PROBABILISTIC INTELLIGENCE

Examples of AI for Simulation and Experimental Data

Authors: Dr. Kevin Cremanns*

*Co-founder & responsible for research
PI Probaligence GmbH

CADFEM GROUP

About PI Probaligence

PI offers:

- Unique self-developed ML algorithms
- (Customized) software products
- Consulting
- Methods development
- Research partnerships
- Training courses for professionals

Our software Stochos

Web application

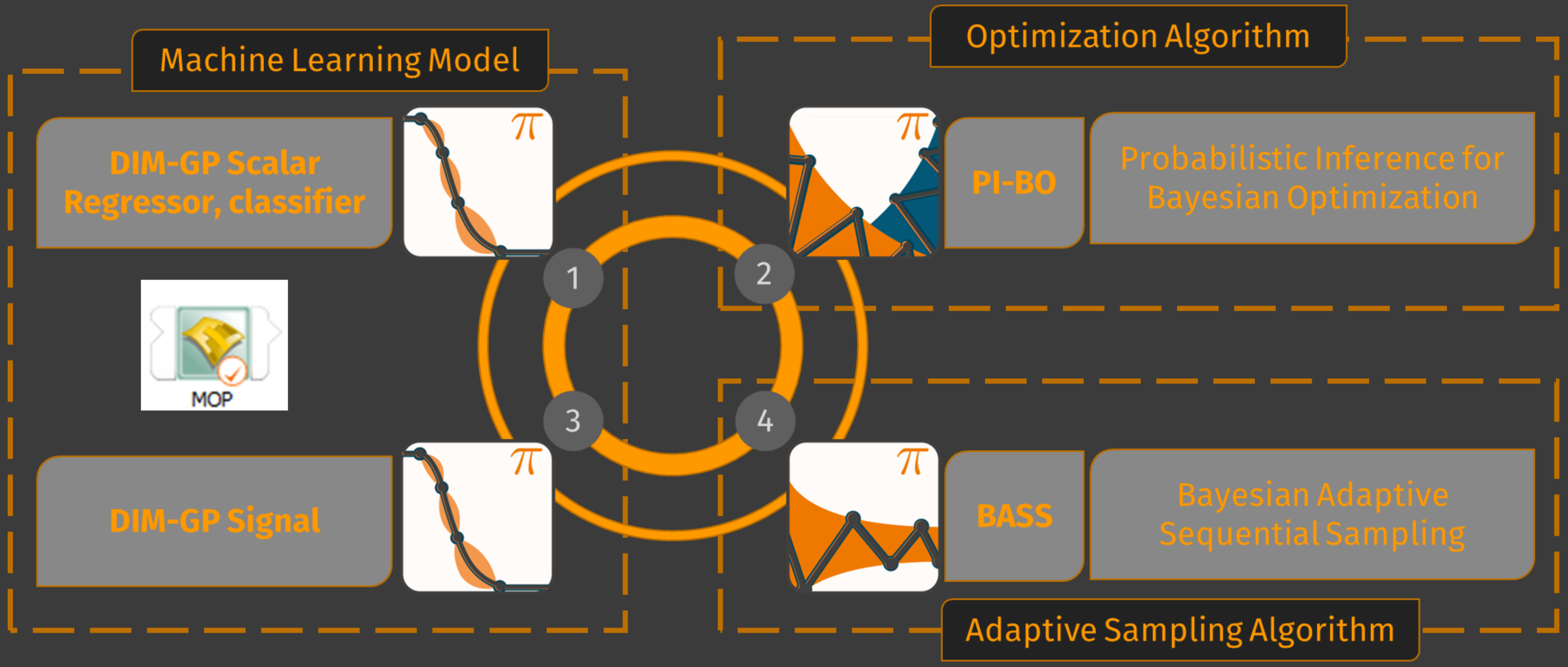
Python module

```python
from stochos.bayesian_optimization import bayesian_opt
from stochos.plot import plot_scatter_mat, plot_sensitivity
from stochos.dimgp import dimgp_regr, pam_regr
from stochos.sensitivity import sobol_indices
import numpy as np
import pandas as pd

"""Define parameter bounds and type (continous "c", discrete "d"). We use
here +-20% based on reference design"""
reference_design = [45, 5, 6, 3, 60, 6, 35, 35, 35, 35, 35, 10, 10, 10, 10 ,10]
bounds = []
for i in range(len(reference_design)):
    tmp_val = reference_design[i]
    bounds.append([tmp_val*0.8, tmp_val*1.2])
types=["c"]*16

"""Define objective, minimization is assumed"""
def objectives(x,models):
    objs, lcbs = [],[]
    for i in range(len(models)):
        y_pred, l_cb, u_cb= models[i].predict(x, CI=0.6827)
        if i == 0:
```

in the fields of design of experiment, probabilistic machine learning, stochastic analysis and optimization.

Part of Ansys OptiSLang (2022)

- Since 2022 parts of **STOCHOS** are integrated in the Ansys OptiSLang (AI+ license required)



Probabilistic Machine Learning - Gaussian Process

Neural networks pros & cons

Advantages:

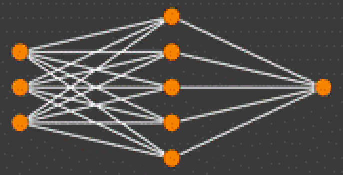
- Fast training and infinite scalability (big data)
- Generalising regression and classification model
- State of the art for speech, image and video recognition

Disadvantages:

- Requires typically many data points for a good prognosis quality → higher hardware requirements
- Prone to overfitting because of many trainable parameters
- Not a probabilistic model (except Bayesian NN) most practical implementations like Drop-NN or Ensembles are not truly Bayesian
- Many hyperparameters (network topology, number of neurons and layers, type of activation function types, optimizer, …) which have an impact on the model learning

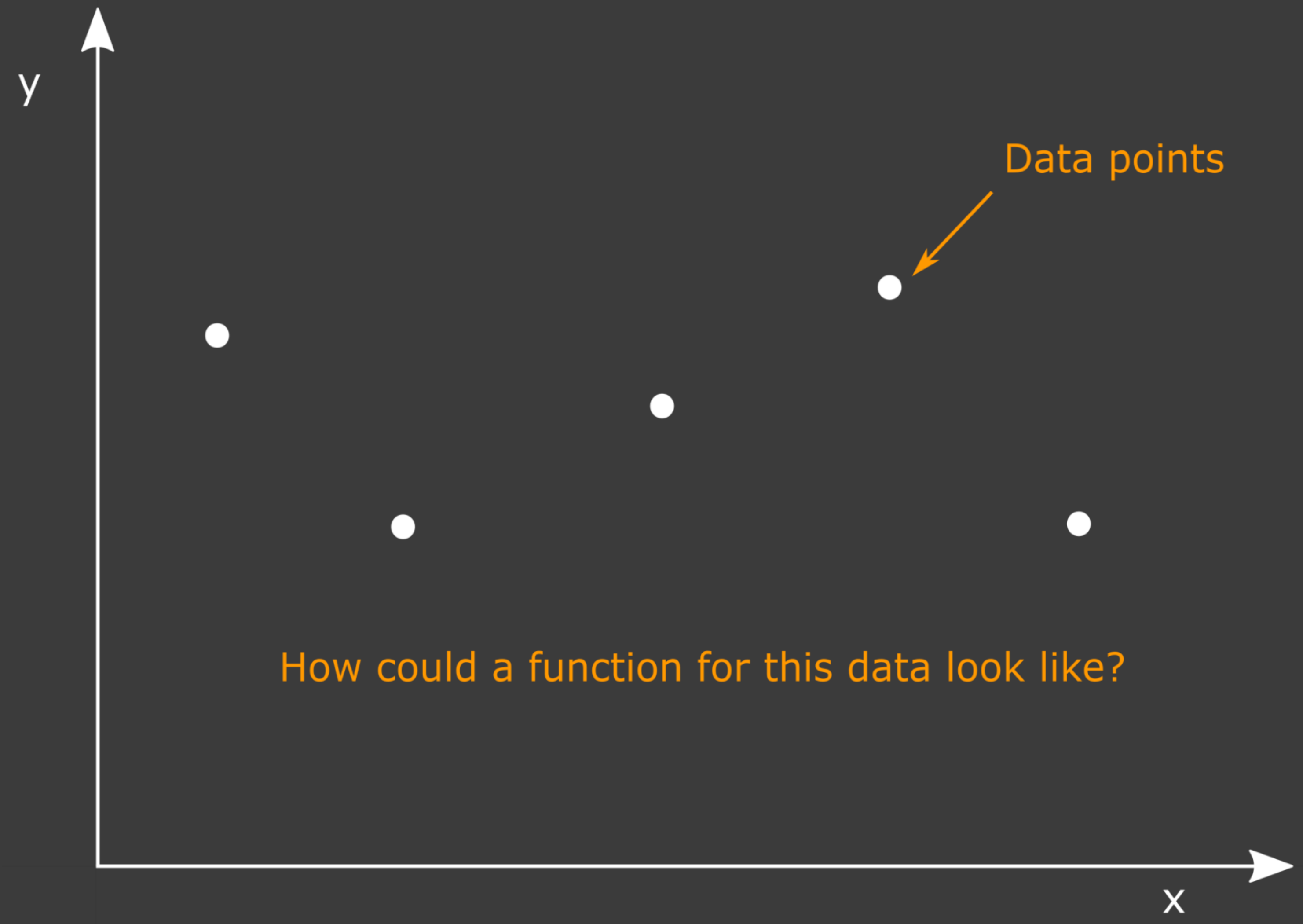
Hyperparameter tuning:

- ML experts (typically too complex for domain experts)
- Expensive trial & error on an HPC
- Cloud Auto-ML

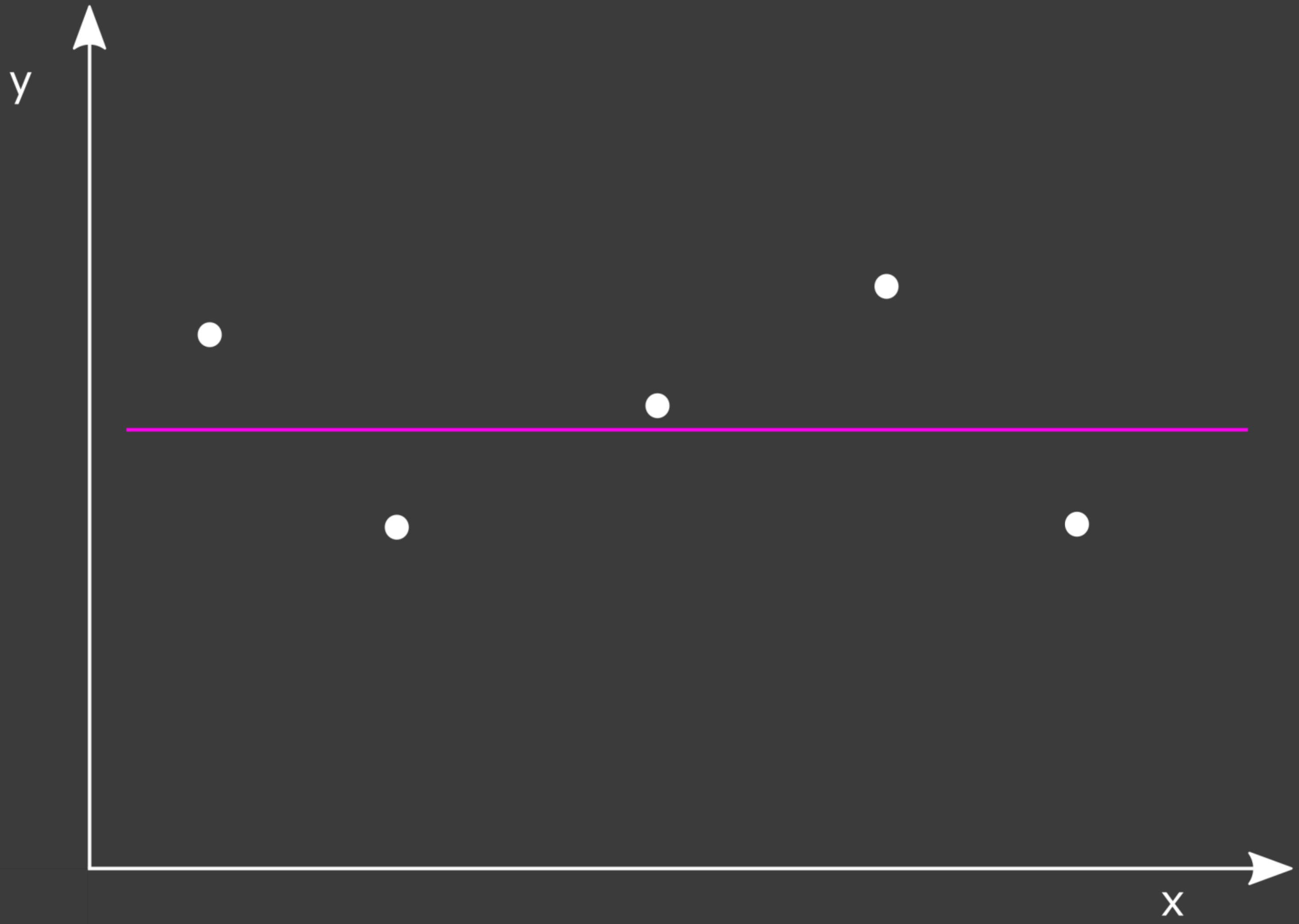


Hyperparameter	Approximate sensitivity
Learning rate	High
Optimizer choice	Low
Other optimizer params (e.g., Adam beta1)	Low
Batch size	Low
Weight initialization	Medium
Loss function	High
Model depth	Medium
Layer size	High
Layer params (e.g., kernel size)	Medium
Weight of regularization	Medium
Nonlinearity	Low

