

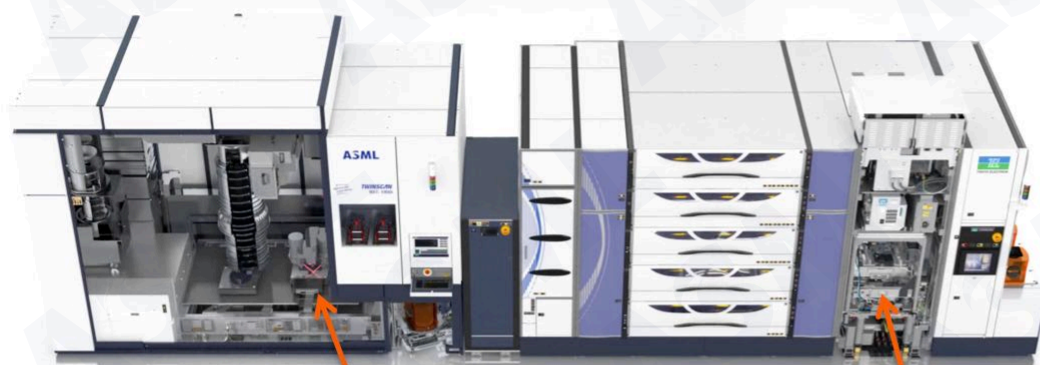


Virtual overlay metrology for fault detection supported with integrated metrology and machine learning

Emil Schmitt-Weaver

Introduction

- All wafers moving through a lithocluster get measured with **TWINSKAN metrology**
- Some wafers leaving the lithocluster get measured with **Integrated Metrology**



All Wafers have TWINSKAN Metrology (alignment, leveling)

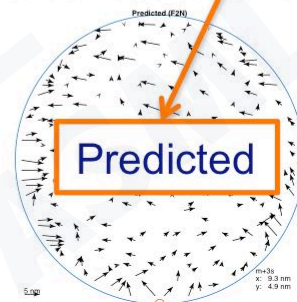
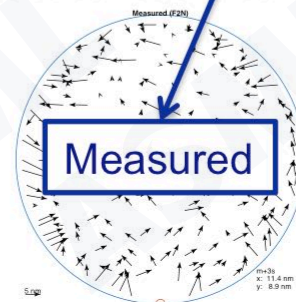
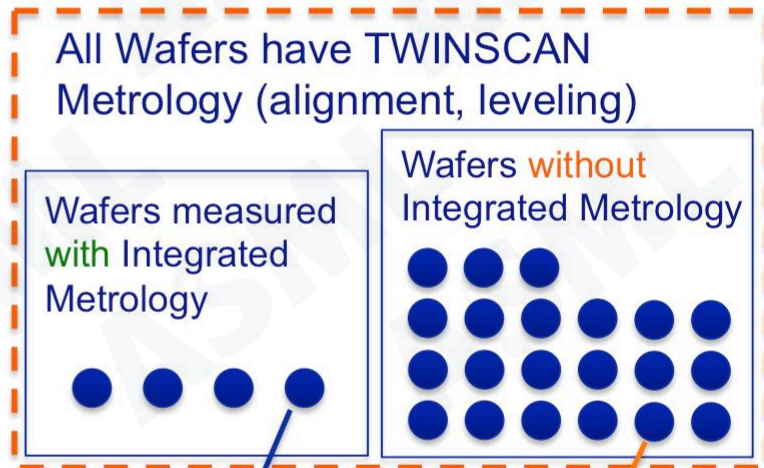
Wafers measured with Integrated Metrology

Wafers without Integrated Metrology

● ●

Introduction

- Our study explores the use of a regression based machine learning technique, known as “function approximation” to **predict overlay behavior for all of the wafers within a lot**



me=3s
x: 11.4 nm
y: 8.9 nm

me=3s
x: 9.3 nm
y: 4.9 nm

Outline

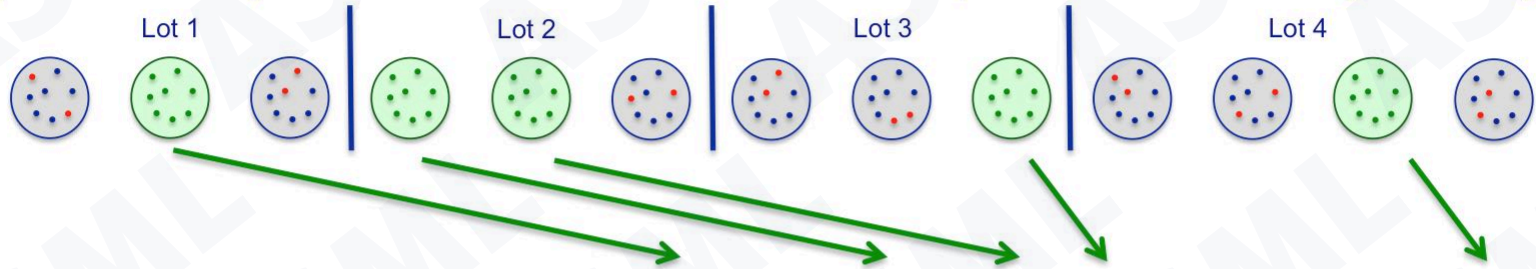
- Introduction
- How the function works
- Data separation into Training and Testing groups
- Training with Bayesian Automated Regularization
- Prediction Vs. Measured Overlay as regression plots
- Precision of Trained Function as a vector map
- Results
- Conclusion

Outline

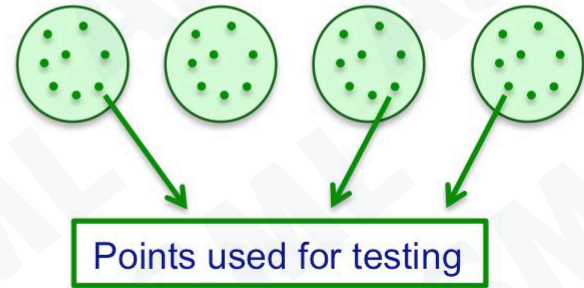
- Introduction
- How the function works
- Data separation into Training and Testing groups
- Training with Bayesian Automated Regularization
- Prediction Vs. Measured Overlay as regression plots
- Precision of Trained Function as a vector map
- Results
- Conclusion

Data separation into Training and Testing groups

1) We randomly select measured wafers from the exposed lots for training & testing

