Bayesian dropout approximation in deep learning neural networks: analysis of self-aligned quadruple patterning

Scott D. Halle,[a] Derren N. Dunn,[a] Allen H. Gabor,[a,*] Max O. Bloomfield,[b] and Mark Shephard[b]
[a]IBM TJ Watson Research Center, Albany, New York, United States
[b]Rensselaer Polytechnic Institute, Troy, New York, United States

Abstract

Background: Predictive estimates of the final process outcome(s) of multistep, coupled processes can be difficult to make based on data measured at the various process steps. Self-aligned quadruple patterning (SAQP) is an example of such a process where the prediction of pitch-walk is desired at the various process steps.

Aims: Be able to both predict pitch-walk values and the uncertainty in the predicted values at SAQP process steps based on optical critical dimension (OCD) spectroscopy outputs (dimensions, angles, thicknesses, and so on) of mandrel, spacer, and other SAQP features.

Approach: Train a neural network using OCD-modeled values of an SAQP process to be able to predict SAQP pitch-walk at early process steps. Use Bayesian dropout approximation (BDA), a methodology using Bayesian inference with stochastic neural networks, to estimate uncertainty in the predicted SAQP pitch-walk.

Results: Able to predict pitch-walk values, and the uncertainty in the predictions, of the final SAQP structure after the deposition of the first spacer. The pitch-walk predictions become more accurate as OCD information from the bottom mandrel RIE and bottom spacer are added as inputs to the BDA network.

Conclusions: In contrast to a single output value that traditional neural networks would predict, BDA makes an estimated distribution of predictions, where the BDA network gives both a most likely value as well as a distribution of potential values. While this paper shows the power of BDA to predict SAQP pitch walk, it is expected that BDA will be a valuable tool to analyze many data sets in semiconductor manufacturing to help improve yield and performance.

 [DOI: 10.1117/1.JMM.21.4.041604]

Keywords: pitch-walk; self-aligned quadruple patterning; deep neural network; Bayesian dropout approximation.

Paper 22008SS received Feb. 21, 2022; accepted for publication Oct. 5, 2022; published online Nov. 8, 2022.

1 Introduction

Self-aligned quadruple patterning (SAQP) is a method for enabling sub-lithographic patterning that has been extensively discussed in the literature.[1–5] While it can be used to pattern many line-space layers, this paper will describe the use of SAQP for the fin layer. Indeed, SAQP is actively employed in semiconductor manufacturing of FinFET devices.[5,6]

The SAQP process employs multiple nonlithographic sidewall spacer image transfers to reduce the pitch to a quarter of the original lithographic pitch. Errors in earlier process steps can propagate through the subsequent deposition and etch steps resulting in unwanted variations in the final structure created with SAQP. Indeed, the difficulty of controlling the complicated

*Address all correspondence to Allen H. Gabor, Allen.Gabor@ibm.com

process sequence of SAQP has been noted by many authors.[3,7,8] One particularly troublesome process-induced variation, the geometric oscillation of the quartered-pitch features, is commonly referred to as pitch-walk.[3]

Chao et al.[6] used optical critical dimension (OCD) data to create a calibrated SAQP measurement model, using a data feedforward approach and verification by reference metrology. A similar approach for measurement of the SAQP pitch-walking has been demonstrated with OCD by Kagalwala et al.,[9] using a virtual reference, instead of the calibrated reference of Chao et al.[6] While the OCD work enables the extraction of precise two-dimensional (2D) measurements of the stack geometric parameters (which will be used in this study) it does not enable reliable pitch-walk predictions of the final SAQP structure to be made early in the SAQP process flow.

Given the complexity of its coupled multistep process, it is inefficient to guard against pitch-walk from SAQP by relying on specifications of individually measured parameters from the different process steps. Thus, rather than specifying limits for individual parameters measured by OCD, it is highly desirable to be able to take those individual parameters, feed them into a model and predict what the pitch-walk value will be if the wafers continue processing. One of the goals of the work described in this paper is to predict the pitch-walk of the final SAQP structure early in the process. This early projection of the pitch walk enables decisions on reworking wafers, scrapping wafers, or feeding forward corrections to future processing steps so that downstream processing bandwidth is not wasted on wafers that will not meet the technology requirements. While other papers mention models to predict pitch walking, our paper is the first to document the usefulness of its predictions at different process steps within the SAQP module. A more detailed comparison of other prediction methodologies will be examined later in this manuscript.

The desire to predict pitch-walk as early as possible in the SAQP process flow led us to the use of deep neural network (DNN) methods. Spurred by the fast implementation of DNNs on GPUs,[10,11] DNNs have been employed in a wide variety of fields over the last decade.[12–14] In particular, they have been highly successful in producing regressions over high-dimensional spaces. In this work, we investigated such a space in the form of the various geometric parameters measured over the process history of wafers making their way through the SAQP module.

One downside of typical DNN regression models is that they act as point estimators, reporting a single prediction for any given input vector, without reflecting the uncertainty associated with the variability of a real manufacturing process followed by a real measurement process. This inability to account for the uncertainty in the prediction would limit the usefulness of pitch-walk predictions for making decisions regarding wafer scrapping, wafer reworking, or making feedforward process corrections. As an example, the pitch walk may be predicted to be 3 nm, where the specification is <2 nm. If the 1σ uncertainty in that prediction is ± 3 nm the decision may be to continue processing the wafer as there is the chance that the wafer will end up within the specifications. On the other hand, if the uncertainty is ± 0.1 nm, the decision may be to scrap the wafers and not waste further downstream processing bandwidth.

2 Methodology

To develop quantitative uncertainty estimates, we employ a DNN methodology adapted from work by Gal and Ghahramani[15] to make predictions of the probability distribution function (PDF) that represents all possible outcomes for pitch-walk at the end of the SAQP process. With an estimate of the entire PDF available, informed decisions can be made regarding the reworking or scrapping of wafers that are not expected to meet a particular target threshold while accounting for uncertainty both due to imperfect measurement and modeling (epistemic uncertainty) and due to the variability inherent in the manufacturing process (aleatoric uncertainty).

The uncertainty represented by the PDF depends on the fraction of the total number of steps in the module providing data to the network making the prediction. Predictions made later in the module, and thus having more input values to the DNN, result in a narrowing of the predicted PDF. We note that this stochastic approach is not tied to the physics of the SAQP module and has broader applicability to model many complex problems related to devices and semiconductor processes.

2.1 *Experimental Measurement and Interpretation*

OCD measurements were obtained at five discrete process steps in the SAQP process (as described in Sec. 3) using the methodology described by Chao et al.[6] The OCD dataset of ~30 wafers was collected from a stable process route under active process control without any experimental splits. For each of the 30 wafers undergoing metrology, 20 sites were measured. After each process step, a number of geometric stack parameters were extracted from OCD.

