in object detection retain the spatial structure, the granularity of voxelization required to discriminate between minor differences in the shape error will make the problem computationally infeasible and hence, limit performance. In the proposed approach, the nominal object is voxelized and each voxel is characterized by real vales of the shape error $\boldsymbol{d_k}$. This is critical in representing the geometric variations with the required granularity for effective RCA. This efficiently retains all information about the spatial structure of the object as well as the components of object shape errors. Given the alignment ensures a fixed orientation, there is no need for data augmentation to achieve rotation invariance.

C. Uncertainty Estimation

Given the uncertainties of the system and the availability of only a limited dataset, a deterministic estimate of function $f(.)$, as shown in (5), is not feasible. Hence, by leveraging Bayesian inference, a prior distribution can be allocated over the space of possible functions $p(f)$, which represents a prior belief of the possible functions $f(.)$. Given a dataset, a likelihood $p(\boldsymbol{y}|f, \boldsymbol{x^s})$ is defined to model the function from which the observation is generated, and hence, given a dataset $(\boldsymbol{x^s}, \boldsymbol{y})$, the posterior distribution over the functions $p(f|\boldsymbol{x^s}, \boldsymbol{y})$ can be inferred. The function is characterized by model parameters ω represented by f^{ω} (weights and biases for neural networks) and the posterior over the function can be inferred by estimating the posterior over the parameters ω. In Bayesian neural networks, this is achieved through Bayes-by-Backprop [23] and Flipout [24]. Given a dataset the posterior can be written as

$$p(w|x_s) = p(y|x_s, w), p(w)/p(y|x_s) \tag{7}$$

For complex models such as deep neural networks, it is not analytically possible to infer the true posterior for all model parameters $p(w|x_s, y)$; hence, an approximating variational distribution $q_\theta(\omega)$ parameterized by θ, such as normal distribution, is used to approximate the posterior. This approach is known as variational inference [25]. The approximating distribution should be as close as possible to the true posterior, which is achieved by minimizing the Kullback–Leibler (KL) divergence with respect to θ

$$KL(q_\theta(\omega)||p(w|x_s)) = \int q_\theta(\omega) \log q_\theta(\omega)/p(w|x_s) d\omega. \tag{8}$$

Using the estimated variational distribution ${q^*}_\theta(\omega)$, the process parameter distribution quantifying the uncertainties for a new data point $\boldsymbol{x^{s*}}$ can be obtained using

$$\begin{aligned} &p\left(\boldsymbol{y^*}|\boldsymbol{x^{s*}}, \boldsymbol{x^s}, \boldsymbol{y}\right) \\ &\quad \approx \int p\left(\boldsymbol{y^*}|\boldsymbol{x^s}, \omega\right) p\left(w|\boldsymbol{x^s}, \boldsymbol{y}\right) d\omega =: {q^*}_\theta\left(\boldsymbol{y^*}|\boldsymbol{x^{s*}}\right). \end{aligned} \tag{9}$$

D. Bayesian 3-D CNN Model Architecture

Building on the work done on voxel-based approaches for 3-D object detection such as VoxNet [17], the research proposes a Bayesian 3-D CNN architecture to enable object shape error estimation. The 3-D convolutions aggregate features from the input, which are then utilized by the fully connected layers and mapped to process parameters. The model consists of three 3-D convolutional Flipout layers, a 3-D max-pooling layer, followed by three fully connected Flipout layers; the final layer estimates parameters of the predictive distribution for all process parameters. The convolution can be represented as

$$\begin{aligned} &v_{ab}^{xyz} \\ &= ReLU \left(\beta_{ab} + \sum_{m} \sum_{p=0}^{P_i - L_i} \sum_{q=0}^{Q_i - M_i} \sum_{r=0}^{R_i - N_i} w_{ab}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)} \right) \end{aligned} \tag{10}$$

where v_{ab}^{xyz} represents the layer output value at position (x, y, z) in the *a*th layer and *b*th feature map. ReLU is the rectified linear unit activation function [26]. β_{ab} represents the bias; m represents the number of filters from the previous layer; (P_a, Q_a, R_a) and (L_a, M_a, N_a) represent the kernel dimensions and stride lengths in the three directions, respectively; w_{ab}^{pqr} represents the weights of the connections. The convolution operation as in (10) is done consecutively for the three convolutional

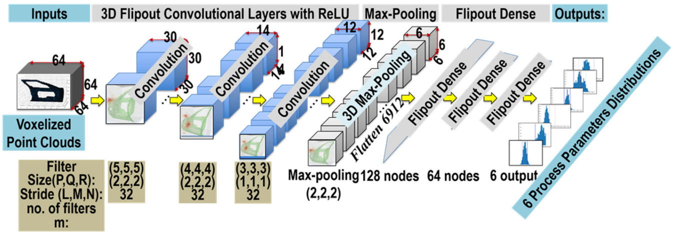

Fig. 2. Bayesian 3-D CNN model architecture of the OSER method.

layers. In 3-D max-pooling operation, the resolution of the feature map is reduced by taking the maximum value of the local neighborhood of the layer outputs. Given the Bayesian framework, each parameter of the Bayesian 3-D CNN model follows a distribution. In the case of neural networks, it is not feasible to assign informative priors; hence, noninformative prior distributions are placed over the model parameters. Each parameter is approximated using variational inference approach assuming that the posterior follows a normal distribution. The overall model has 1 997 286 trainable parameters. Output nodes have linear activation units. Fig. 2 shows the proposed Bayesian 3-D CNN model architecture with annotated hyperparameters.

E. Architecture Selection and Optimization

Hyperparameter optimization for Bayesian 3-D CNNs is done to maximize performance and eliminate architectures that are more likely to overfit. As this is computationally intensive, in order to perform optimization in a computationally feasible manner, the following steps were involved.