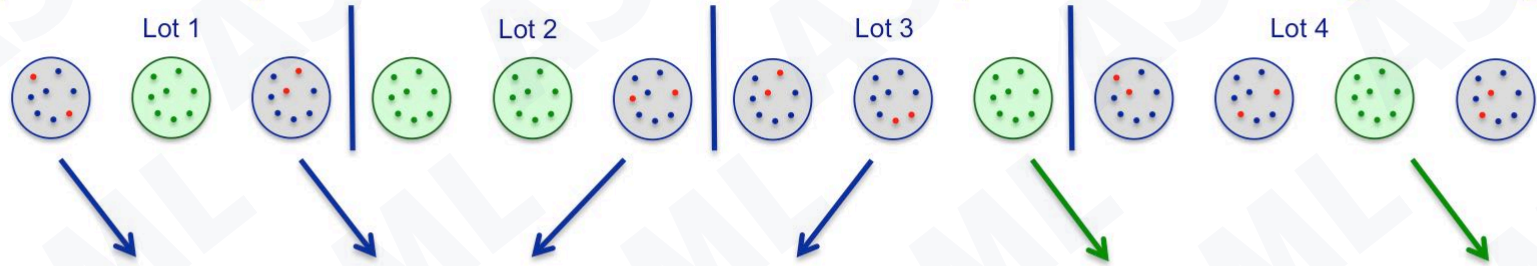


2b) Wafers for testing

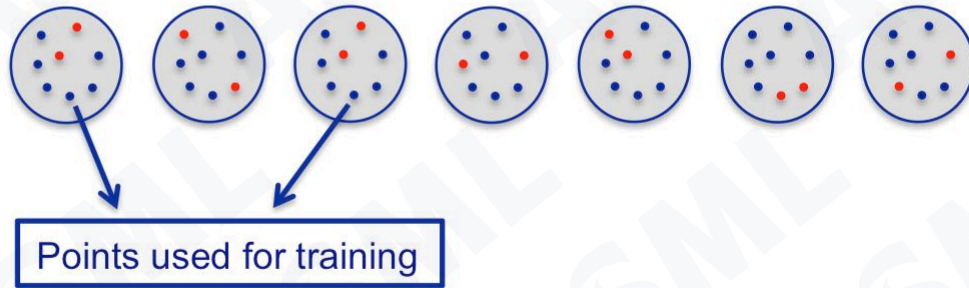


Data separation into Training and Testing groups

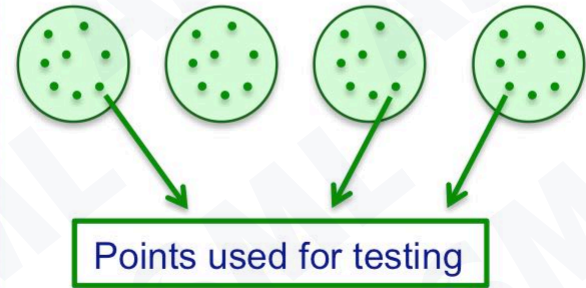
1) We randomly select measured wafers from the exposed lots for training & testing



2a) Wafers for training

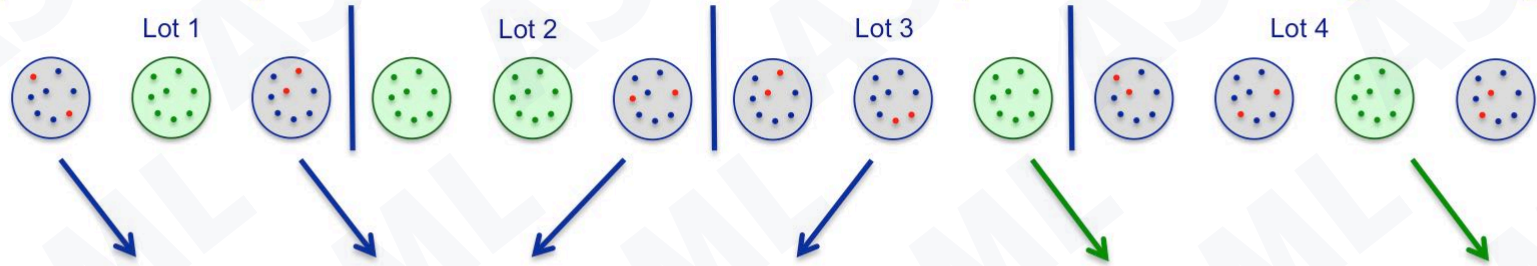


2b) Wafers for testing

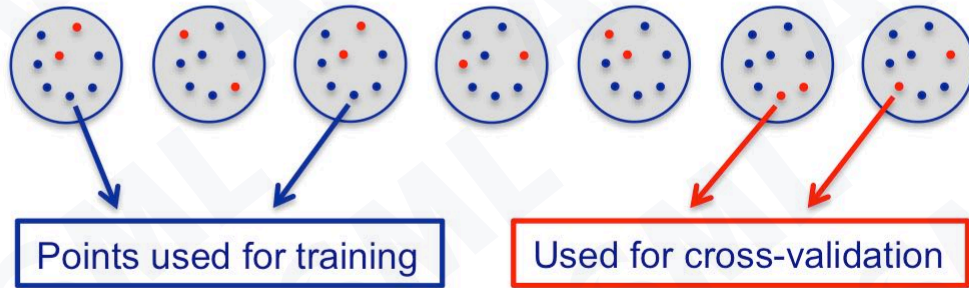


Data separation into Training and Testing groups

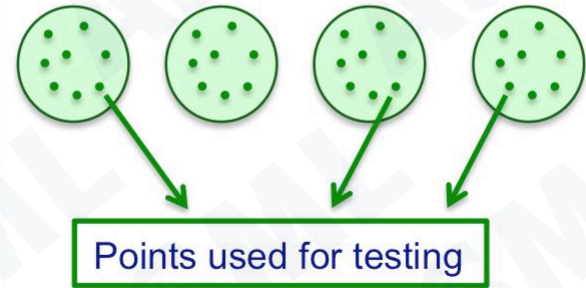
1) We randomly select measured wafers from the exposed lots for training & testing