For modeling, up to 16 parameters were used as inputs to the DNN. These 16 parameters were a subset of the parameters measured with OCD at the five discrete process steps and will be described in more detail in Sec. 3. Note, the 30 wafers used for this study only included wafers in which all the OCD measurements were available for all five process steps, i.e., wafers scrapped before the final fin pitch-walk measurement were not included. Additional culling of the data included filtering with a goodness-of-fit parameter threshold.

2.2 *Bayesian Dropout Approximation Approach*

The typical approach for developing a regression model based on a DNN is to first identify a training set consisting of correlated vectors of inputs and outputs. The network topology and activation functions are then chosen, with floating point representations of weights and biases stored at each node. These weights and biases are adjusted in an optimization loop to allow the network to reproduce the behavior of the training set. In this construction, the predictions of the network are point estimates of the regressed quantities that are fully determined by the input vectors.

An estimate of the error of a trained network can be made by averaging the error in the predictions made on a validation set and assigning that error to all predictions. However, this averaging results in a global estimate for the network as a whole that is not a function of the input. Because training data cannot represent the entire input space and because of the inability of any real training process to capture training data perfectly, the ability of a network to make predictions is better for some inputs than others, often by orders of magnitude.

To extract estimates of the uncertainty from a regression network, in this study we exploit an interpretation proposed by Gal and Gharamani[15,16] of a standard neural network regularization technique known as a dropout. In standard dropout,[17] during any given training step, each node has a probability p of being multiplied by zero, effectively severing its connection to the rest of the network. During inference, that is when the network is used to make predictions, the output of each node is multiplied by $1/(1-p)$. Empirically, it is found that dropout decreases the tendency of networks to overfit and increases the performance of a trained network on test data not in the training data set.[18,19]

Within the reinterpretation of the dropout technique due to Gal and Gharamani,[15,16] here referred to as Bayesian dropout approximation (BDA), the training of the network progresses as with standard dropout. However, any inference includes the stochastic multiplication by zero with the same probability p as was used during training and without the correction factor of $1/(1-p)$. Typically, the network is sampled with different dropout vectors many times for any given input vector, as demonstrated schematically in Fig. 1(a), and the statistics of the outputs are computed with a sample mean serving as a predicted value and the sample standard deviation serving as a measure of the uncertainty in that prediction. Figure 1(b) shows an example of the statistics that can be generated through this type of procedure.

A careful reading of the appendix of Ref. 15 will allow the reader to understand the sampling during inference as a Monte Carlo integration of the product of the likelihood and the posterior distributions, within the context of a variational inference approach to computing the Bayesian distribution. Alternatively, and perhaps more intuitively, the repeated stochastic inference may be thought of as sampling an ensemble of networks each of slightly different topology and each consistent with the training data set. In this interpretation, in the limit of long training and a large number of samples, the distribution of predictions arises from the variability in the data-generating process being reflected in the variability in the training data set.

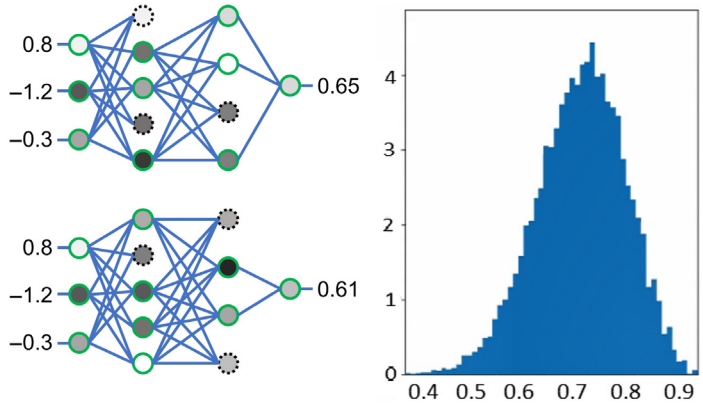


Fig. 1 (a) A schematic of DNNs employing the BDA approach. Both networks have identical inputs, but 33% of nodes are zeroed out randomly (shown with black dashed borders), giving different network outputs. (b) A typical histogram of outputs after a large number of samples with identical inputs.

We have implemented the BDA algorithm in Python, using the Tensorflow package[19,20] and a custom, well-instrumented layer class that implements dropout during training and inference. Using this software, a small library of stochastic neural networks, as discussed in Sec. 4, was trained, based on 500 data points, and expanded to 10,000 data points via the data augmentation technique defined in Sec. 2.3; 20% of these points were selected at random and reserved for model validation. This library contains networks with inputs taken from each of the first four OCD measurement steps. In all cases, the network outputs being regressed were the pitch-walk measurements ($\alpha - \beta$ and $\alpha - \gamma$), as discussed in Sec. 3, from the final fin step. Inputs from each step were selected based on the likelihood of affecting this pitch walk, though this selection is made permissively, erring on the side of inclusion rather than exclusion. The computational cost of including input with a low gain is minimal during inference. These networks were trained using a 15% dropout, heuristically tuned to give average squared z-scores near unity, and an Adam stochastic optimization algorithm[21] with hand-tuned learning rates.

As will be discussed more fully in Sec. 4, to make a pitch-walk prediction based on a set of OCD parameters, those parameters are used as an input vector to the trained network and 100 to 1000 samples are taken to generate an output probability distribution. On consumer-level laptops, such a computation is likely to be complete in tens of seconds, making it ideal for inline applications. The sample mean and sample standard deviation of those outputs are then used as a predicted value and uncertainty.

2.3 *Data Augmentation*

Data augmentation is a well-known strategy[22] to increase the size of a data set to aid the optimization routine during network training by filling in gaps in the parameter landscape. We augment our data set through the creation of new, albeit not independent, data points by interpolating existing data points. We interpret the output features as a vector function on the n-dimensional space represented by the n input features. We can then construct a simplex in this input space using $n + 1$ data points and represent the output vector as a finite element field on that simplex with linear basis functions.[23] To create a new data point, we choose a point within that simplex, representing the input vector, and interpolate the basis functions to find the corresponding output vector. In lieu of creating a full finite element mesh of the n-dimensional space, we randomly select sets of $n + 1$ data points to form simplices and reject any set for which the resulting simplex does not meet restrictions on size and quality. These restrictions are determined heuristically to avoid interpolating across too large a distance in parameter space. It is unknown if the data augmentation technique introduces bias.