Step 1 – Set baseline: VoxNet [16], which is a 3-D CNN architecture used for object detection consisting of two 3-D convolutional layers, one max-pooling layer, and two fully connected, is set as the baseline. A dataset consisting of 1500 samples is generated to conduct k-fold cross-validation ($k = 6$). The hyperparameters are split into two categories: Category one consists of the number of convolutional layers $N_C = \{2, 3, 4\}$ and number of dense layers $N_D = \{1, 2, 3\}$ and category two consisted of the number of filters in each 3-D convolutional layer, filter size for each 3-D convolutional layer, and number of hidden units in each dense layer.

Step 2 – Grid search for category one hyperparameters: In this step, grid search for category one hyperparameters are conducted and each selection is evaluated using k-fold cross-validation (see Fig. 3). For computational feasibility, category two hyperparameters are kept constant and equal to the VoxNet architecture values. $N_C = 3$ and $N_D = 3$ were obtained as the optimal hyperparameters having the minimum cross-validation mean absolute error (MAE) average of 0.08 mm.

Step 3 – Hyperband for category two hyperparameters: The optimal values for category one hyperparameters are fixed and further hyperband [27] is leveraged to obtain the optimal values for category two hyperparameters given its ability to speed up the random search process through adaptive resource allocation and early stopping.

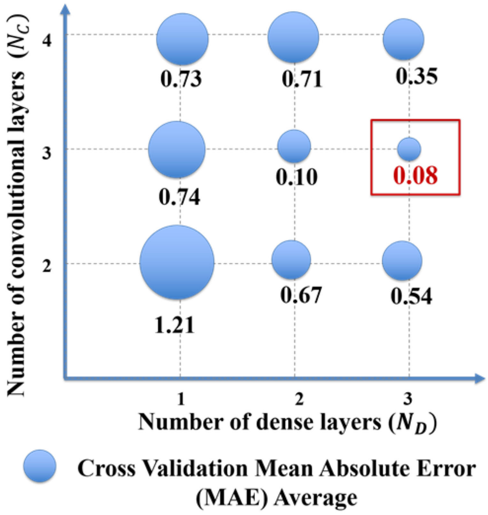

Fig. 3. Grid search for category one hyperparameters.

TABLE I
OBJECT DETECTION AND OSER COMPARISON

Object Detection (VoxNet)	Object Shape Error Detection (OSER)	Rationale
Single-channel binary occupancy input voxel of dimension $32 \times 32 \times 32$	Multi-channel real-valued object shape error input voxel of dimension $64 \times 64 \times 3$	High data dimensionality of shape error $(\boldsymbol{P}, \boldsymbol{d}^s)$ [requirement 1)]
Two 3D Convolutional layers, one dense layer and categorical output layer	Three 3D Convolutional layers, three dense layers and real-valued output layer	Increased model capabilities to handle non-linearity, collinearity and high fault multiplicity [requirements 2)–4)]
Deterministic Layer	Bayesian Flipout Layer	Uncertainty quantification for corrective actions and closed-loop sampling for faster convergence [requirements 5) and 6)]

Step 4 – Deterministic to Bayesian model: The final step includes replacing the deterministic layer with Bayesian Flipout layers and then training using Bayes-by-Backprop. Various learning rates and prior distributions for the model weights were tested. Standard normal distribution provided the best balance between weight initialization and weight exploration, which was inferred by conducting an uncertainty versus error calibration study. The training hyperparameters that provided the best uncertainty calibration and ensured that the model performance was greater than or equal to the deterministic counterpart were selected as the final Bayesian 3-D CNN architecture training hyperparameters. The key changes from the baseline architecture of object detection that enable the fulfillment of the aforementioned six requirements are summarized in Table I.

F. Model Training and Deployment

Training of the model is done in a closed-loop framework using data generated by VRM. The key motivation behind using a closed-loop framework as opposed to an open-loop framework

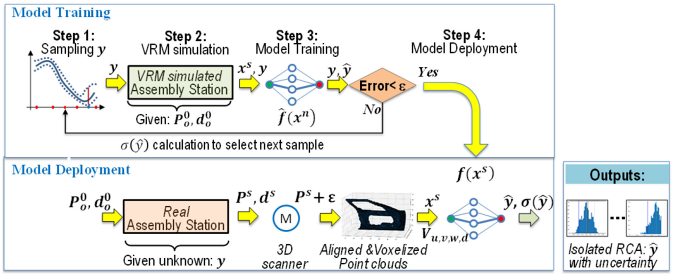

Fig. 4. Model training and deployment framework.

is to minimize the bottle-neck computation, i.e., multiphysics simulation using the VRM model. Although this increases the number of training iterations, nonetheless, the overall time of VRM simulation and training is significantly reduced, as fewer samples need to be generated. The key steps of the proposed framework are summarized as follows (see Fig. 4).

Step 1 – Sampling: Process parameters $\boldsymbol{y}$ are sampled from the allowable ranges. Latin Hypercube Sampling [28] is used to generate initial process parameter sample values given it distributes samples optimally across the $h-$dimensional process parameter space by stratifying the possible ranges. The consecutive sets of samples are generated using Monte Carlo (MC) sampling based on the uncertainty $\boldsymbol{\sigma}(\hat{\boldsymbol{y}})$ of the model.

Step 2 – VRM simulation: The samples are used as input to the VRM to simulate the assembly process and generate the output mesh from which the point cloud and deviations of each point are extracted after the desired stage s of the assembly system, as in (4), $\boldsymbol{x^s} = \{(\boldsymbol{p_k}, \boldsymbol{d_k^s})\}$.

