eXplainable Artificial Intelligence Application in Metrology

Eleni Lavasa, Theodore Dalamagas, Giorgos Giannopoulos, Vasilis Gkolemis, Anargyros Tzerefos

ATHENA Research Center
December 2022

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957362

Optical Industrial Metrology - Background

Industrial Quality Control

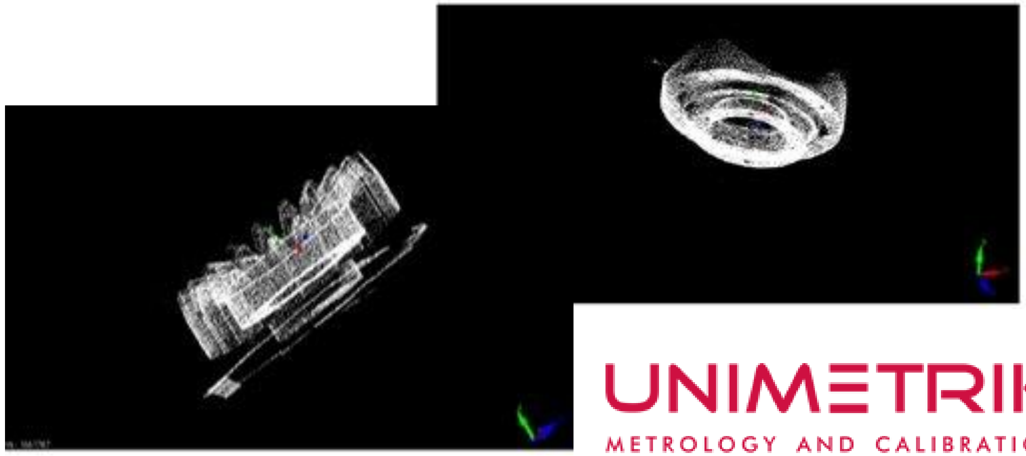
- Manufactured parts for automotive, aeronautic, energy sectors
- Dimensional/positional measurements: Definition of dimension/position & tolerance

Instrument: 3D laser scanner

- User-defined parametrization of scanning conditions
- Point Cloud generation

Software: M3 metrological software

- Point cloud processing
- Measurement using two methods (complete Point Cloud / geometry definition points)

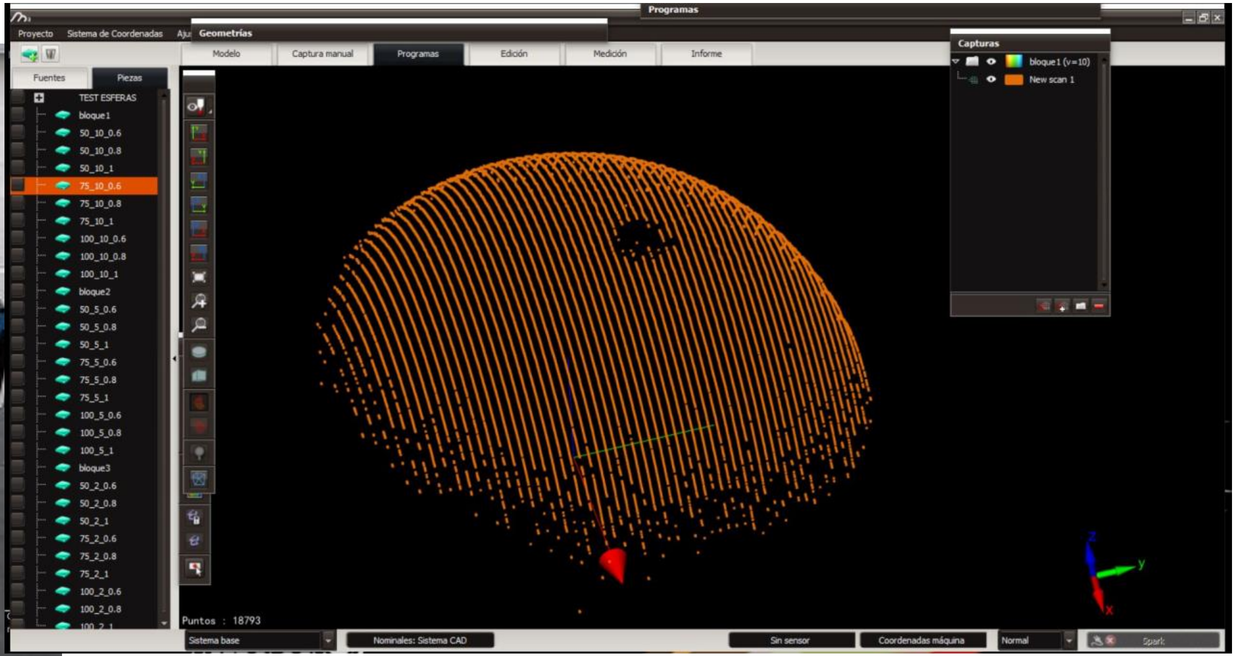


UNIMETRIK
METROLOGY AND CALIBRATION

Measurement Workflow

Step 1: Preliminary analysis of the object under study & selection of laser setup

- ➢ scanning parameters & laser orientation



Step 2: Configuration of scanning conditions & generation of Point Cloud

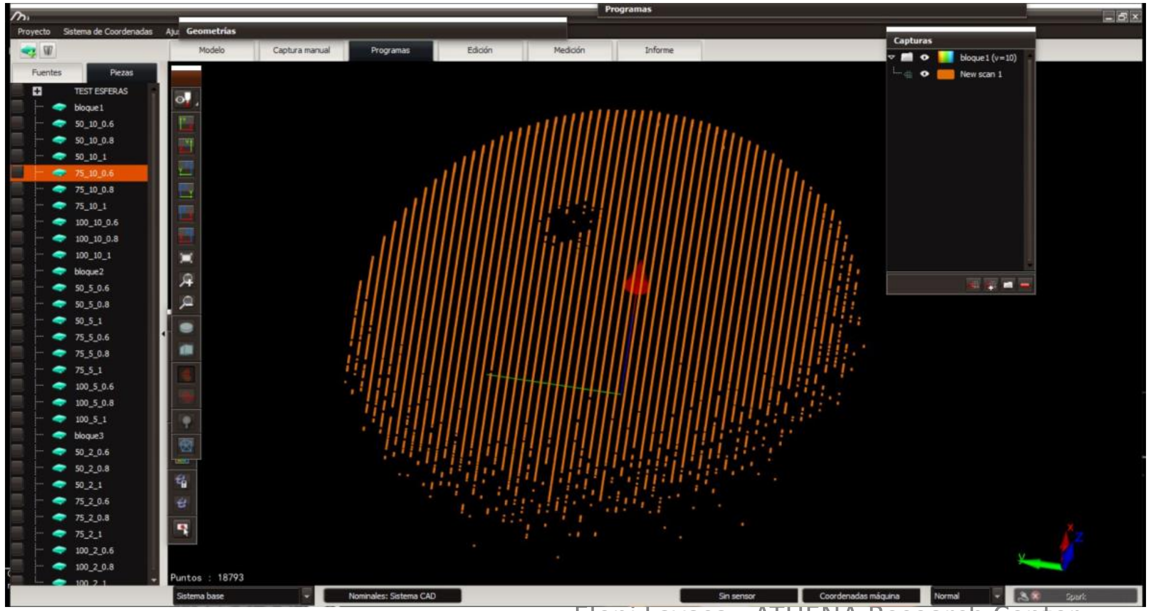
Measurement Workflow

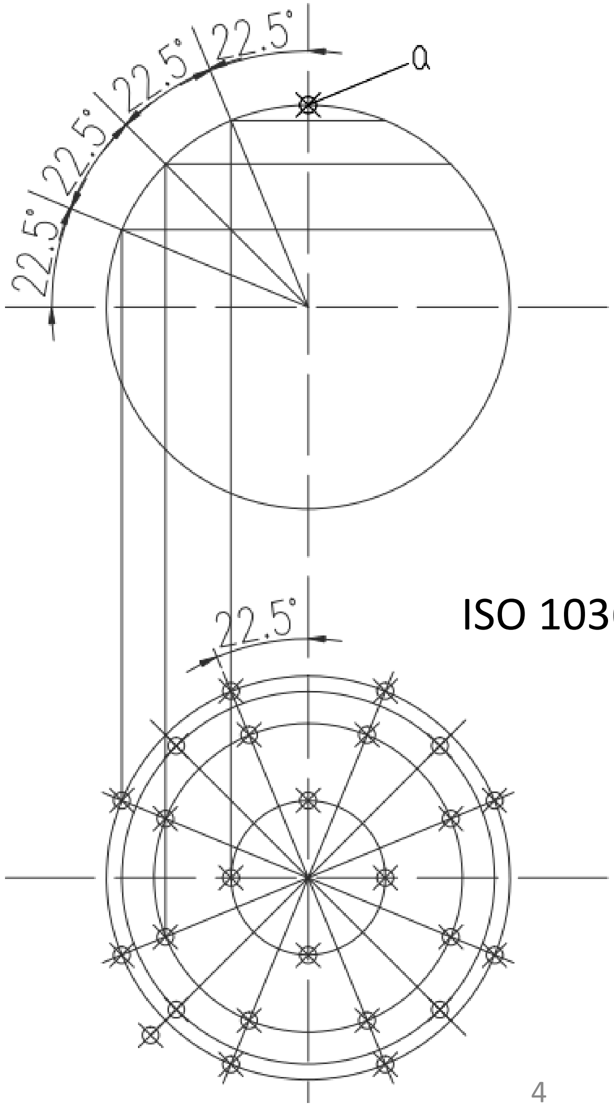
Step 3: Point cloud preprocessing

- Outlier removal
- Noise filtering
- Surface Segmentation

Step 4: Dimensional/positional measurement

- Method#1 (CONSTRUCT) (Geometry definition points)
- Method#2 (EXTRACT) complete Point Cloud





ISO 10360

Use case description

Issues:
- Results depend on metrologist's experience (inconsistency of results)
- Wasted time on iterative analysis, each project from scratch

Use Case 1: Measurement plan parameter optimization

Objective: Optimization of scanning parameters, to achieve minimum error/uncertainty in dimensional measurement

Accelerate/Facilitate ***Step 1: Preliminary analysis***

Use Case 2: Point Cloud optimization

Objective: Optimization of the 3D point cloud in terms of quality & size

- **Size**: minimum # of points sufficient to perform accurate measurement, & preserve shape/geometry information
- **Quality**: Point cloud denoising / filtering / segmentation

Accelerate/Facilitate ***Steps 3&4: Point Cloud pre-processing & Dimensional measurement***

Initial Approach: Supervised Learning Setting

Objective: Model instrument's anisotropic accuracy

- in response to *different surface orientations*
- in relation to *scanning parameters*

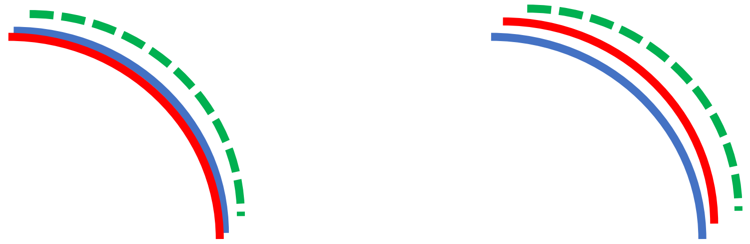
Approach: Supervised Regression task

- Prediction of error in Point capturing
- Access to *Ground Truth* needed

Method: Use measurements of *Calibrated artifacts*

- Actual dimensions precisely known (~1μm), i.e. actual ≈ nominal
- Validated shape perfection
- Ground Truth: actual Point positions located on artifact's nominal surface

calibrated *non-calibrated*



Nominal => dimension in CAD design
Actual => true dimension
Measured => dimension estimated by measurement

target variable

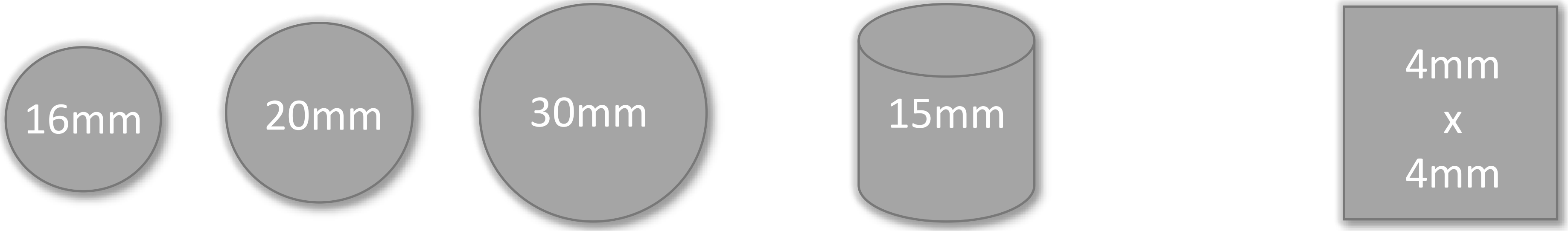
Point Measurement deviation (error):

PointDev = measured - nominal

Data Sources Description

Geometries: Sphere | Cylinder | Plane

Ground truth: Diameter | Diameter | Distance to laser



Scanning parameters

1. *Lateral Density* — low (50), medium (75), high (100)
2. *Scan Direction Density* — low (2), medium (5), high (10)
3. *Exposure time* — low (0.6), medium (0.8), high (1.0)

Laser setup

Source orientation $\vec{L}$

$\vec{L}$ *= constant (Sphere & Plane)*

$\vec{L}1, \vec{L}2$ *= constant & symmetrical wrt axis (Cylinder)*

Detector (CMOS) viewing direction $\vec{V}$

Vertical distance $Z_l \approx 90mm$

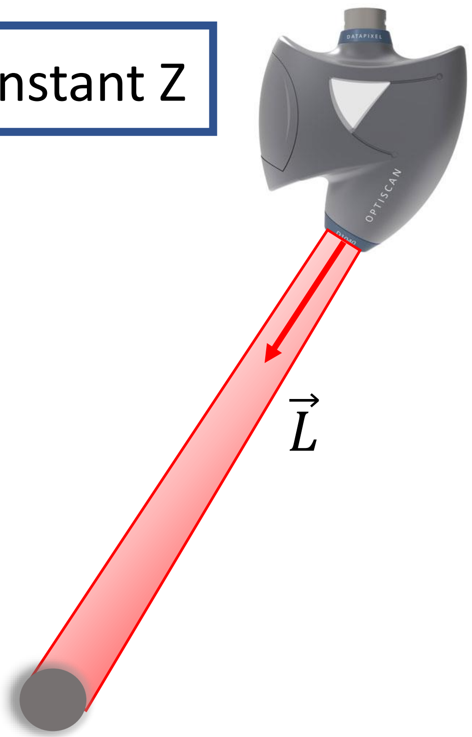
$\vec{V}$ *at 30° wrt* $\vec{L}$ *(rotate over axis of movement)*