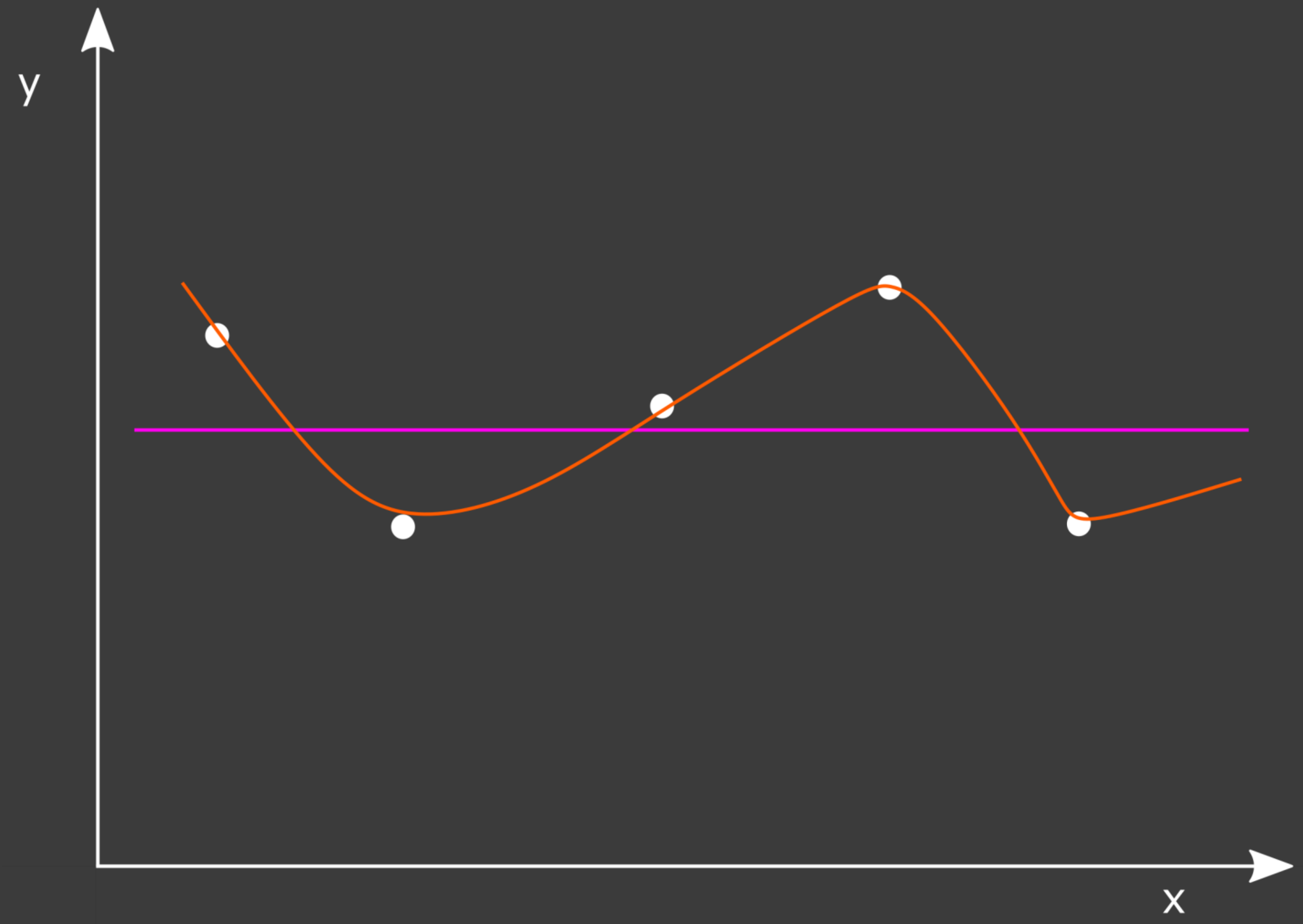
Gaussian process



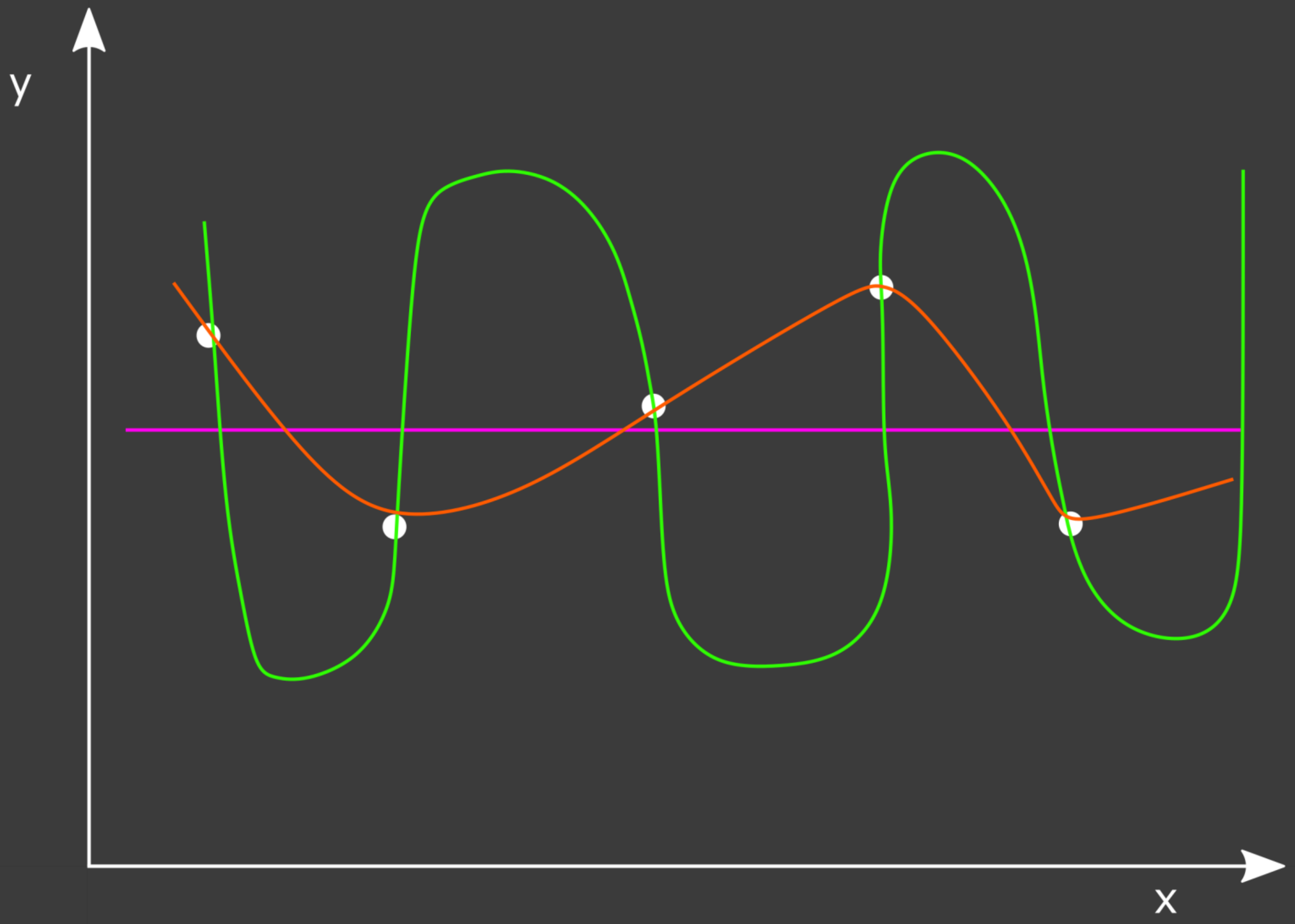
Gaussian process



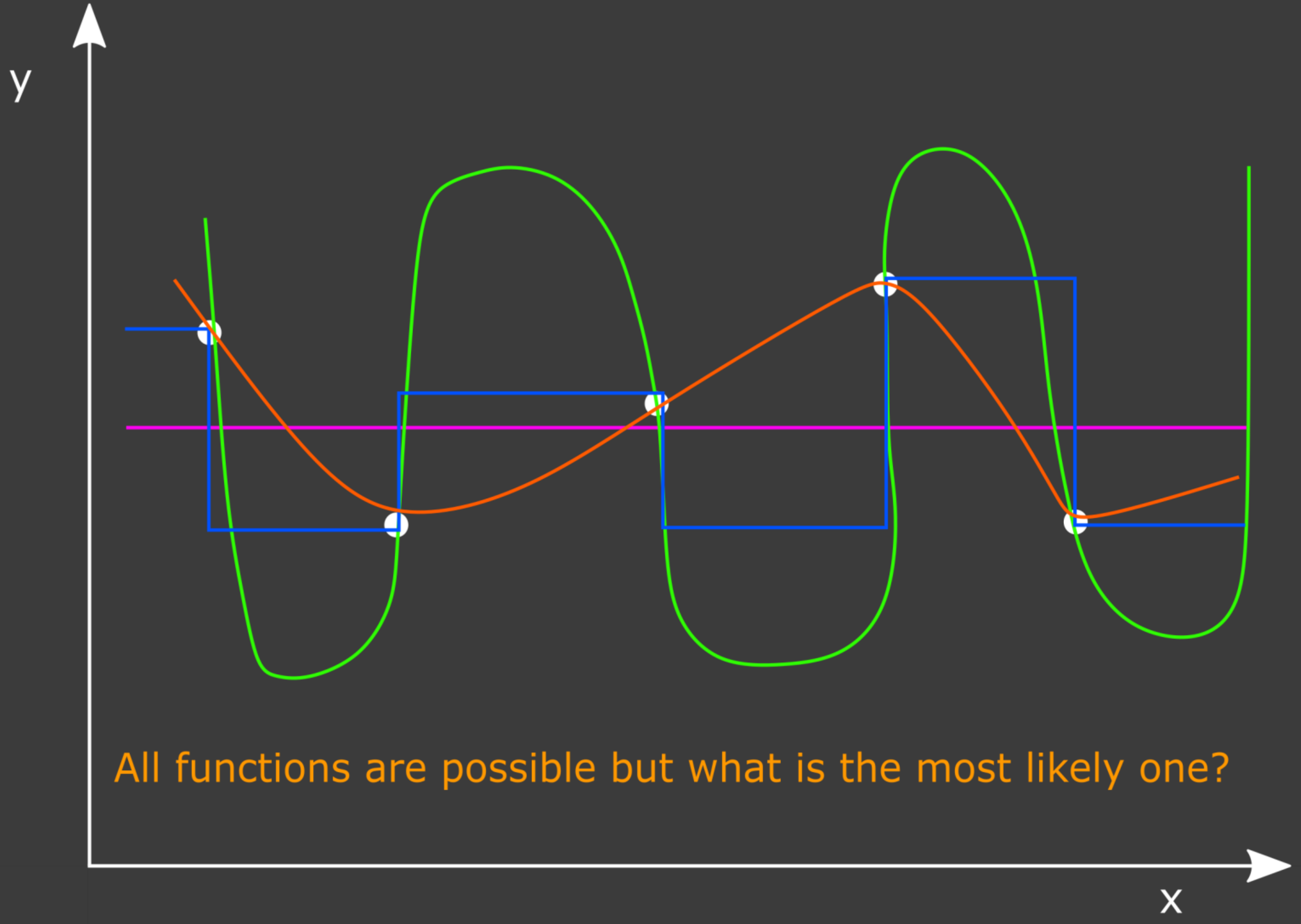
Gaussian process



Gaussian process



Gaussian process

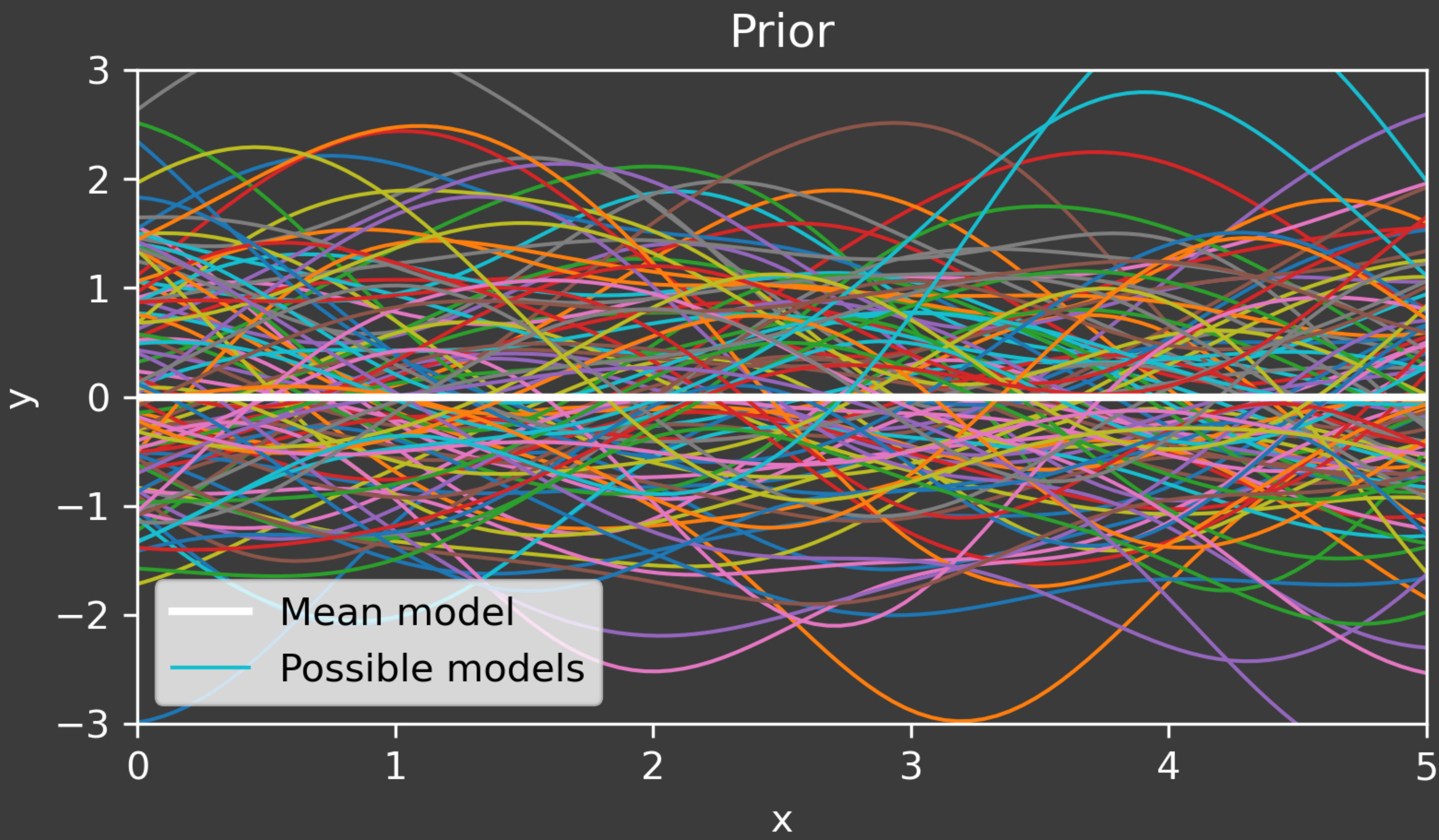


All functions are possible but what is the most likely one?

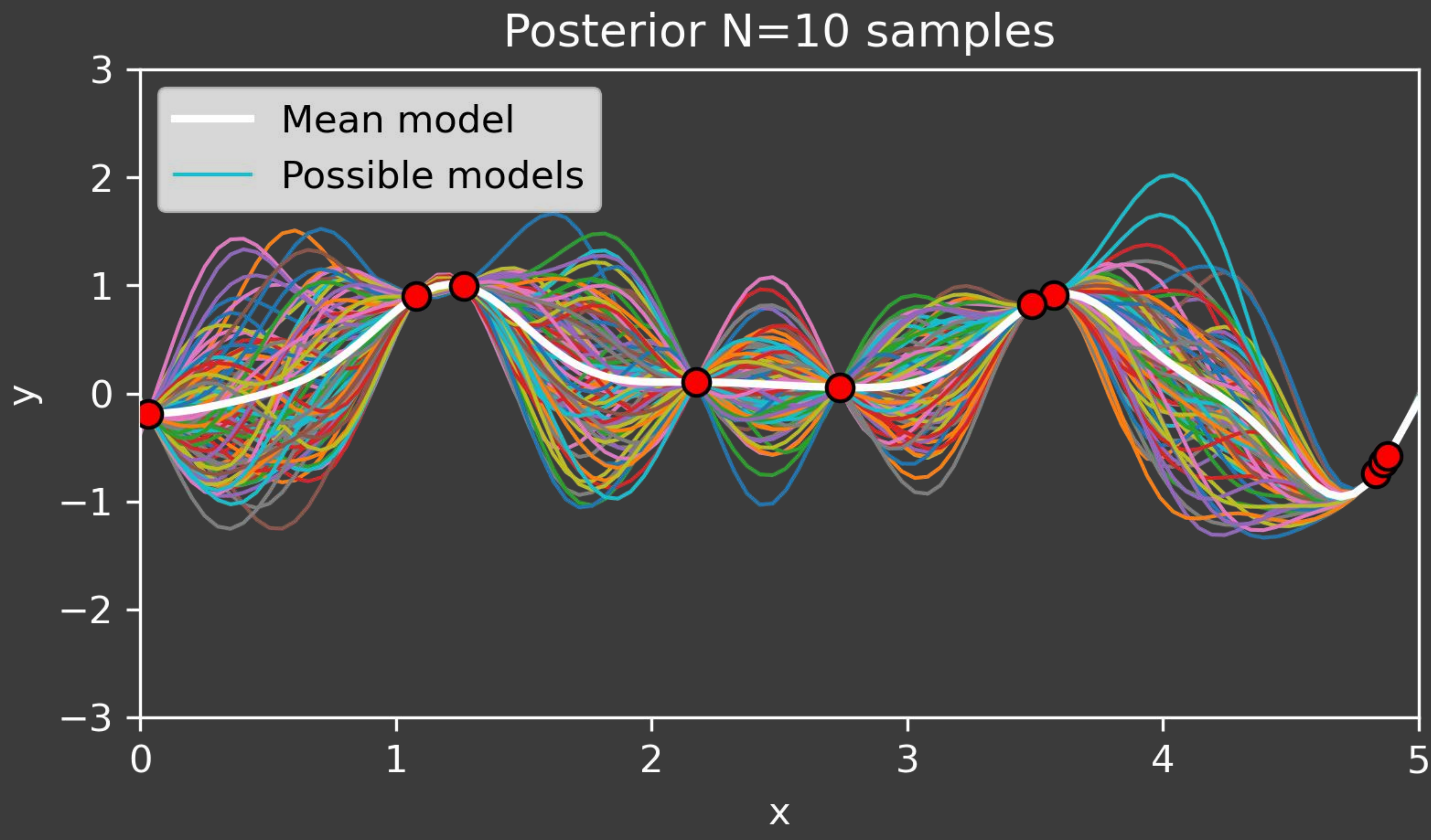
Gaussian process

Why not making a distribution over functions and choose the most likely one to describe our data?