Step 3 – Model training: The point cloud and deviation data of object shape errors along with the respective process parameters $(\boldsymbol{x^s} = (\{(\boldsymbol{p_k}, \boldsymbol{d_k^s})\}, \boldsymbol{y})$ are used for model training. Note that $\boldsymbol{x^s}$ is voxelized $\{\boldsymbol{V_{u,v,w,d}}\}$ before it is used for training. Bayes-by-Backprop and Flipout are applied for model training. The loss function optimized while training comprises of the sum of KL divergence for each layer (9) and the negative log-likelihood (11) of the predictive distribution

$$-\ln L = 1/2[\ln(|\boldsymbol{\Sigma}|) + (\boldsymbol{y} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}) + h\ln(2\pi)]. \tag{11}$$

The KL divergence term quantifies the divergence between the standard normal prior and the learnt posterior and hence, prevents overfitting by penalizing weights for diverging from the prior. Group normalization [29] with four groups is used after each convolutional layer. This also prevents overfitting and accounts for the small minibatch size due to GPU memory size constraints and aids in stabling the training process. Weights of the network are initialized using normal initializer [30]. The Adam method for stochastic optimization was used to optimize the loss function while training [31]. The initial learning rate is fine-tuned to $\alpha = 0.0005$ and monotonic KL annealing was leveraged to ensure the model initially learns the object shape error and process parameter relations before applying the KL penalty for uncertainty quantification. The learning rate fine-tuning, monotonic KL annealing, and ReLU activations prevent gradient vanishing. The predictive distribution is modeled as a multivariate normal $\mathcal{N}_h$ with h components (the same number of components as the number of process parameters h), $\boldsymbol{y} \sim \mathcal{N}_h(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where each component corresponds to a process parameter; hence, the mean across all components of the multivariate distribution corresponds to the set of process parameters $\boldsymbol{y}$. The distribution is assumed to have a diagonal covariance matrix $\boldsymbol{\Sigma}$. The scale parameters in the diagonal are assumed to be fixed since the noise has been assumed to be negligible.

After each iteration of training, the model is evaluated on the validation set. For evaluation, MC sampling from the model is done and the sample means $\bar{\boldsymbol{y}}$ and standard deviations (SDS) $\boldsymbol{\sigma}(\hat{\boldsymbol{y}})$ are estimated for each process parameter. $\boldsymbol{\sigma}(\hat{\boldsymbol{y}})$ represents the epistemic uncertainty while the fixed scale parameters of the predictive distribution represent the known aleatoric uncertainty [19]. Given the assumption of negligible measurement noise, aleatoric uncertainty is considered to be negligible and hence, the overall uncertainty in the prediction can be assumed to be equal to epistemic uncertainty $\boldsymbol{\sigma}(\hat{\boldsymbol{y}})$. This uncertainty is used for sampling in the next iteration.

MAE between the model estimates $\hat{\boldsymbol{y}} = \bar{\boldsymbol{y}}$ and actual value $\boldsymbol{y}$ across all process parameters h (12) are used as the metrics for model performance evaluation given the ease of interpretation and given that the model outputs are continuous and real valued. Training is stopped when MAE is below the required threshold ϵ. The threshold value for this metric is determined based on the quality requirements for a specific product as required by design tolerances and the accuracy of the measurement system. The model is trained within the measurement system accuracy. For example, automotive body assembly process tolerances are within [−1 mm, 1 mm], and the 3-D optical scanner used has a repeatability of 0.05 mm and accuracy within 0.15 mm

$$\text{MAE} = \sum_h |\boldsymbol{y} - \hat{\boldsymbol{y}}|/h. \tag{12}$$

Step 4 – Model deployment: After training, the model can be deployed within an actual system. The data collected from the 3-D scanner $\boldsymbol{P^s}$ is aligned to obtain point cloud and deviations $\boldsymbol{x^s} = \{\boldsymbol{p_k}, \boldsymbol{d_k^s}\}$ and then, voxelized $\boldsymbol{V_{u,v,w,d}}$ before it can be given to the trained model for conducting RCA inference. Inferencing estimates the process parameters for a given $\boldsymbol{x^s}$ (5) using MC sampling from the trained model. Using these samples, process parameters (distribution mean) $\bar{\boldsymbol{y}}$ and their uncertainty (distribution SD) $\boldsymbol{\sigma}(\hat{\boldsymbol{y}})$ can be estimated. The sample mean $\bar{\boldsymbol{y}}$ is considered as the model estimate $\hat{\boldsymbol{y}}$, while $\boldsymbol{\sigma}(\hat{\boldsymbol{y}})$ quantifies the uncertainty. Further, the RCs can be inferred as a subset of $\hat{\boldsymbol{y}}$ (6). The work has been implemented using Python 3.7 and TensorFlow - GPU 2.0 [32] and TensorFlow-Probability 0.8. A python library, Bayesian Deep Learning for Manufacturing [33] has been developed to validate and replicate the results of the methodology. For this article, both, the data generation and evaluation of the OSER methodology have been done using

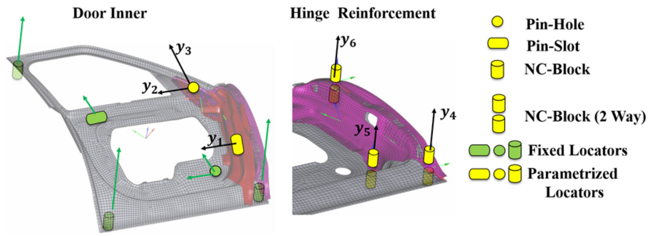

Fig. 5. Assembly process parameters.