Z_l *= constant for all measurements*

Data Sources Description

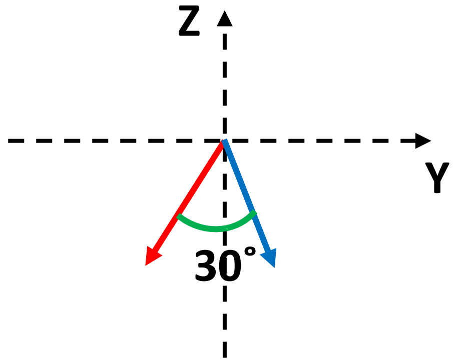
Rotary Head of Optical System

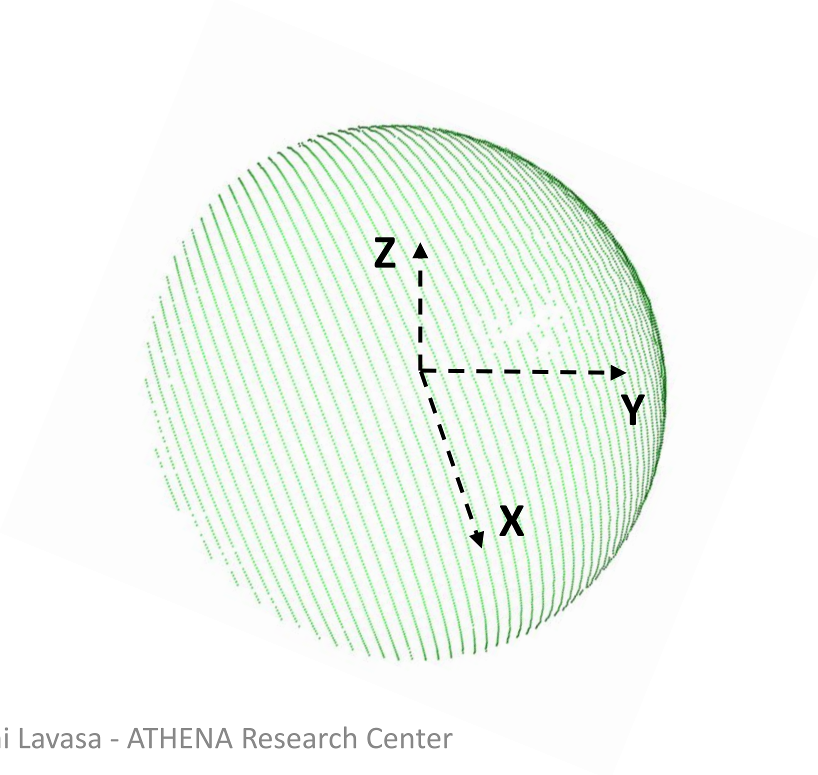
Head moves along Y-axis at constant Z



$\vec{L}$ = *Laser Source orientation vector*

$\vec{V}$ = *Detector (CMOS) viewing direction*



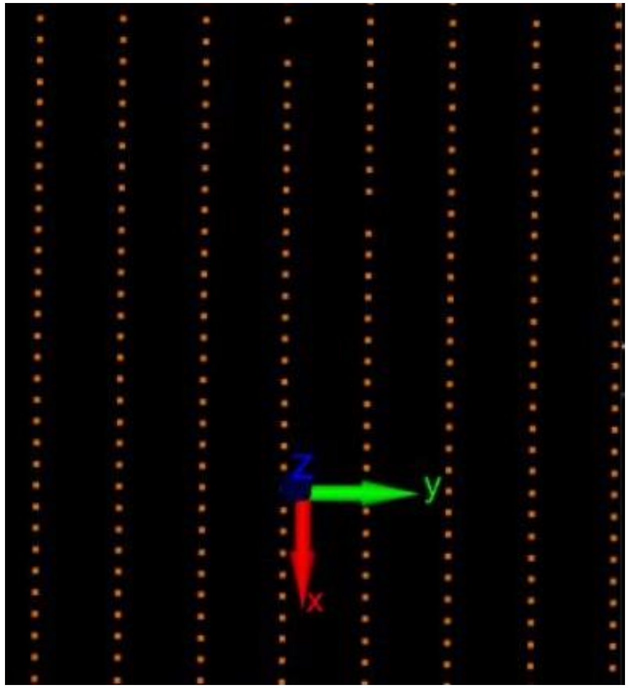


Data Sources Description

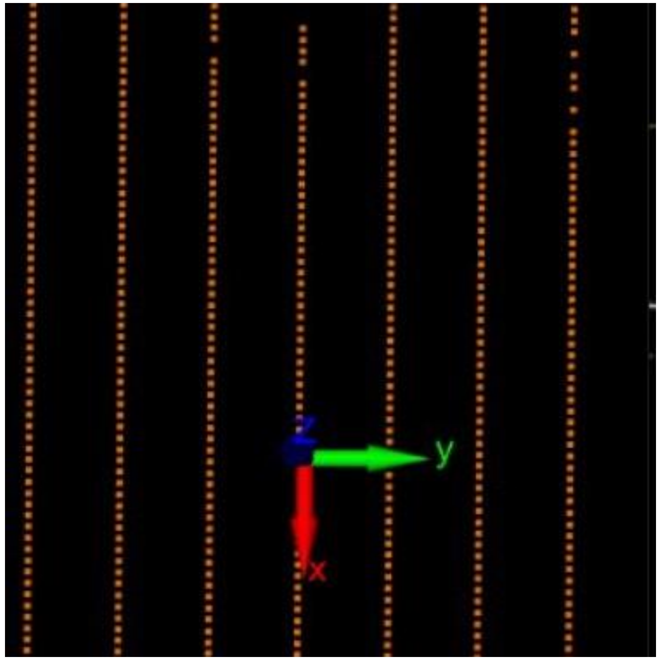
Scanning Parameters: ***Lateral Density***

Increasing point density along each line (lateral direction)

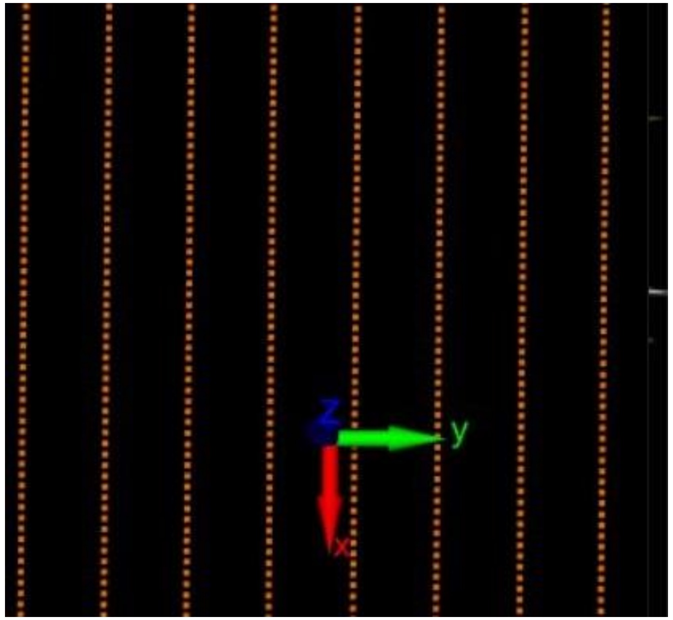
50



75



100



Data Sources Description

Scanning Parameters: *Scan Direction Density*

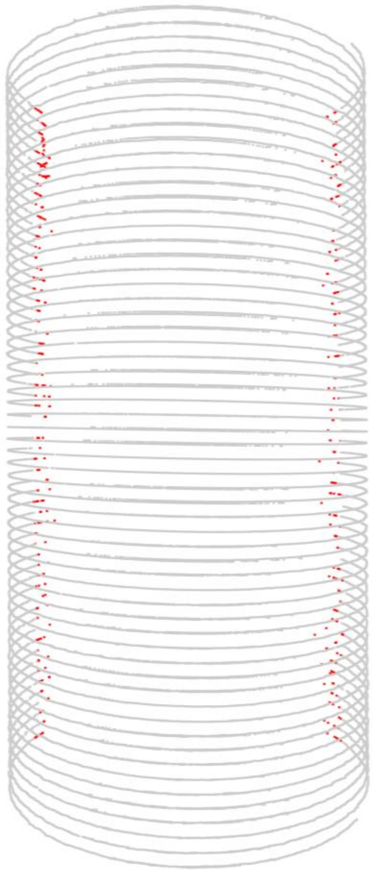
Decreasing point density in the direction of movement (sparse lines)

..increasing velocity of rotary head movement

10

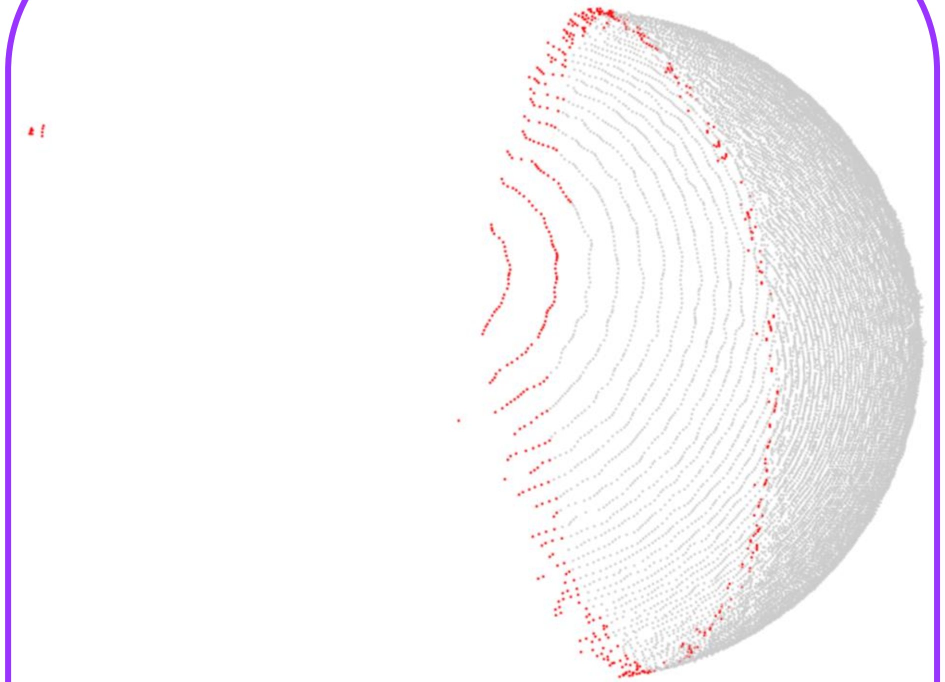
5

2



Statistical outlier removal

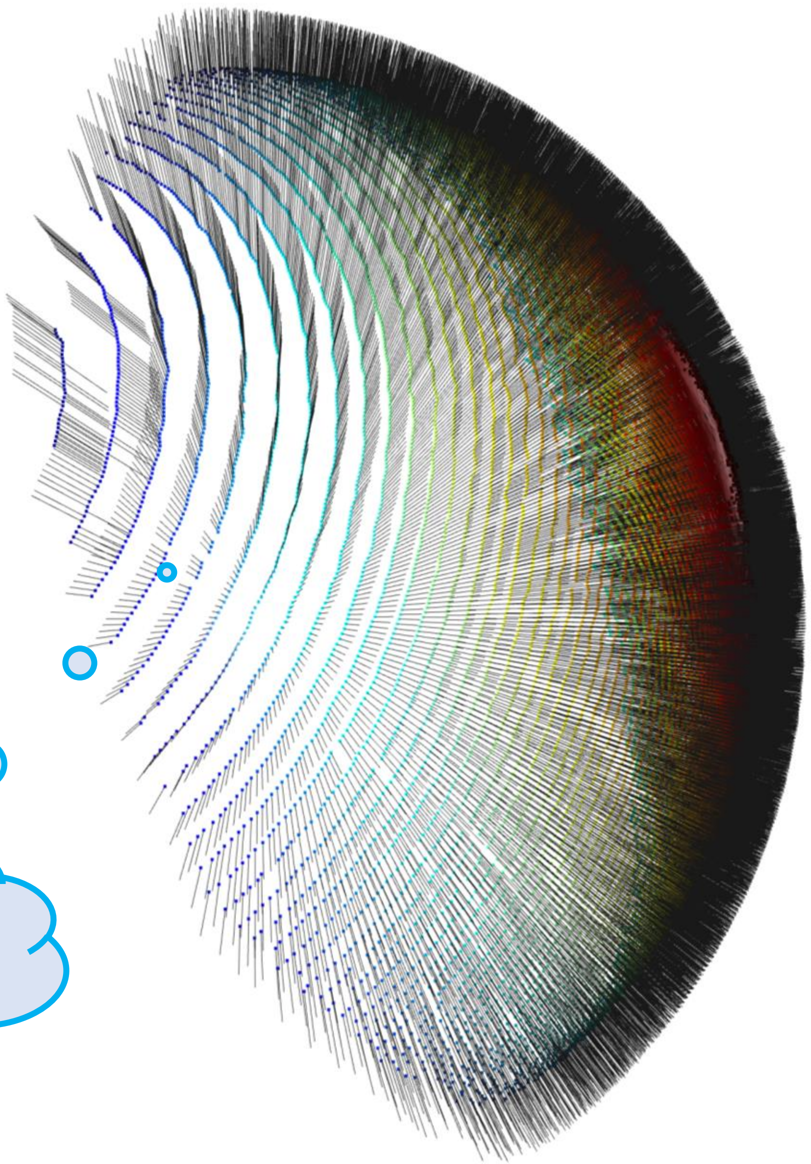
- Based on distance to nearest neighbors (number of NN, std)
- inliers: red
- outliers: grey



Estimation of normal vectors $\vec{N} = [Nx, Ny, Nz]$

- **surface orientation** at each point
- Based on nearest neighbors within a given radius (number of NN, radius)

Ensure alignment of normal vectors !