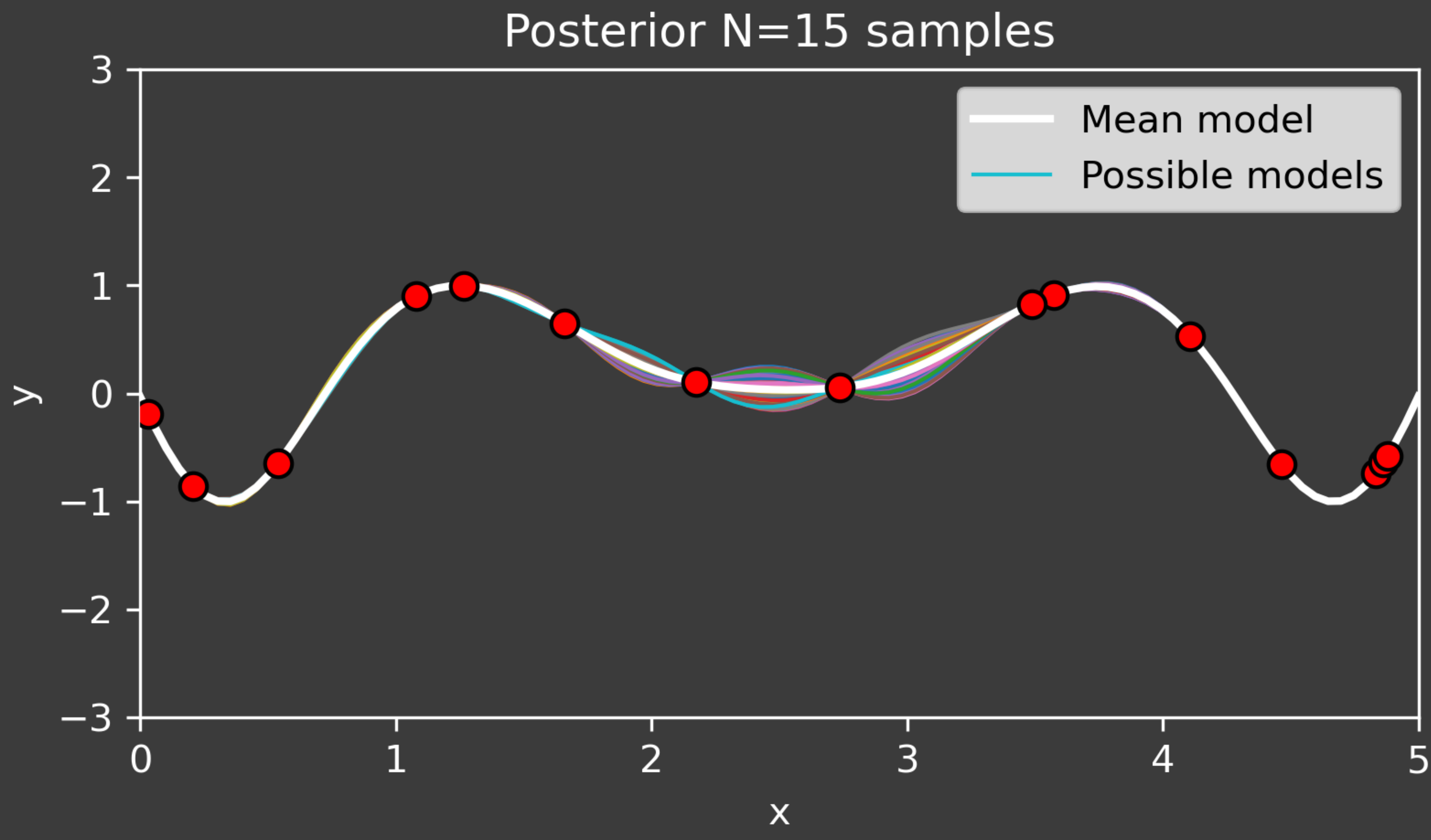
Gaussian process



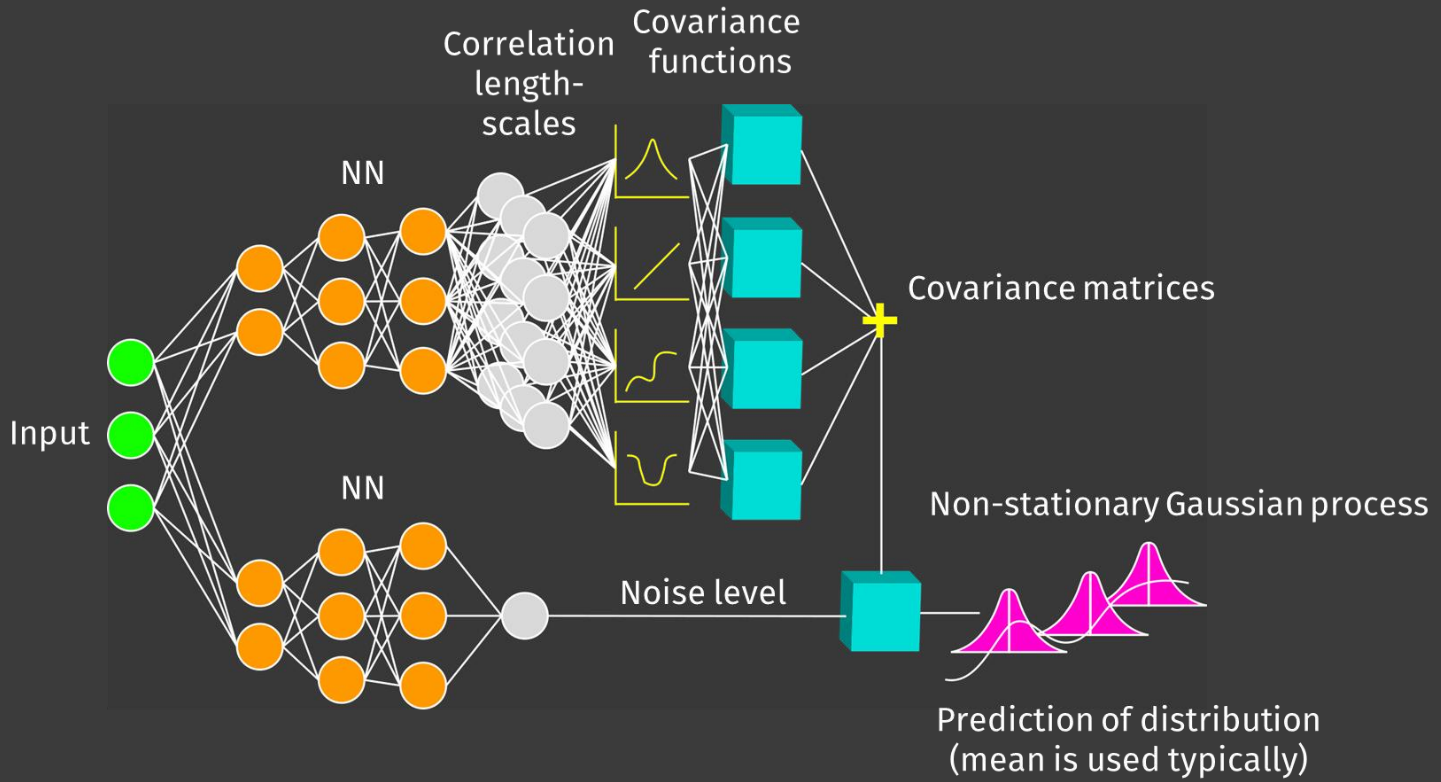
Gaussian process

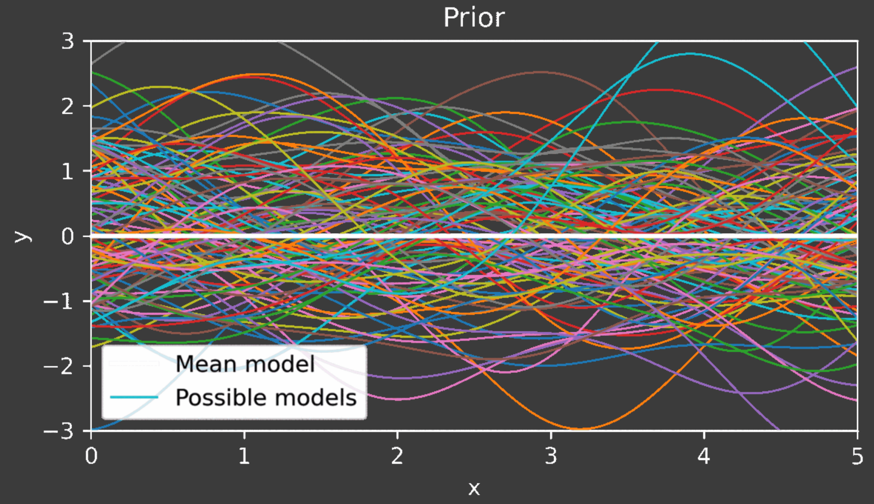


Gaussian process



Deep infinite mixture of Gaussian Processes (DIM-GP)





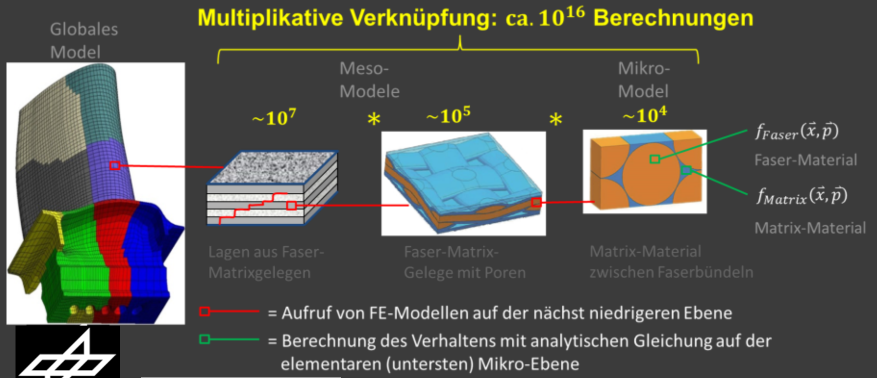
- **Non-stationary probabilistic model**
- **No settings (no expert knowledge)**
- **Can be used for various forms of data**
- **Requires little data for good results**
- **Automatic noise handling**
- **Low hardware requirements (no cloud, data remains with the customer)**

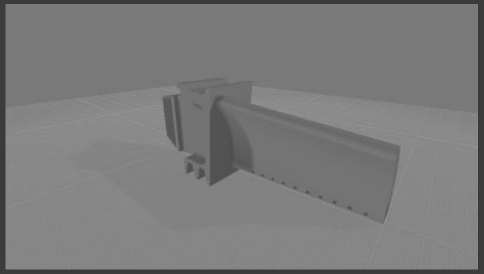
Unique combination of neural networks + Gaussian process

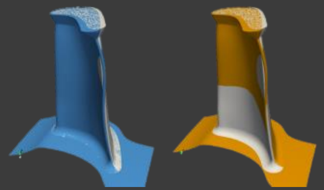
Usable data with DIM-GP

- Scalars, signals, fields, tensors, images, meshes can be used as input / partially as output:

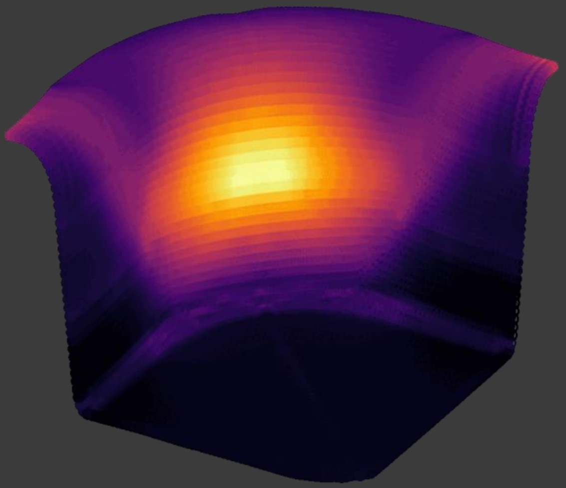
x1	x2	x3	x4	x5	x6	x7	y1	y2	y3
27	90	0.5	13.5	51	2	3.5	21.3277356	20759.878	29198.3713
27.6775	123.975	0.895	13.37	49.52	1.855	3.143	23.671277	26584.1086	44739.0519
31.7725	121.275	0.865	12.27	49.76	1.645	3.675	23.9304892	27406.0836	29920.9292
28.0025	115.425	0.255	12.93	48.32	1.505	4.095	27.7219304	21129.948	28484.1233
28.8475	125.775	0.735	12.39	48.16	1.635	4.151	24.0487891	23760.1779	26910.5454
31.9675	113.175	0.995	12.03	45.68	2.275	3.899	25.4821736	26816.4716	28359.8146
29.0425	129.825	0.625	12.05	50.56	1.665	3.913	23.1483514	25908.9534	29674.944
28.5875	110.925	0.585	13.49	49.04	2.305	3.717	26.3814237	20551.2131	40532.4755
29.3675	128.025	0.135	12.99	51.52	2.145	3.535	22.7670509	23111.2386	39604.8048
27.0275	97.425	0.935	13.17	50.48	1.525	3.073	22.9899555	26302.2032	30536.5236
26.2475	92.475	0.915	13.57	48.96	1.875	3.605	24.1296008	19847.4432	27214.5355
31.6425	120.375	0.375	13.45	46.96	1.605	3.227	24.2098935	25843.3629	46968.0151
29.1725	132.075	0.115	12.17	47.44	1.905	2.835	23.2732693	36018.1975	40974.4874