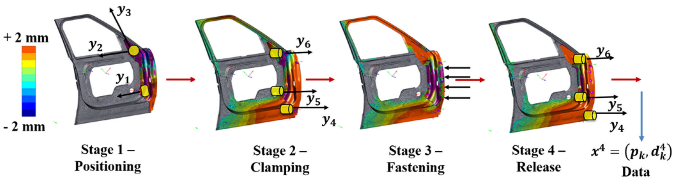

Fig. 6. PCFR stages of the assembly process.

VRM. Two Nvidia Tesla V100 32 GB GPUs are used for model training and deployment.

III. Case Study

A. Assembly Setup

For verification and validation of the proposed OSER approach, an automotive assembly of two components, namely, the door inner and hinge reinforcement are selected. The assembly setup and parameters are shown in Fig. 5. The assembly process is controlled by the six $(h = 6)$ parametrized process parameters $y_1, y_2, \ldots, y_6$ (depicted using yellow symbols in Fig. 4). Assembly parameters such as pin-hole, pin-slot, and NC blocks for the door inner are considered constant (depicted using green symbols in Fig. 5) and are not parameterized. Data are collected after stage $s = 4$. The point cloud is characterized by $n = 10\,841$ points, which are preprocessed and voxelized to $(u, v, w) = (64, 64, 64)$ voxel grids. The deviation features $\boldsymbol{d}$ include deviations in all directions for all points $(\tilde{x}_k, \tilde{y}_k, \tilde{z}_k)$. The assembly consists of four stages (see Fig. 6): Stage 1 involves positioning (i) the door inner on the pin-hole, pin-slot, and the three NC blocks (not parameterized; marked in green in Fig. 1) and (ii) hinge reinforcement using the pin-slot (y_1), pin-hole (y_2, y_3); Stage 2 comprises of clamping two parts together using three NC blocks with clamps (y_4, y_5 and y_6); Stage 3 involves fastening/joining the two parts using self-piercing riveting (SPR); and finally, Stage 4 involves releasing the clamps (y_4, y_5 and y_6) after the fastening is completed. The training range for all process parameters is $[-1 \text{ mm},\ 1 \text{ mm}]$ while the validation range is $[-2 \text{ mm},\ 2 \text{ mm}]$. Point cloud and deviation data $\{(\boldsymbol{p_k}, \boldsymbol{d_k^4})\}$ are collected after release, i.e., Stage 4 (see Fig. 5). The data are voxelized $\{\boldsymbol{V}_{64,64,64,3}\}$ and used as model input and the process parameters $y_1, y_2, \ldots,\ y_6$ are used as model outputs.

After starting with 200 initial samples for model training, 200 samples are adaptively added during each iteration of the closed-loop training based on the uncertainty estimates, and the model is trained on the combined dataset including all previous samplings to ensure that there is no catastrophic forgetting (using 200 samples provided an optimal trade-off between VRM simulation time and model training time). These samples and outputs are used for training the Bayesian 3-D CNN model. The diagonal scale parameters for all process parameters in the covariance matrix are fixed at 0.001. A validation set of 300 samples is generated within the validation range, and after each iteration, the trained model is evaluated on the validation set. During the evaluation for each of the 300 samples, 1000 MC samples are drawn from the trained model. The sample means are considered as the estimate for the process parameters while the sample SDs quantify the uncertainty for each process parameter for the given sample. RCs can be inferred from the process parameter estimates. The closed-loop training is stopped when average MAE across all process parameters for the validation set is below the threshold which is selected to be 0.05 mm for automotive assembly applications as the impact of variations less than 0.05 mm is not detectable by the 3-D scanner. After this, the model is ready for deployment with measurement data collected from 3-D optical scanners, followed by alignment and voxelization. For each measurement, MC samples from the trained Bayesian 3-D CNN model can be drawn to estimate process parameter mean and SDs (uncertainty). Measurement data collection is done using WLS400A mounted on an ABB robot.

In summary, the industrial assembly process selected for case study consists of the following.

1) High dimensionality point cloud (10 841 points).
2) Nonlinearity, as induced by fixturing (N-2-1, where $N =$ 6), two compliant parts, and part-to-part interactions (door inner to hinge reinforcement).
3) Collinearity induced by fixturing, as locators y_4, y_5, and y_6 are within 5° of collinearity (−3° to 2° deviation from y-axis).
4) High fault multiplicity, as we take into consideration 6-sigma defects at the level of variation within 3-D scanner accuracy (<0.05 mm) that significantly increases fault multiplicity from zero to six process parameters manifesting errors (100% fault multiplicity).

The door assembly requirements are: 1) *product:* Design tolerances of door assembly: $< -1.0,\ 1.0 >$ [mm]; 2) *process*: Fixturing calibration and commissioning is achieved within $< -0.1;\ 0.1 >$ [mm]; and 3) *shape error detection*: Using the 3-D optical scanner for measurement.

Key performance indicators (KPIs) used for assessment of the results are as follows: 1) MAE <0.05 mm and 2) R^2>0.95 for the model to have the capability to explain more than 95% variance in the process parameters under the assembly system requirements 2)–4).

B. Results

The KPIs of model performance are summarized for all $y_1, \ldots, y_6$ in Fig 7. The model convergence is shown in Fig. 8.