OPEN3D

Data Sources Description

Calculation of the target variable

at point P:

$P(X, Y, Z) =$ *measured point coordinates*

$P'(Xnom, Ynom, Znom) =$ *projected point coordinates*

(nominal surface)

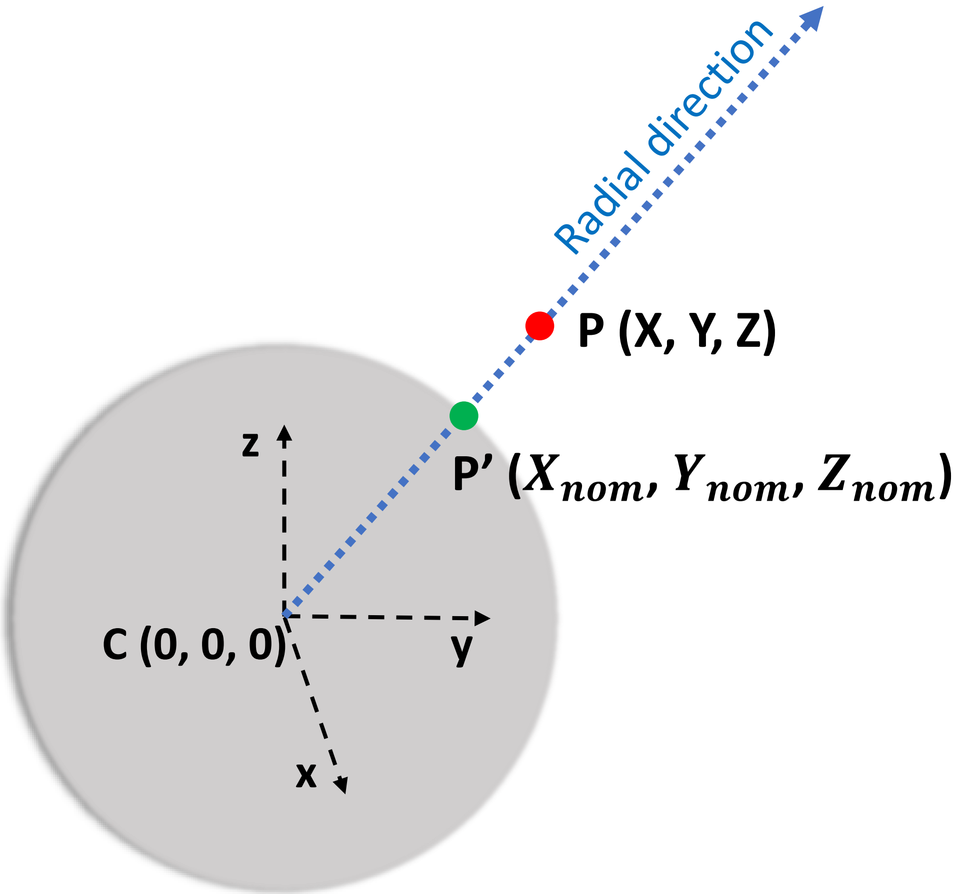
|CP| = R = **measured** point radial distance (spheres)

(cylinder: axial, plane: vertical)

|CP'| = Rnom = **nominal** radial distance = Nominal Radius

PointDev = |P'P| = R – Rnom

= **point deviation from nominal surface**



$\vec{L}$ = *Laser Source orientation vector*
$\vec{V}$ = *Detector (CMOS) viewing direction*

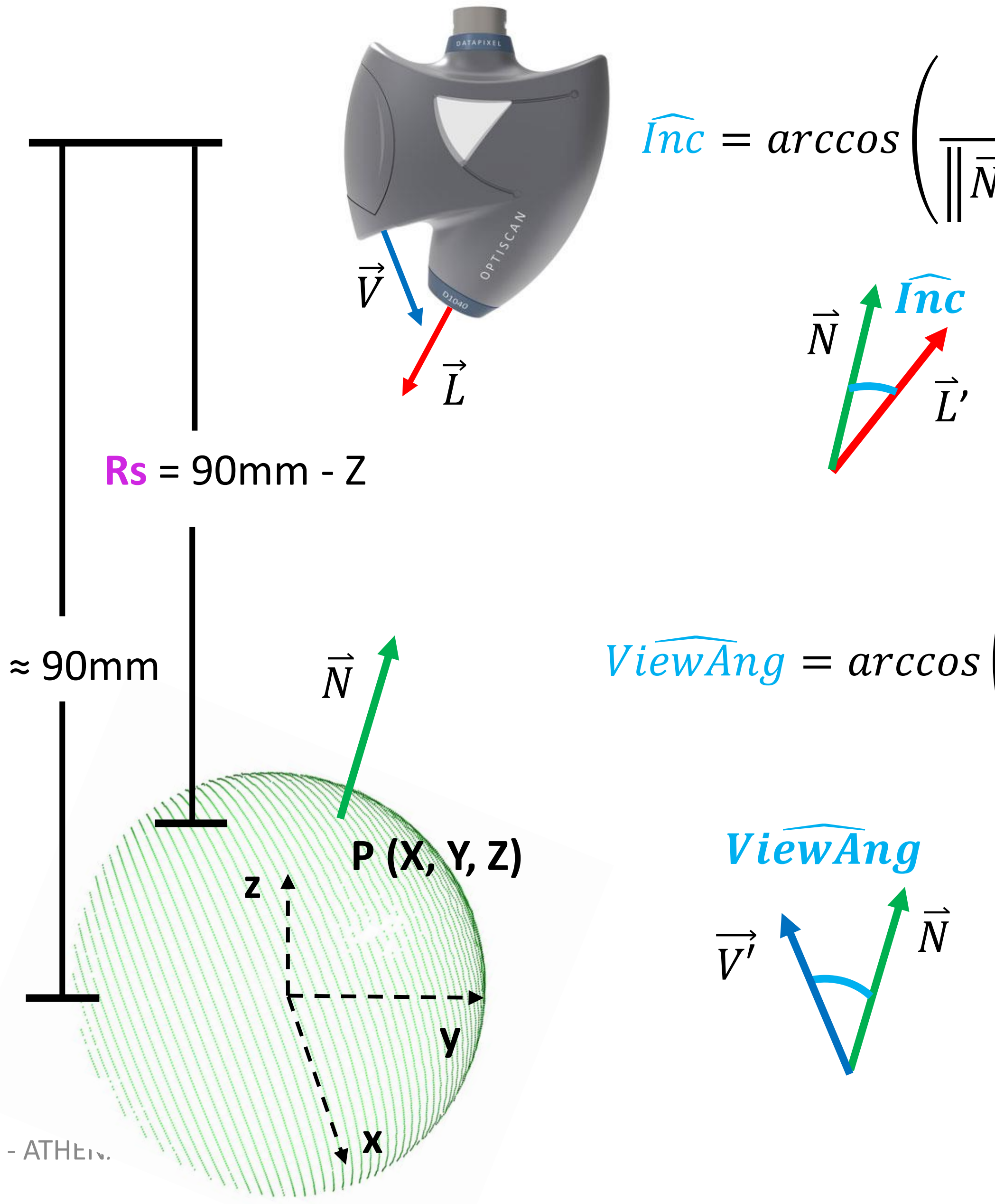
at point P:

$\vec{N} = [Nx, Ny, Nz] =$ *surface orientation*

$\widehat{Inc}$ = *incidence angle of light on surface*

$\widehat{ViewAng} = viewing\ angle\ from\ CMOS$

R_s = *Vertical distance to laser source*



$$\widehat{Inc} = arccos\left(\frac{\vec{N}.\vec{L}'}{\|\vec{N}\|.\|\vec{L}'\|}\right)$$

$$\widehat{ViewAng} = arccos\left(\frac{\vec{N}.\vec{V}}{\|\vec{N}\|.\|\vec{V}\|}\right)$$

Data Sources Description

$\vec{L}$ = *Laser Source orientation vector* $= [I, J, K]$

$\vec{V}$ = *Detector (CMOS) viewing direction* $= [I, J', K]$

at point P:

$\vec{N} = [Nx, Ny, Nz]$ $=$ *surface orientation*

$\overrightarrow{Ori} = \vec{N} - \vec{L}$ = *[OriX, OriY, OriZ]*

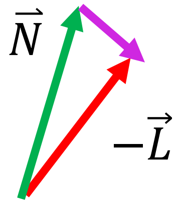
= *Orientation difference vector (surface & laser)*

$\overrightarrow{OriCMOS} = \vec{N} - \vec{V}$ = *Orientation difference vector*

(surface & CMOS sensor)

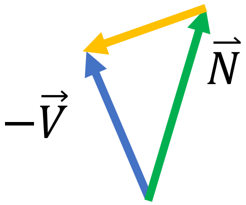
OriYcmos = *Ny – J'*

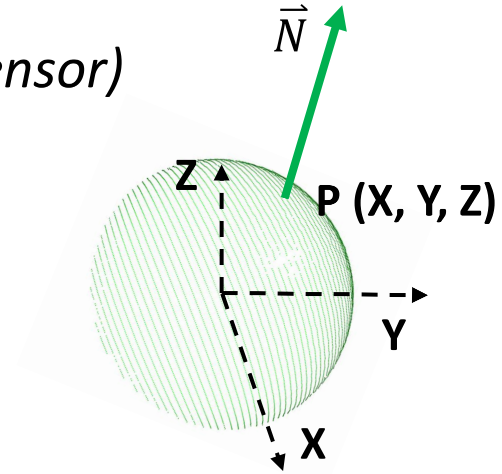
$\widehat{Ang}$ = *4-quadrant angle*



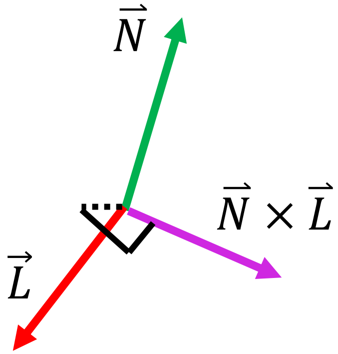
$\overrightarrow{Ori} = \vec{N} - \vec{L}$

$\overrightarrow{OriCMOS} = \vec{N} - \vec{V}$





$$\widehat{Ang} = arctan2d\left(\vec{N} \times \vec{L}, \vec{N}.\vec{L}\right)$$

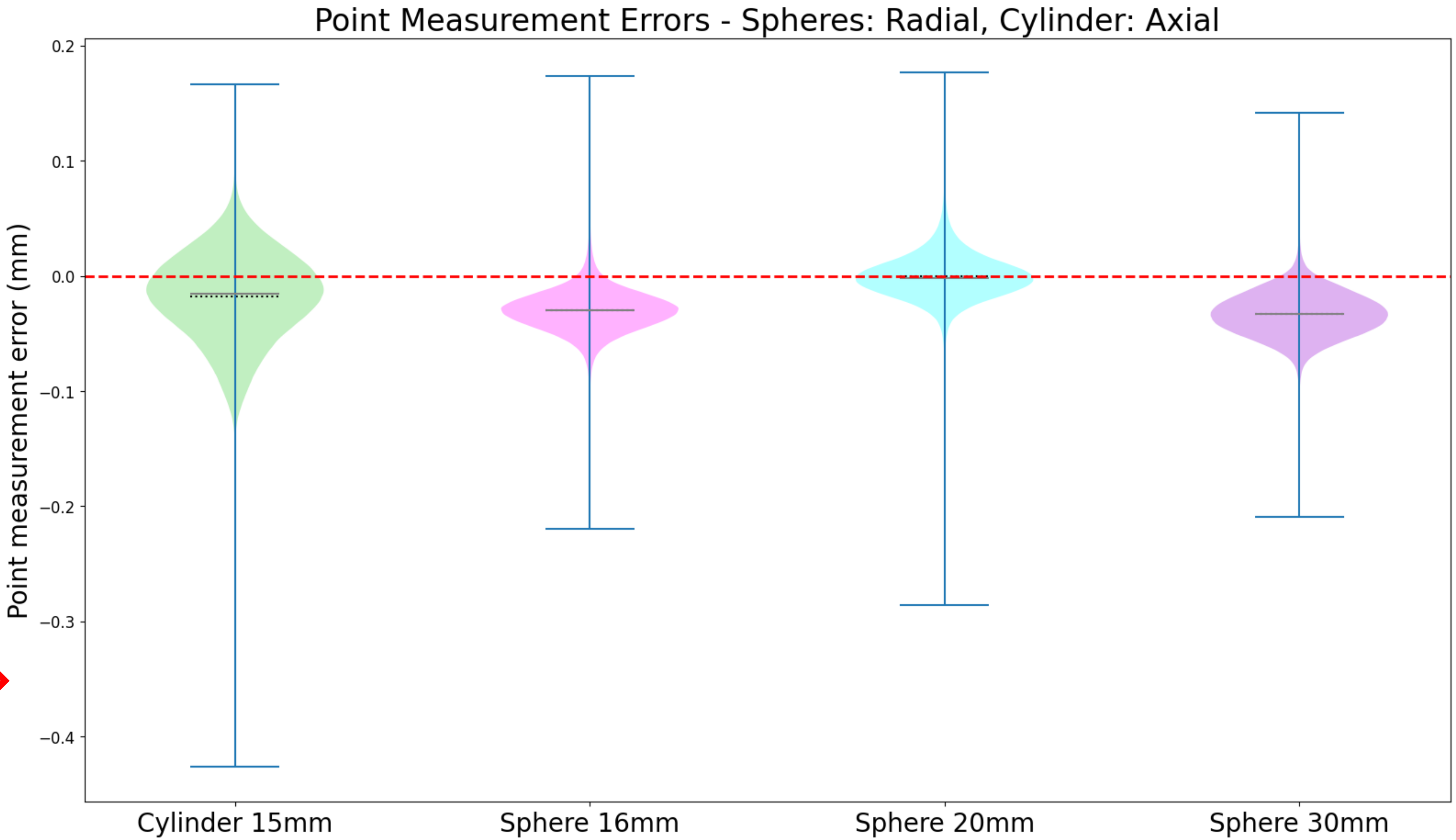


Data exploration & Analysis

Studying the behavior of the instrument

- Errors in point capture tend to be negative
- Instrument tends to under-estimate the dimension of objects