2a) Wafers for training



2b) Wafers for testing

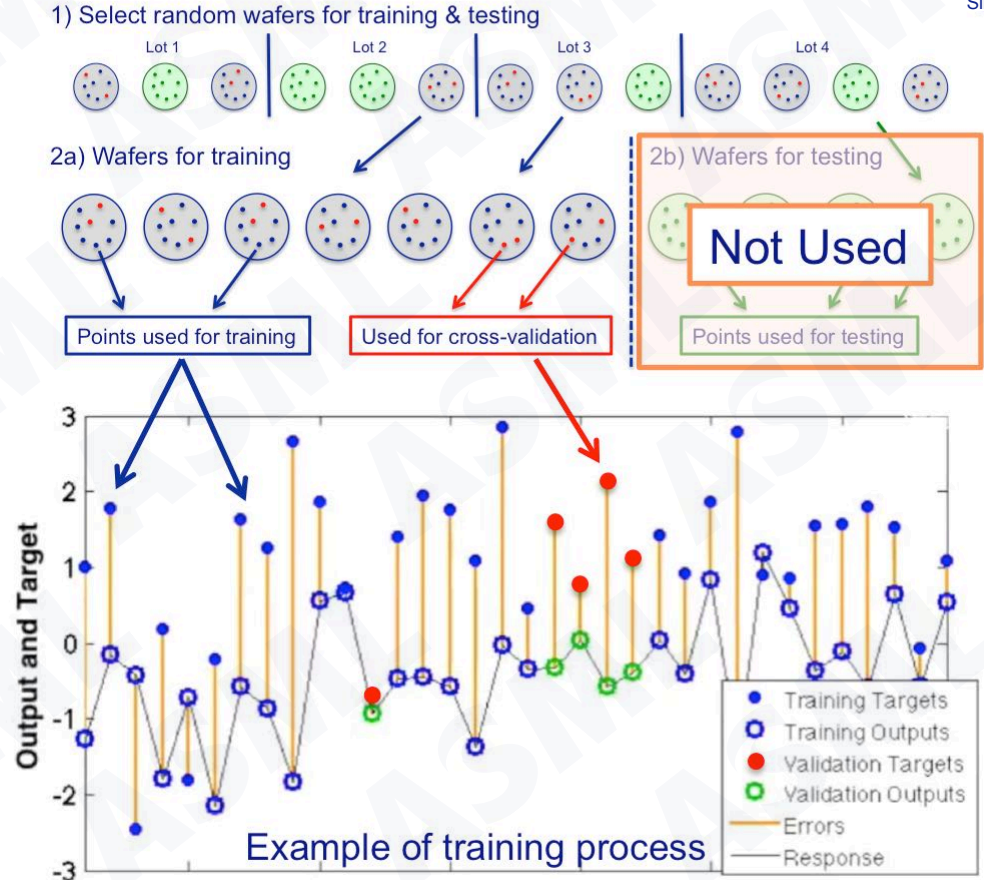


Outline

- Introduction
- How the function works
- Data separation into Training and Testing groups
- Training with Bayesian Automated Regularization
- Prediction Vs. Measured Overlay as regression plots
- Precision of Trained Function as a vector map
- Results
- Conclusion

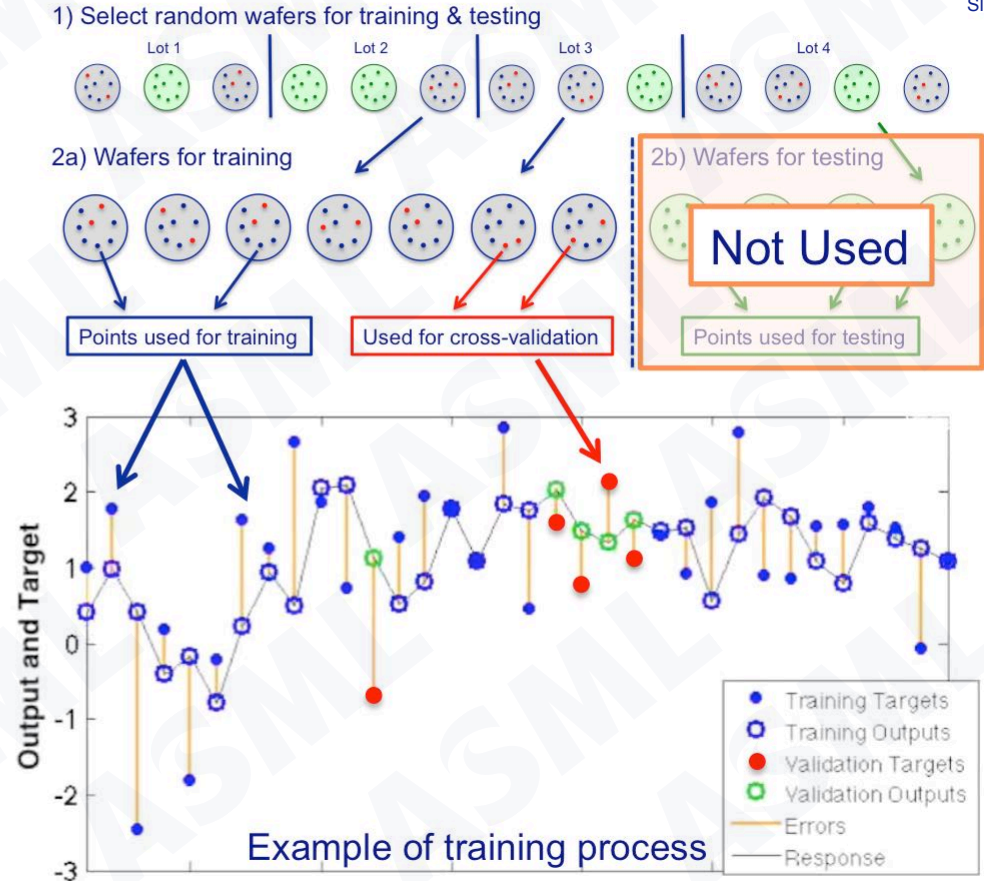
Training with Bayesian Automated Regularization

- To reduce the likelihood of overfitting the training dataset, a Bayesian framework with automated regularization is employed



Training with Bayesian Automated Regularization

- To reduce the likelihood of overfitting the training dataset, a Bayesian framework with automated regularization is employed
- The training process repeats in cycles until convergence, which is when the sum-squared error, the sum squared weights, and the effective number of parameters reach a constant value or till the cycle limit is reached
- If the cycle limit is reached before convergence a new random sample of points is selected for training and cross-validation points within the training dataset, while starting iterations toward convergence where the previous cycle stopped