3 Self-Aligned Quadruple Patterning

An SAQP fin process can result in pitch-walk, which is defined as a variation of the space-width between neighboring fins. These space-width differences have previously been defined by the geometrical SAQP model by Chao et al.[6] The process steps of this SAQP model are shown in Fig. 2. The five distinct process steps correspond to the OCD measurement steps used in this work, including the following: (1) the top mandrel after lithography and RIE etching (TM_{RIE}), (2) the top mandrel postspacer deposition (TM_{SP}), (3) the top spacer etch followed by top mandrel pull and RIE etching to form the bottom mandrel (BM_{RIE}), (4) the bottom mandrel post spacer deposition (BM_{SP}), and (5) the final fin formation at fin reveal (FIN). Three different space-widths formed between the FINs at the final step of the SAQP process, as shown in Fig. 2, are indicated by space-width designations α, β, γ used by Kagalwala et al.[9] To be specific, the OCD measurement steps one to five are measurement steps at five distinct process steps that occur sequentially in the SAQP flow.

For the SAQP DNN training, the input dataset is based on measurements of the geometric parameters from optical scatterometry (OCD). The elucidation of the geometric SAQP model shown in Fig. 2 not only illustrates the fin space-widths, α, β, γ but also can be used to formulate the network topologies comprised of different process-step parameters, measured by OCD, that are used in the training of the DNNs. The geometrical SAQP model describes how the parameters such as mandrel widths and space widths between the mandrels can influence the space-width differences in the FIN structures. In this work, the severity of pitch walk is evaluated by looking at the values of $\alpha - \beta$ and $\alpha - \gamma$.

Each process step of the SAQP can be approximated by an analytical equation based on geometrical process parameters and their physical relationships. For example, the OCD data for each process step can be solved by an analytic equation at each successive step in the process sequence. The equations can be evaluated independently or by passing the output geometric parameters from one process equation step to the next. For the later case, a series of time-ordered sequential analytical equations[6,24] can represent the SAQP process. While the analytic approach is ideal for fitting and extracting parameters from the OCD measurements, it does not make predictions about fin pitch-walking. Specifically, the space widths at the FIN step are only determined from the evaluation of the FIN analytic approximation to FIN OCD data, i.e., the last step of the SAQP sequence. It is the aim of this work to predict pitch-walk values, using data from earlier processing steps as interpreted using the analytic approach.

Pitch-walk occurs when the space-widths α, β, and γ are not equal. Nonidealities in the pattern transfer steps of the SAQP process can result in errors that result in pitch-walking.

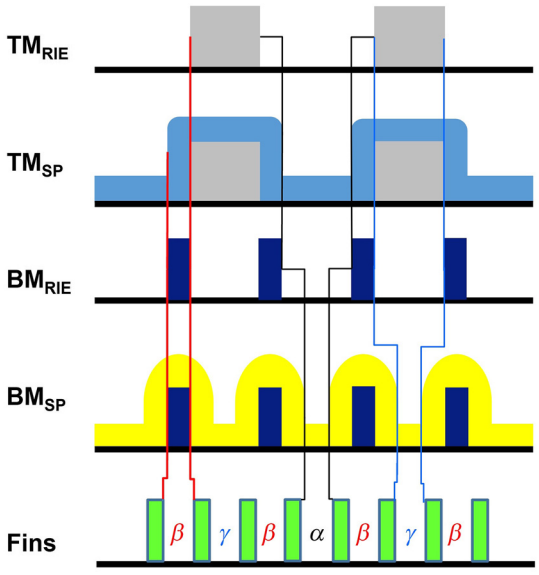


Fig. 2 A schematic of the process steps for SAQP.

Table 1 The sign (– or +) of the gains between the process steps TM_{RIE}, TM_{SP}, BM_{RIE}, and BM_{SP}, the SAQP spacing parameters of α, γ, and β. An entry of 0 indicates little or no dependency.

Process step	α space	γ space	β space
TM_{RIE}	–	+	0
TM_{SP}	–	0	+
BM_{RIE}	–	–	+
BM_{SP}	–	–	0

The unique dependencies of pitch-walk on the nonidealities of the discrete processing steps are now elaborated. From the geometrical SAQP model shown in Fig. 2, we can define the relationship between increasing or decreasing the top mandrel, top mandrel spacer, bottom mandrel, and bottom mandrel spacer widths on the fin space-widths α, β, and γ.

- α has an inverse relationship with the following parameters: the top mandrel width in TM_{RIE}, the width of the spacer in TM_{SP}, the bottom mandrel width in BM_{RIE}, and the width of the spacer in BM_{SP}, i.e., as top mandrel CD, top spacer width, the bottom mandrel CD and bottom spacer width increase the α space decreases.
- β increases as both the width of the spacer in TM_{SP} and the bottom mandrel width (BM) in BM_{RIE} increase. In contrast, an increase in both the top mandrel width in TM_{RIE} and the width of the spacer in BM_{SP} should have no impact on β.
- γ increases as the top mandrel width in TM_{RIE} increases. However, γ has an inverse relationship with the bottom mandrel width in BM_{RIE} and the width of the spacers in BM_{SP}. The width of the spacer in TM_{SP} does not have an impact on γ.

Process excursions or variability of the different structural parameters during the SAQP process can contribute to the pitch-walk at the FIN step. The relationship trends between the geometric structural parameters at each SAQP process step and the fin space-width parameters α, β, and γ are illustrated in Table 1. Using the table, one can determine what SAQP processing deviations will contribute to pitch walking as measured by $\alpha - \beta$ and $\alpha - \gamma$. As an example, increasing either the bottom mandrel width in BM_{RIE}, or the width of the spacer in BM_{SP}, is found to decrease both α and γ, and thus even though BM_{RIE} and BM_{SP} are not at nominal, the $\alpha - \gamma$ pitch walk parameter will not be impacted. However, because increasing BM_{RIE} increases the β space, the $\alpha - \beta$ pitch walk parameter will be impacted. Similarly, the $\alpha - \beta$ pitch walk parameter will be impacted by BM_{SP} since α has a negative relation and β has a neutral (no change) relationship.

The process sensitivities shown in Table 1 and the magnitude of the expected process errors can enable a deeper understanding of what drives pitch walking. For example, since the spacer deposition process in both top and bottom mandrels has an extremely tight process control, it is expected to have a lower impact than the mandrel size on the fin space-widths. Therefore, the top mandrel features with opposite sign contributions are expected to have a large influence on the $\alpha - \gamma$ pitch walk parameter. Since the magnitude of the pitch-walk FIN space-width differences $\alpha - \beta$ and $\alpha - \gamma$ are experimentally determined, we chose these two parameters as the output layer for the DNN.