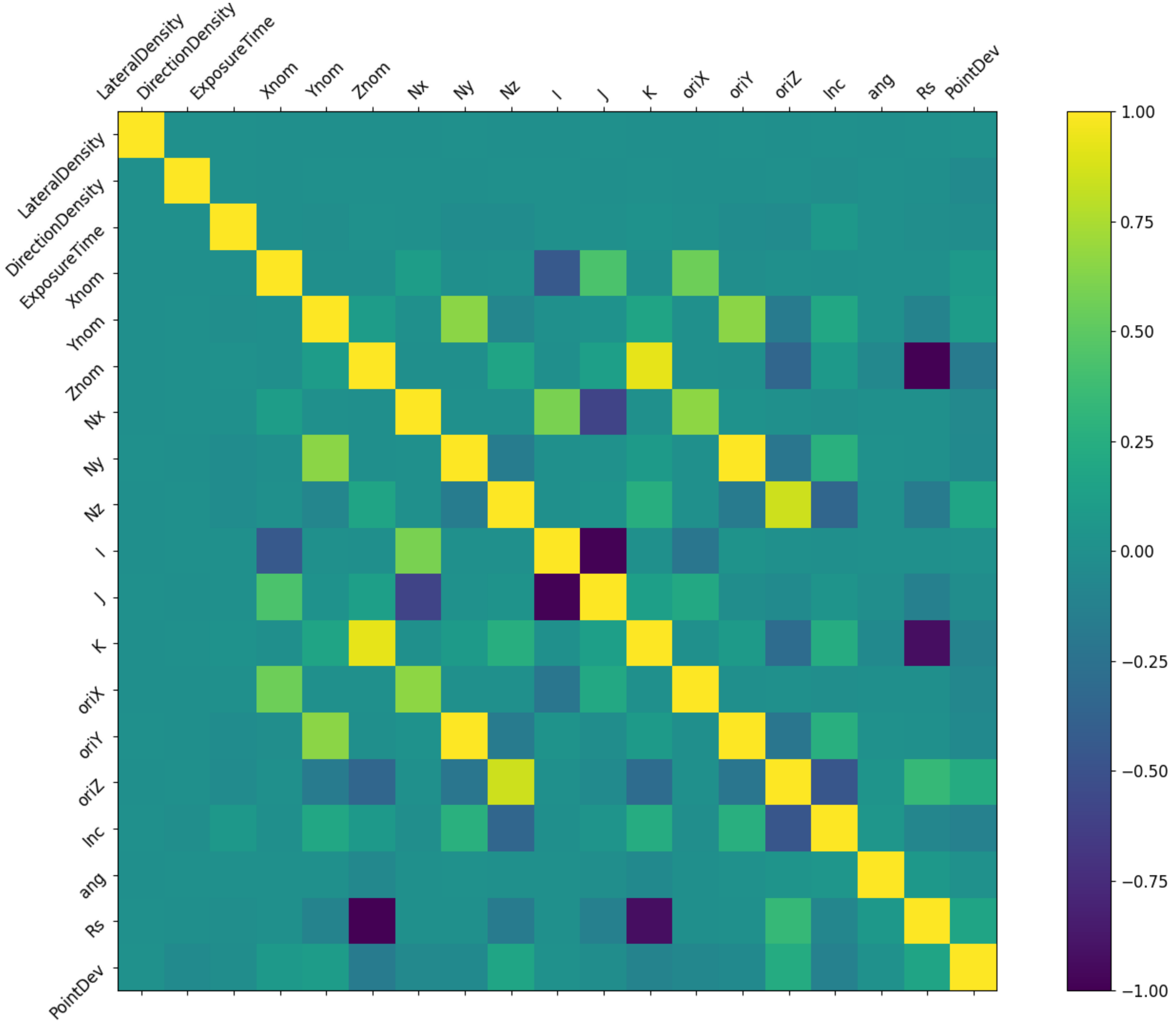
__Mean Point Error__
-0.024 ± 0.053 mm



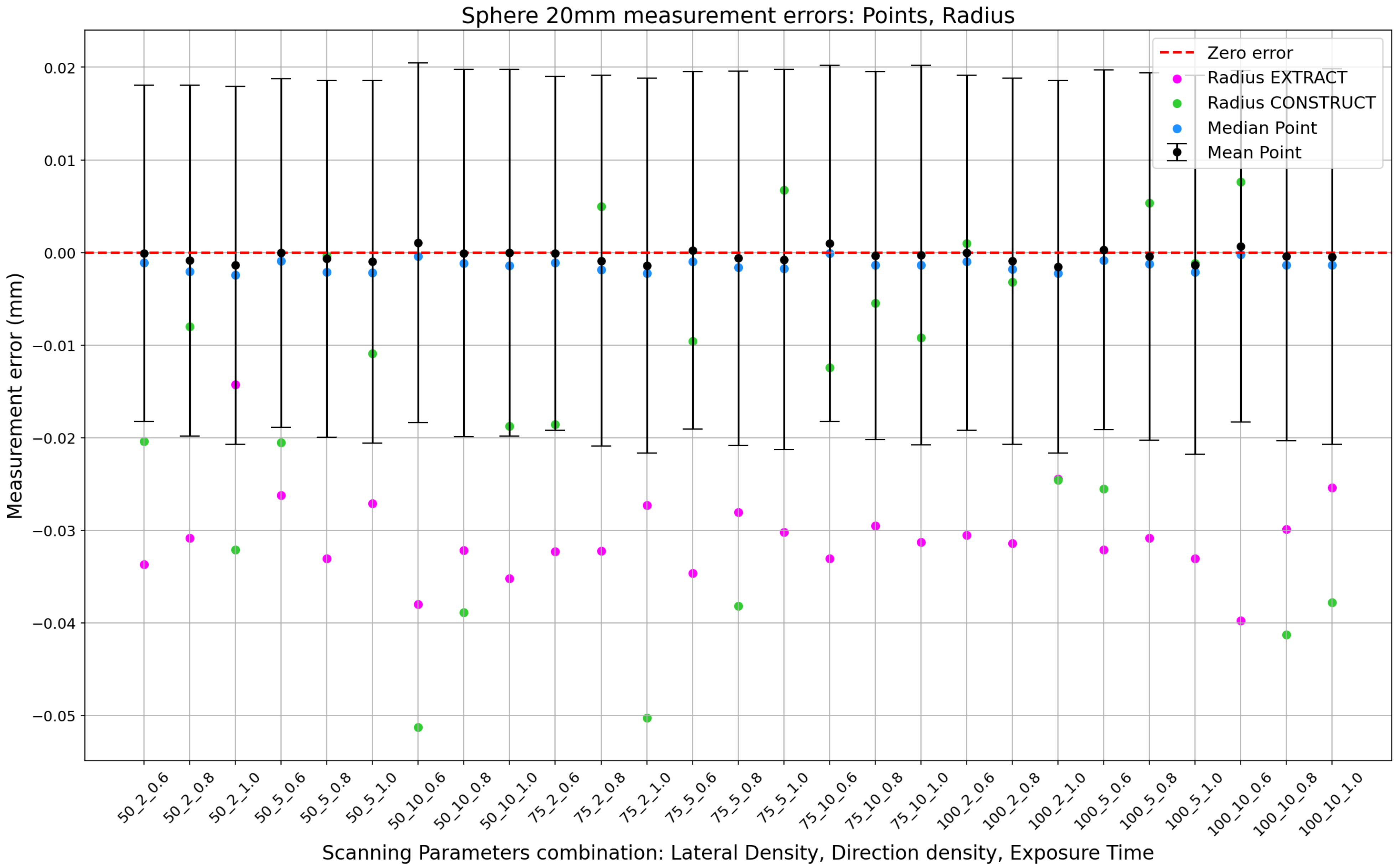
Data exploration & Analysis

Investigating (anti)correlations amongst features

- Scanning parameters (Lateral Density, Direction Density, Exposure Time) are not related to the target variable, Point measurement error (PointDev)
- The z-component of the normal vector (Nz) and the orientation difference vector (oriZ) show minor correlation to PointDev
- Strong correlations are due to calculation formulas or bias induced by specific geometries (sphere & cylinder)

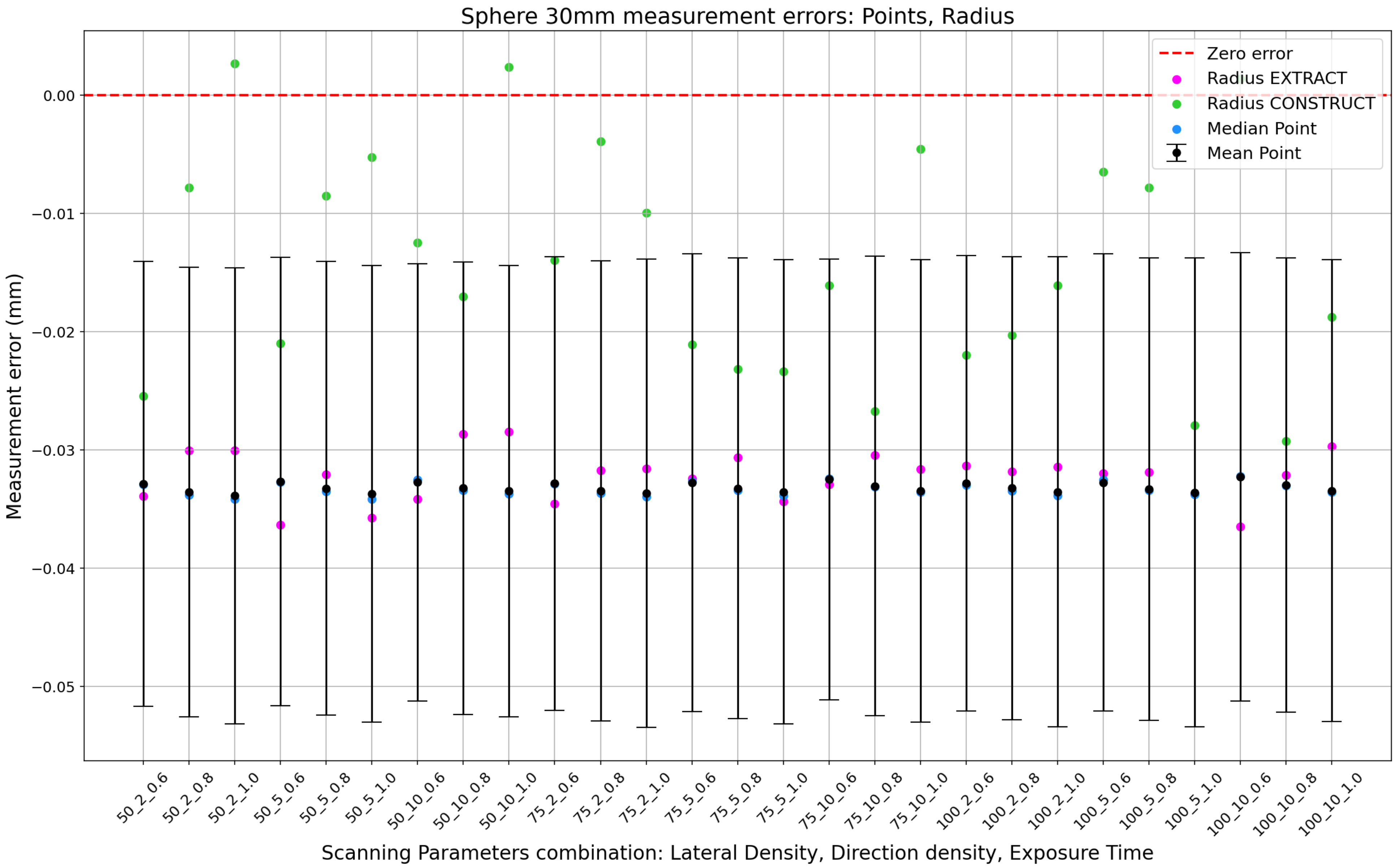


Data exploration & Analysis



- Statistical distribution of points: captures well actual surface
- Mean of radial distribution: good estimate of actual radius
- Both methods under-estimate sphere radius, method EXTRACT (Point Cloud) performs worse

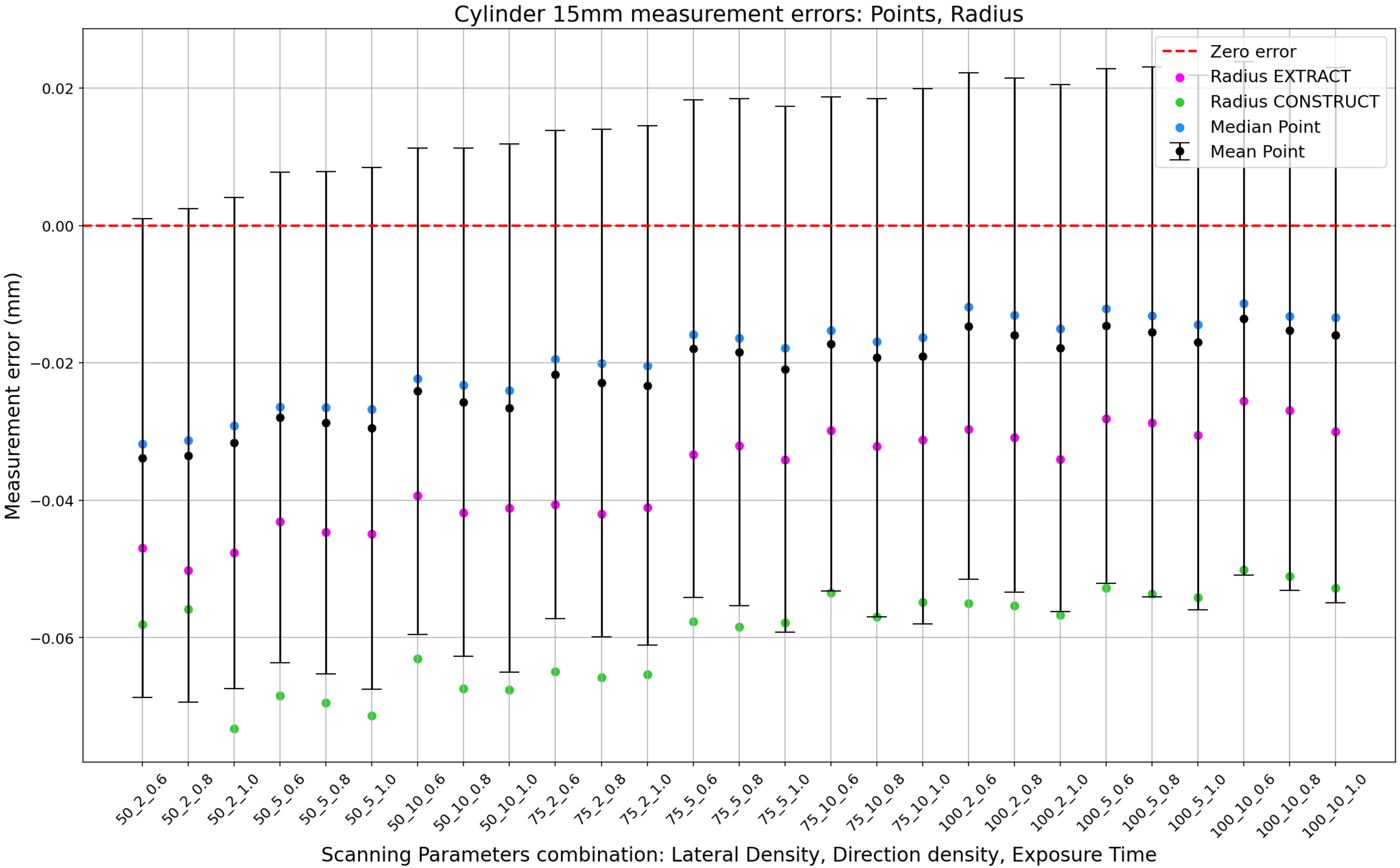
Data exploration & Analysis



- ➢ Statistical point distribution: well below actual surface
- ➢ Mean of distribution: bad estimate of radius

- ➢ Method EXTRACT (Point Cloud) under-estimates radius
- ➢ Method CONSTRUCT (Geometry Definition points) performs better

Data exploration & Analysis



- ➢ Statistical point distribution: below actual surface
- ➢ Mean of point radial distribution: good estimate of radius

- ➢ Both methods consistently under-estimate radius, Method EXTRACT (Point Cloud) performs better
- ➢ Cylinder captured better with high Lateral Density

Machine Learning Pipeline

Step 1

- Random selection of N points/point cloud
- Feature & target scaling to [0, 1] interval

Step 2

- Selection of algorithms (SVM, MLP, Decision Tree, Random Forest)
- Wide hyper-parameter grid

Step 3

- Randomized Grid search cross-validation
- Nested Cross-Validation scheme (10 fold)