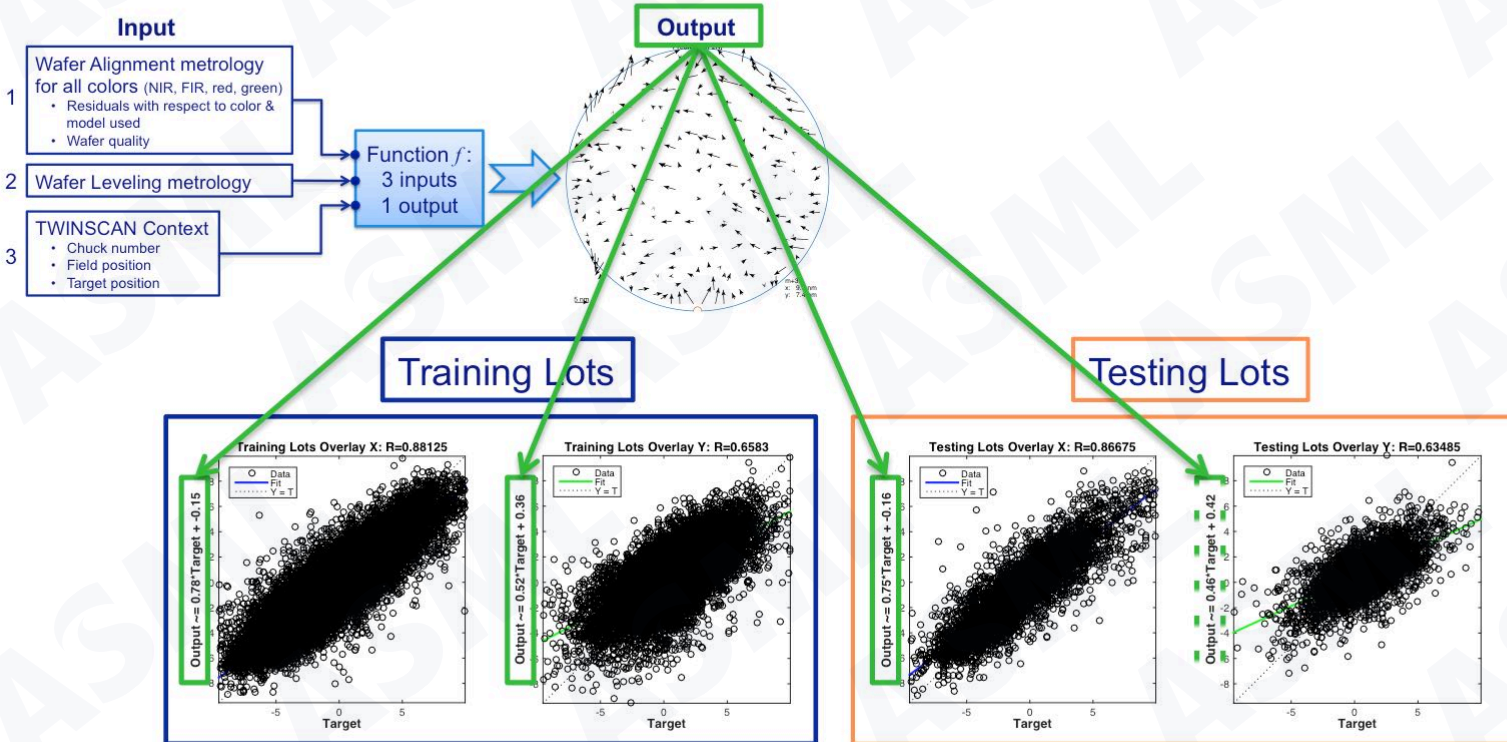


Outline

- Introduction
- How the function works
- Data separation into Training and Testing groups
- Training with Bayesian Automated Regularization
- Prediction Vs. Measured Overlay as regression plots
- Precision of Trained Function as a vector map
- Results
- Conclusion

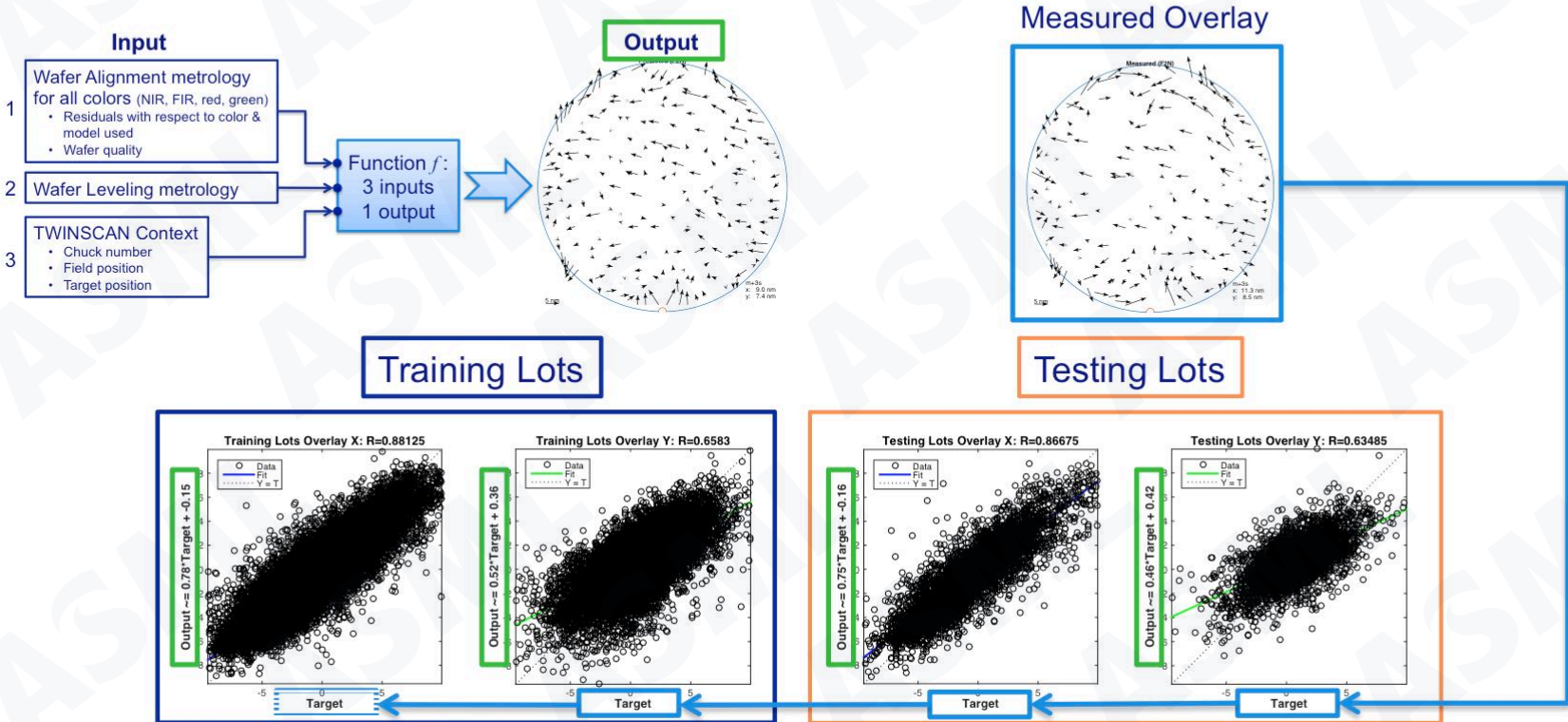
Prediction Vs. Measured Overlay as regression plots