4 Network Topology FOR SAQP

As previously mentioned, the different network topologies used for training DNNs in this study are comprised of different process-step parameters based on the geometric model for the SAQP process sequence. It should be emphasized that because geometrical rules and relationships are not built into the DNN, in contrast to a geometrical SAQP model, the SAQP DNN is not a physics-based model. In an analytic approach, the process parameters are determined by sequentially evaluating the equations in the SAQP process sequence. In contrast, each network created

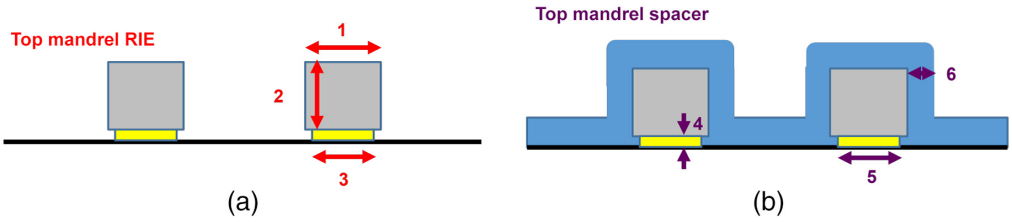


Fig. 3 The stack geometry parameter definitions for the six-input network. (a) geometry parameter definition and (b) geometry parameter definition.

in this work is trained to regress pitch-walk metrics against a subset of measured quantities chosen from the OCD measurement step over the process history of the wafer. These measurements are taken independently, so the choice need not be based on process-sequence order.

When choosing which quantities are well-suited to be input into our DNNs, two characteristics were considered: (1) the quantity should represent an aspect of the intermediate geometry that impacts pitch-walk as measured in the final step of the process and (2) quantities measured later in the process delay the use of the network for predictions. The more quantities included that possess the first property, the narrower the predicted PDF and the more certain the pitch-walk prediction can be. Rather than strike a compromise in this trade-off between certainty and early prediction, we constructed three different networks with three increasingly complete input vectors (of sizes 6, 10, and 14), with each network regressing the same pitch-walk metrics. The output of each of these three networks yields predicted final pitch-walk values well ahead of the final fin RIE step.

Using Python-based TensorFlow,[19] these different networks are trained with three hidden layers. The nodes of the output layer for all these networks are the pitch-walk metrics $\alpha - \beta$ and $\alpha - \gamma$. For all three SAQP networks in this study, the number of nodes in the three hidden layers is 100, 100, and 50, respectively. Next, the descriptions for the three networks are given.

The six-input network contains inputs only from the top mandrel, with three geometric parameters each from TM_{RIE} and TM_{SP}. Figure 3(a) shows the top mandrel stack at TM_{RIE} where the geometric stack parameters 1 to 3 correspond to top mandrel top width, top mandrel height, and undercut hardmask layer bottom width, respectively. Figure 3(b) shows the top mandrel stack at TM_{SP}, where the geometric stack parameters 4 to 6 correspond to undercut hardmask layer height, undercut hardmask layer bottom width, and the sidewall spacer width along the top mandrel. Figure 4 illustrates the six-input network with TM_{RIE} and TM_{SP} parameters from Fig. 3 as input. Note that as mentioned earlier, the two nodes on the output layer are both final fin pitch-walk parameters, $\alpha - \beta$ and $\alpha - \gamma$.

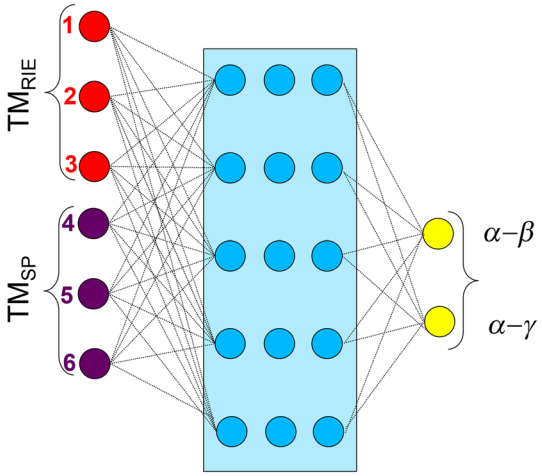
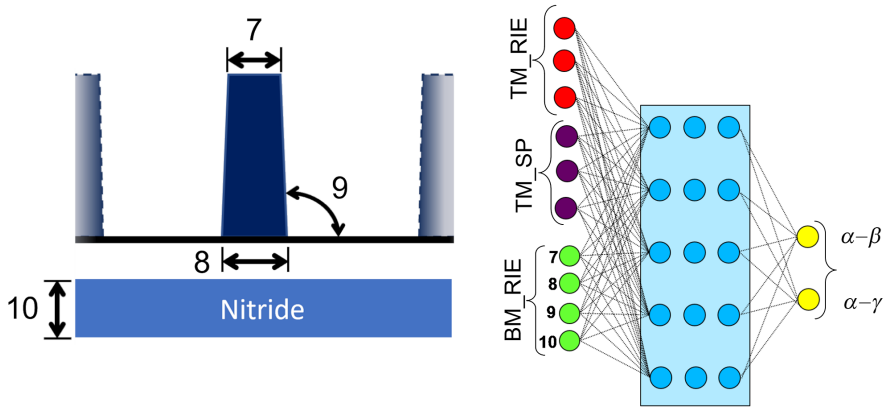
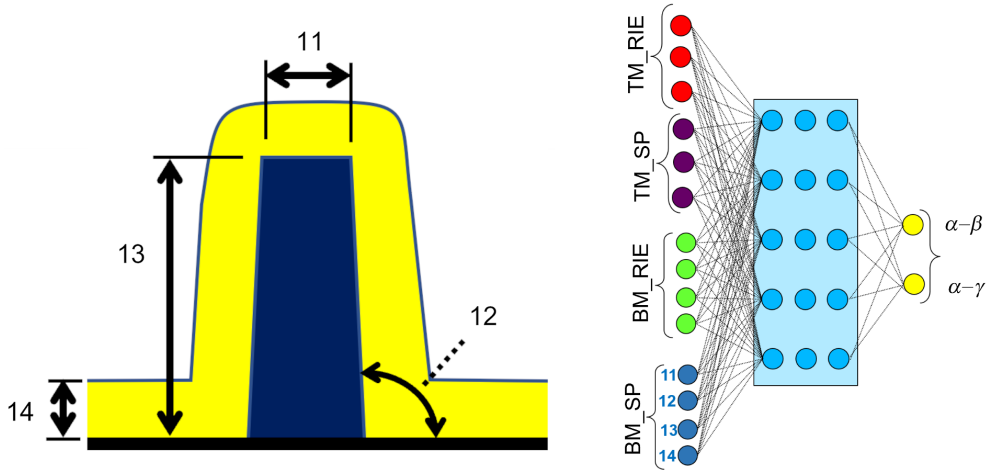


Fig. 4 Schematic of six-input network with inputs from TM_{RIE} and TM_{SP}, as defined in Fig. 3.