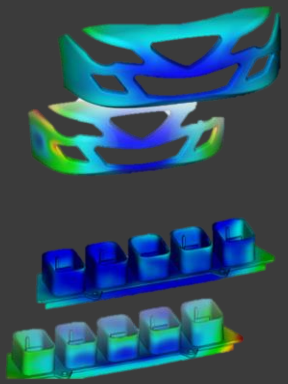


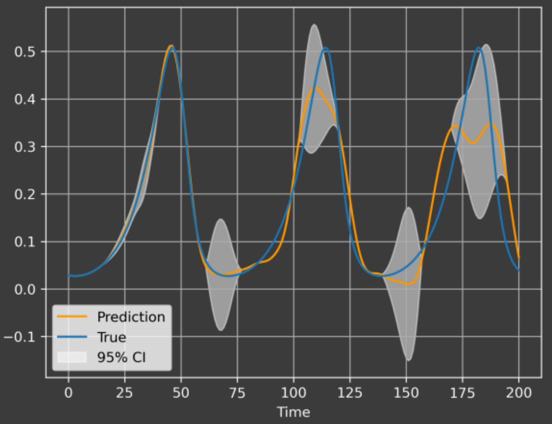


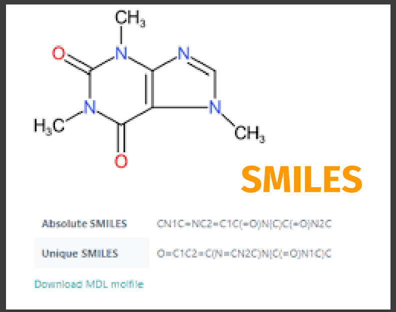


Live FEM & CFD

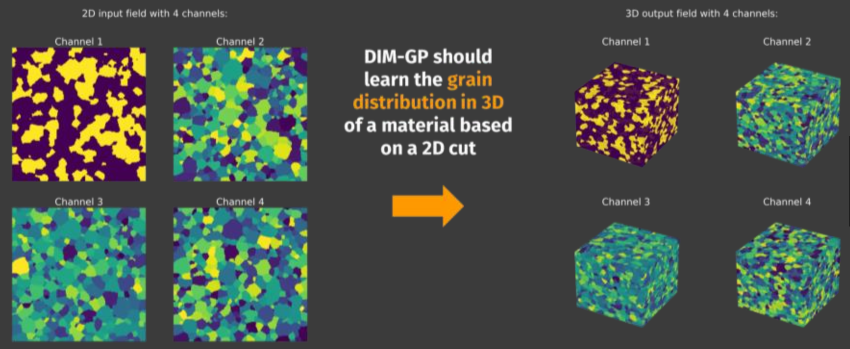








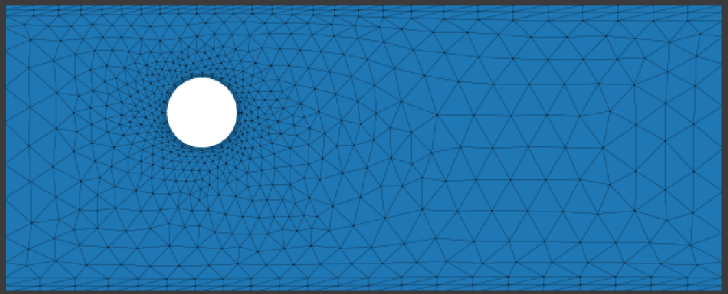
Molecule information



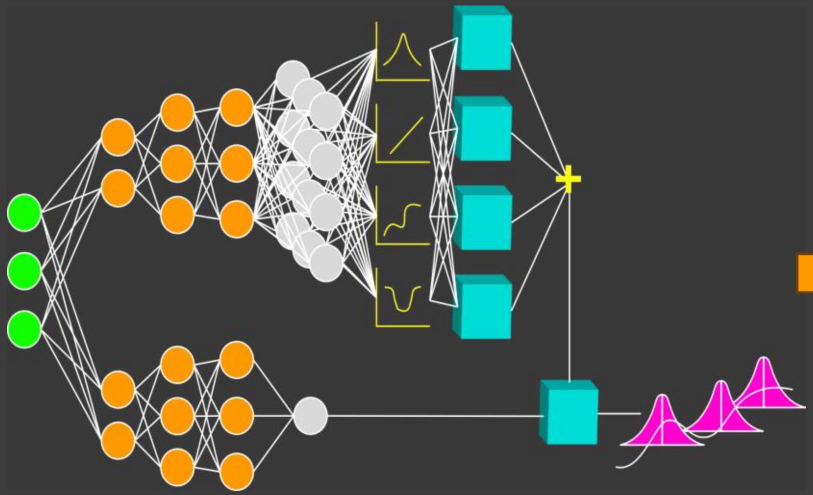
Geometric Deep Infinite Mixture of Gaussian Processes

Input

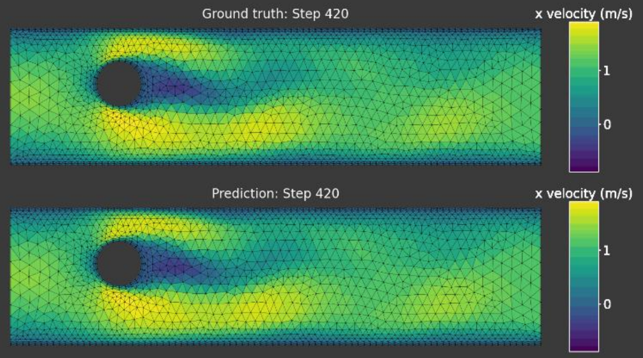
Mesh node positions + initial node features / boundary conditions (e.g. stress, velocity, …) + optional global features



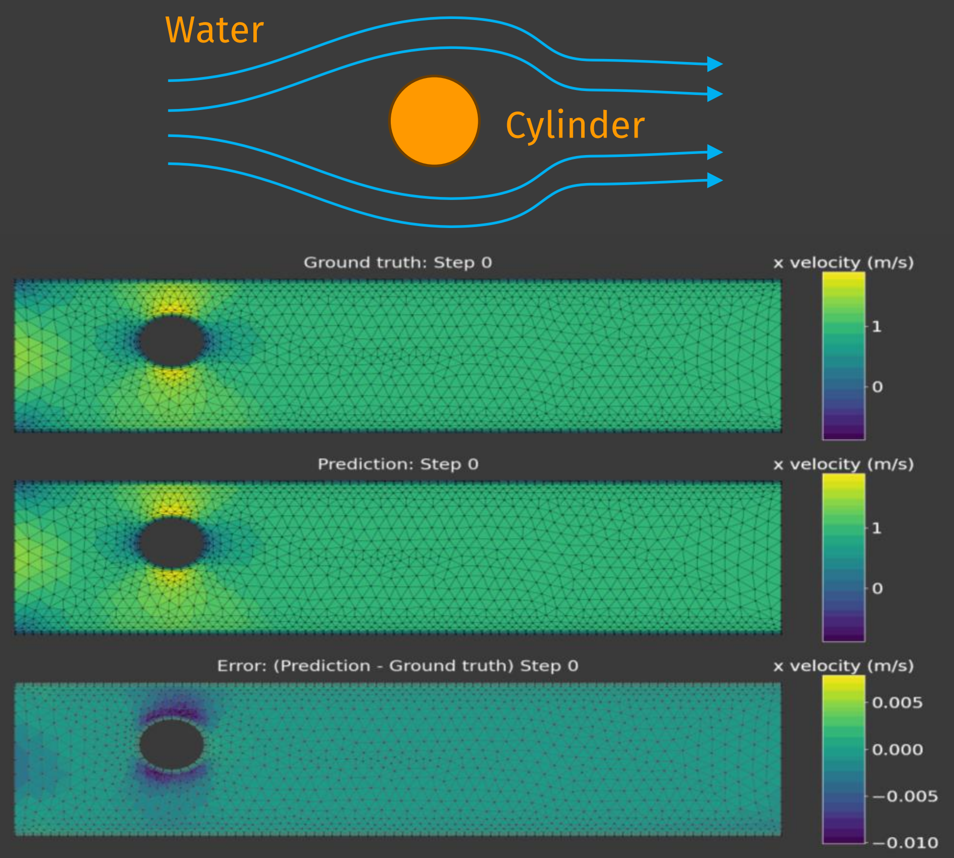
Geometric DIM-GP

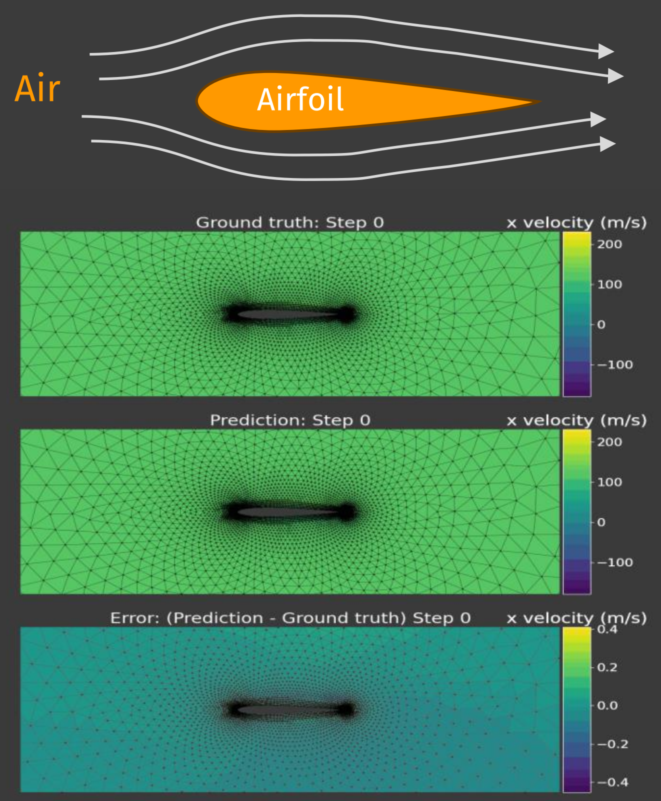


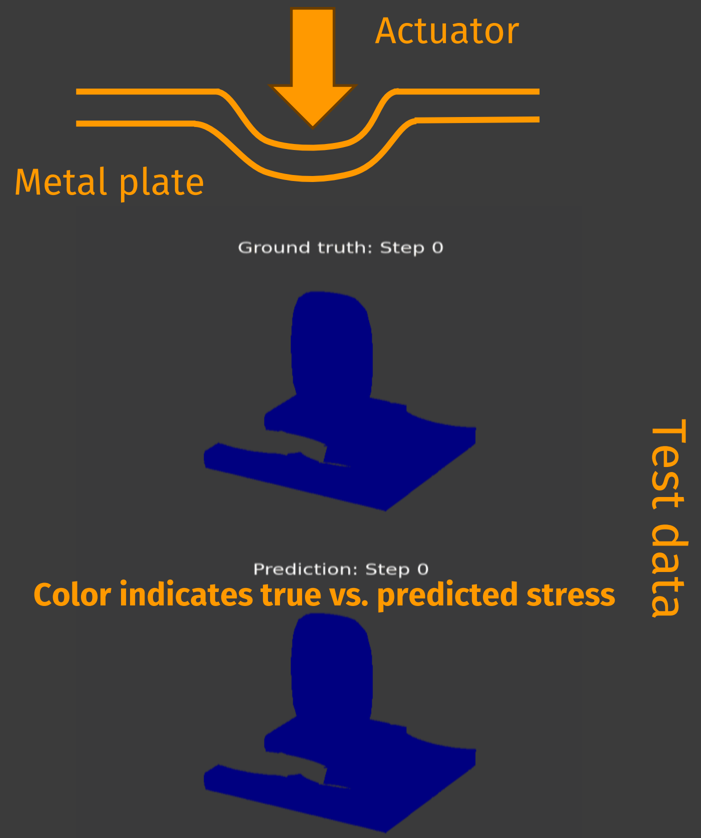
ML-based predictions of (transient) FEM / CFD results



2D / 3D transient FEM / CFD







- 9,867 training steps = 1 Epoch on 5 samples
- 1 NVIDIA 4090 GPU training time 5 minutes
- 1 CPU (8 cores) training time 12 minutes
- RMSE 1-step prediction: 1.54 x 10e-3

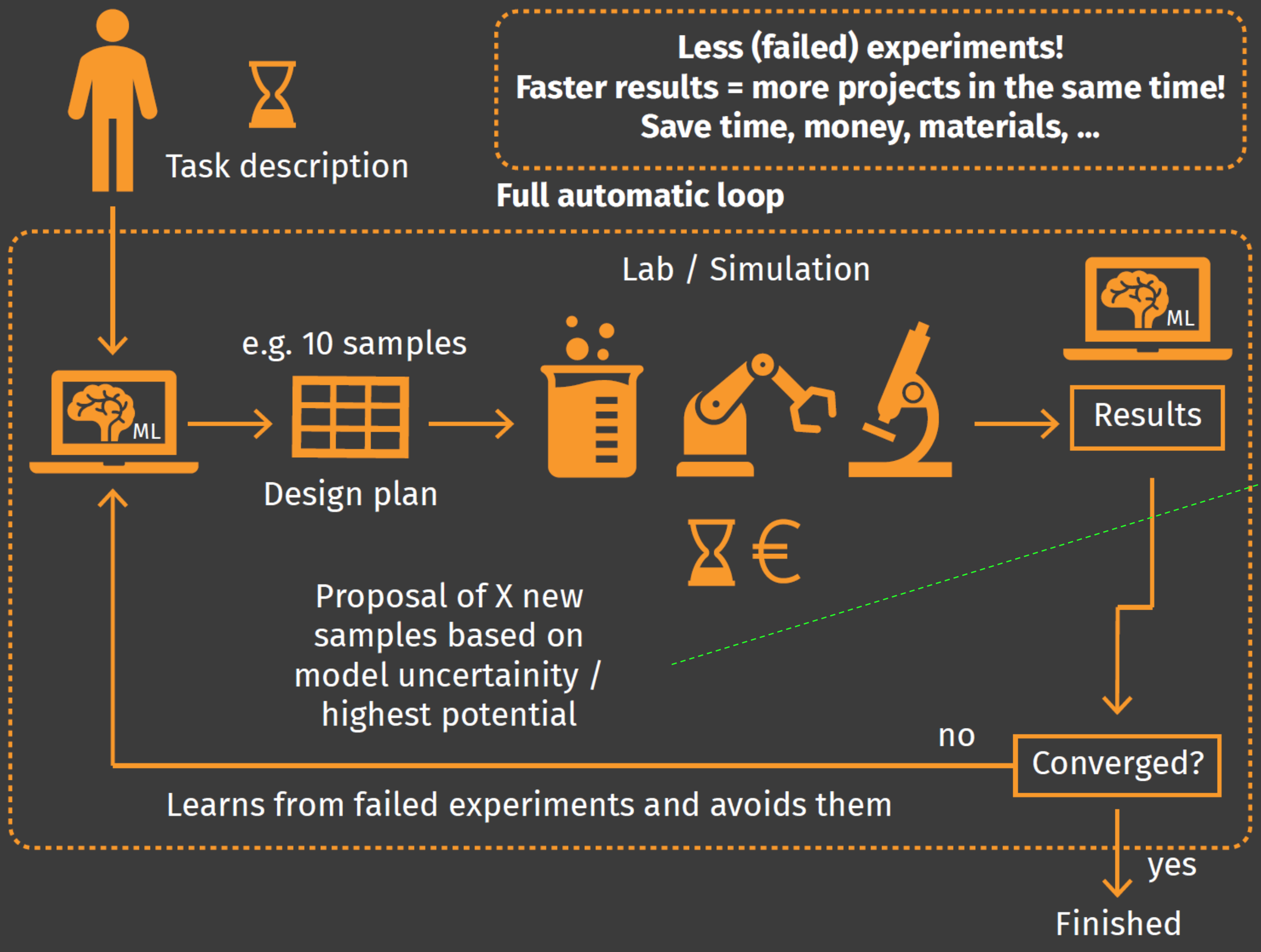
- 30,606 training steps = 1 Epoch on 5 samples
- 1 NVIDIA 4090 GPU training time 14 minutes
- 1 CPU (8 cores) training time 32 minutes
- RMSE 1-step prediction: 0.05

- 3,035 training steps = 1 Epoch on 5 samples
- 1 NVIDIA 4090 GPU training time 2 minutes
- 1 CPU (8 cores) training time 7 minutes
- RMSE 1-step prediction: 0.55 x 10e-4

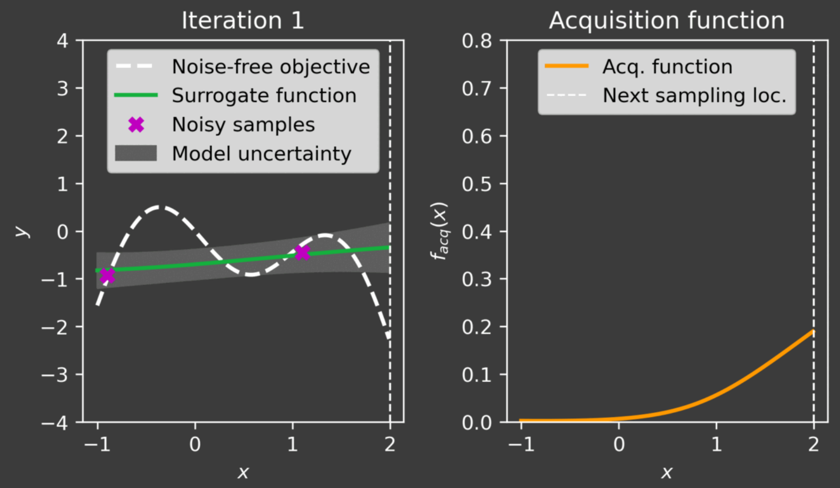
Adaptive optimization & DoE

Efficient adaptive optimization / design of experiment

Adaptive search of next optimum



Single objective: search maximum of y



Total 5 adaptation with 3 formulations = 15 formulations

Final adaptation

Multi-fidelity modeling & optimization

What is multi-fidelity data?