- Point-to-point correlation between the functions predicted Output vs. Measured



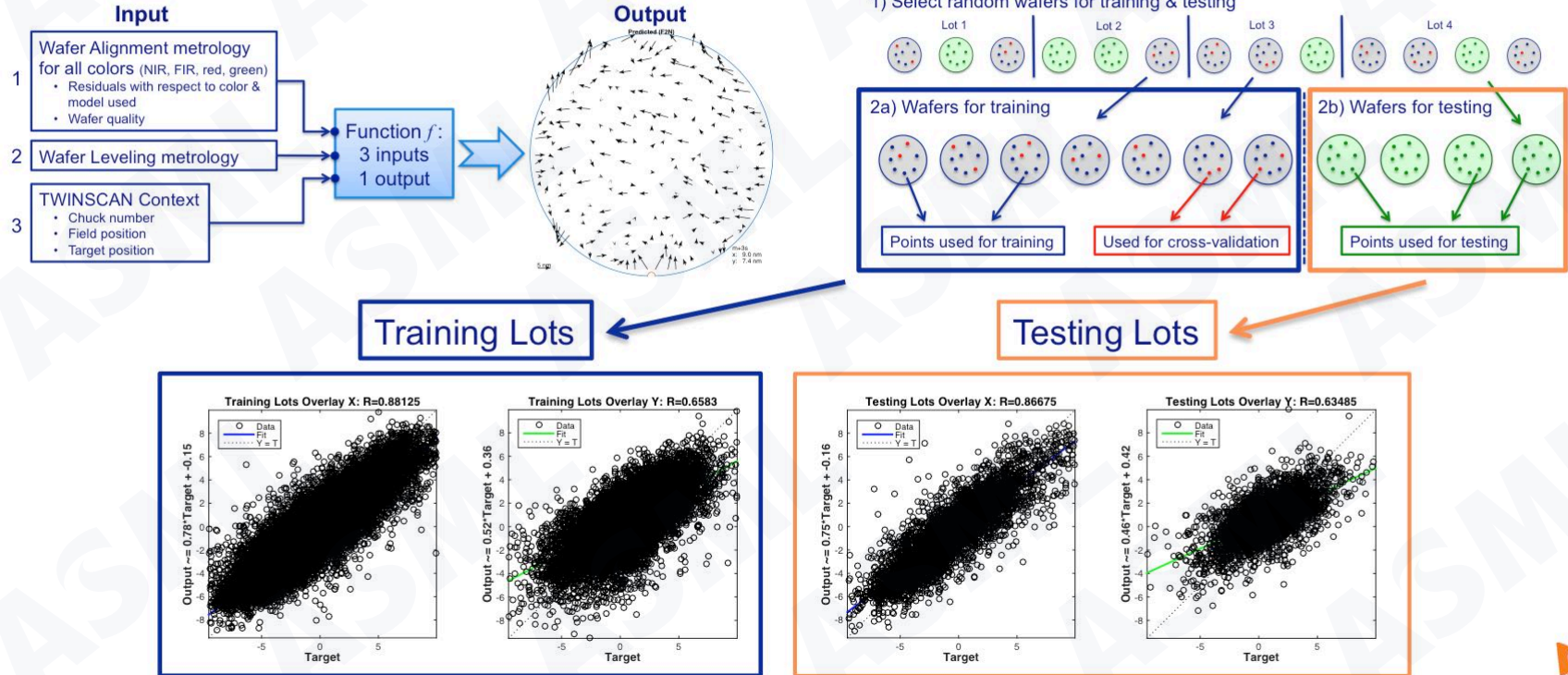
Prediction Vs. Measured Overlay as regression plots

- Point-to-point correlation between the functions predicted Output vs. Measured



Prediction Vs. Measured Overlay as regression plots

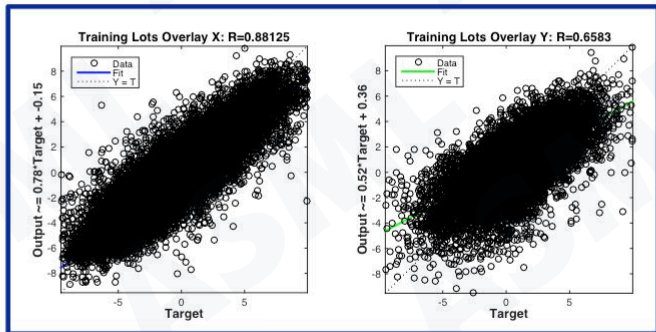
- Point-to-point correlation between the functions predicted Output vs. Measured
- Wafers from both the Training and Testing Groups are used



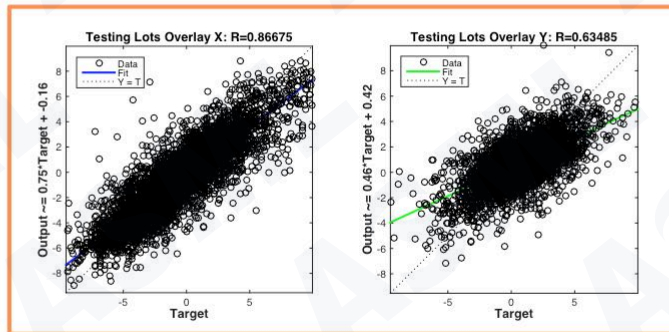
Prediction Vs. Measured Overlay as regression plots

- Because the **Testing group** of wafers are “blind” to the training process we can use R-values to judge the performance of the trained function.
- R-values of the testing wafers \approx to the R-values of the training wafers.
 - Overlay X (0.88) and Y (0.66) of the training wafers
 - Overlay X (0.87) and Y (0.63) of the **testing** wafers
- From this we conclude that the network generalized well with the automated regularization algorithm.

Training Lots



Testing Lots

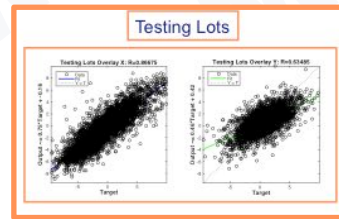
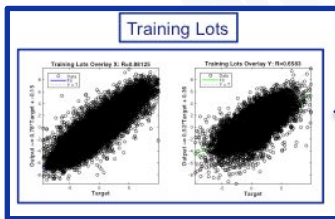
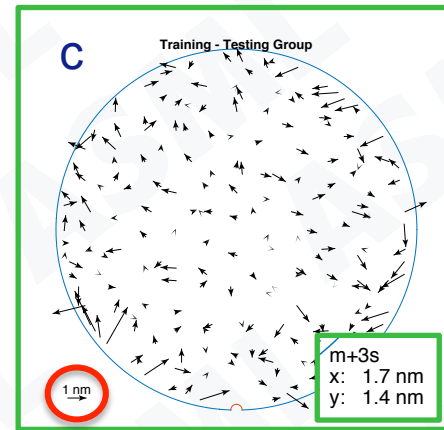
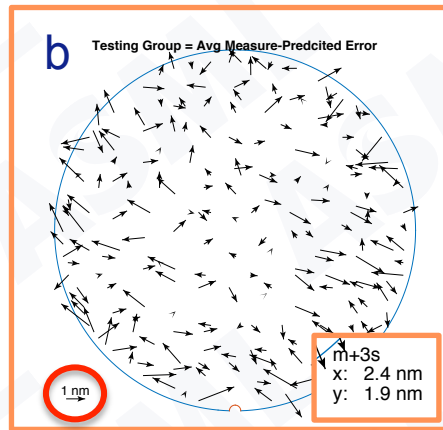
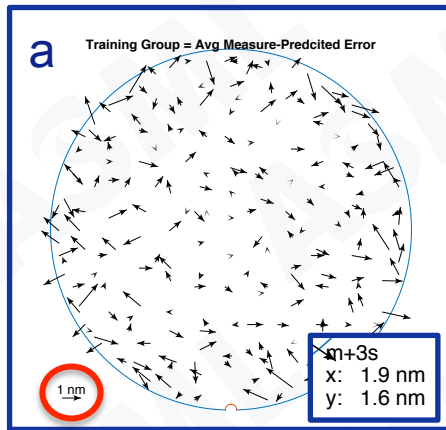


Outline

- Introduction
- How the function works
- Data separation into Training and Testing groups
- Training with Bayesian Automated Regularization
- Prediction Vs. Measured Overlay as regression plots
- Precision of Trained Function as a vector map
- Results
- Conclusion

Precision of Trained Function as a vector map

- Noise between the measured and predicted overlay is relatively consistent for both Training and Testing groups
- Consider the error as a plus or minus contribution per wafer coordinate position of any prediction from the trained function



Point - Point delta
between a) and b)