(a) BM_{RIE} geometry parameter definitions (b) 10-parameter network schematic

Fig. 5 (a) A schematic defining the stack geometry parameter definitions for the bottom mandrel BM_{RIE} for (b) the 10-input network.



(a) BM_{SP} geometry parameter definitions (b) 14-parameter network schematic

Fig. 6 (a) A schematic defining the stack geometry parameter definitions for the bottom mandrel BM_{SP} for (b) the 14-input network.

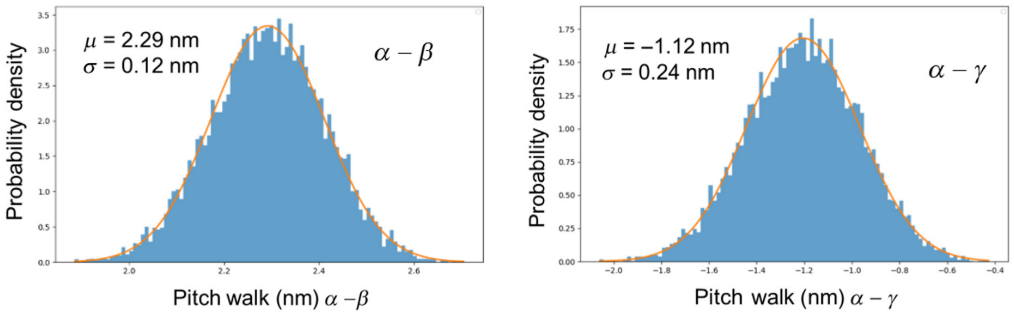
The 10-input network contains all input layer nodes from the 6-input network plus four additional geometric parameters from bottom mandrel step at BM_{RIE}. Figure 5 shows the bottom mandrel stack at BM_{RIE}, where the geometric stack parameters 7 to 10 correspond to bottom mandrel top width, bottom mandrel bottom width, bottom mandrel sidewall angle, and stack nitride thickness, respectively.

Likewise, the 14-input network contains all input layer nodes from the 10-input network plus four additional geometric parameters from bottom mandrel step at BM_{SP}. Figure 6(a) shows the bottom mandrel stack at BM_{SP}, where the geometric stack parameters 11 to 14 correspond to bottom mandrel top width, bottom mandrel sidewall angle, bottom mandrel height, and sidewall spacer width, respectively.

5 Results and Discussion

5.1 *BDA Predictions from Centroid of Input Data*

In this section we examine the output of a fully trained n-parameter SAQP DNN network using the methodology described in Sec. 2.2. By sampling the output of the forward-solve inference of



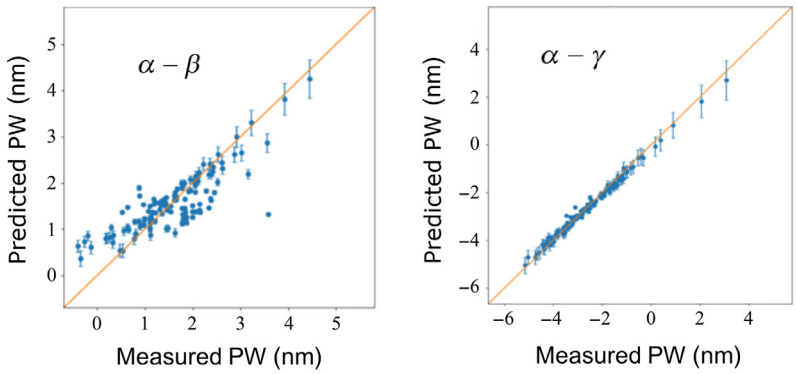
(a) Probability density for $\alpha - \beta$ (b) Probability density for $\alpha - \gamma$

Fig. 7 The probability densities predicted for (a) $\alpha - \beta$ and (b) $\alpha - \gamma$ with the six-parameter network with inputs having their respective mean values.

a given BDA network at a particular set of inputs, we can predict the distribution of the pitch-walk metrics $\alpha - \beta$ and $\alpha - \gamma$ we expect from a system with those inputs. For demonstration purposes, here we take the mean of each of the input parameters, which is the centroid of the input data, as a "typical site." Sampling the six-parameter network at this input centroid, we determine histograms of the probability density for $\alpha - \beta$ [Fig. 7(a)] and for $\alpha - \gamma$ [Fig. 7(b)]. As the distribution is approximately Gaussian, it is useful to interpret the mean and sample standard deviation of the network output as a predicted process output and prediction uncertainty. This yields predicted uncertainties for $\alpha - \beta$ and $\alpha - \gamma$ of 0.12 and 0.24 nm, respectively. For a given dropout rate, prediction uncertainty serves as a convenient measure of the underlying distribution of the experimentally measured pitch-walk.

The usefulness of this measure of prediction uncertainty can be accessed from the correlation of the experimentally measured pitch-walk metrics as a function of the predicted pitch-walk. This correlation scatter plot is shown in Fig. 8(a) for $\alpha - \beta$ and Fig. 8(b) for $\alpha - \gamma$. The calculated 1σ values of the predicted pitch walk are shown. The selected dropout rate places 50% to 80% of z-scores within the $[-1, 1]$ interval for all test datasets.[25] This condition for a usable dropout rate allows for internal comparisons of uncertainty within data sets with a familiar scale of units. Note that the scale of the plots is displayed in nanometers and is not normalized for comparison. The uncertainty (i.e., 1σ values) at both the lower and upper values of the correlation scatterplot are significantly larger than those values that are clustered at the center of the plots. The data at the center of the plots, with a larger number of experimental values, are better determined.

It is observed that the mean values of the predicted $\alpha - \beta$ are not as precisely predicted at the $\alpha - \gamma$. This observation from the six-input network, reflecting only parameters from the top mandrel, is consistent with the previous discussion of SAQP. The $\alpha - \gamma$ pitch-walk is determined primarily from the top mandrel, whereas the $\alpha - \beta$ pitch-walk value is inherently determined



(a) Model correlation for $\alpha - \beta$ (b) Model correlation for $\alpha - \gamma$

Fig. 8 Correlation of the measured pitch-walk (PW) as a function of the predicted output of the six-input network for both (a) $\alpha - \beta$ and (b) $\alpha - \gamma$ with inputs having their respective mean values.