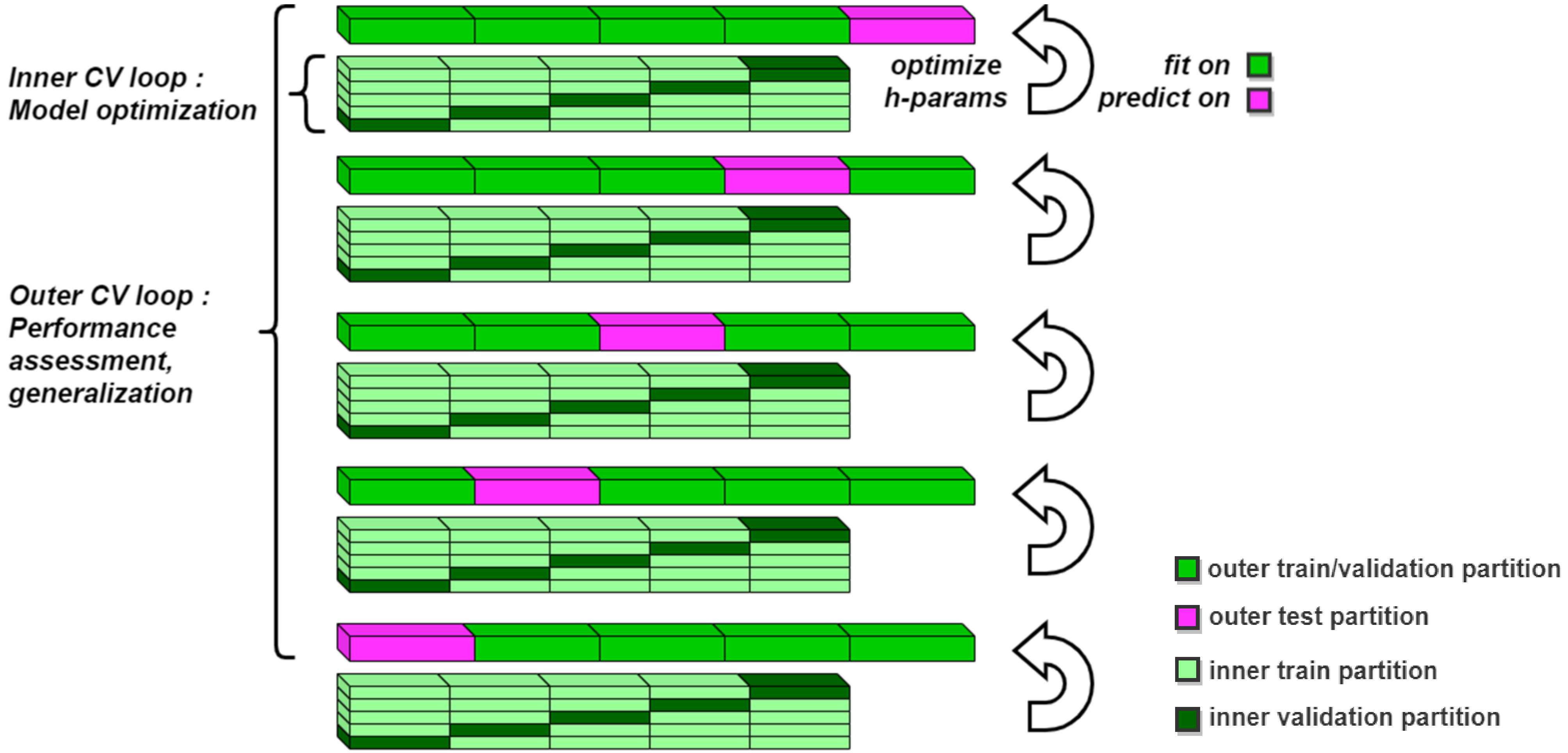
Step 4

- Optimal model retrieval based on mean validation scores

Model training, optimization & Validation

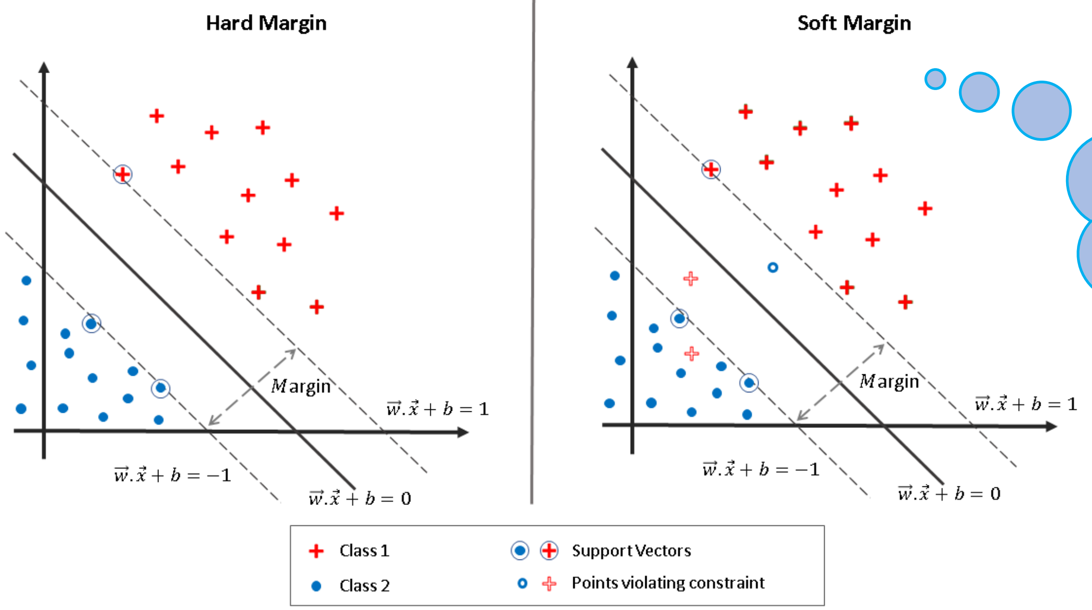
Nested Cross-Validation scheme

Self-consistent methodology to train/optimize models & assess their performance

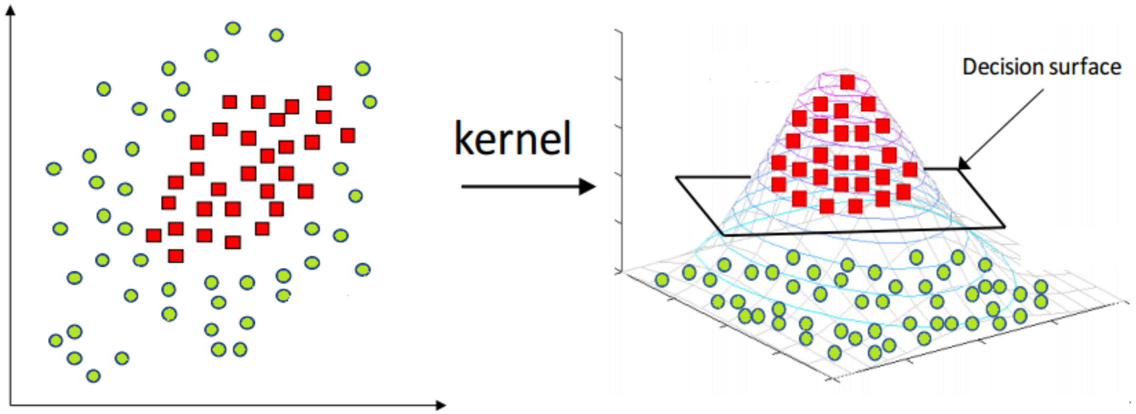


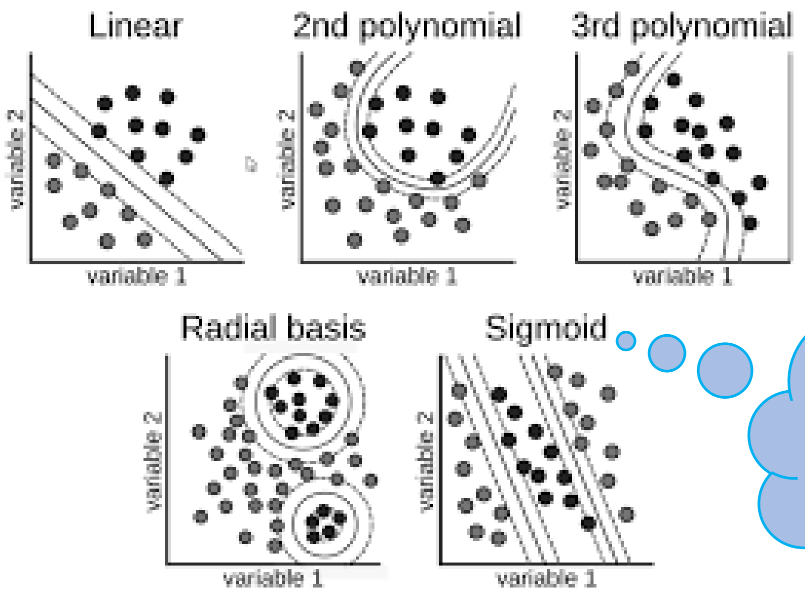
Model intuition – Support Vector Machine (SVM)

- Finds Optimal separating hyperplane (decision surface) for linearly separable patterns
- Optimal Hyperplane => maximum margin
- Support vectors => data points closest to the decision surface (points most difficult to discriminate – specify optimum location of decision surface)
- Extend to not linearly separable patterns: transformations of original data into new space – Kernel function



Hyper-parameter ***C*** controls margin
Large C => small margin



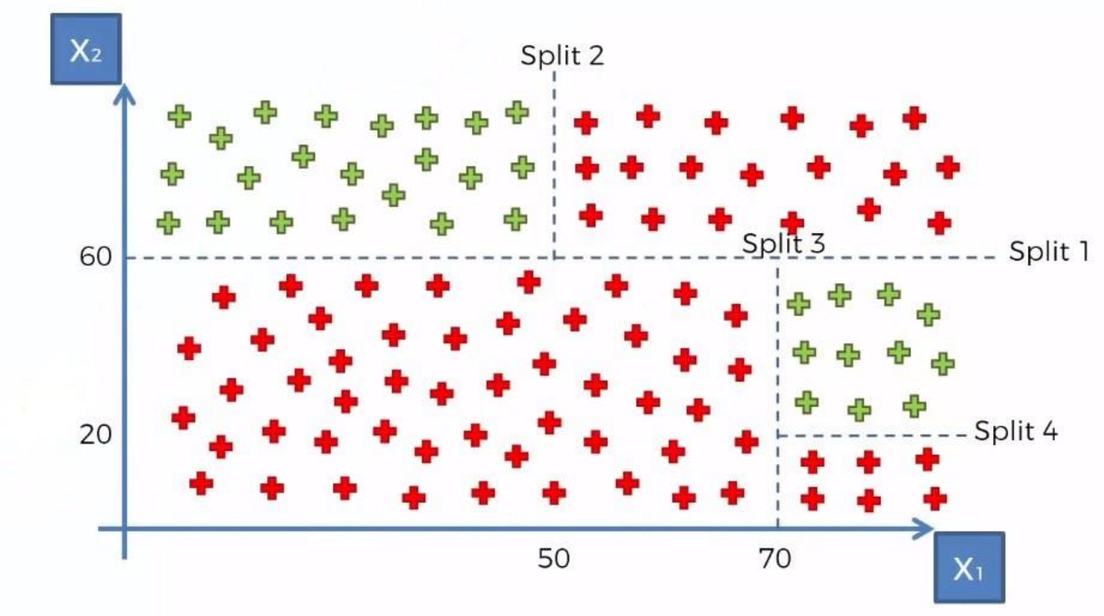


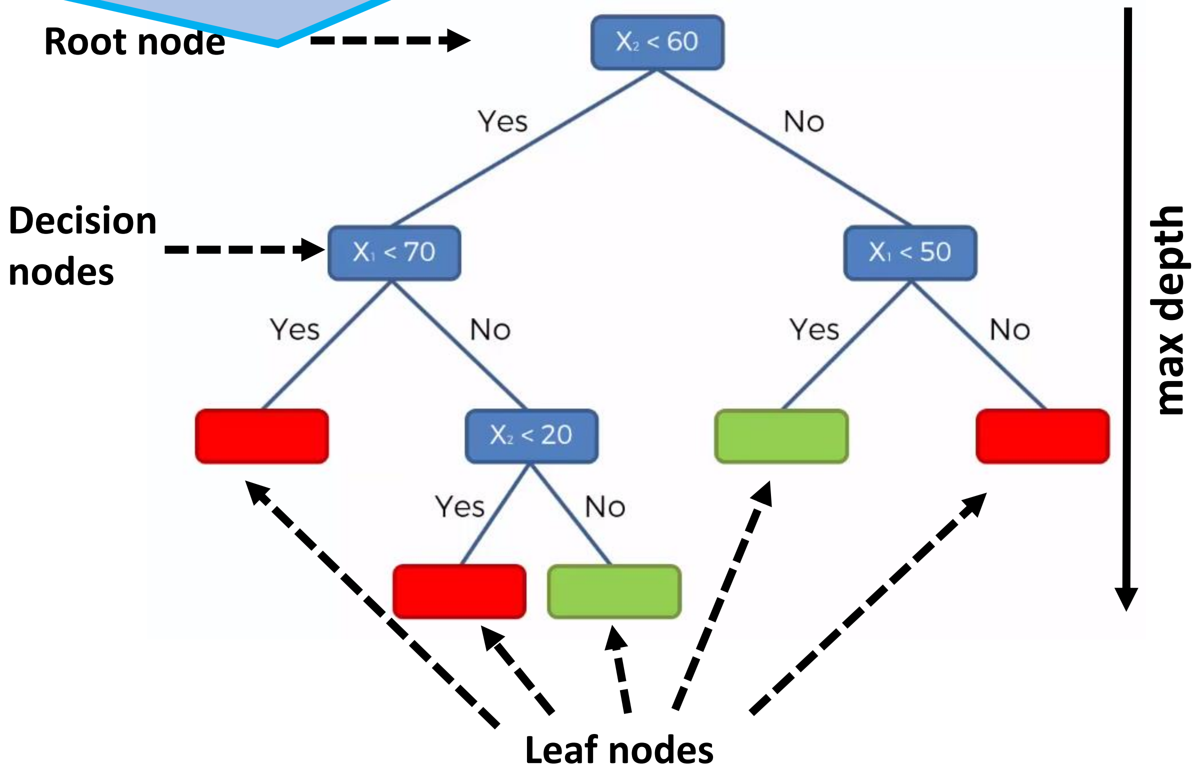
H-parameter ***kernel*** controls kernel function

Model intuition - Decision Tree

- Splits data points to maximize information gain (minimize impurity) in the resulting data partitions
- Optimal partitions contain as much as possible information and less randomness

Hyper-parameters (regression)
- *max_depth*: maximum depth of the tree (max # of decision levels)
- *splitter*: strategy to perform splits (“best”, “random”)
- *split_criterion*: function to measure quality of split (“squared_error”, …)
- *min_samples_split*: minimum # of samples to split decision nodes
- *min_samples_leaf*: minimum # of samples at leaf nodes