The model converges with average MAE across all process parameters equal to 0.05 (below the required threshold) and

TABLE II
BENCHMARKING RESULTS

		Requirement 1)	Requirement 2)–4)				Requirement 5)	Requirement 6)			
	Models		Accuracy (**MAE**)		Goodness-of-fit (***R***^{2})			Sampling	CAE Simulations Time (minutes)	Model Training Time (minutes)	Total Training Time (minutes)
			Mean	SD	Mean	SD					
Proposed	OSER (Bayesian 3D CNN)	**3D Shape Error Voxelization**	**0.05**	0.03	**0.98**	0.01	**Aleatoric & Epistemic Uncertainty**	**Epistemic uncertainty sampling (2000 samples)**	4400	424	4824
	OSER (3D CNN)	**3D Shape Error Voxelization**	**0.05**	0.01	**0.98**	0.009	No	Random Sampling (4000 samples)	8800	268	9068
Deep Learning	Multi-View 2D CNNs (MV-CNNs)	6 face projection and 2D gridding	0.08	0.02	0.94	0.01	No	Random Sampling (4000 samples)	8800	321	9121
	Depth Based 2D CNNs	1 face projection and 2D gridding	0.12	0.04	0.93	0.02	No	Random Sampling (3000 samples)	6600	248	6848
	Deep Dense Neural Networks	Flattening	0.28	0.09	0.91	0.07	No	Random Sampling (5000 samples)	11000	358	11358
Machine Learning	Gradient Boosted Trees	PCA	0.26	0.08	0.93	0.08	No	Random Sampling (3000 samples)	6600	120	6720
	Random Forests	PCA	0.29	0.09	0.92	0.08	No	Random Sampling (3000 samples)	6600	136	6736
	Support Vector Regression	PCA	0.38	0.09	0.85	0.1	No	Random Sampling (2500 samples)	5500	180	5680
Currently used linear approaches	Regularized Linear Regression	PCA	0.41	0.01	0.76	0.01	No	Random Sampling (1600 samples)	3300	10	3310

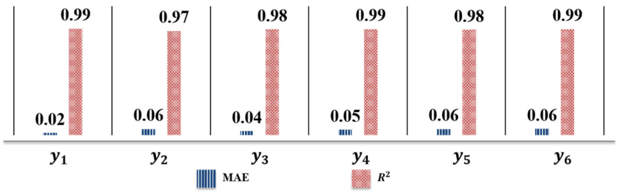

Fig. 7. MAE and R^2 across all process parameters.

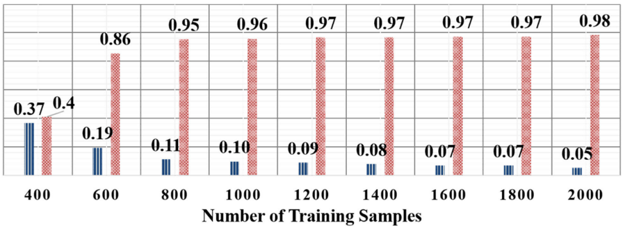

Fig. 8. Bayesian 3-D CNN OSER convergence.

average R^2 equal to 0.98 after ten iterations of closed-loop training, which included a total of 2000 samples being generated adaptively. For validation purposes, this study trained both Bayesian 3-D CNN and a deterministic version of the model, i.e., 3-D CNN with the same architecture as in Fig. 2.

C. Benchmarking and Discussion

The benchmarking analysis is conducted by using the six requirements as listed in Section I. The case study and results along with analysis of collinearity, multiplicity, and uncertainty are used to demonstrate the capabilities of the proposed approach to fulfill the aforementioned requirements.

The benchmarking analysis of the proposed 3-D OSER approach is discussed on two levels.

1) OSER Versus Currently Used Approaches at Production Phase When Point Cloud Data are Available:

The benchmarking is conducted for two scenarios: 1) RCA and 2) RCA with uncertainty quantification.

a) RCA: As discussed in Section I, the state-of-the-art models used for assembly process RCA, such as [9] and [34], are linear and can be classified as regularized linear regression approaches (see Table II). Hence, their upper limit performance can be estimated by using regularized linear regression on all point deviations $\boldsymbol{d}$ within the point cloud. They also use a limited number of sampled points from the point cloud on a single part (less than 100 out of >10 000), which additionally limit their performance for assembly processes. The OSER methodology validation against the six requirements as presented in Section I is as follows. Requirement 1) is fulfilled by the proposed voxelization approach which ensures that irrespective of the dimensionality of the point cloud, it is transformed into a sparse tensor of dimensions (64, 64, 64, 3) which preserves information in terms of the object spatial structure and point deviation features. This also enables the application of the OSER-based models that require a regular data structure as input. Second, the model performance of the state-of-the-art regularized linear regression approaches is at $\mathrm{MAE} = 0.41$ mm and $R^2 = 0.76$ (see Table II), which is unsatisfactory as compared to the required MAE<0.05, R^2 >0.95. This is because the regularized linear regression model can explain only the linear variance in the system. By comparison, the proposed OSER model demonstrates good performance at $\mathrm{MAE} = 0.05$, $R^2 = 0.98$, hence fulfilling requirements 2)–4). Fig. 9 compares the performance of regularized linear regression (i.e., upper limit for state-of-the-art approaches) with the proposed OSER approach under different levels of fault multiplicity and collinearity. For example, in scenarios 1–3 (fault multiplicity up to 50%), both approaches

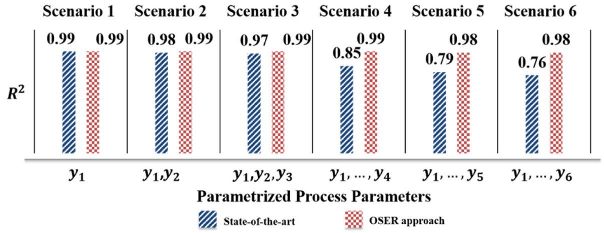

Fig. 9. Performance under different levels of fault collinearity and multiplicity (from single fault y_1; and two-fault y_1, y_2; … up to all parameters being simultaneously at fault $y_1, y_2, \dots, y_6$).