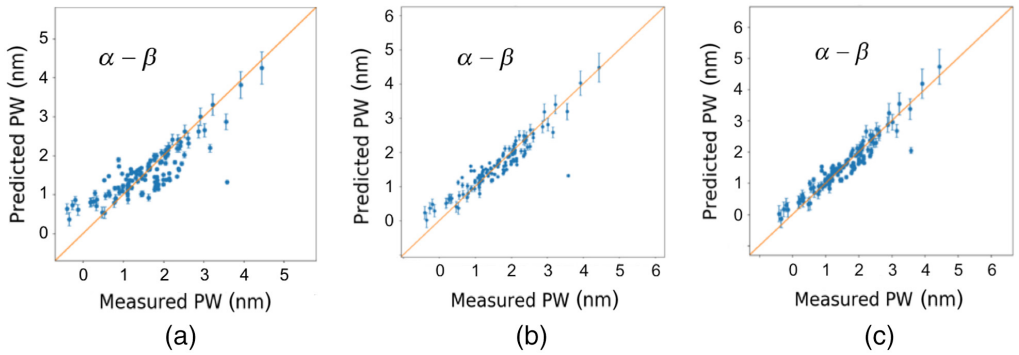


Fig. 9 Correlation of the measured PW metric $\alpha - \beta$ with the predicted PW from the networks for points in the test data set (a) six-input network, (b) 10-input network, and (c) 14-input network.

from both the top and bottom mandrel parameters, where the later parameter is not defined for this network. Figure 9 shows the correlation plots using the 6 parameter, where only top mandrel geometric parameter as inputted (Fig. 9a), as well as the 10 parameter (Fig. 9b) and 14 parameter (Fig. 9c) where bottom mandrel geometric parameters are also inputted. As these bottom mandrel parameters are added the delta between the measured and predicted values decreases.

5.2 *BDA Predictions from Individual Input Data Points*

In the previous discussion, we examined results from the forward-solve inference with different n-parameter networks for an artificial site defined by the mean (or centroid) of the geometric parameters. Using this approach, we gain an insight to the overall behavior of the SAQP networks. Alternatively, applying the forward-solve inference to individual experimental wafer/chip-sites is a more realistic use-case for applying the pitch-walk prediction. The forward-solve inference is applied to an arbitrary single chip location of a wafer with different n–parameter networks, allowing a comparison of pitch-walk predictions at different process steps of the SAQP process. These histograms of the probability density for both the $\alpha - \beta$ and $\alpha - \gamma$ pitch-walks are shown in Fig. 10 for each n-parameter network. Figure 10 shows a comparison of the predicted distributions to the experimental mean of the chip-site data, indicated by a red-colored vertical bar. By visually comparing the means of the predicted distributions to the experimental means, we can easily see that for both $\alpha - \beta$ and $\alpha - \gamma$ pitch-walks, increasing the number of input layer/nodes in the network improves the agreement between the predicted and

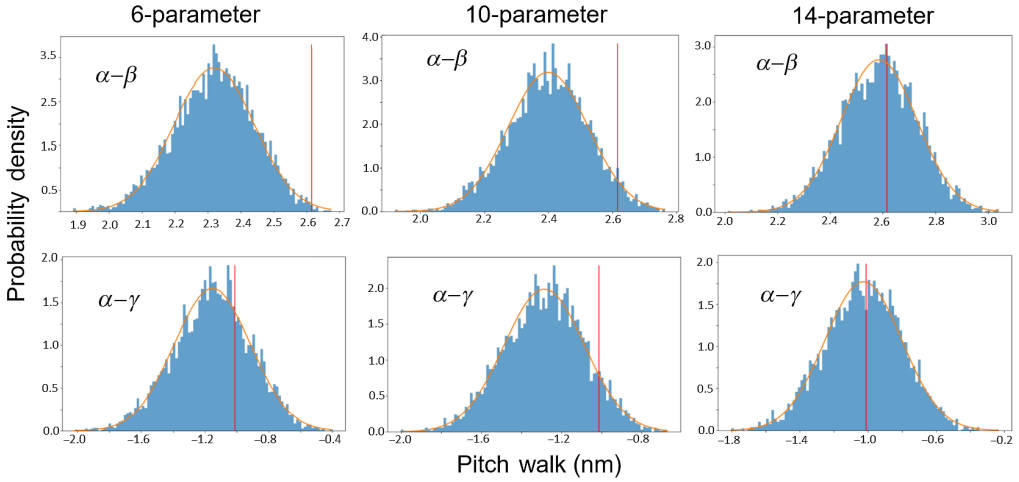


Fig. 10 The outcome of the probability density of the $\alpha - \beta$ and $\alpha - \gamma$ pitch-walk for an arbitrary wafer site location is shown for the 6-input, 10-input, and 14 parameter networks. The experimental mean of the chip-site data is indicated by a red-colored vertical bar.

experimental means. From the 6-input, 10-input, to 14 parameter networks, the improvement between the predicted and observed value for $\alpha - \beta$ pitch-walk is substantial, with a reduction of 2σ. This result is consistent with our previous discussion, in which the scatterplots of the predicted versus measured $\alpha - \beta$ pitch-walk improve with increasing the number of parameters. The pitch-walk uncertainty for both the $\alpha - \beta$ and $\alpha - \gamma$ does not significantly improve or degrade with increasing number of input layers. The application of pitch-walk predictions over a number of sites on a wafer, relative to a defined threshold value, can be applied to a predictive disposition process.

5.3 *Methodology: Sensitivity to Input Parameters*

A more nuanced understanding of the SAQP DDN is gained by exploiting the feedforward-solve inference under BDA, giving insight into the parameter sensitivity of the network. A methodology for gauging the sensitivity of input layer parameters to DNN is briefly explored here.

The predicted distribution of an output parameter is generated by systematically varying a chosen input layer parameter (η) over a small range, where the impact of η on the output. Figure 11 shows two 2D histograms, generated using the fourteen parameter network, of the probability density distribution for $\alpha - \gamma$ pitch-walk as a function of two different η, where η projects into the page and is allowed to vary over a range of 0.25 standard deviations. In Fig. 11(a), η is chosen to be one of the BM_{RIE} parameters, and the distribution of predicted $\alpha - \gamma$ pitch walk is shown to be rather insensitive to variations in η. However, when η is chosen to be a TM_{RIE} parameter in Fig. 11(b), we see that the predicted pitch-walk distribution is quite sensitive and the histogram responds strongly.

The area of high probability density presumably corresponds to the tight distribution of the parameter where the network was trained. These pitch-walk sensitivities shown here are consistent with our understanding of SAQP. The pitch-walk $\alpha - \gamma$ should be sensitive to certain top mandrel parameters, and not sensitive to bottom mandrel parameters. In principle, this analysis can be extended to n-dimensional input parameters. n-dimensional sensitivity is graphically complex and understanding parameter interactions are beyond the scope of this work. In summary, the ability to interrogate a DNN with this methodology is a computationally fast and powerful tool for understanding process sensitivities.