Fidelity spectrum

Low-fidelity models ⟷ **High-fidelity models**

- **Coarse physical resolutions**
- **Fast runtime**
- **Low cost**

- **Fine physical resolution**
- **Slow runtime**
- **High cost**

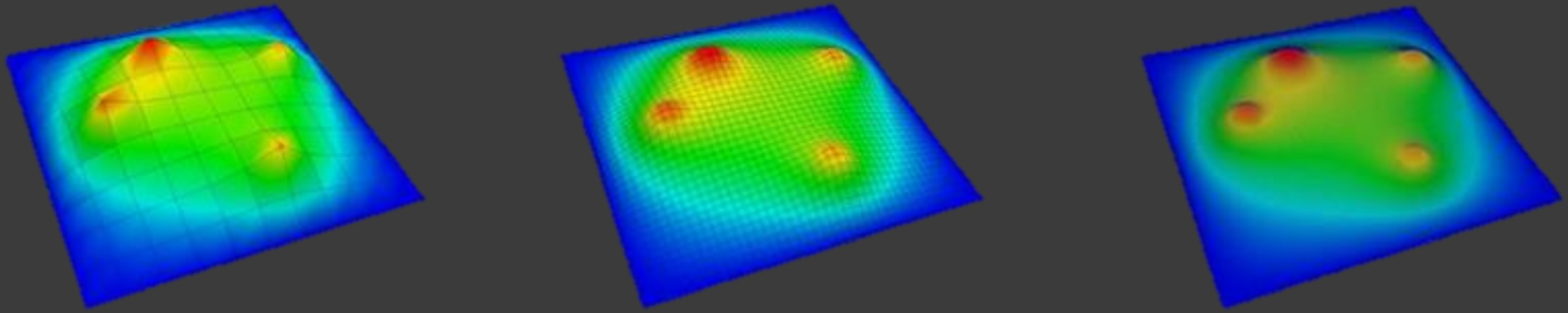
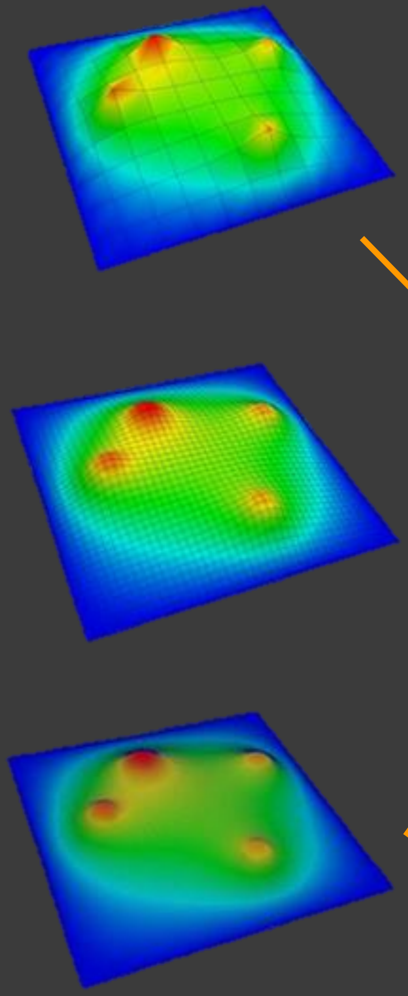
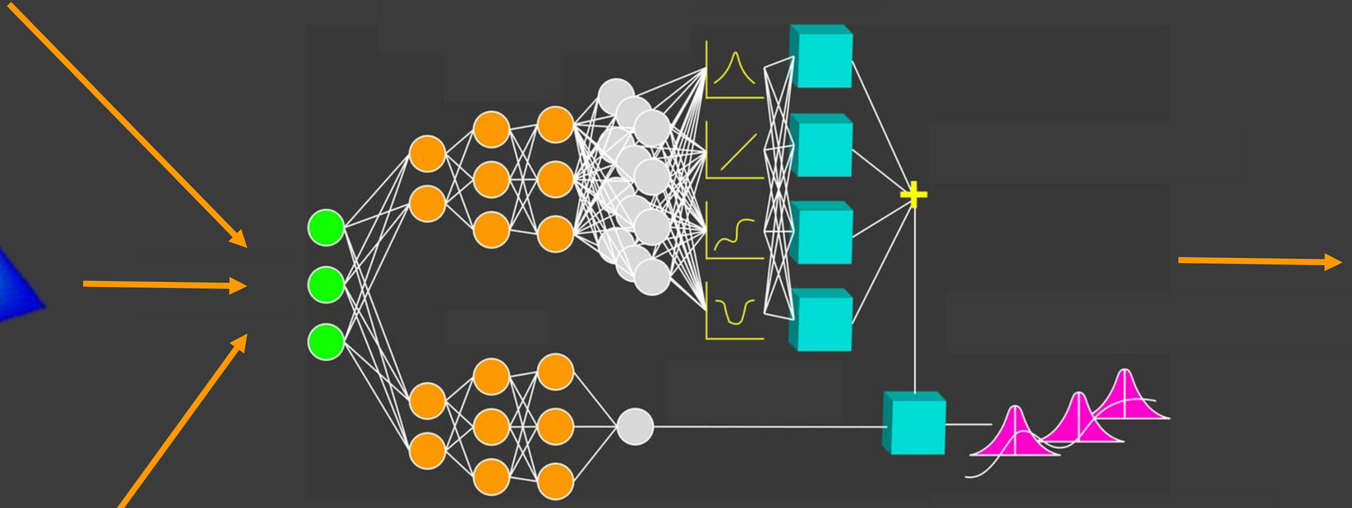


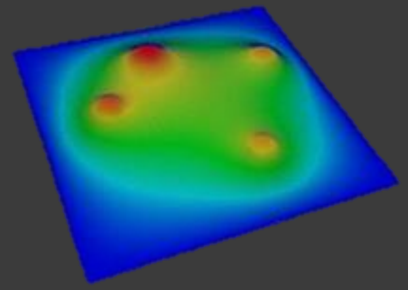
Image source: Aydin, Roland Can, Fabian Albert Braeu, and Christian Johannes Cyron. "General multi-fidelity framework for training artificial neural networks with computational models." *Frontiers in Materials* 6 (2019): 61.

What is multi-fidelity modeling?



Model utilizes information from all fidelity sources and predicts in the highest fidelity quality





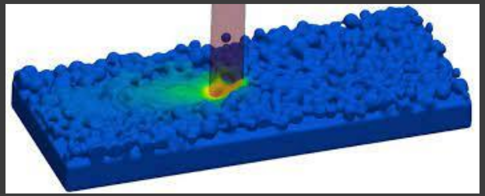
Advantages:
- **Less expensive data points required**
- **Calibration between simulation and real experiment**

What is multi-fidelity modeling?

Real experimental data

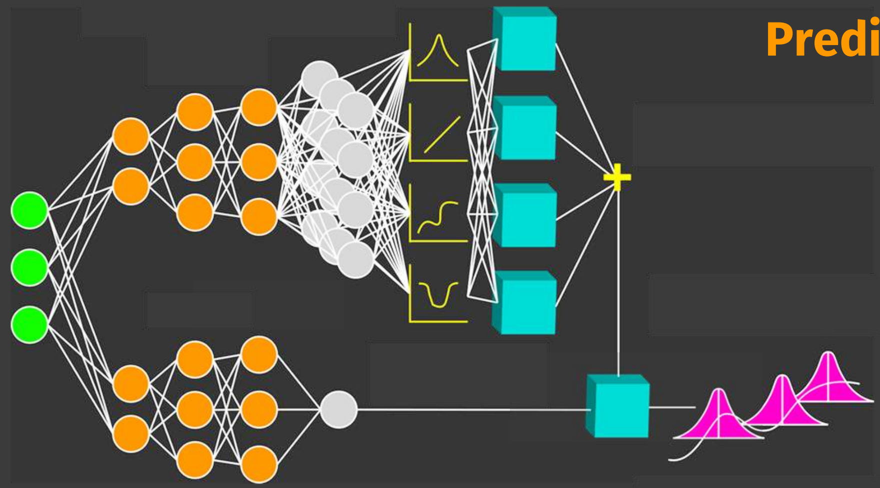
© Fraunhofer ILT

Simulation data



© Fraunhofer IWM

Model utilizes information from all fidelity sources and predicts in the highest fidelity quality

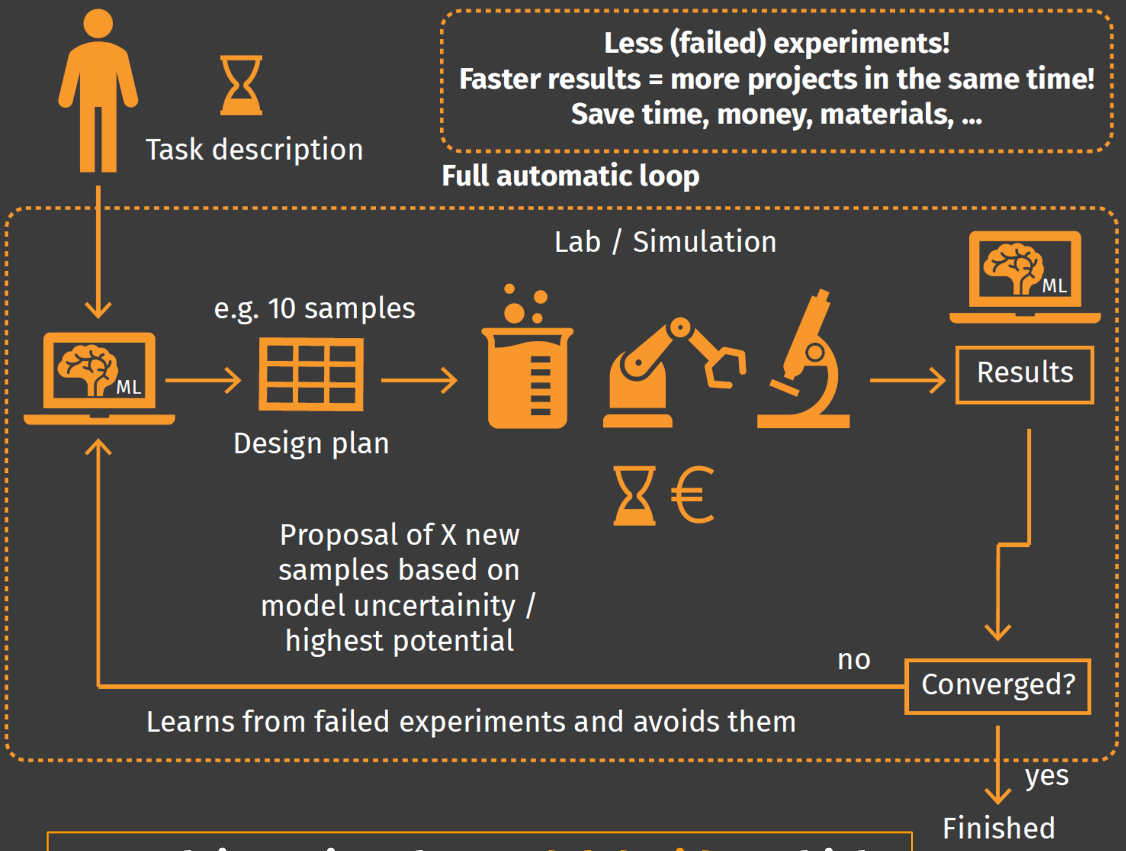


Prediction in quality of real experiment

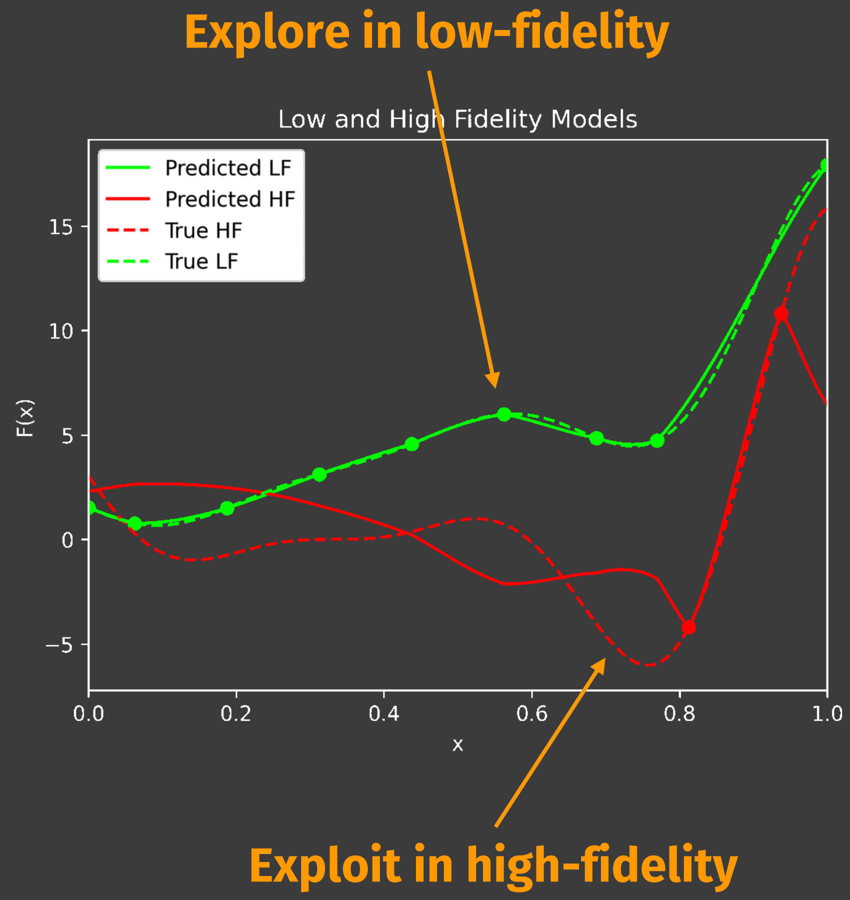
© Fraunhofer ILT

Advantages:
- **Less expensive data points required**
- **Calibration between simulation and real experiment**

Multi-fidelity optimization



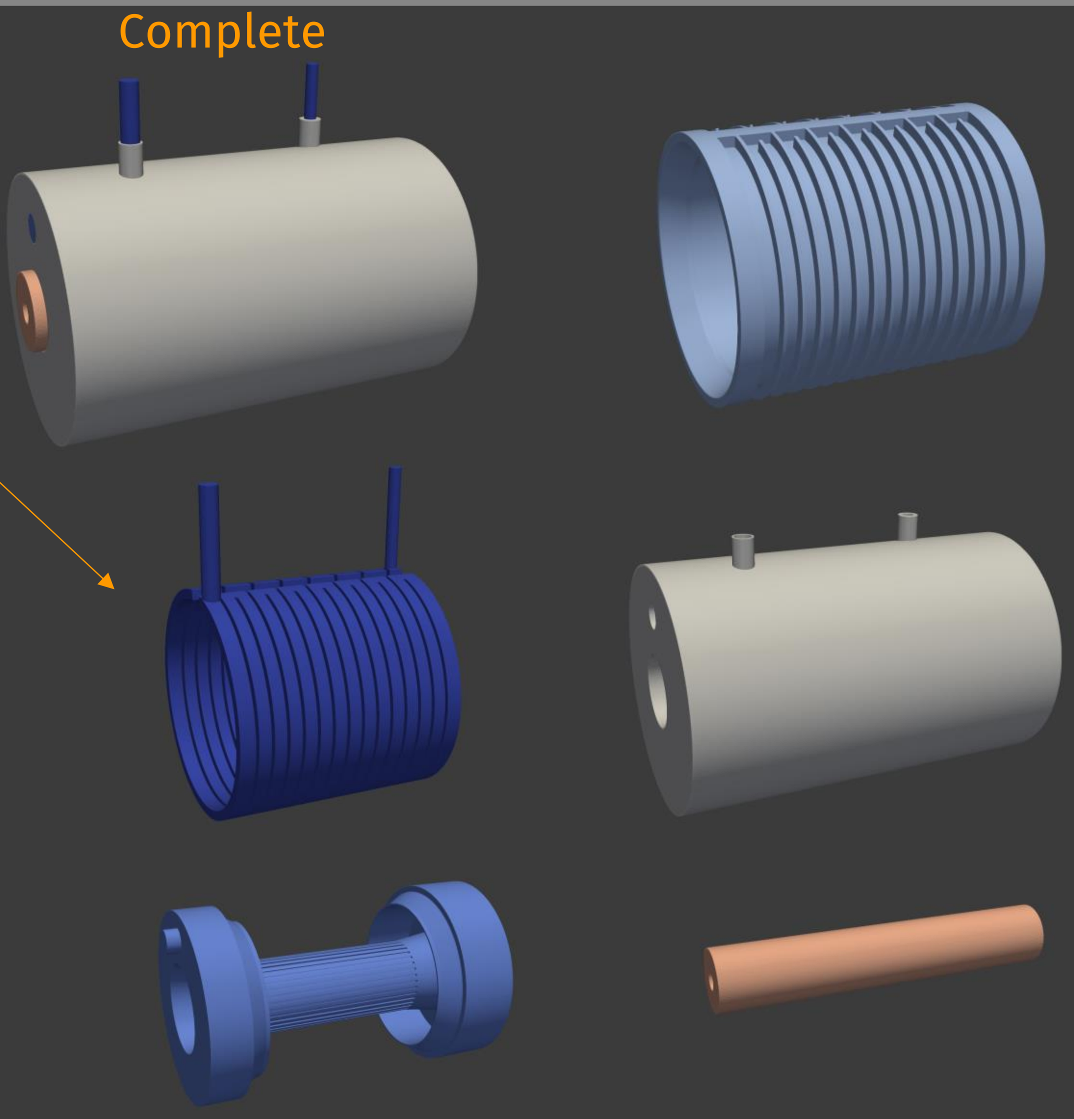
In each iteration the model decides which fidelity level is needed based on user specified costs