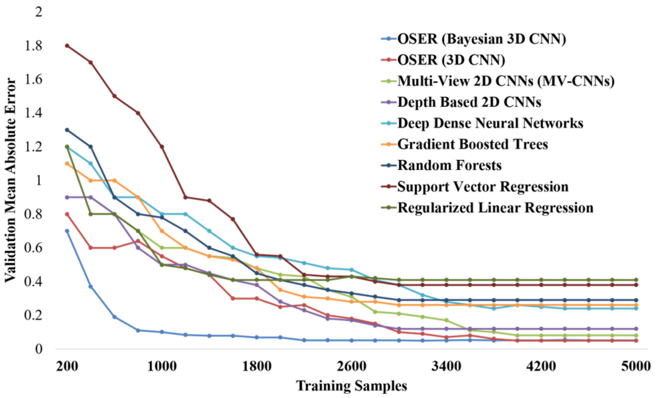

Fig. 10. Convergence comparison for all benchmarking models.

have similar performance. However, in scenarios 4–6, as the fault multiplicity increases to 4, 5, and 6 parameters being simultaneously at fault, i.e., 100% of parameters, and also induced by the designed collinear relation between process parameters and input, the performance of the linear model decreases, while the OSER approach exhibits performance above the required threshold (R^2 >0.95).

The benchmarking also comprehensively assesses the OSER against existing deep learning and machine learning techniques [35] in ways that are not currently used for RCA of assembly processes (see Table II). This article implemented these techniques and applied them for the aforementioned case study. CNN-based deep learning methods where selected as they retain spatial information while learning which is essential for object shape error estimation. Each model is compared in its ability to fulfill the aforementioned six requirements. Table II shows the implemented benchmark approaches and the results. The multiview 2-D CNN (MV-CNN) [36] approach considers six 2-D projections of the object shape error. Gridding is performed on each projection. Each projection has an input dimension of 64 × 64 × 3. Each of the six projections is given as input to the MV-CNN consisting of six heads. These are pooled before the fully connected layers. Depth-based CNN [37] considers a single projection along the y-axis with dimension 64 × 64 × 4. The first three channels consist of the shape error ($\boldsymbol{d}$) while the fourth channel consists of the y-coordinate of the nominal point-cloud $\boldsymbol{P_o}$. Both 2-D CNN-based approaches have same hyperparameters as the OSER (but only considering 2-D convolutions and 2-D max-pooling). The deep dense neural network is given the flattened vector of shape error $\boldsymbol{d} = \text{Flatten}\,\{(\tilde{x}_k, \tilde{y}_k, \tilde{z}_k)\}$ as input. It consists of two hidden layers (1024, 512 nodes, respectively) with ReLU activations and a linear output layer with six nodes. All machine learning models take a transformed input of $\boldsymbol{d} = \text{Flatten}\,\{(\tilde{x}_k, \tilde{y}_k, \tilde{z}_k)\}$ as input. PCA is used for the transformation and reduced features explaining 99% of the variance are retained. Comparison for requirement 1) is done on the basis of transformation used for input. Shape error voxelization retains information pertaining to the 3-D structure and shape error features, while projection retains only 2-D spatial structure. Flattening eliminates all information related to the spatial structure while PCA also eliminates information explaining 1% of the variance. Comparison for requirements 2)–4) are done on performance attributes, namely, accuracy (MAE) and goodness-of-fit (R^2). Comparison for requirement 5) is done on the ability to quantify and segregate uncertainties. Finally, comparison for requirement 6) is done on overall training time, which is inclusive of the CAE simulation time and model training time. Although the proposed model has higher model training time, the overall training time is significantly lesser due to the ability to leverage the epistemic uncertainty to generate informative samples leading to faster convergence with only ∼2000 samples. All other models are trained using random sampling until convergence. Fig. 10 summarizes the convergence of the entire set of benchmarking models. The hyperparameters of the machine learning models were optimized using grid search. For statistical quantification of accuracy and goodness-of-fit, 20 runs of training and testing are conducted using a set of 4000 randomly sampled data points for training and 300 for validation within the validation range. The mean and SD for each model, averaged across six process parameters, have been reported. The model performance of the proposed OSER model is significantly better in terms of accuracy and goodness-of-fit. Result from ANOVA followed by *post hoc* Tukey-HSD test at 95% significance level considering two sources of variations (model type and process parameter) showed the differences to be statistically significant. This comes at the expense of increased model complexity.

b) RCA with uncertainty quantification: As discussed in Section I, the identified RCA frequently leads to costly corrective actions conducted in the manufacturing environment; therefore, it is crucial, especially for 6-sigma faults to have decision-driven RCA directed toward informing choices by uncertainty quantification of solving problems. The OSER methodology provides the SD of the predicted process parameter distributions $\boldsymbol{\sigma}(\hat{\boldsymbol{y}})$ that quantifies this uncertainty hence, fulfilling requirement 5). Although the performance of the OSER with 3-D CNN and OSER with Bayesian 3-D CNN models are similar, the latter can quantify and segregate the aleatoric and epistemic uncertainty while estimating the process parameters. To demonstrate the capability of the model in quantifying the uncertainty on unseen samples, evaluation is done on 500 samples within the training range [−1 mm, 1 mm] and 500 samples outside of the training range [−2 mm, 2 mm]. The SD across all observations has been averaged and compared for each process

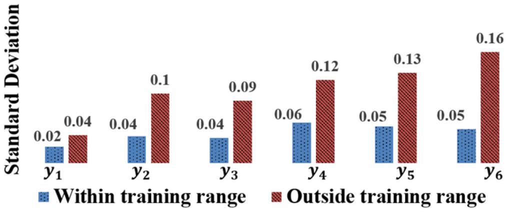

Fig. 11. Process parameter distribution SDs.

parameter $y_1, \ldots, y_6$. Results are shown in Fig. 11. Additionally, the epistemic uncertainty estimates enable closed-loop training, reducing overall training time.