5.4 *Comparison to Other Prediction Approaches*

Other architectures exist for modeling a multistep process. For example, a predictive modeling conceptual framework using classifiers has been discussed by Stich et al.[26] In the framework from Stich et al. either machine learning or neural net classifiers are used to model yield on a process tool. This proposal also suggests that a cascading classifier approach, for sequential

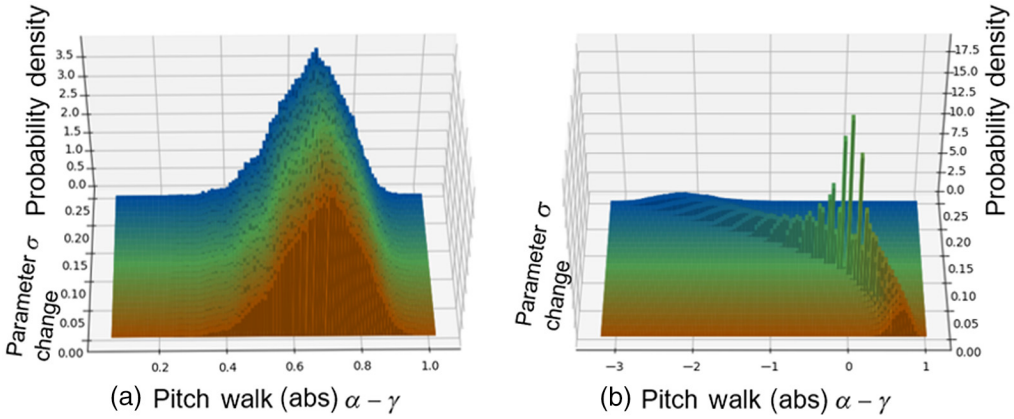


Fig. 11 (a) and (b) Two 2D histograms, generated using the fourteen parameter network, of the probability density distribution for $\alpha - \gamma$ pitch-walk as a function of two different η, where η projects into the page and is allowed to vary over a range of 0.25 standard deviations.

process tools, with feedforward corrections into the process recipe might be achieved. This ambitious approach has not yet been demonstrated for a complex production process flow such as SAQP.

Ren et al.[5] have recently published on the importance of a predictive model for pitch walk. Their approach uses a process-based analytical model based on the following two types of inputs: (a) metrology values of line widths and spacer thicknesses and (b) tool-specific characterization of the key process parameters, which gives experimental distributions of the process data, e.g., etch bias as a function of process temperature. The intent of the predictive analytical approach is twofold dependent on the following: (a) feedback control early in the SAQP process flow to improve PW control and (b) to experimentally assess the PW variance for different process control knobs (such as etch temperature).

In contrast to this analytical model, our fully empirical model cannot be explanatory of the underlying mechanism at play. However, while both approaches require measuring and providing key parameters, the analytical approach requires more domain experience to winnow the available parameters to a relevant set. When using the BDA approach, selection of parameters for inputs can be quite permissive, as the network will tune the gains of irrelevant parameters toward zero in the training process with enough data. There is a small cost in the number of network parameters to be trained and the number of floating point operations to be executed during inference by allowing less relevant parameters to become inputs to the model, but given the computationally lightweight nature of the BDA method, we feel this is a reasonable cost for the resulting physical agnosticism and low implementation initialization effort. Additionally, as the complexity of a process increases, the balance tilts further in the direction of the empiricism and mechanistic implementation of the BDA approach, as even high-level domain experts are hard-pressed to identify all possible relevant parameters and all the higher-order interactions between them. Such complexity can arise from large numbers of steps in a process module, an application that the BDA is well-suited for and allowing probabilistic yield assessments early in the process flow. As previously discussed, this probabilistic assessment could facilitate either re-work or the early removal of wafers that exceed the desired tolerances early in the process flow.

6 Conclusions

We have described in detail the complexity of SAQP and the need to have a predictive model for both the mean and the uncertainty of the pitch-walk prediction. Elaboration of the network input layers, the top and bottom mandrel geometric stack parameters, and their contribution to three types of space-widths α, β, and γ in the final pattern is given. The relationship of the stack parameters to the output layers of the network, pitch-walk $\alpha - \beta$ and $\alpha - \gamma$, is demonstrated. We defined the relevant network topologies and the input layers for modeling SAQP: six-input top mandrel only network, and 10 and 14 parameter networks including both top and bottom mandrels. Modeling the complex SAQP process with a stochastic DNN achieved a very good correlation of measured to predicted pitch-walk values despite the fact that the DNN model contains no knowledge of the physics of SAQP. The use of the BDA to perform Bayesian inference is an effective, easy-to-implement, and computationally fast method for making sophisticated predictions about the pitch-walk observed in SAQP. These predictions provide quantitative uncertainties and can be used in further business-relevant calculations for process outcomes.

The predicted pitch-walk for the n-parameter network gives a range for uncertainty from the probability density that is not found to significantly change by increasing the size of the input layer nodes. However, increasing the number of input layer nodes/parameters does improve the overall goodness-of-fit of the model predictions to the measured values for pitch-walk $\alpha - \beta$. Thus, increasing the number of input parameters does improve the pitch-walk prediction, (i.e., the predicted to experimental mean). For the available dataset, it is not unreasonable that the magnitude of the pitch-walk uncertainty is dominated by the top mandrel process and that further downstream processes do not increase or decrease the uncertainty. Thus, the ability to predict both the mean and uncertainty for the pitch-walk outcome early in SAQP process flow is a powerful methodology, which could be deployed for reliable manufacturing process disposition.

While the network input nodes in this SAQP study were limited by design intent, the methodology with the BDA approximation can easily be scaled to a much larger set of input parameters and network sizes. We have demonstrated that DNNs can be effectively used to model the complexity of SAQP and with the use of the BDA approximation provide actionable results. This approach can be readily extended to modeling other complex patterning processes, such as self-aligned double patterning and self-aligned octuple patterning. In addition, there are a host of opportunities to deploy the methodology demonstrated here to other device-yield scenarios. Uncertainty estimation has a large literature, and for future work, it would also be of interest to investigate competing approximation techniques such as Gaussian processes demonstrated in Ref. 24.