Outline

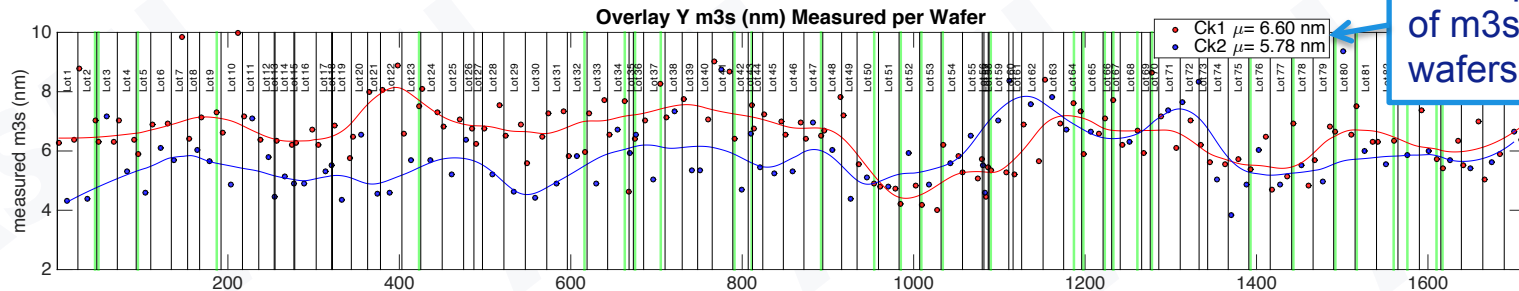
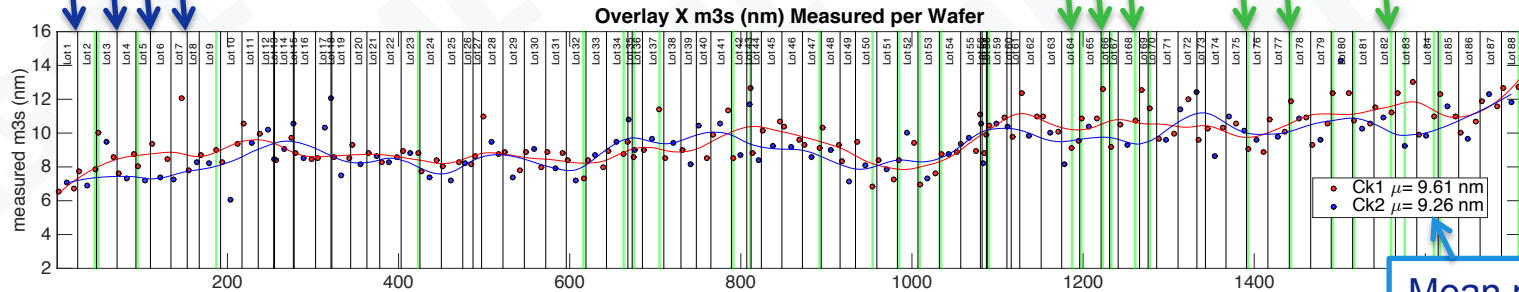
- Introduction
- How the function works
- Data separation into Training and Testing groups
- Training with Bayesian Automated Regularization
- Prediction Vs. Measured Overlay as regression plots
- Precision of Trained Function as a vector map
- Results
- Conclusion

Results – Measured Data

- SK hynix provided the on product overlay data for our proof book analysis.
- Process for the 20nm DRAM layer was intentionally manipulated as it was prepared for high volume production by the integration team

Black lines denote start and end to lot

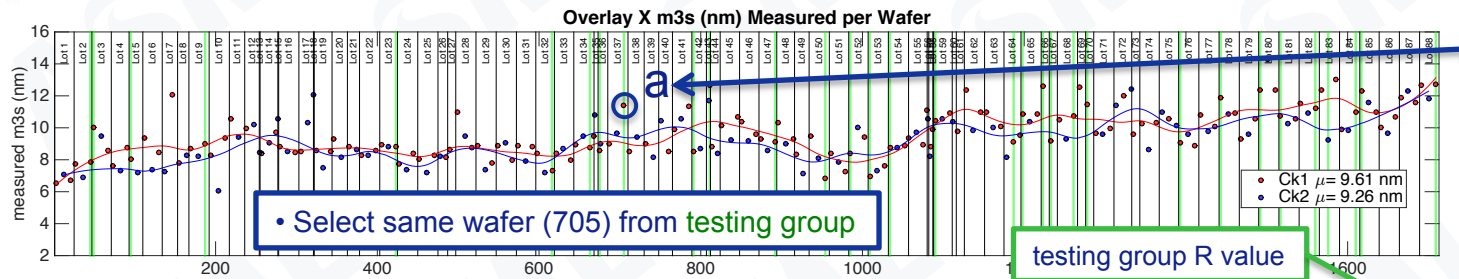
Green lines denote wafers testing group



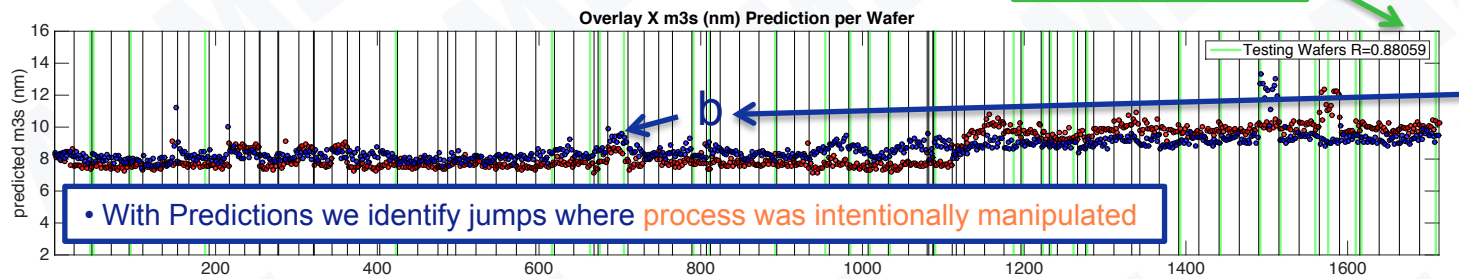
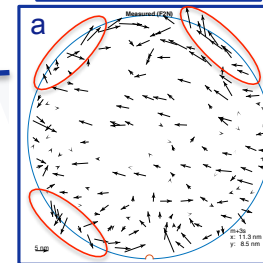
Mean per chuck of m3s from all wafers measured

Results – Overlay X

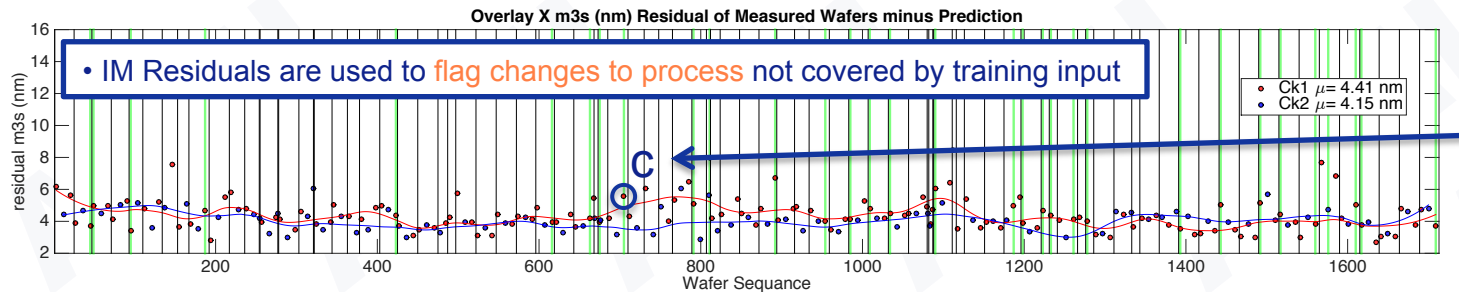
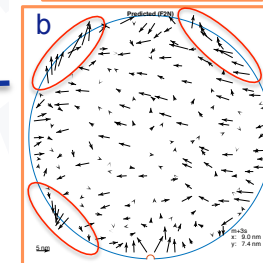
- Measured, Predicted & Residual Integrated Metrology (IM)