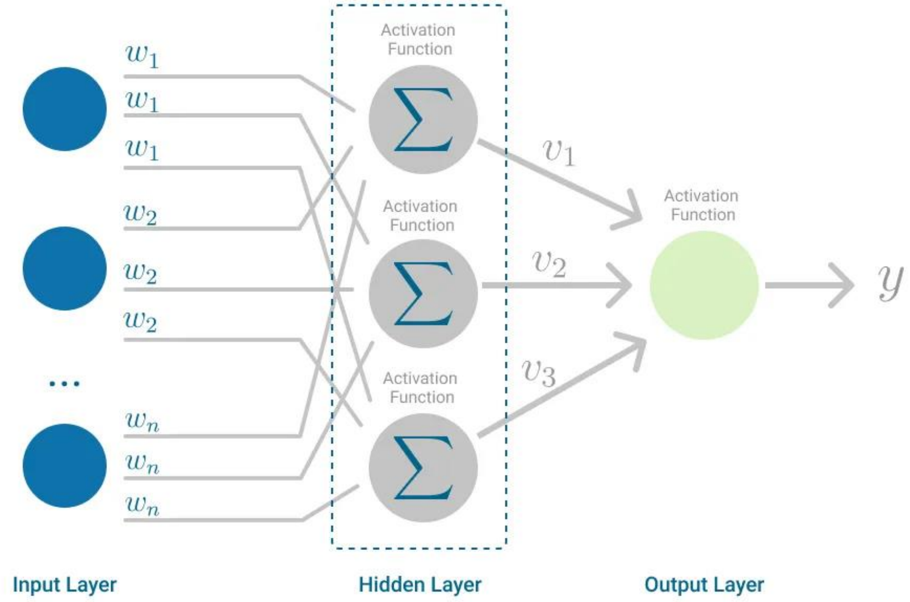


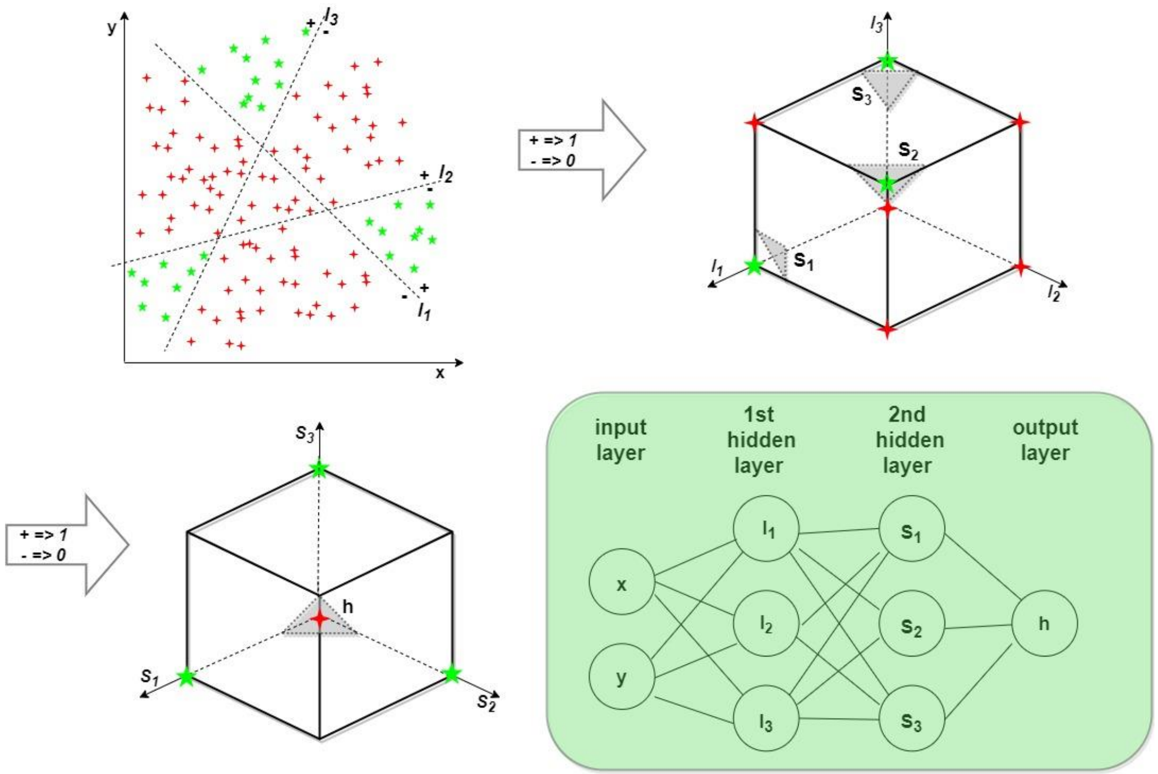


Model intuition - Multi-layer Perceptron (MLP)

- Sequentially transforms input into higher dimensional spaces: samples can be discriminated
- Neuron: basic unit of computation, receives inputs & weights (for each input)
- At each neuron: inputs combined in a weighted sum
- If weighted sum exceeds predefined threshold: neuron activation, output production
- Threshold represented by activation function

Hyper-parameters
- *hidden_layer_sizes*: the number of neurons in each hidden layer
- *activation*: Activation function(“logistic”, “relu”…)
- *solver*: the algorithm for weight optimization (“sgd”, “adam”, …)





ML Experimental results – Model performance

Optimal SVM model

Hyperparameters = {

kernel: RBF,

C: 10.0}

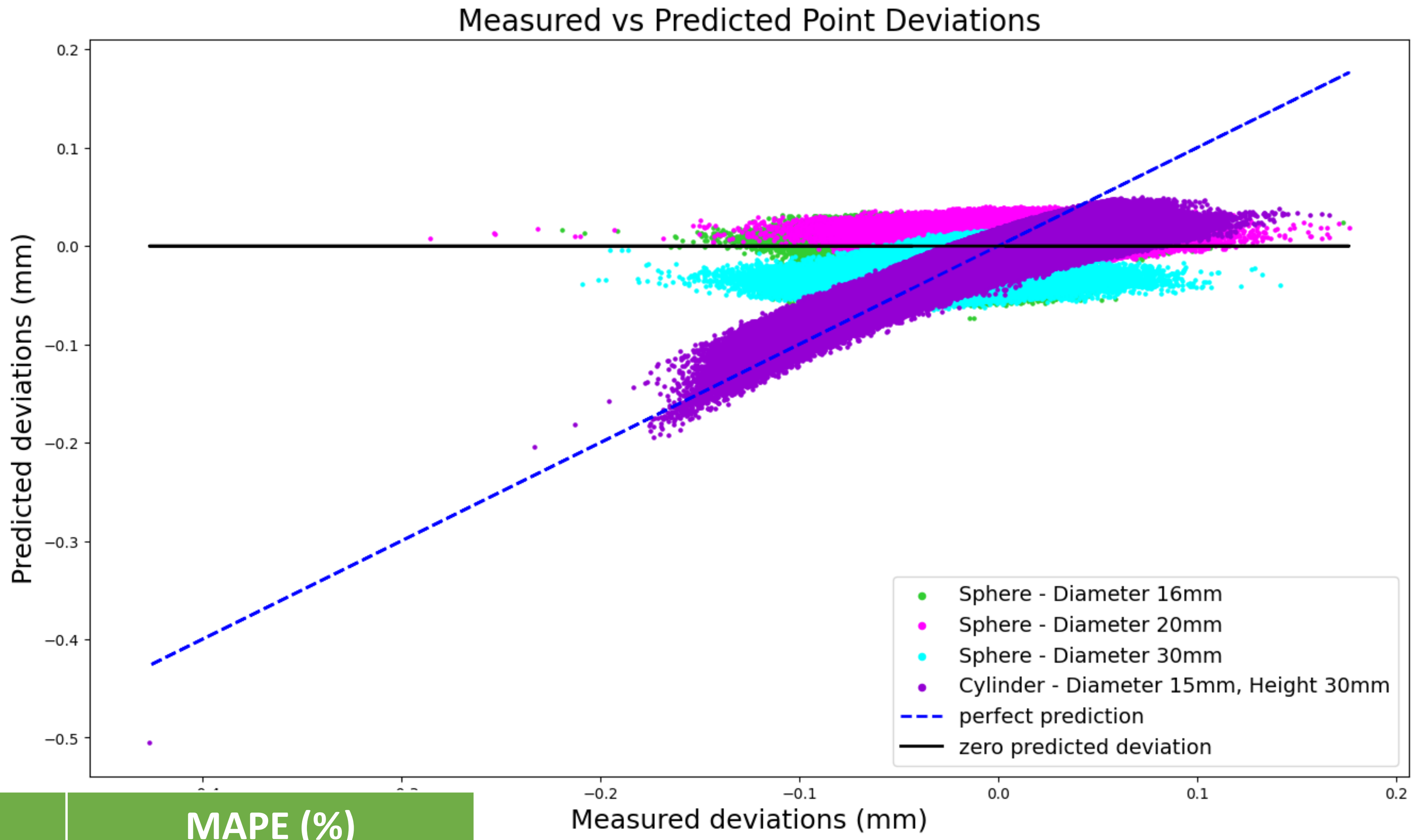
Predictions on spheres

Large deviations from actual

Flat distribution

Predictions on cylinder

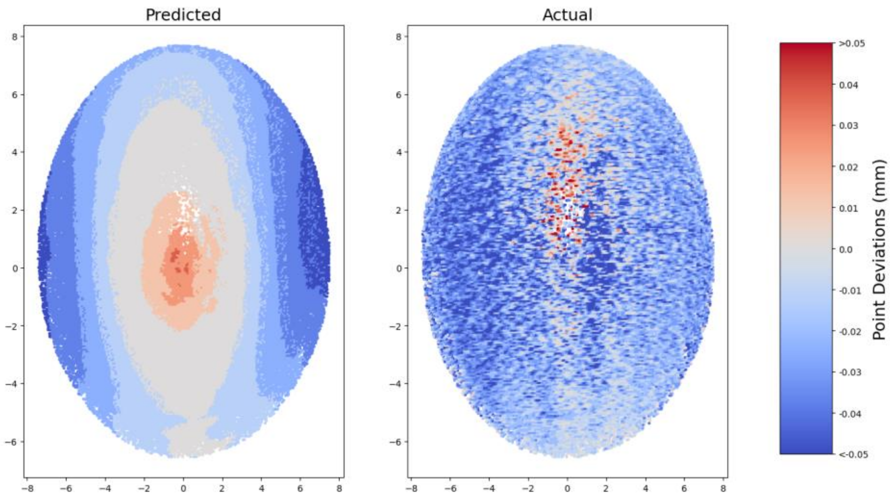
better approximation of truth



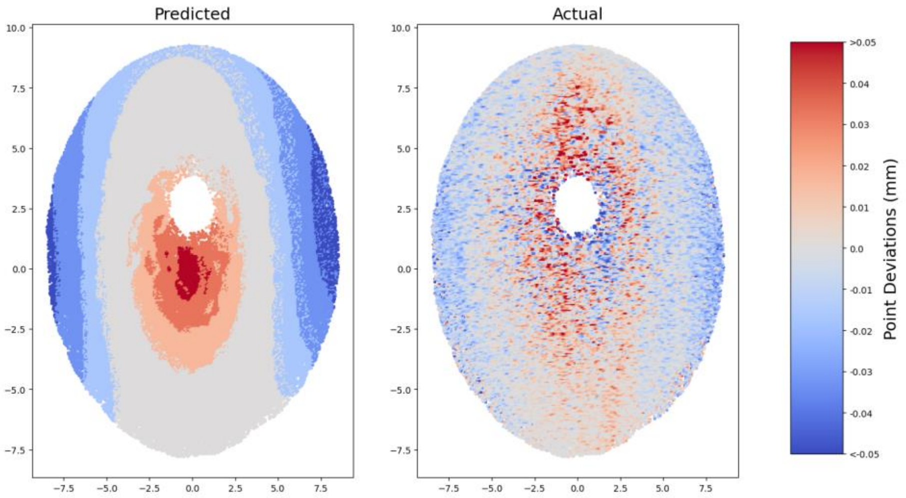
	MAE (mm)	MAPE (%)
Mean Validation score	0.026 ± 0.002	3.24 ± 0.03
Mean Test score	0.028 ± 0.004	3.4 ± 0.1

ML Experimental results – Visualize predictions

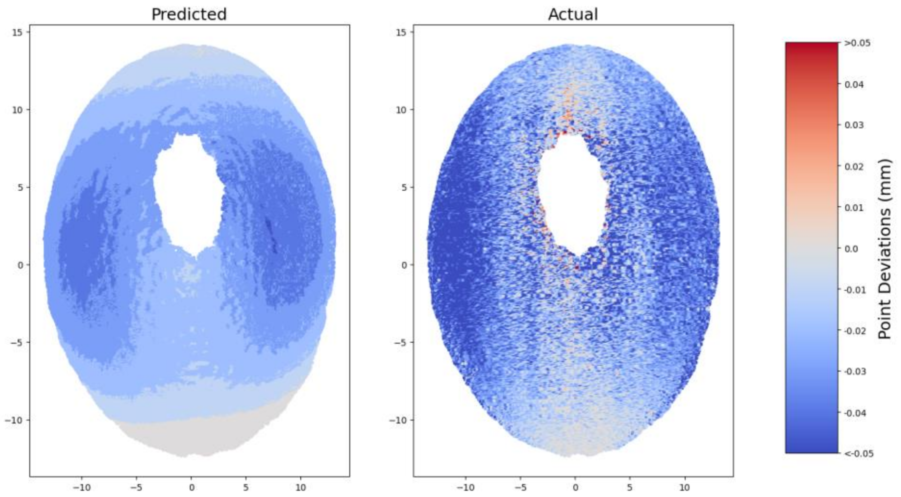
Sphere 16.0026mm, Lateral Density=50, Direction Density=2, Exposure Time=1.0



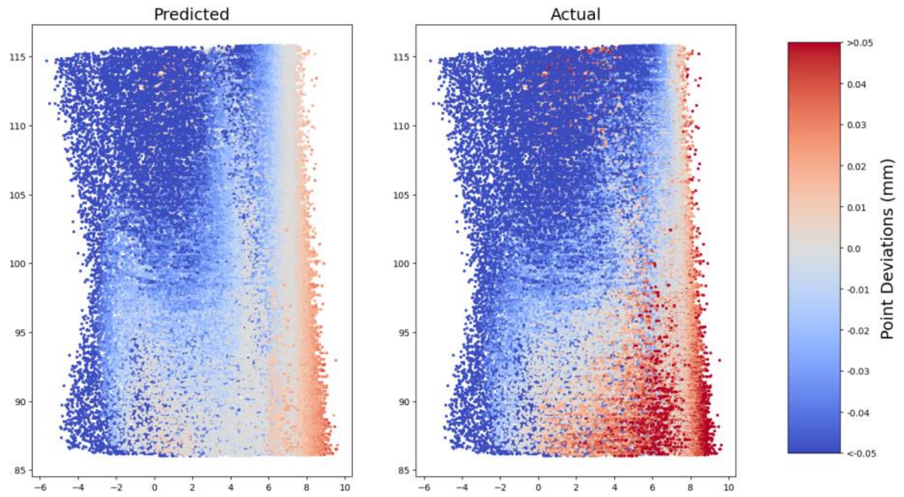
Sphere 20.0118mm, Lateral Density=50, Direction Density=2, Exposure Time=1.0