Acknowledgments

The authors would like to acknowledge both Drs. Geng Han and Robert Baseman for their useful insights and IBM for funding this project with the CCI (Center for Computational Innovations) at RPI. Finally, we thank SPIE and the Advanced Etch Technology for Nanopatterning IX Conference for the opportunity to present our work in this area.[27]

References

1. B. Mebarki et al., "Innovative self-aligned triple patterning for 1x half pitch using single "spacer deposition-spacer etch" step," *Proc. SPIE* **7973**, 79730G (2011).
2. Y. Chen, Q. Cheng, and W. Kang, "Technological merits, process complexity, and cost analysis of self-aligned multiple patterning," *Proc. SPIE* **8326**, 832620 (2012).
3. T. Standaert et al., "BEOL process integration for the 7 nm technology node," in *IEEE Int. Interconnect Technol. Conf./Adv. Metallization Conf. (IITC/AMC)*, IEEE, pp. 2–4 (2016).
4. A. Yeoh et al., "Interconnect stack using self-aligned quad and double patterning for 10 nm high volume manufacturing," in *IEEE Int. Interconnect Technol. Conf. (IITC)*, pp. 144–147 (2018).
5. H. Ren et al., "Advanced process control loop for SAQP pitch walk with combined lithography, deposition and etch actuators," *Proc. SPIE* **11325**, 392–400 (2020).
6. R. Chao et al., "Advanced in-line metrology strategy for self-aligned quadruple patterning," *Proc. SPIE* **9778**, 977813 (2016).
7. A. Raley et al., "Self-aligned blocking integration demonstration for critical sub-40 nm pitch MX level patterning," *Proc. SPIE* **10149**, 101490O (2017).
8. N. Felix et al., "EUV patterning successes and frontiers," *Proc. SPIE* **9776**, 97761O (2016).
9. T. Kagalwala et al., "Scatterometry-based metrology for SAQP pitch walking using virtual reference," *Proc. SPIE* **9778**, 97781W (2016).
10. K.-S. Oh and K. Jung, "GPU implementation of neural networks," *Pattern Recognit.* **37**(6), 1311–1314 (2004).
11. H. Jang, A. Park, and K. Jung, "Neural network implementation using iCUDA and OpenMP," in *Digital Image Comput.: Tech. and Appl.*, pp. 155–161 (2008).
12. K. T. Schütt et al., "Unifying machine learning and quantum chemistry with a deep neural network for molecular wavefunctions," *Nat. Commun.* **10**(1), 5024 (2019).
13. M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *PLoS One* **11**(6), e0155781 (2016).
14. M. Jhuria, A. Kumar, and R. Borse, "Image processing for smart farming: detection of disease and fruit grading," in *IEEE Second Int. Conf. Image Inf. Process. (ICIIP-2013)*, IEEE, pp. 521–526 (2013).
15. Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: representing model uncertainty in deep learning," in *Proc. 33rd Int. Conf. Mach. Learn. (ICML-16)* (2016).
16. Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: appendix," https://arxiv.org/abs/1506.02157 (2015).
17. G. E. Hinton et al., "Improving neural networks by preventing co-adaptation of feature detectors," https://arxiv.org/abs/1207.0580 (2012).

18. N. Srivastava et al., "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014).
19. M. Abadi et al., "Tensorflow: a system for large-scale machine learning," in *12th USENIX Symp. Oper. Syst. Des. and Implement. (OSDI 16)*, pp. 265–283 (2016).
20. M. Abadi et al., "TensorFlow: large-scale machine learning on heterogeneous systems," 2015, tensorflow.org.
21. D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," https://arxiv.org/abs/1412.6980 (2014).
22. T. Dao et al., "A kernel theory of modern data augmentation," https://arxiv.org/abs/1803.06084 (2018).
23. T. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Dover Civil and Mechanical Engineering, Dover Publications (2012).
24. S. D. Halle, K. Yeo, and D. Derren, "US patent application for predictive multi-stage modeling for complex process control," 20210049241 (2019).
25. W. Mendenhall and T. Sincich, *Statistics for Engineering and the Sciences*, Pearson Prentice-Hall (2007).
26. P. Stich et al., "Yield prediction in semiconductor manufacturing using an AI-based cascading classification system," in *IEEE Int. Conf. Electro Inf. Technol. (EIT)*, pp. 609–614 (2020).
27. S. D. Halle et al., "Bayesian dropout approximation in deep learning neural networks: analysis of self-aligned quadruple patterning," *Proc. SPIE* **11329**, 113290B (2020).

Scott D. Halle received his BA degree from Wesleyan University, his MSEE degree from Columbia University, and his PhD from Massachusetts Institute of Technology, followed by an NSF postdoctoral fellowship in the Department of Physics, University of Tokyo. Currently, he is working in the Computation Lithography Group, focused on aspects of modeling, optical proximity corrections and resolution enhancement techniques for advancing EUV technology. Previously, he was with the Advanced Lithography Research Group. His research has contributed extensively to advanced nanometer scale lithographic patterning methods and measurement techniques, including both experimental and computational methods for the development of EUV lithography. He is a senior scientist at IBM Research, Watson Research Center, Albany, New York, United States.

Derren N. Dunn is a computational patterning team lead at IBM's Albany Nanotechnology Laboratory, where he leads a team of engineers responsible for migrating electronic design automation workflows to public clouds. These workflows are focused on advanced resolution enhancement technology, VLSI design, and the interaction of computational patterning solutions with advanced node mask technologies. Prior to serving as computational patterning team lead, he held several team lead and management positions within IBM's Semiconductor Research and Development Center.

Allen H. Gabor received his PhD in materials science and engineering from Cornell University based on his work on block copolymer photoresists in 1996. He has worked in the field of lithography at Arch Chemicals, GLOBALFOUNDRIES and IBM. This work has included photoresist development, CD control, overlay minimization and 193 dry, immersion and EUV insertion. He is the author of more than 50 journal papers and holder of over 30 patents. He currently serves on the program committee for SPIE Extreme Ultraviolet (EUV) Lithography Conference and is a member of SPIE. He is a senior technical staff member at IBM.

Max O. Bloomfield has worked as a research staff at Rensselaer Polytechnic Institute since receiving his PhD in chemical engineering in 2007. His professional activities have focused on simulation of semiconductor processes at multiple scales, on models of chemical engineering unit operations, and on advanced machine learning-based and neural network-based techniques. He works closely with the Center of Computational Innovation at RPI, with Sandia National Labs, and with a variety of industrial partners.

Mark Shephard is the Samuel A. Johnson '37 and Elizabeth C. Johnson Professor of Engineering and Director of the Scientific Computation Research Center (SCOREC) at Rensselaer Polytechnic Institute. His research activities have led to well recognized and applied contributions on the areas of automatic mesh generation of CAD geometry, automated and adaptive analysis methods, and parallel adaptive simulation technologies. He has published more than 250 papers and graduated 24 PhDs.