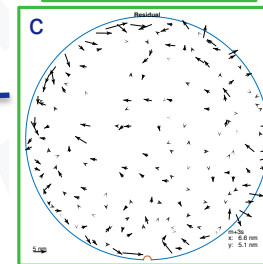
Measured Testing w705



Predicted Testing w705

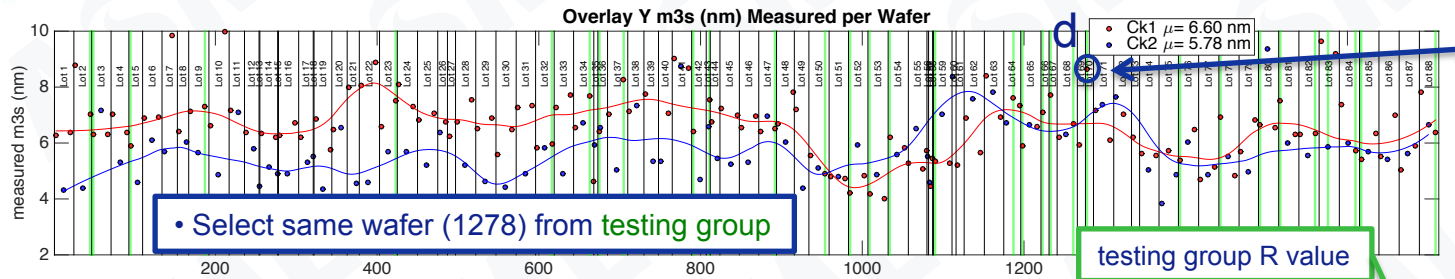


Residual Testing w705

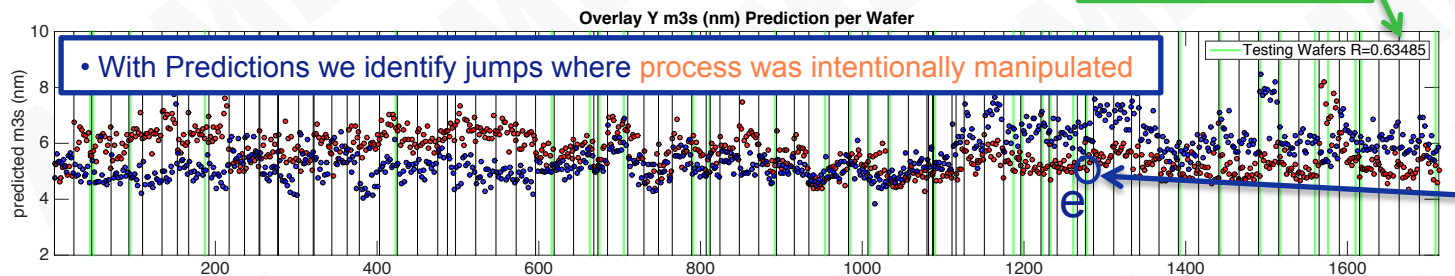
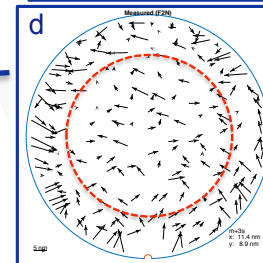


Results – Overlay Y

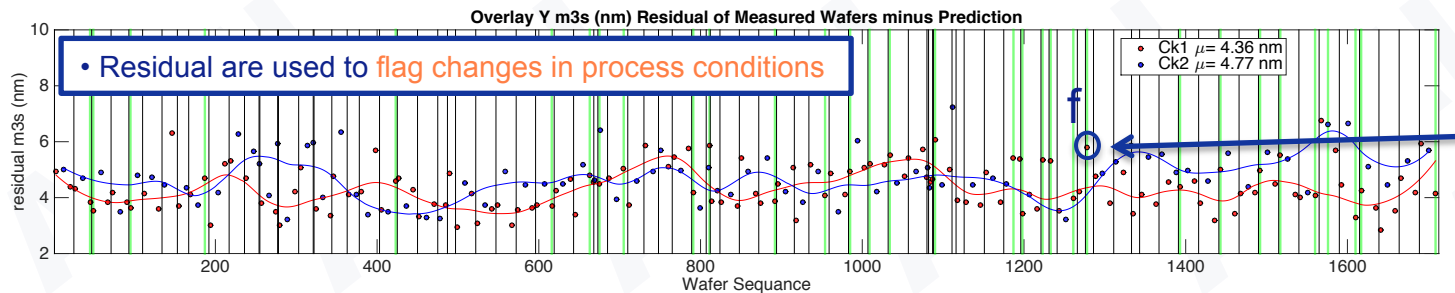
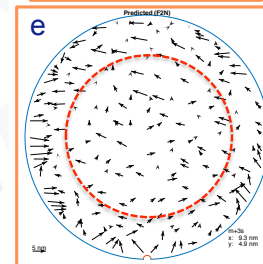
- Measured, Predicted & Residual Integrated Metrology (IM)