Sphere 30.0085mm, Lateral Density=50, Direction Density=2, Exposure Time=1.0

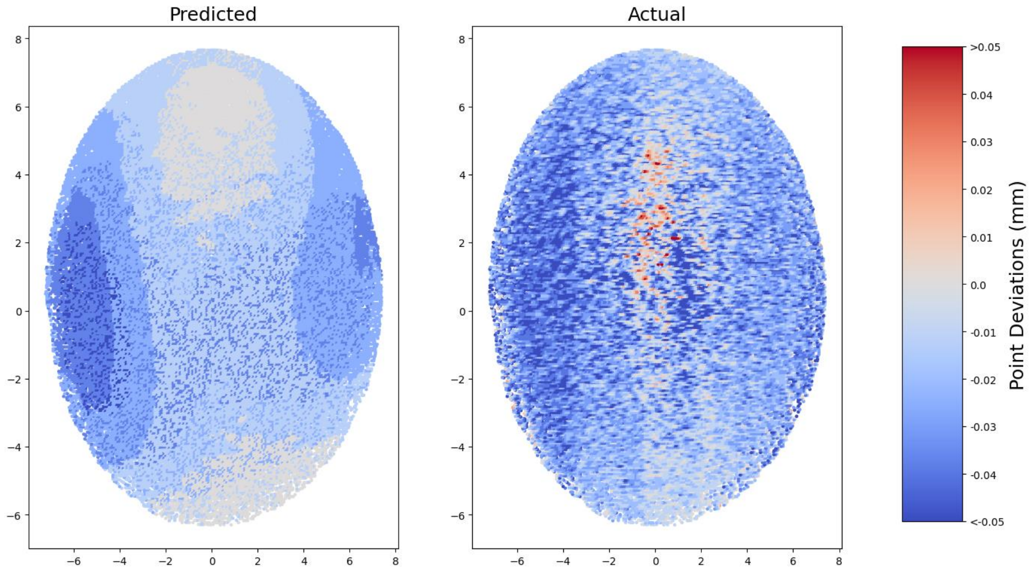


Cylinder 15.0219mm, Lateral Density=50, Direction Density=2, Exposure Time=1.0

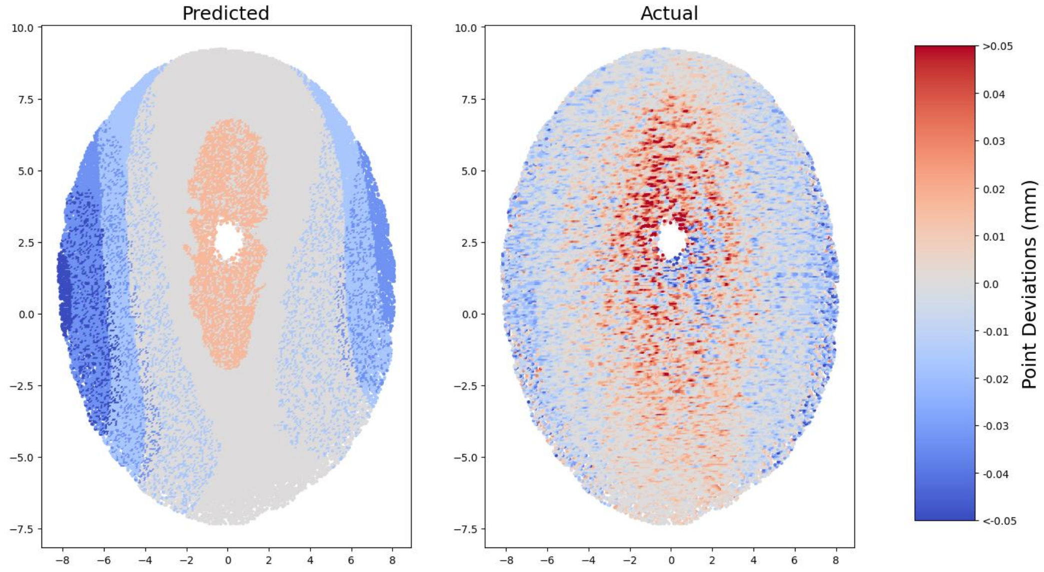


ML Experimental results – Visualize predictions

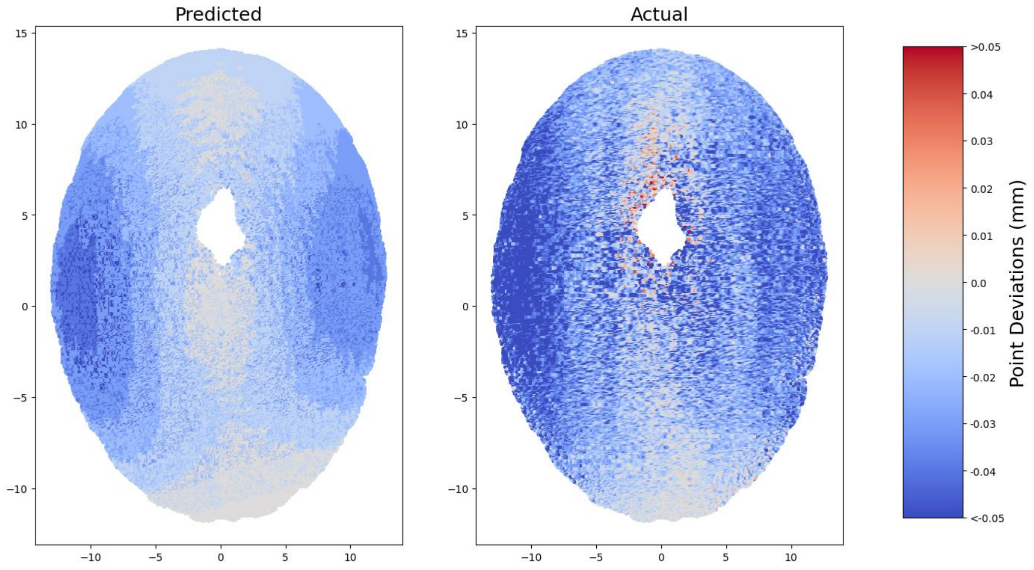
Sphere 16.0026mm, Lateral Density=50, Direction Density=2, Exposure Time=0.6



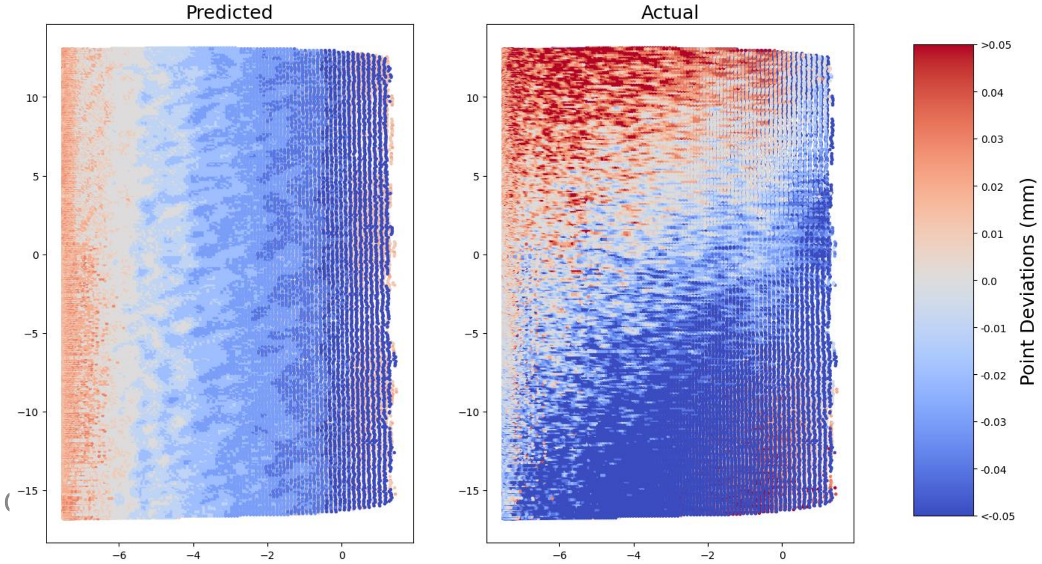
Sphere 20.0118mm, Lateral Density=50, Direction Density=2, Exposure Time=0.6



Sphere 30.0085mm, Lateral Density=50, Direction Density=2, Exposure Time=0.6



Cylinder 15.0219mm, Lateral Density=50, Direction Density=2, Exposure Time=0.6



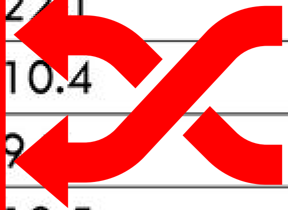
Model Explanations - Feature Importance methods

Permutation Feature Importance

- Permute each feature by random shuffle of values
- Compute the decrease in model performance scores
- Repeat N times
- Calculate mean decrease & std in performance scores

Computed independently for each feature …
Permutation Importance scores **do not** add up to 1

	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	9
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9

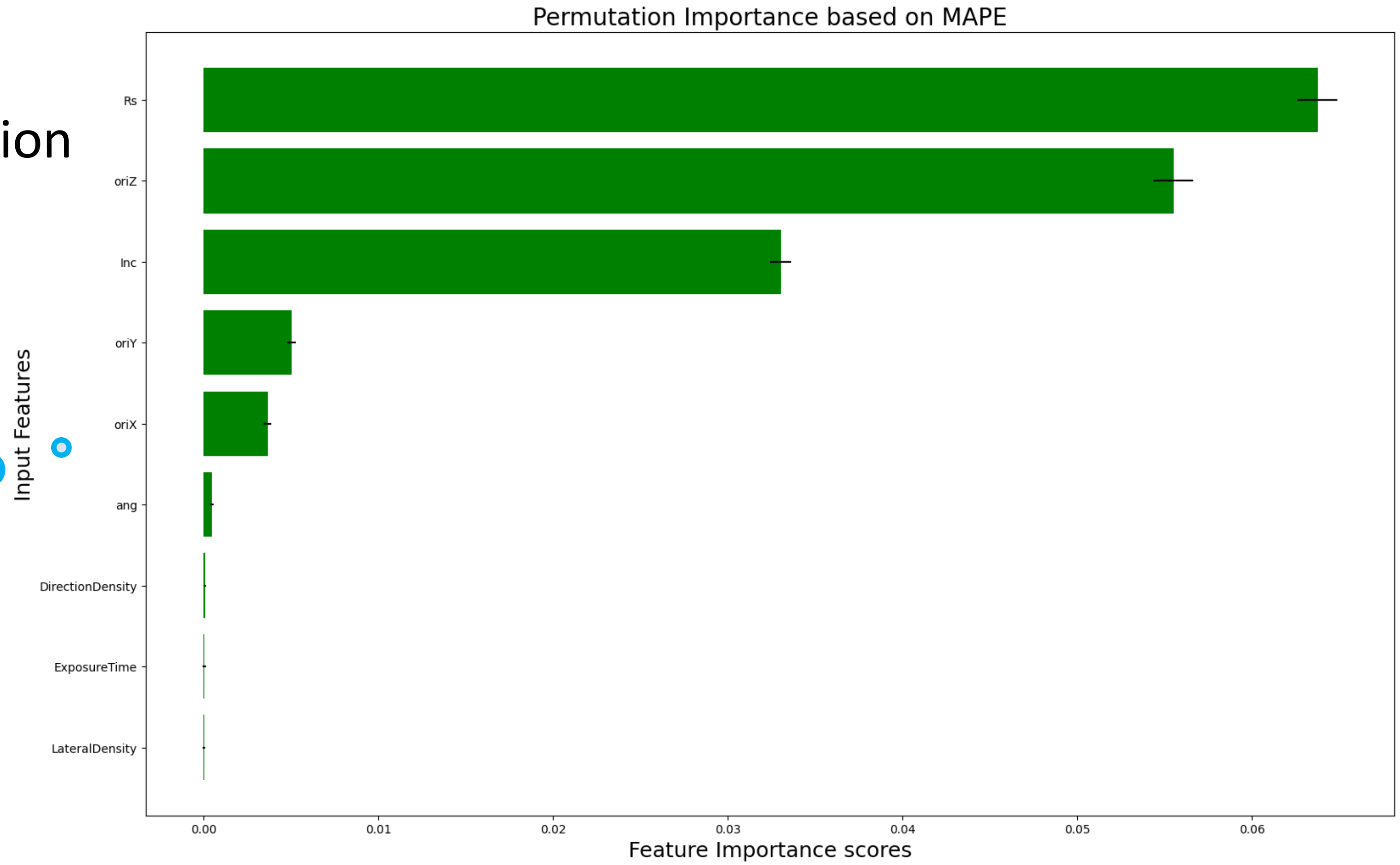


Model Explanations

Features with high effect on model's output:

- *Rs* = Distance to laser source
- *oriZ* = z-component of Orientation difference vector
- *Inc* = Incidence angle

Very low importance scores **< 7%**

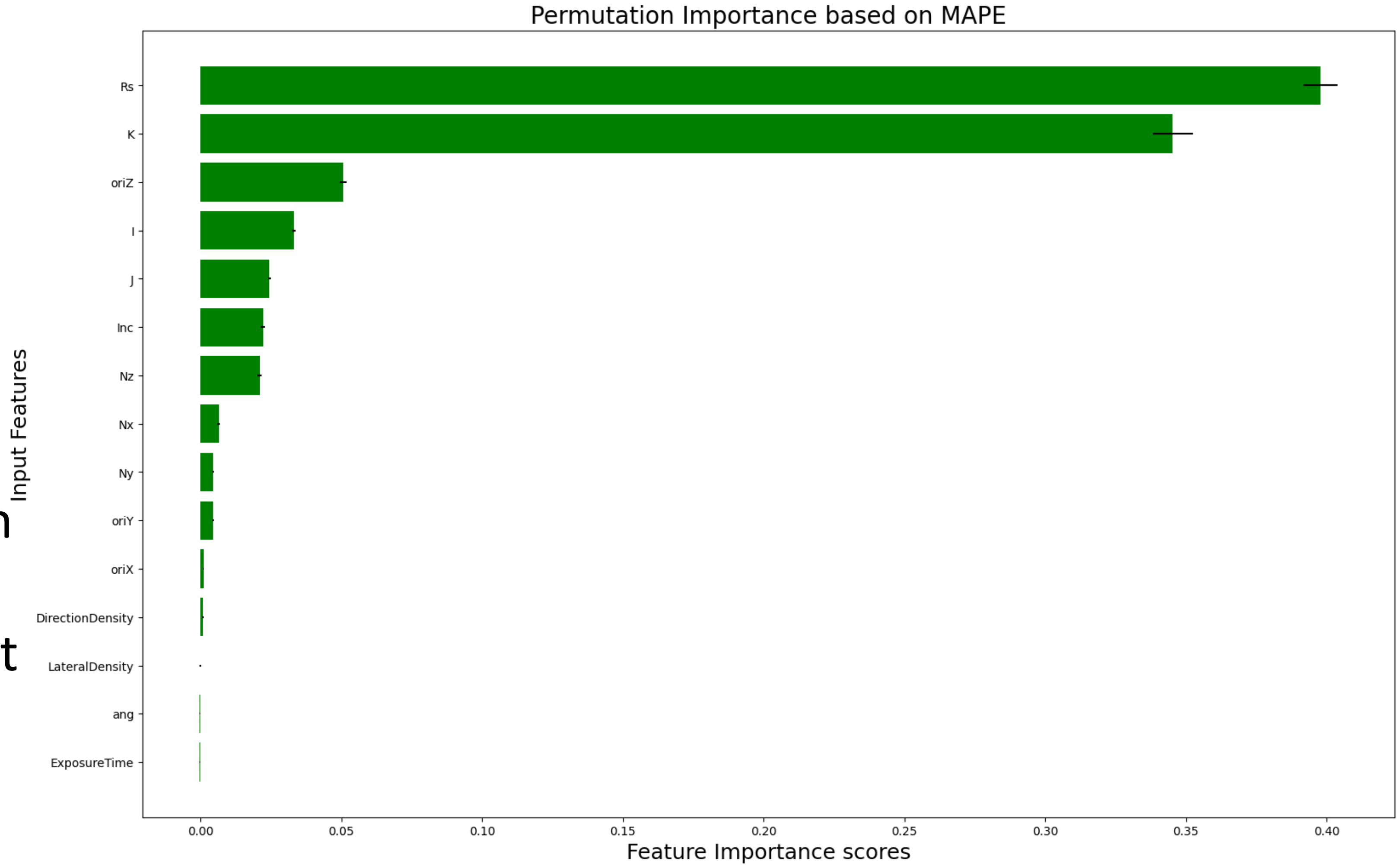


Model Explanations

Features with high effect on model's output:

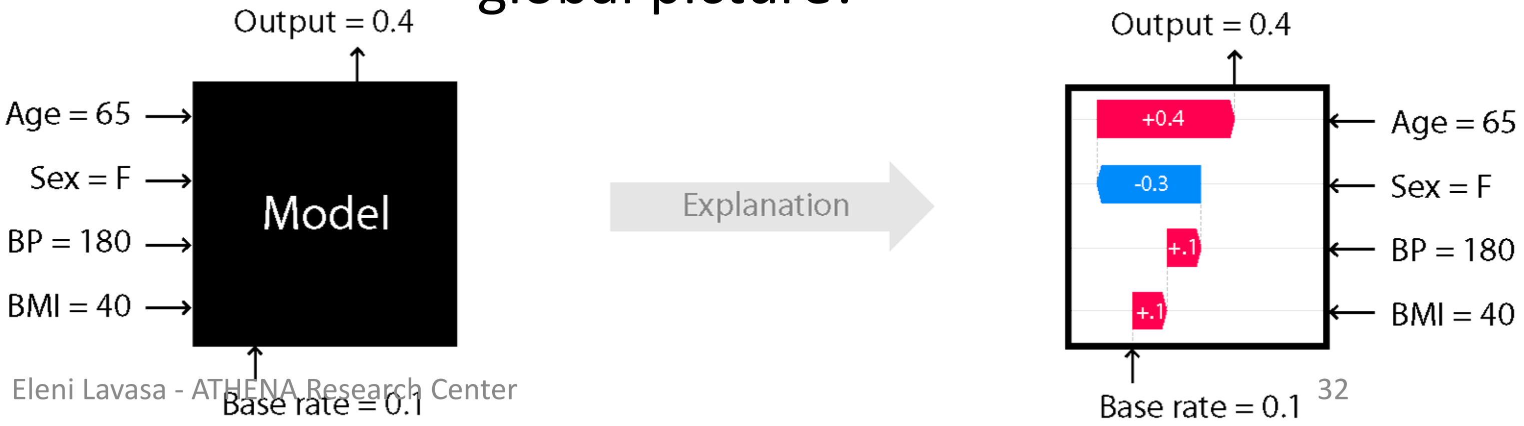
- ➢ Rs = Distance to laser source
- ➢ K = z-component of laser source direction

- ➢ Features *Rs, K* have ~8-10 times higher importance than the rest
- ➢ Features *Rs, K* have significant impact on model's predictions: 35-40%



SHAP – SHapley **A**dditive ex**P**lanations

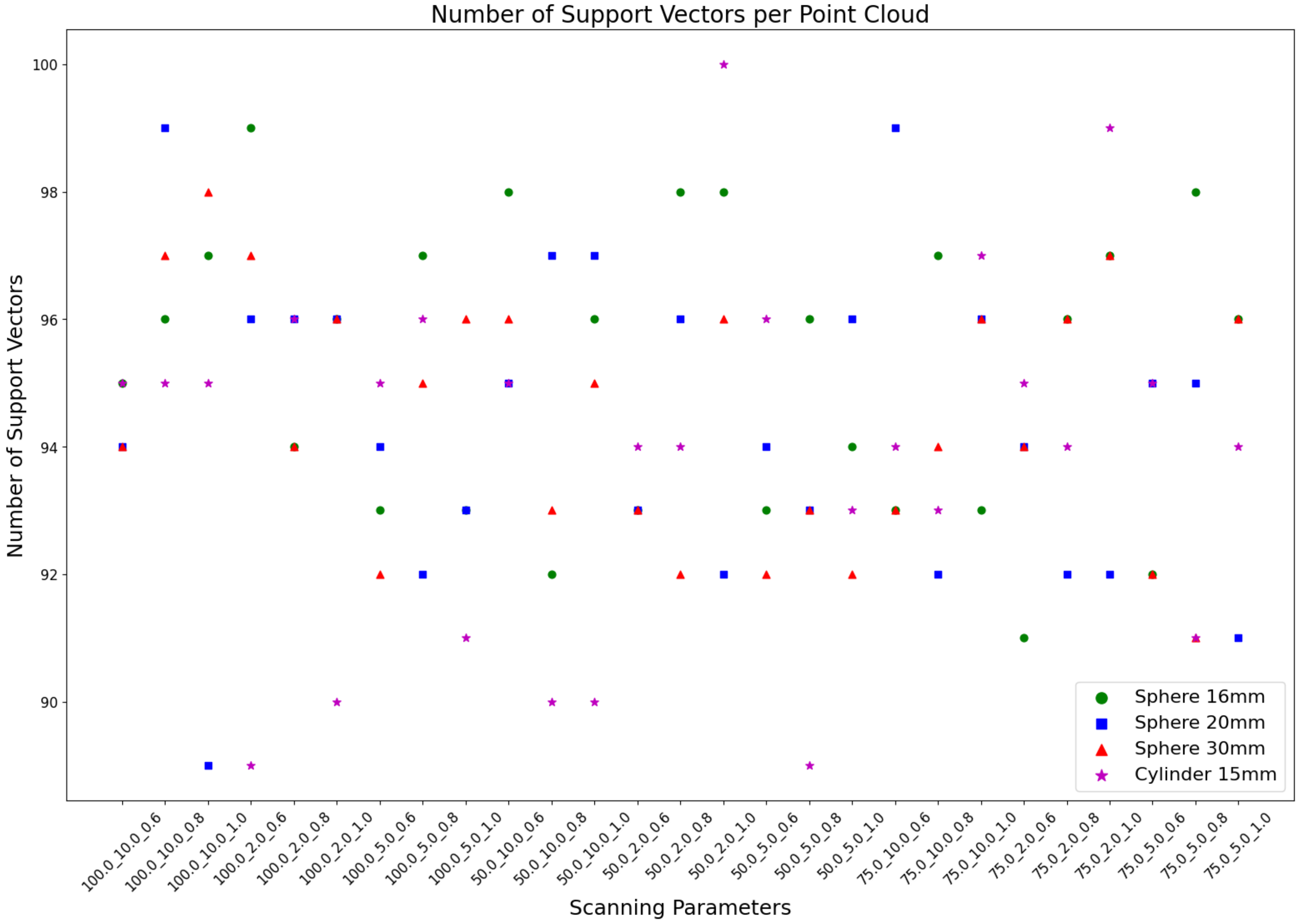
- Based on Shapley values from cooperative game theory
- Shapley quantifies the contribution of each player to the outcome of the game, considering every possible coalition of players
- SHAP quantifies the contribution each feature brings to each prediction of the ML model, considering all possible combinations of input features
- Builds one predictive model per combination of features, sequentially including more features: estimates the marginal contribution of each feature to the final outcome
- Aggregation of local explanations: global picture!



Model Explanations

Retrieve Support Vectors (SV):

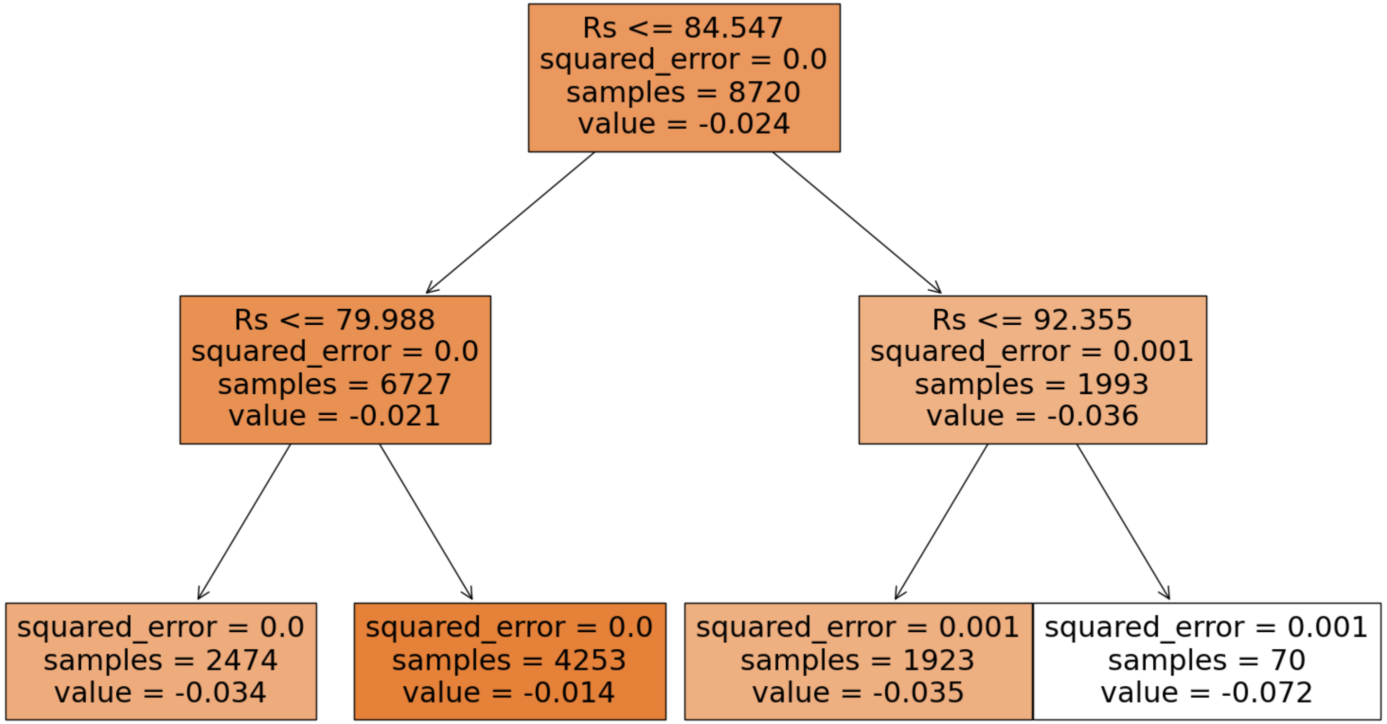
- 90-100 SVs per Point Cloud
- Training samples per Point Cloud = 100
  - SVs ≈ training samples!



Model Explanations – Global surrogate

Decision Tree to explain trained SVM

- Depth = 2
- MAE = 0.014 (mm)



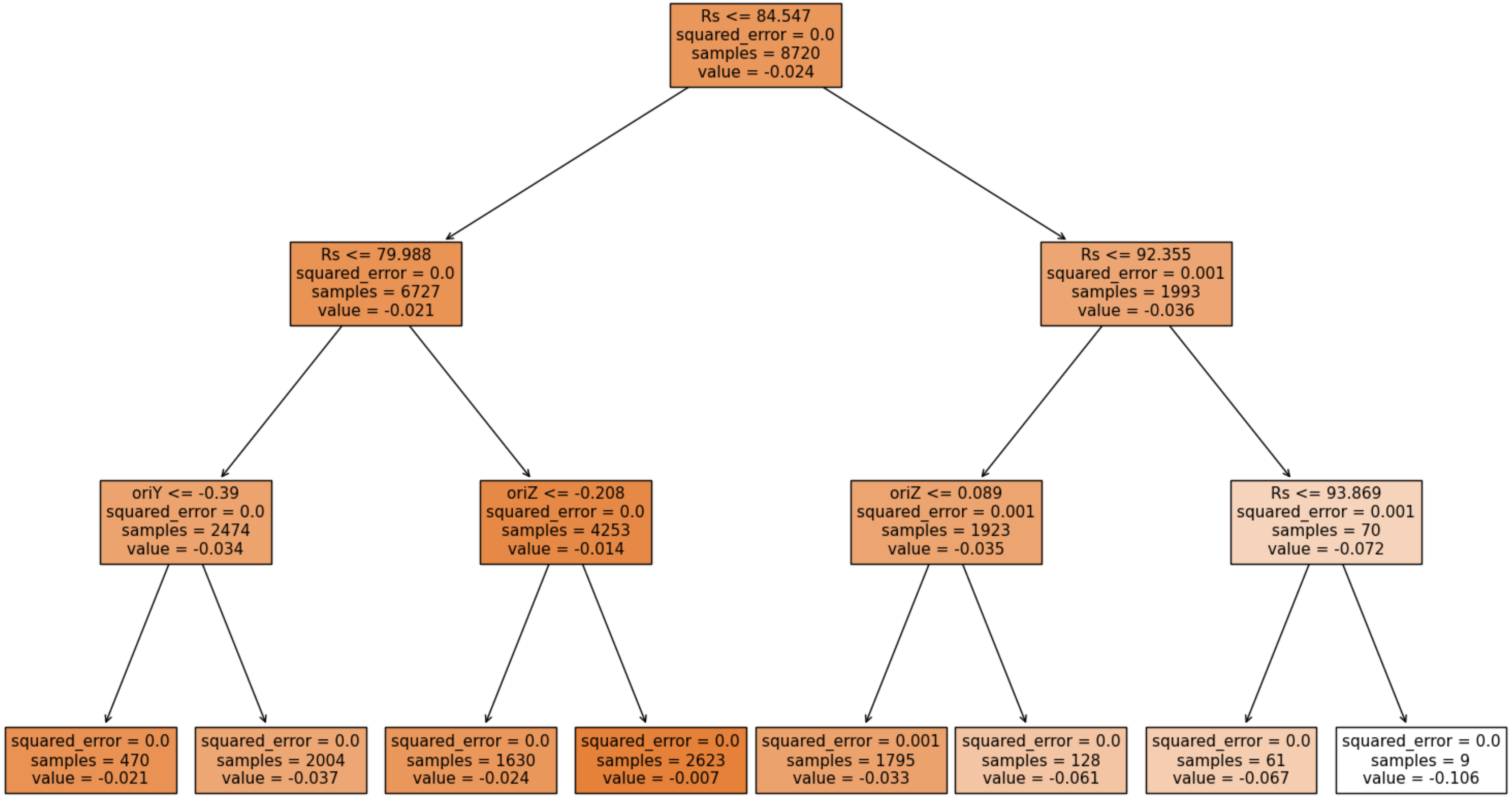
Decision Rules

```
|--- Rs <= 84.55
|   |--- Rs <= 79.99
|   |   |--- value: [-0.03]
|   |--- Rs >  79.99
|   |   |--- value: [-0.01]
|--- Rs >  84.55
|   |--- Rs <= 92.35
|   |   |--- value: [-0.03]
|   |--- Rs >  92.35
|   |   |--- value: [-0.07]
```

Model Explanations – Global Surrogate

Decision Tree to explain trained SVM

- Depth = 3
- MAE = 0.012 (mm)



Decision Rules

```
|--- Rs <= 84.55
|   |--- Rs <= 79.99
|   |   |--- oriY <= -0.39
|   |   |   |--- value: [-0.02]
|   |   |--- oriY >  -0.39
|   |   |   |--- value: [-0.04]
|   |--- Rs >  79.99
|   |   |--- oriZ <= -0.21
|   |   |   |--- value: [-0.02]
|   |   |--- oriZ >  -0.21
|   |   |   |--- value: [-0.01]
|--- Rs >  84.55
|   |--- Rs <= 92.35
|   |   |--- oriZ <= 0.09
|   |   |   |--- value: [-0.03]
|   |   |--- oriZ >  0.09
|   |   |   |--- value: [-0.06]
|   |--- Rs >  92.35
|   |   |--- Rs <= 93.87
|   |   |   |--- value: [-0.07]
|   |   |--- Rs >  93.87
|   |   |   |--- value: [-0.11]
```

Conclusions – Next Steps

Prediction of point-wise measurement accuracy

- Extraction of additional features
- More measurements needed (different geometries, fine granulation of scanning parameters)
- Repeat experiments for concave/complex geometries (different laser setup)

Relate point deviations to the target of the measurement

- Use Open3D to fit nominal geometries, calculate dimension & estimate uncertainty

New problem statement

- Unsupervised / semi-supervised learning methods
- Model data on graphs & apply Graph ML techniques

Thank you for your attention!