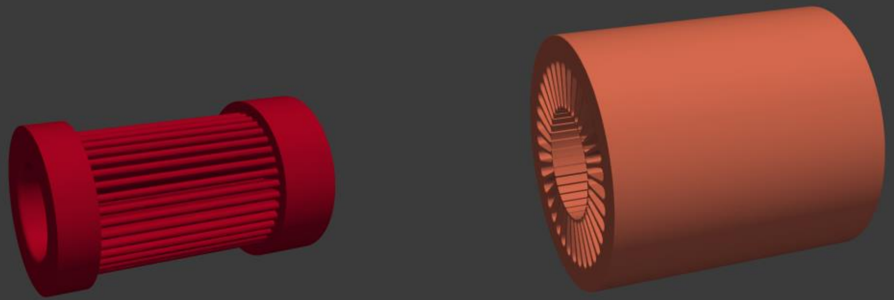


Applications

E-Motor Cooling

- Temperature prediction for a E-Motor based on different cooling channel geometries
- 5,366,013 nodes / 2,817,502 elements
- 34 training samples / 4 test samples
- Field input parameters:
  - Node positions (only the non-parameterized geometry)
- Field output parameters:
  - Temperature
- Training time approx. 5-6 h on a single (cheap) GPU

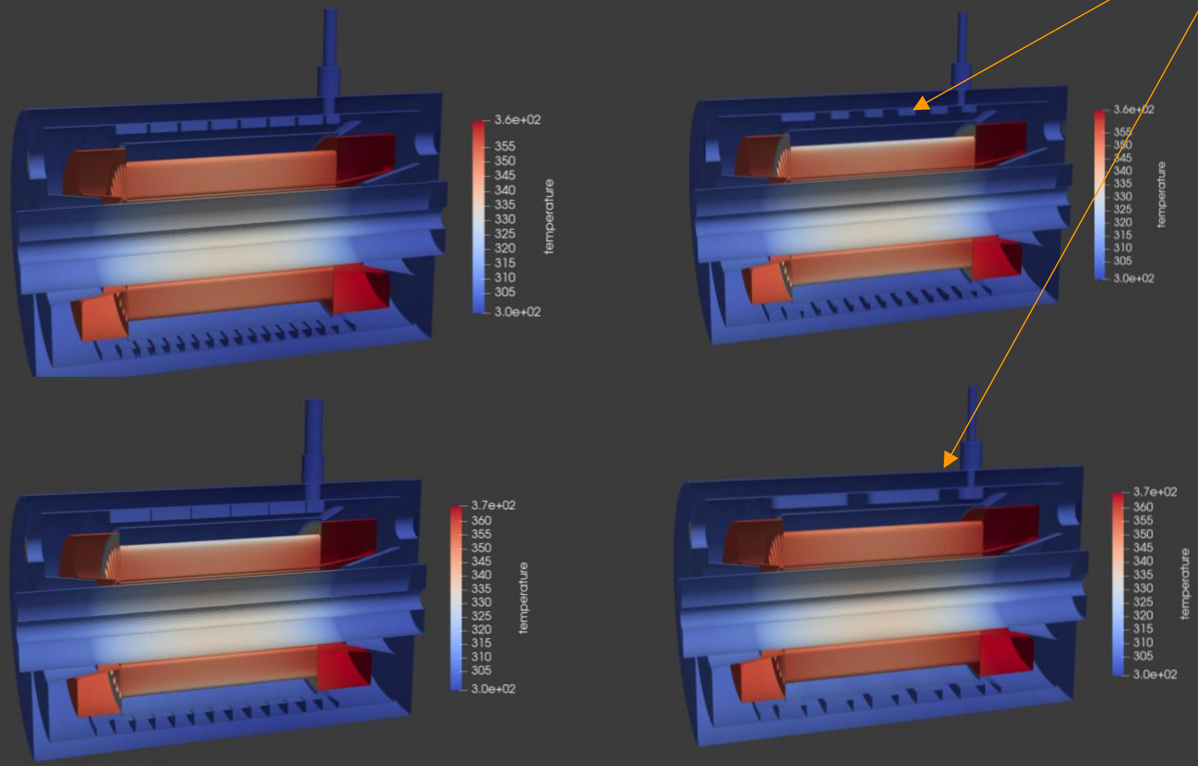




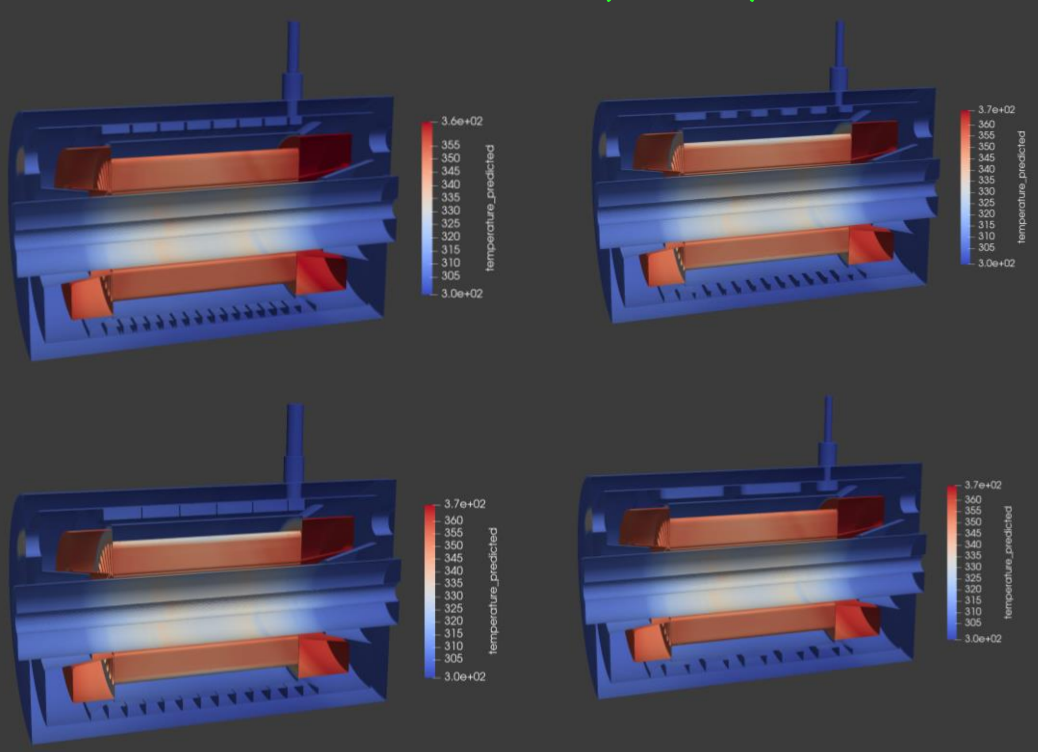
E-Motor Cooling

Test designs (new cooling channel geometries)

Simulation (6-10h)



Prediction (30 sec)



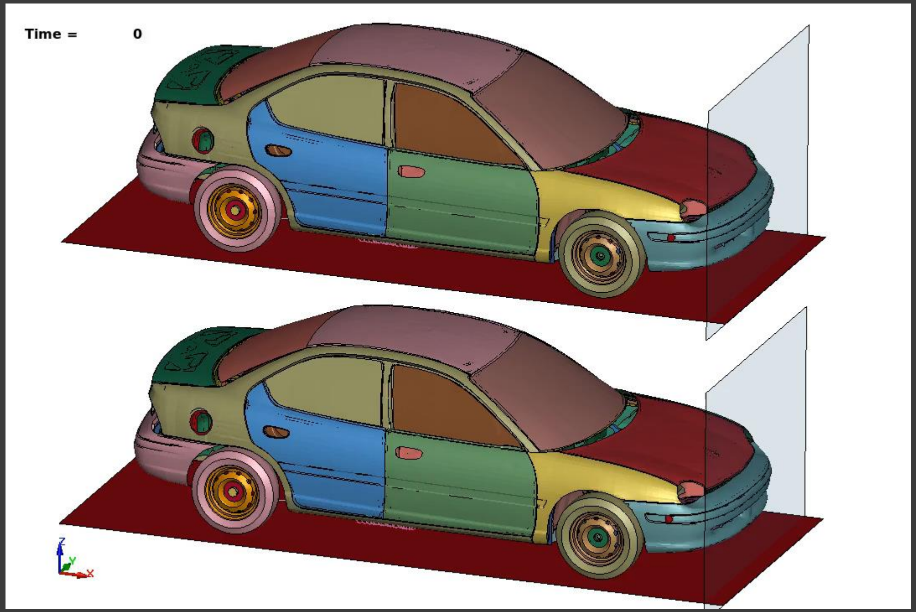
Field	Abs. Error		Rel. Abs. Error [%]		R2	
	Mean	Std	Mean	Std	Mean	Std
Temperature	1.395	0.312	0.419	0.09	0.993	0.002

Mean over all nodes and all test designs

Crash-test

- 1 simulation consists of 50 time steps, 283,791 nodes per time step → 14,189,550 nodes per simulation
- Training data: 32 simulation / Test data: 5 simulation
- Training time of DIM-GP: 21 seconds
- Changing input: thickness of the shells

DIM-GP Prediction



17.5 s (rolling prediction)
or < 1 s if training done with time step as input

Crash simulation

0.5 h (20 cores CPU)

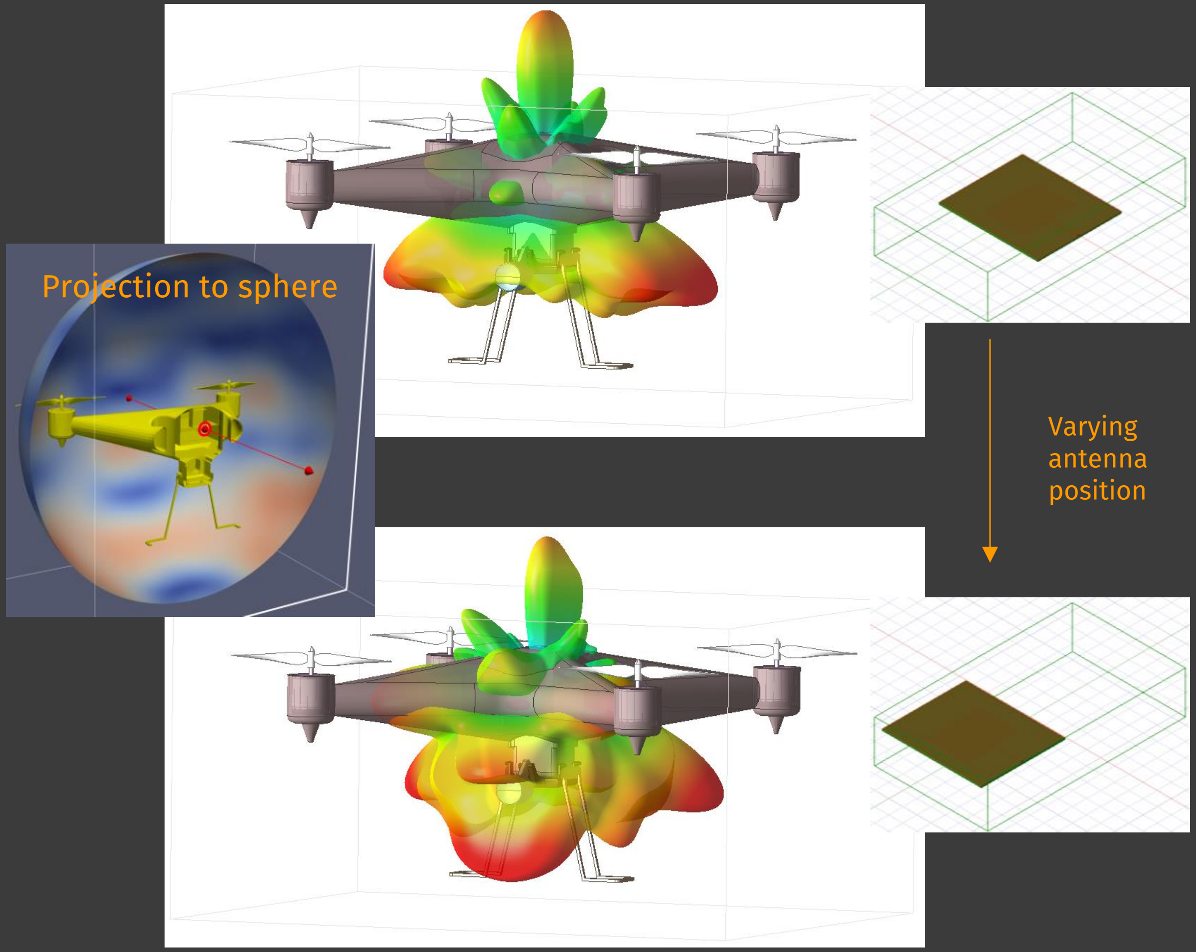
Mean absolute percentage error over all 5 test designs over all 50 time steps over all nodes: 5,73%

Drone Antenna

- 45 design variants
  - 37 for training / 4 for test / 4 validation (not used by DIM-GP)
  - 65,808 nodes / 66,464 elements

- Field input parameters:
  - Node positions, normals

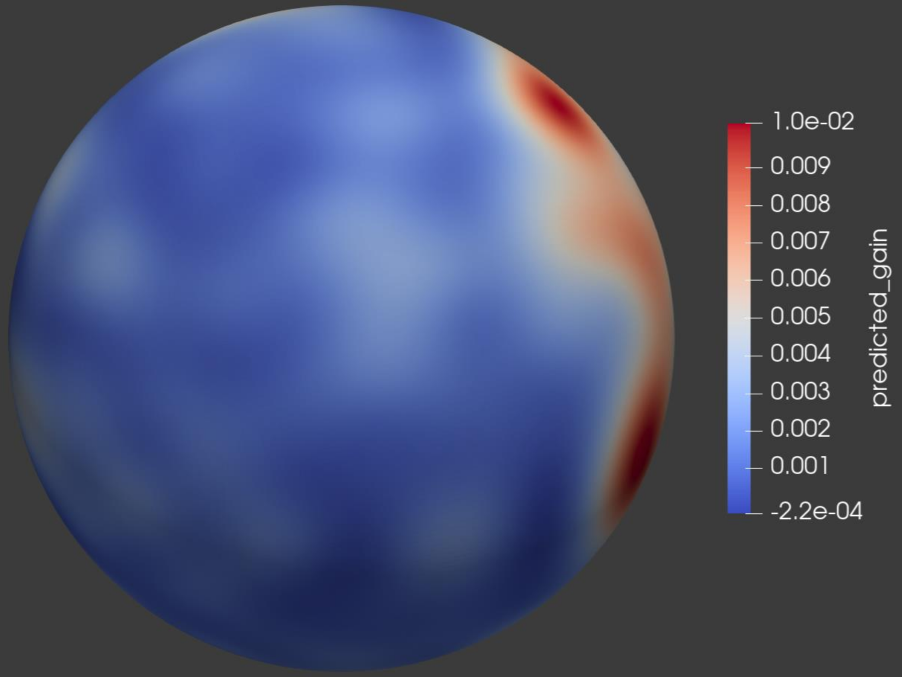
- Field output parameters:
  - Radiation pattern (rEtotal)
  - Realized gain