2) OSER Versus Approaches at Design Phase When No Point Cloud Data are Available: In manufacturing environments, the availability of a comprehensive dataset inclusive of all fault scenarios is not feasible; hence, augmenting the dataset with high-fidelity multiphysics simulation enables training and deployment of deep learning approaches during the design phase of a new product/production system introduction. Given that the proposed OSER approach transforms the simulation mesh nodes output and scanned point cloud output to the same voxelized shape error that is compatible with 3-D CNN, it enables this integration, hence fulfilling requirement 6). This provides the capability for modeling and simulation of the assembly process and conducting system diagnosability and resilience analysis. Currently, none of the approaches provide this capability for object shaper error RCA at the design phase.

IV. Conclusion

This article presented an OSER approach which is relevant to manufacturing industries where dimensional and geometric variations can be quantified as object shape errors. This is also relevant to areas such as robotics, computer-aided detection, stamping, machining, and additive manufacturing where RCA of dimensional variations translates to estimating object shape error patterns and relating them to process parameters. Transfer learning can be leveraged for application in these domains with exponentially lesser training samples [18], a focus for future work. The proposed approach leveraged a Bayesian 3-D CNN model trained within a closed-loop framework using a multiphysics simulation (VRM) model, to estimate shape errors and relate them to process parameters while quantifying uncertainty. This could then be deployed on real data collected from 3-D surface scanners and thereby, enabling more effective and efficient decision-making for control and correction of manufacturing systems. The approach was benchmarked against state-of-the-art assembly RCA models and other machine learning models to highlight statistically significant better model performance while fulfilling the manufacturing system design requirements. Leveraging such automated RCA models ensured early estimation and elimination of process variations before they become defects, which could improve the quality and productivity of the system by reducing scrap and machine downtime. This also eliminated the need for trial-and-error approaches for RCA, which is often ineffective and inefficient. Future work aims to explore scaling up the work to multistation assembly systems. Various encoder–decoder-based CNN architectures, such as U-Net [38] and Pointnet [16], that enable process parameter estimation for a heterogeneous set of process parameters, i.e., continuous and categorical as well as enable object shape error estimation in-between stages/stations could be explored to comprehensively perform RCA on multistation systems. Approaches for uncertainty guided continual learning will also be explored that enable transfer learning to different manufacturing systems while simultaneously retaining knowledge of previous assembly systems. The future work also aims to develop a life-long continual learning approach leveraging Bayesian 3-D CNNs, which is crucial for continuously changing manufacturing environments.

References

[1] D. Ceglarek and J. Shi, "Dimensional variation reduction for automotive body assembly," *Manuf. Rev*, vol. 8, pp. 139–154, 1995.

[2] S. Sinha *et al.*, "3D convolutional neural networks to estimate assembly process parameters using 3D point-clouds," *Proc. SPIE*, vol. 11059, 2019, Art. no. 110590B.

[3] W. Huang *et al.*, "Stream-of-variation (SOVA) modeling II: A generic 3D variation model for rigid body assembly in multi station assembly processes," *J. Manuf. Sci. Eng.*, vol. 129, no. 4, pp. 323–333, 2006.

[4] H. Wang and D. Ceglarek, "Variation propagation modeling and analysis at preliminary design phase of multi-station assembly system," *Assem. Autom.*, vol. 29, no. 2, pp. 154–166, 2009.

[5] D. Ceglarek and J. Shi, "Fixture failure diagnosis for autobody assembly using pattern recognition," *J. Eng. Ind.*, vol. 118, pp. 55–66, 1996.

[6] D. Ceglarek and J. Shi, "Fixture failure diagnosis for sheet metal assembly with consideration of measurement noise," *J. Manuf. Sci. Eng.*, vol. 121, no. 4, pp. 771–777, Nov. 1999.

[7] Y. Ding, D. Ceglarek, and J. Shi, "Fault diagnosis of multistage manufacturing processes by using state space approach," *J. Manuf. Sci. Eng.*, vol. 124, no. 2, pp. 313–322, 2002.

[8] S. Jin, Y. Liu, and Z. Lin, "A Bayesian network approach for fixture fault diagnosis in launch of the assembly process," *Int. J. Prod. Res.*, vol. 50, no. 23, pp. 6655–6666, 2012.

[9] K. Bastani, B. Barazandeh, and Z. Kong, "Fault diagnosis in multistation assembly systems using spatially correlated Bayesian learning algorithm," *J. Manuf. Sci. Eng.*, vol. 140, no. 3, 2017, Art. no. 031003.

[10] G. Arvanitis, A. Lalos, and K. Moustakas, "Robust and fast 3D saliency mapping for industrial modeling applications," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 1307–1317, Feb. 2021.

[11] L. D. Xu, C. Wang, Z. M. Bi, and J. Yu, "AutoAssem: An automated assembly planning system for complex products," *IEEE Trans. Ind. Informat.*, vol. 8, no. 3, pp. 669–678, Aug. 2012.

[12] Q. Rong, J. Shi, and D. Ceglarek, "Adjusted least squares approach for diagnosis of ill-conditioned compliant assemblies," *J. Manuf. Sci. Eng.*, vol. 123, no. 3, pp. 453–461, 2001.

[13] F. Adly *et al.*, "Simplified subspaced regression network for identification of defect patterns in semiconductor wafer maps," *IEEE Trans. Ind. Informat.*, vol. 11, no. 6, pp. 1267–1276, Dec. 2015.

[14] Z. Kong, D. Ceglarek, and W. Huang, "Multiple fault diagnosis method in multistation assembly processes using orthogonal diagonalization analysis," *J. Manuf. Sci. Eng.*, vol. 130, no. 1, 2008, Art. no. 011014.

[15] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi, "Machine learning for predictive maintenance: A multiple classifier approach," *IEEE Trans. Ind. Informat.*, vol. 11, no. 3, pp. 812–820, Jun. 2015.