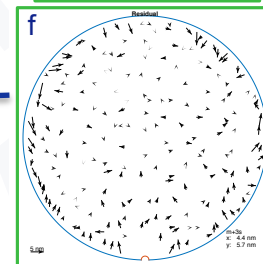
Measured Testing w1278



Predicted Testing w1278



Residual Testing w1278

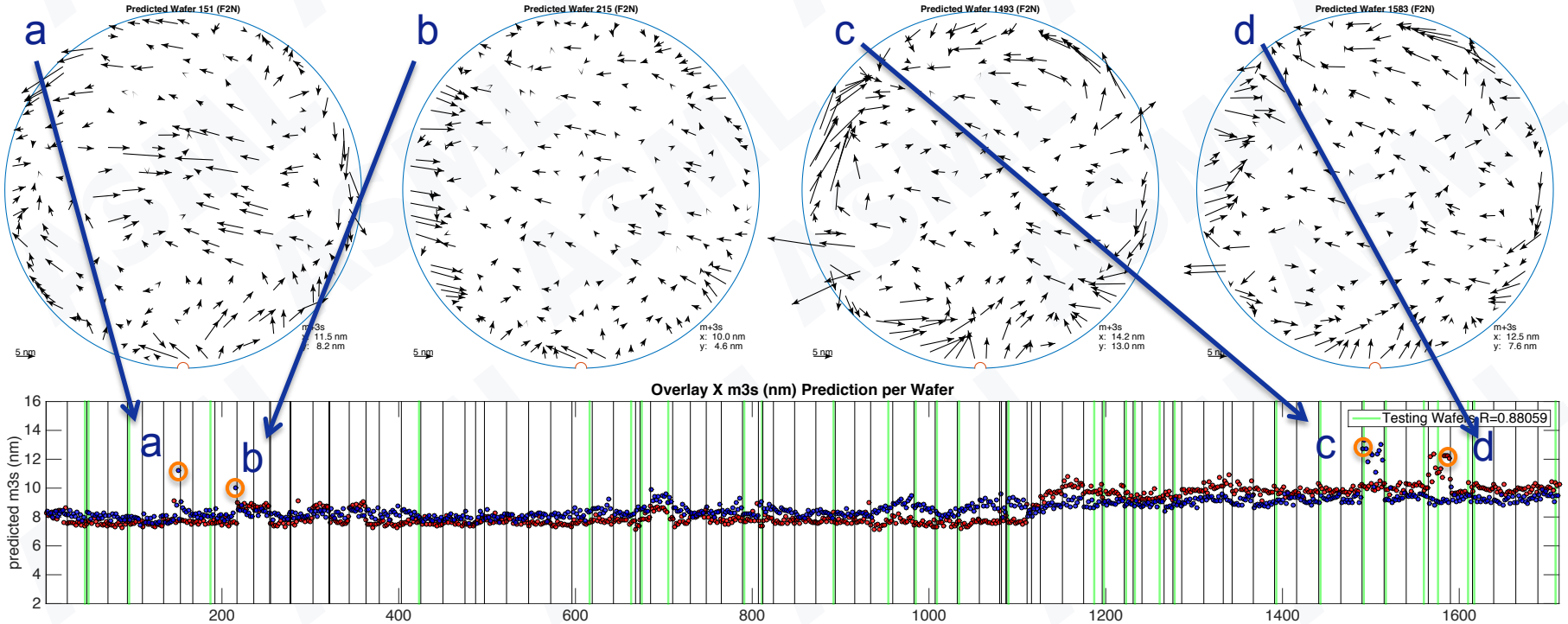


Outline

- Introduction
- How the function works
- Data separation into Training and Testing groups
- Training with Bayesian Automated Regularization
- Prediction Vs. Measured Overlay as regression plots
- Precision of Trained Function as a vector map
- Results
- Conclusion

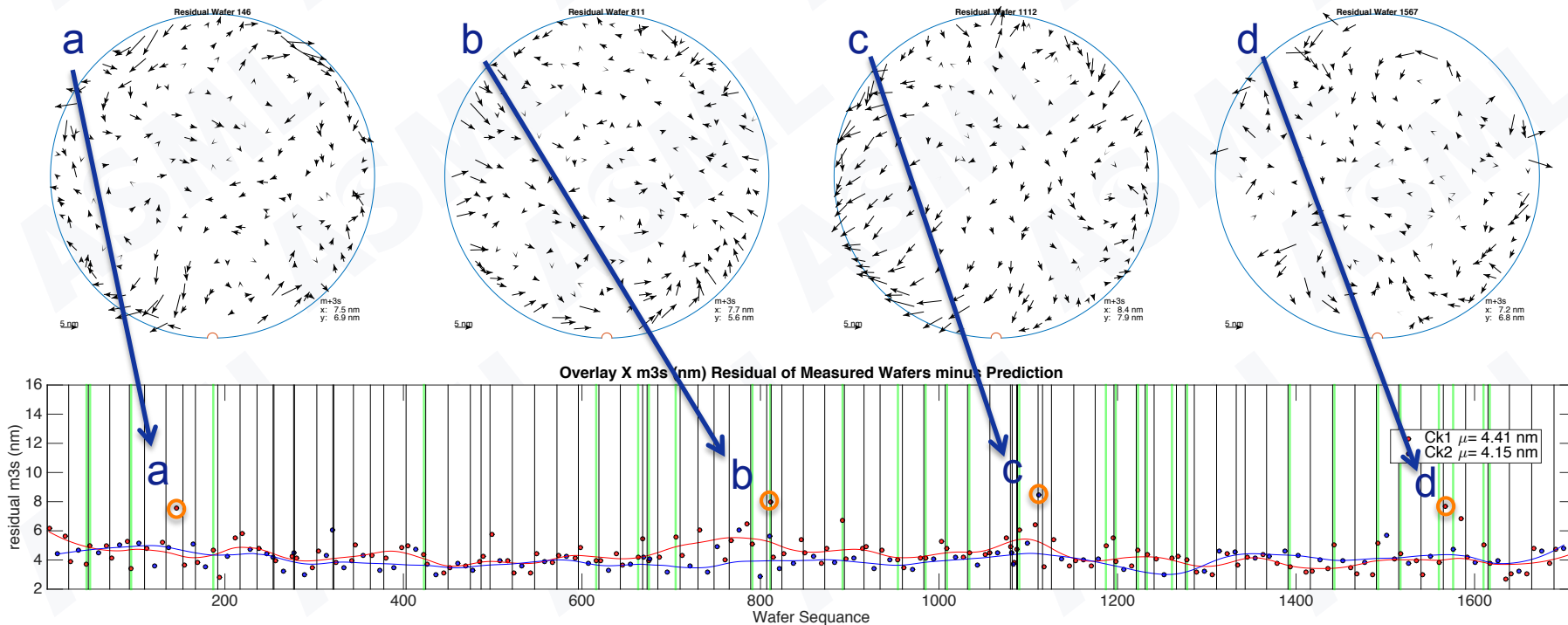
Conclusion 1

- With the Predictions we identify jumps in the overlay data where process was intentionally manipulated by the integration team



Conclusion 2

- With **Residuals** we flag wafers from IM.
 - Something other than inputs we trained with is effecting the overlay signature
 - This can be used to remove a wafer from APC or to trigger an investigation



Moving forward

- Work on this subject is open to users with interest in exploring the application, both in production and development environments
- Topics of interest include exploring effect;
 - Fab context from outside the lithocluster has on the overlay prediction
 - Increasing the number of parallel works and neurons has toward improving the R value (correlation coefficient) between the predicted output and target values in the testing dataset

Coauthors SK hynix: Hong-Goo Lee, Min-Suk Kim, Sang-Jun Han, Myoung-Soo Kim, Won-Taik Kwon & Sung-Ki Park

Coauthors ASML: Kevin Ryan, Thomas Theeuwes, Kyu-Tae Sun, Young-Wan Lim, Daan Slotboom, Michael Kubis & Jens Staecker

ASML

A decorative graphic consisting of several thin, curved lines that sweep from the bottom left towards the top right, creating a sense of motion and flow.

Acknowledgements: The authors would like to thank ASML colleagues Jan Mulkens, Marcel Beems, Wolfgang Henke, Henry Megens, Peter ten Berge, Paul Luehrmann, Dick Verkleij, Frank van de Mast, Christophe Fouquet & coworkers in the Holistic Applications Development group for their assistance & feedback