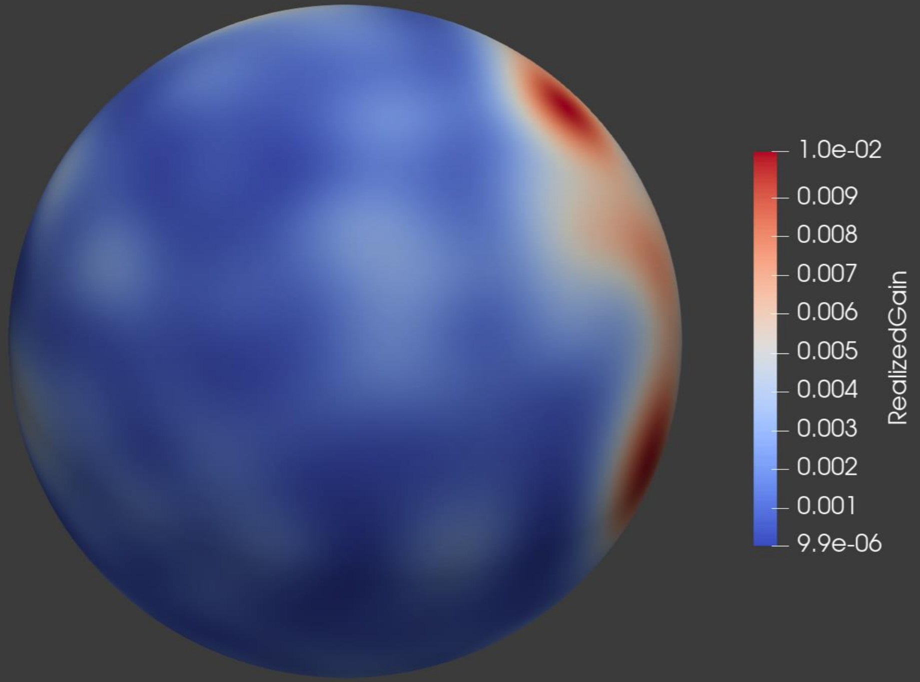
Drone Antenna

Field	Training time [sec]	Mean Abs. Error	Mean R2
RealizedGain	6.6 (GPU) / 10 (CPU) Nvidia 4090	0.023	0.962
rEtotal		0.00027	0.969

Prediction



Simultion



Multiscale simulation (ongoing research project with DLR & SGL Carbon)

- Macro scale FEM:
  - Deformation
  - Temperature
- Micro scale FEM:
  - Material tensor
- Model learns to predict the material tensor based on temperature from macro scale
- Macro and micro scale should be replaced by ML model to speed up calculations
- Trained model on cube

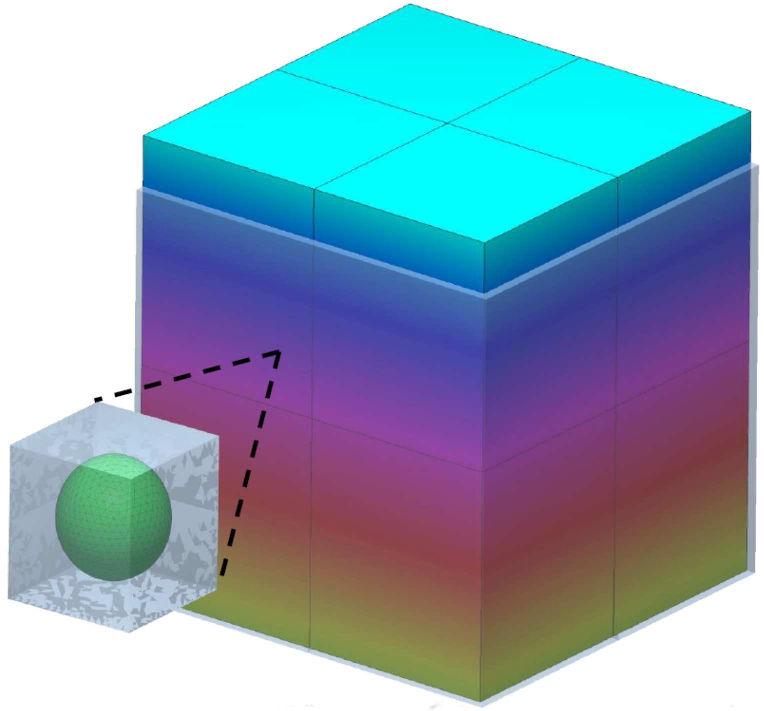
AP2.1 Parametrische FE-Modelle auf allen Ebenen
Beispielsimulation Input KI-Modell

DLR

Materialtensor

- Als Datenbasis wird der Materialtensor (Output) gewählt, welche von der Temperatur (Input) abhängig ist.

$$\overline{\mathbb{C}}_2^{238} = \begin{bmatrix} 169524 & 76592.4 & 77392.1 & -20.0479 & -6.02169 & -1.5958 \\ 76592.4 & 169510 & 77411.8 & -12.471 & 5.51499 & 8.43605 \\ 77392.1 & 77411.8 & 165837 & 15.0372 & 26.7541 & -1.16823 \\ -20.0479 & -12.471 & 15.0372 & 46464.4 & 13.7983 & 15.4344 \\ -6.02169 & 5.51499 & 26.7541 & 13.7983 & 44726.2 & -1.45661 \\ -1.5958 & 8.43605 & -1.16823 & 15.4344 & -1.45661 & 44701.2 \end{bmatrix}$$



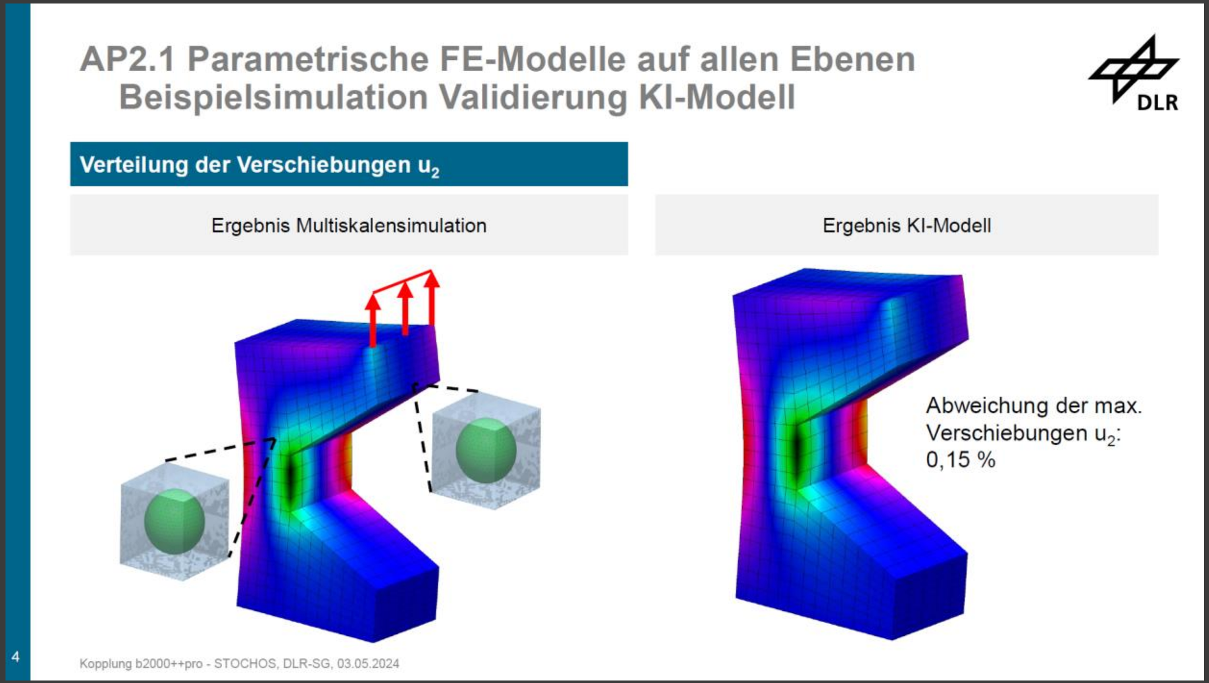
Simulationsergebnis Multiskalensimulation [DLR-SG]

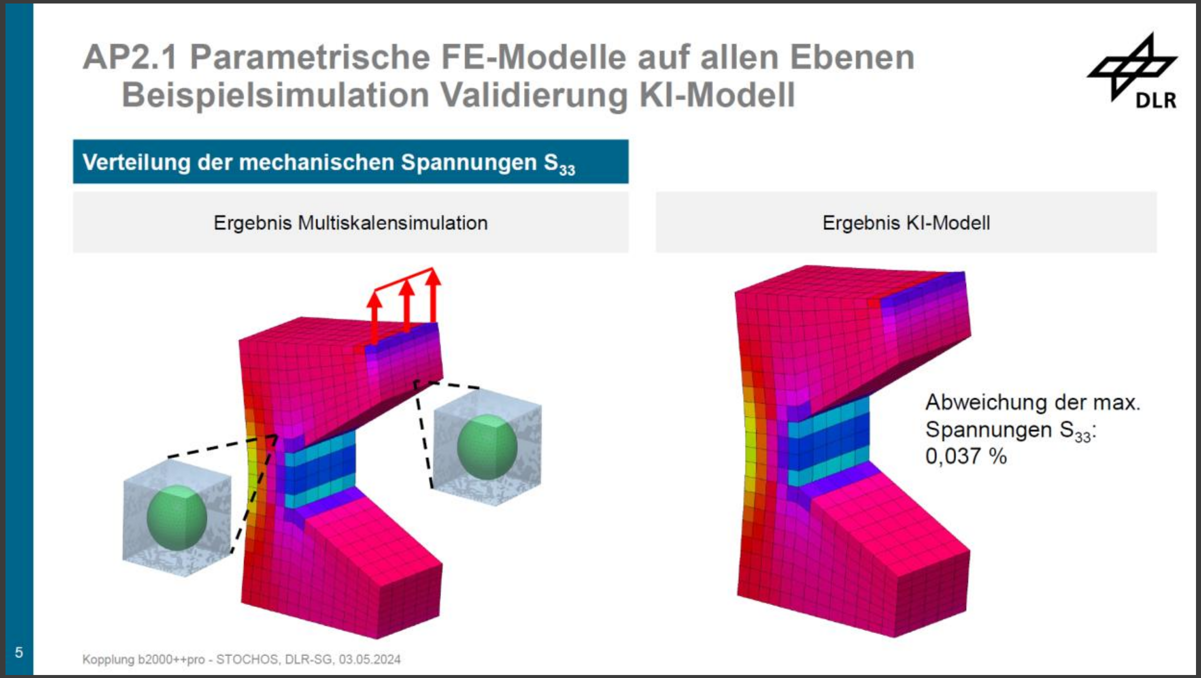
3

Kopplung b2000++pro - STOCHOS, DLR-SG, 29.04.2024

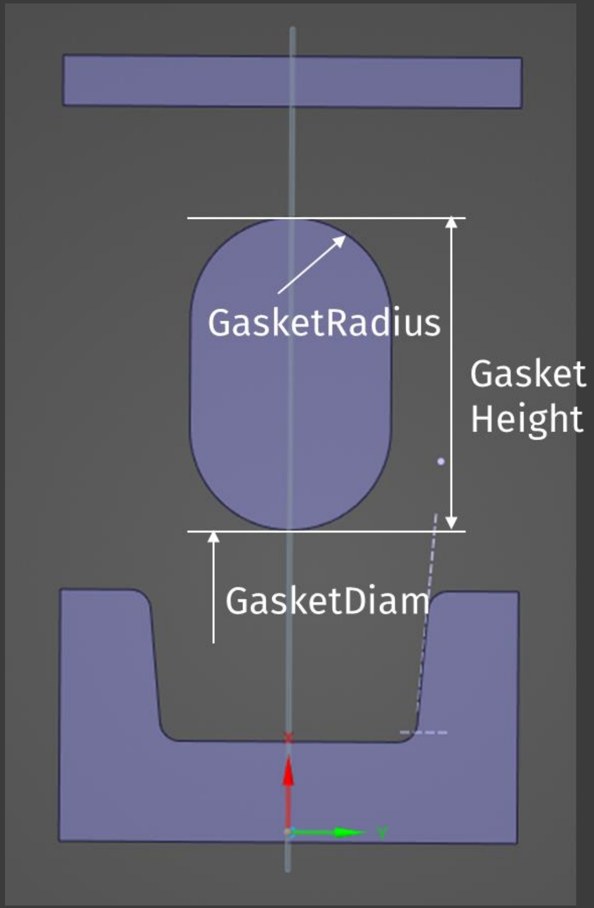
Multiscale simulation (ongoing research project with DLR & SGL Carbon)

- Trained model on cube has been used on a c-beam geometry





Gasket compression



B1_D15 ShaftLength (fixed)
initial_interference 0.1mm

Parameters:
P P14 / B1_D18 GasketRadius „R“ (fixed?)
P P2 / B1_D14 GrooveAngel: +/-20°
P P17 / B1_D12 GrooveWidthScale: >1.2..2.0
P P7 / B2_D3 GasketInnerdiameter >10*R?
P P22 / Shaft2WallClearance (fix?)
P P18 / GasketSlenderness (=2*R/GasketHeight0) : 0.33 .. 1.0[slender..round
P P19 / deformation_ratio (=GasketHeightAssembly/GasketHeight0) 0.6..0.9 -> high..low deformation
P P23 / GasketInterference (diametral) Durchdringung beim Einlegen
P P20 / GrooveRadiusRatio 0.1..1.0[

Dependent parameters
P6/B1_D12 GrooveWidth= GrooveWidthScale*2*R
P15 / B2_D1 GasketHeight0 = 2*R/GasketSlenderness = 2*P14/P18
GasketHeightAssembly=GasketHeight0*deformation_ratio
P3 / B1_D17 GrooveDepth=GasketHeightAssembly-Shaft2WallClearance
=GasketHeight0*deformation_ratio
= P15*P19
P1 / B1_D18 GrooveRadius=GrooveDepth/(2-2*sin(GrooveAngel[GRAD]))*GrooveRadiusRatio
=P3/(2-2*sin(P2))*P20
P5 / B1_D15 ShaftDiameter=GasketInnerdiameter+2*GrooveDepth+2*GasketInterference
=P7+2*P3+2*P23
P11 / B3_D1 WallInnerDiameter=ShaftDiameter+2*Shaft2WallClearance
=P5+2*P22
P12 / B8_P1 t0OffsetGroove_x+ =GasketInterference+initial_interference
=-2*P23+0.1 [mm]
P13 / B9_P1 t0OffsetWall_x- =-2*GasketRadius = -2*P14

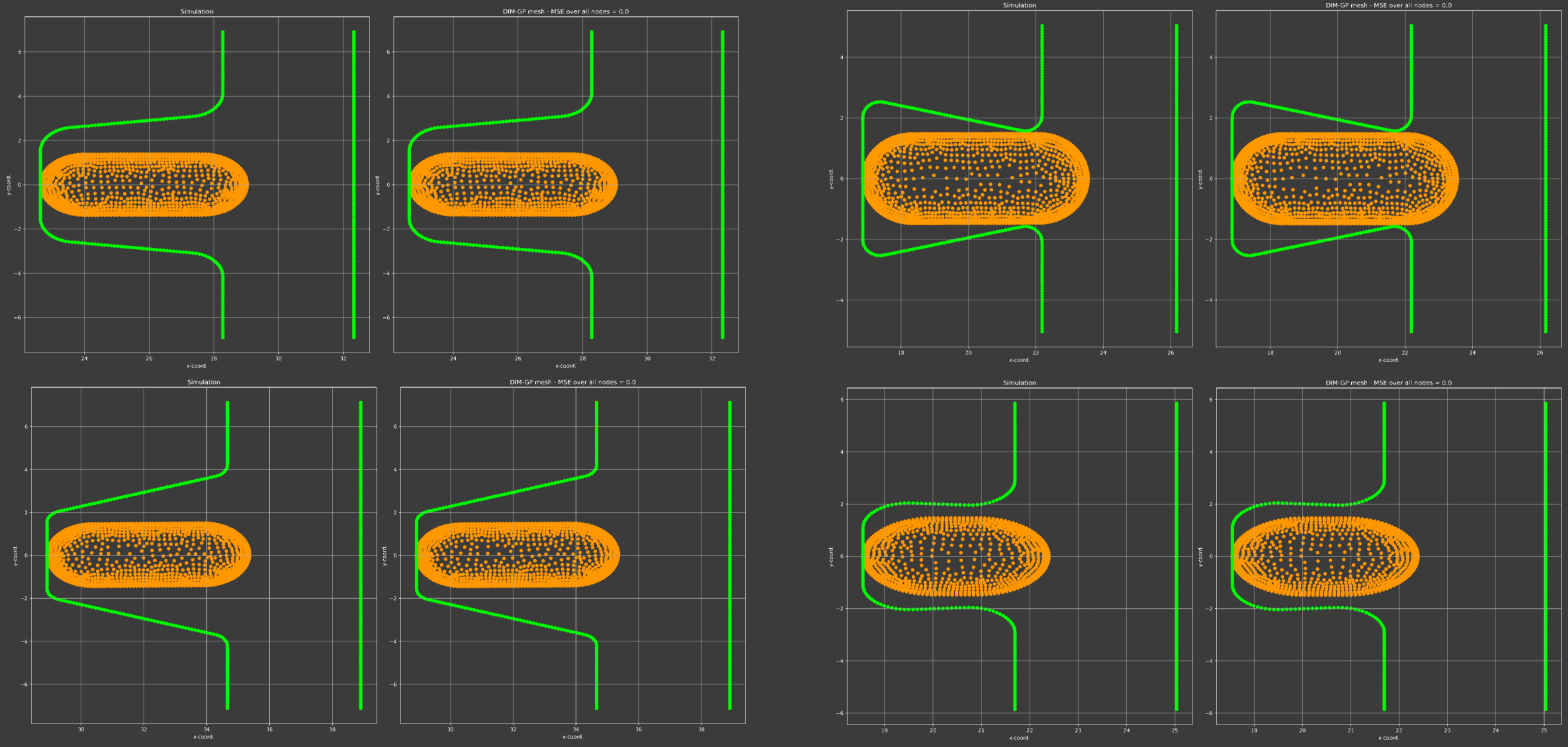
check GasketHeightAssembly>GrooveHeight?