[16] C. R. Qi *et al.*, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 77–85.

[17] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2015, pp. 922–928.

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

[19] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Proc. Neural Inf. Process. Syst. Conf.*, 2017, pp. 5574–5584.

[20] C. Leibig *et al.*, "Leveraging uncertainty information from deep neural networks for disease detection," *Sci. Rep.*, vol. 7, no. 1, 2017, Art. no. 17816.

[21] Y. Yang, W. Li, T. A. Gulliver, and S. Li, "Bayesian deep learning-based probabilistic load forecasting in smart grids," *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4703–4713, Jul. 2019.

[22] P. Franciosa *et al.*, "A novel hybrid shell element formulation (QUAD+ and TRIA+): A benchmarking and comparative study," *Finite Elem. Anal. Des.*, vol. 166, 2019, Art. no. 103319.

[23] C. Blundell *et al.*, "Weight uncertainty in neural networks," in *Proc. 32nd Int. Conf. Mach. Learn.*, vol. 2, 2015, pp. 1613–1622.

[24] Y. Wen *et al.*, "Flipout: Efficient pseudo-independent weight perturbations on mini-batches," in *Proc. 6th Int. Conf. Learn. Representations*, 2018.

[25] D. M. Blei *et al.*, "Variational inference: A review for statisticians," *J. Amer. Statist. Assoc.*, vol. 112, no. 518, pp. 859–877, 2016.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Neural Inf. Process. Syst. Conf.*, 2012, pp. 1097–1105.

[27] L. Li *et al.*, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 6765–6816, 2017.

[28] M. Stein, "Large sample properties of simulations using Latin hypercube sampling," *Technometrics*, vol. 29, no. 2, pp. 143–151, 1987.

[29] Y. Wu and K. He, "Group normalization," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.

[30] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.

[31] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.

[32] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. 12th USENIX Conf. Oper. Syst. Des. Implementation*, 2016, pp. 265–283.

[33] S. Sinha, P. Franciosa, and D. Ceglarek, "Bayesian deep learning for manufacturing," 2020. [Online]:Available. https://github.com/sumitsinha/Deep_Learning_for_Manufacturing

[34] D. Ceglarek and P. Prakash, "Enhanced piecewise least squares approach for diagnosis of ill-conditioned multistation assembly with compliant parts," *Proc. Inst. Mech. Eng. B J. Eng. Manuf.*, vol. 226, no. 3, pp. 485–502, 2012.

[35] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[36] H. Su *et al.*, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 945–953.

[37] J. Han, H. Chen, N. Liu, C. Yan, and X. Li, "CNNs-based RGB-D saliency detection via cross-view transfer and multiview fusion," *IEEE Trans. Cybern.*, vol. 48, no. 11, pp. 3171–3183, Nov. 2018.

[38] X. Li, Y. Jiang, M. Li, and S. Yin, "Lightweight attention convolutional neural network for retinal vessel segmentation," *IEEE Trans. Ind. Informat.*, vol. 17, no. 3, pp. 1958–1967, Mar. 2021.

Sumit Sinha received the B.Tech. degree in industrial and systems engineering from the Indian Institute of Technology (IIT), Kharagpur, Kharagpur, India, in 2017. He is currently working toward the Ph.D. degree in deep learning for manufacturing with Warwick Manufacturing Group, University of Warwick, Coventry, U.K.

From 2017 to 2018, he was a Data Scientist with ZS Associates, Evanston, IL, USA, in the application of data science in pharmaceutical sales and marketing. He was also briefly with the Hong Kong University (HKU), Hong Kong, and Chongqing University, China, working on data science projects. His research interests include Bayesian deep learning and digital twins of manufacturing assembly systems.

Pasquale Franciosa received the Ph.D. degree in mechanical engineering from the University of University of Naples Federico II, Italy, in 2010.

He is an Assistant Professor with the University of Warwick, Coventry, U.K., and the Head of the Laser Welding Applications Laboratory, Warwick Manufacturing Group, University of Warwick. He has been Principal Investigator (PI) and Co-Investigator (Co-I) on several funded projects with a total income to the University of Warwick of circa £2.8M since 2015. He has authored or coauthored more than 80 papers. His research interests include smart manufacturing, process monitoring, closed-loop control, applications of machine learning/artificial intelligence, and multidisciplinary optimization, with specific attention on assembly systems and laser processes.

Dr. Franciosa was the recipient of four Best Paper Awards. He is a Member of the Editorial Board of the *ASTM Smart and Sustainable Manufacturing Systems* journal.

Dariusz Ceglarek received the Ph.D. degree in mechanical engineering (ME) from the University of Michigan-Ann Arbor, Ann Arbor, MI, USA, in 1994.

He is an EPSRC Star Recruit Research Chair with the University of Warwick, Coventry, U.K. He was a Professor with IS&E, University of Wisconsin-Madison, Madison, WI, USA. He has been mechanical engineering (ME)/Co-Investigator (Co-I) on research grants of over £30M: NSF/NIST/EPSRC/InnovateUK/APC/EU-FP7/Curie and industry. He has authored or coauthored more than 180 papers and is listed by Stanford University among top 2% of the world's leading scientists. His research interests include smart manufacturing and data mining/AI for root cause analysis across design, manufacturing, and service.

Prof. Ceglarek is a Fellow of College International pour la Recherche en Productique (CIRP). He was the recipient of several Best Paper Awards, the 2018 JLR "Innovista" Award for the most innovative "piloted technology," the EPSRC Star Award, and the NSF CAREER Award. He was the Associate Editor for *ASTM Smart and Sustainable Manufacturing Systems*, the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, and the *ASME Journal of Manufacturing Science and Engineering*.