Transient Structural (A1)			
P1	(B1_D18) GrooveRadius	0.350553	mm
P2	(B1_D14) GrooveAngle	5	degree
P3	(B1_D17) GrooveDepth	3.2	mm
P4	(B1_D16) Shaftlength	7	mm
P5	(B1_D15) ShaftDiameter	70.4	mm
P6	(B1_D12) GrooveWidth	2.2	mm
P7	(B2_D3) GasketInnerDiameter	60	mm
P8	Material Constant C10	1.25	MPa
P9	Incompressibility Parameter D1	0.016	MPa^-1
P11	(B3_D1) WallInnerDiameter	71.4	mm
P12	(B8_P1) t0OffsetGroove_x+	-1.9	mm
P13	(B9_P1) t0OffsetWall_x-	-2	mm
P14	(B2_D2) GasketRadius	1	mm
P15	(B2_D1) GasketHeight	4	mm
P10	PoissonsRatio	0.49	
P17	GrooveWidthScale	1.1	
P18	GasketSlenderness	0.5	
P19	deformation_ratio	0.8	
P20	GrooveRadiusRatio	0.2	
P22	Shaft2WallClearance	0.5	mm
P23	GasketInterference	2	mm

PARAMETERS		
B1_D18	GrooveRadius	2mm
B1_D14	GrooveAngle	5.00000000000003°
B1_D17	GrooveDepth	14mm
B1_D16	Shaftlength	7mm
B1_D15	ShaftDiameter	50mm
B1_D12	GrooveWidth	30mm
B2_D3	GasketDiameter	60mm
B3_D1	WallInnerDiameter	140mm
B8_P1	t0OffsetGroove_x+	-1mm
B9_P1	t0OffsetWall_x-	-2mm
B2_D2	GasketRadius	10mm
B2_D1	GasketHeight	31mm

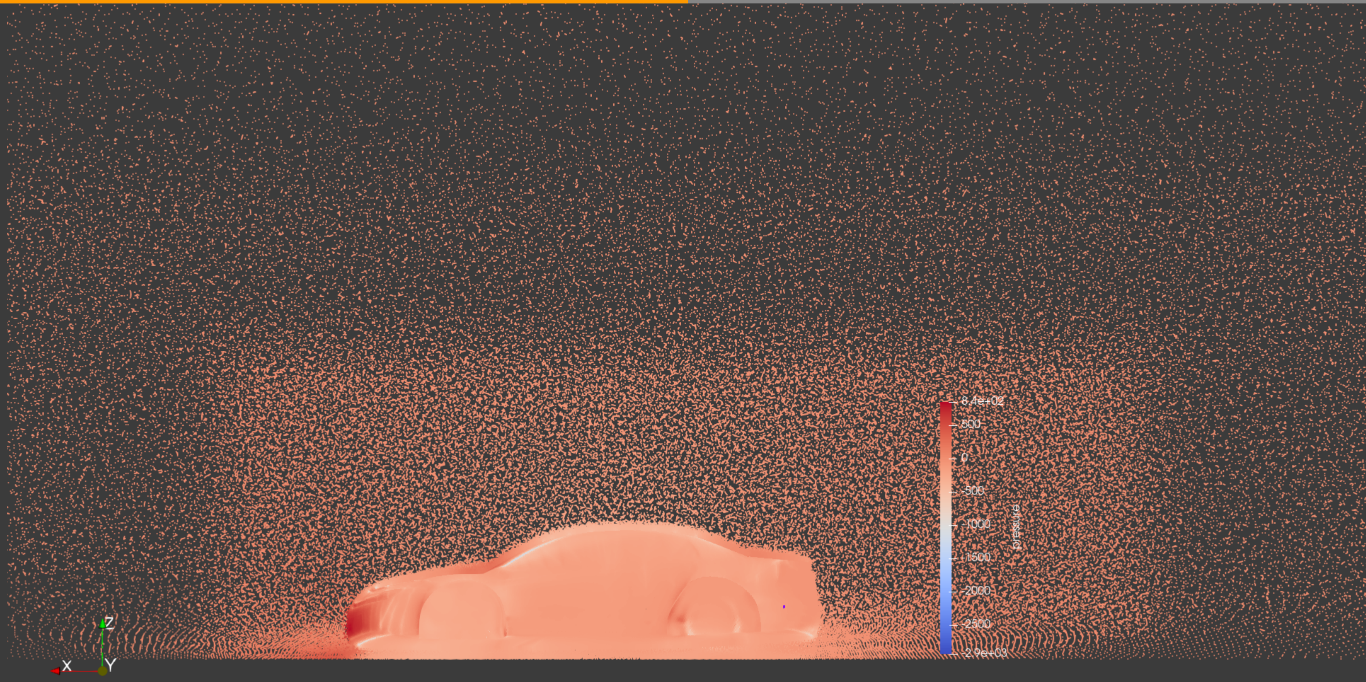
Gasket compression

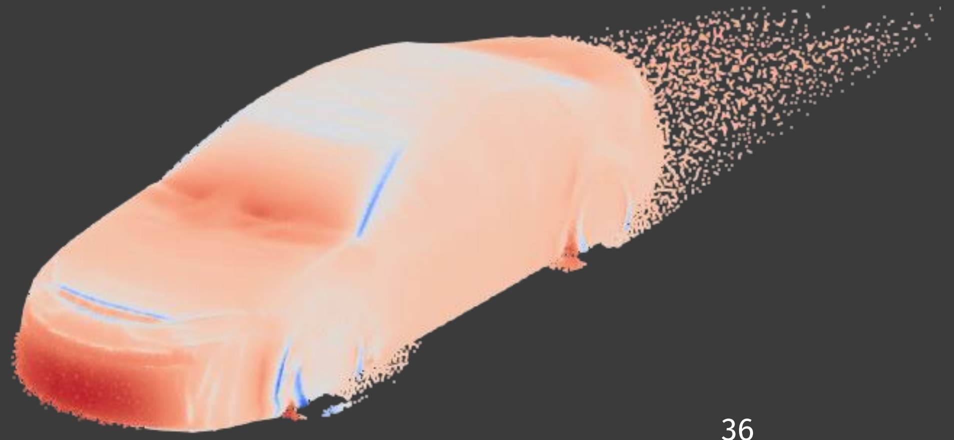
Simulation Prediction Simulation Prediction



Data overview

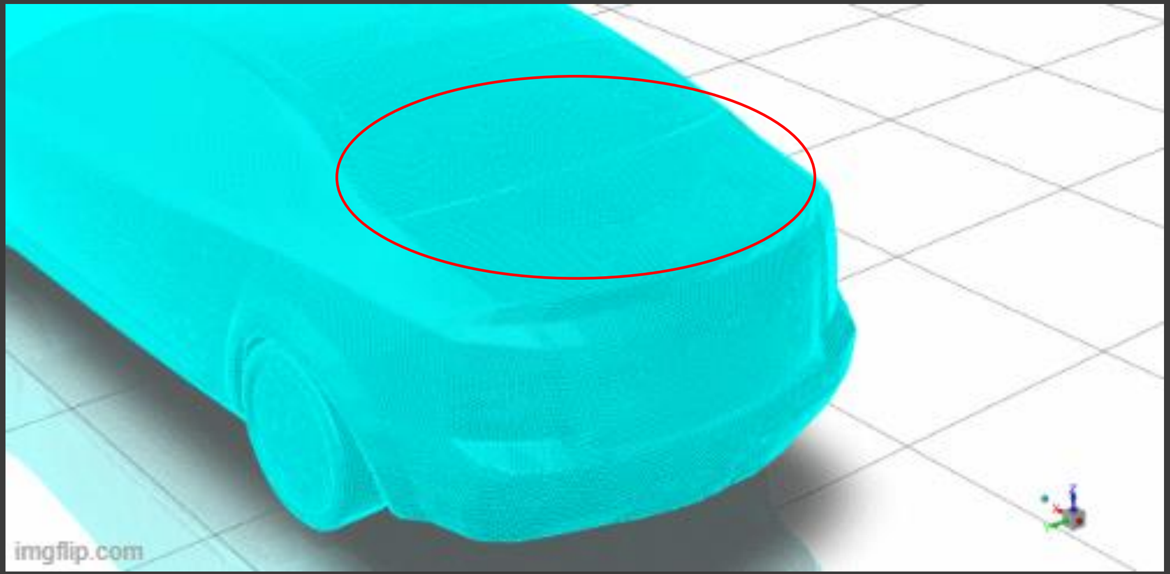
- 51 design variants
  - 45 for training / 6 for test
  - 1,542,308 nodes
- Input parameters:
  - x,y,z-coordinates of nodes,
  - global geometric parameters: back light angle and boat tail angle (see next slide)
- Output parameters:
  - pressure

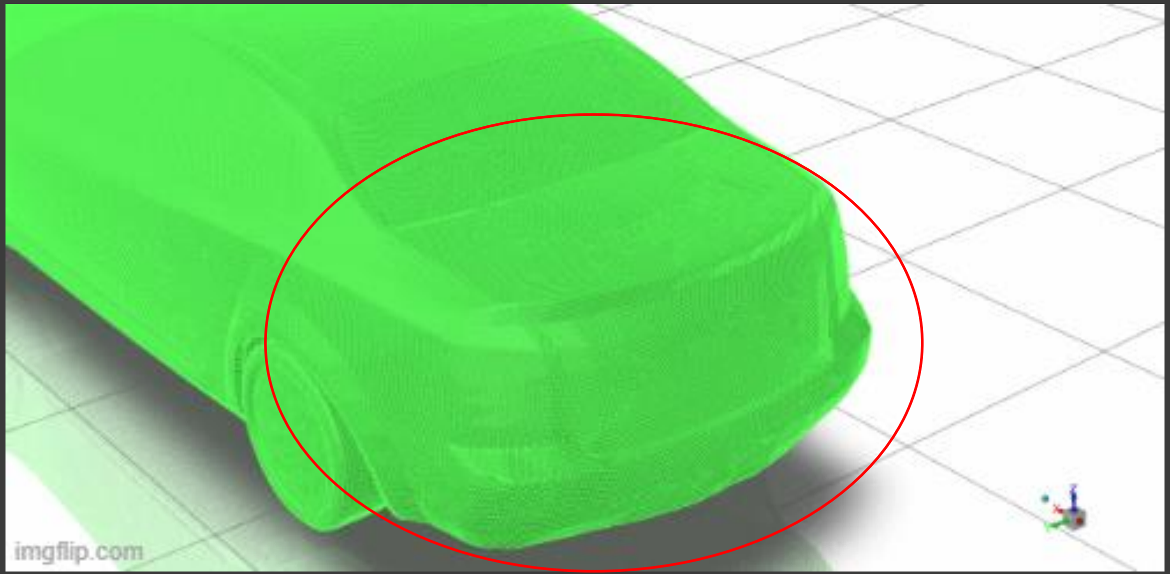




Visualization of the geometric changes

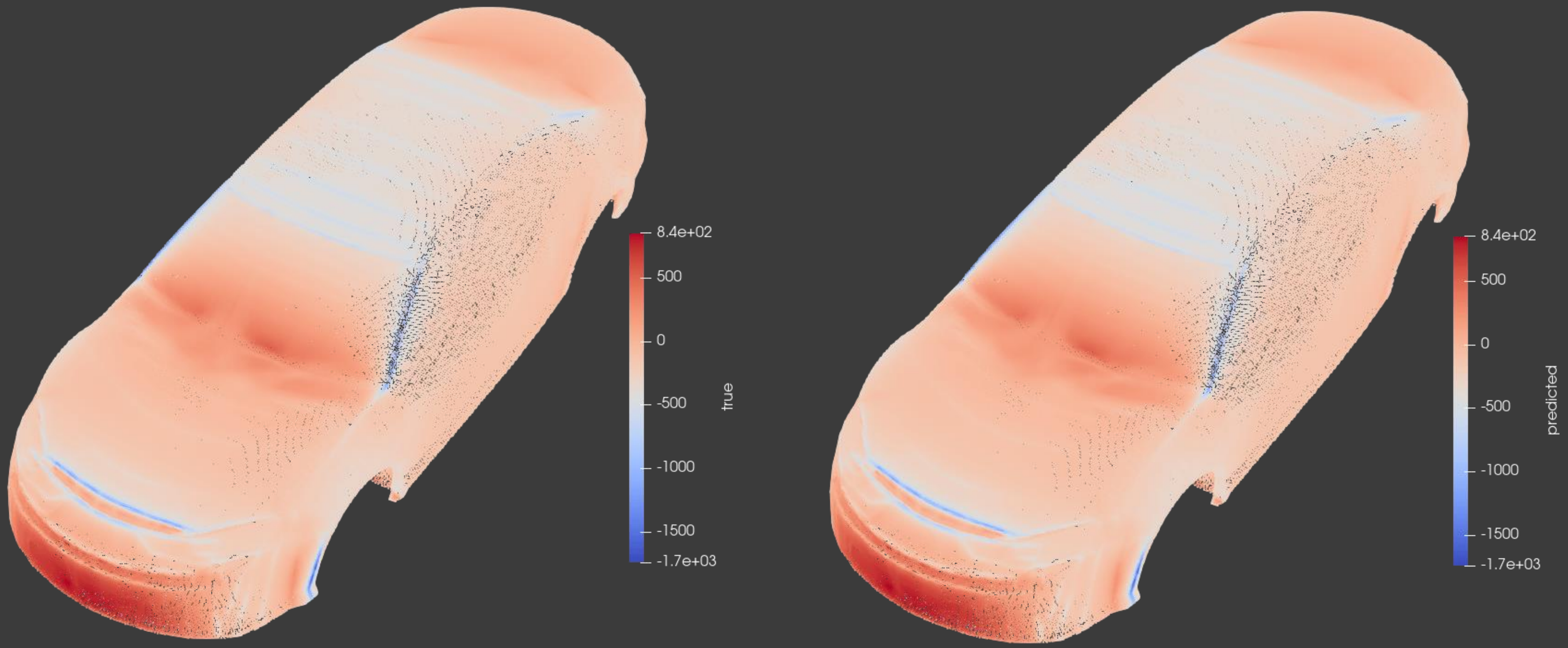
Variation of back light angle and boat tail angle





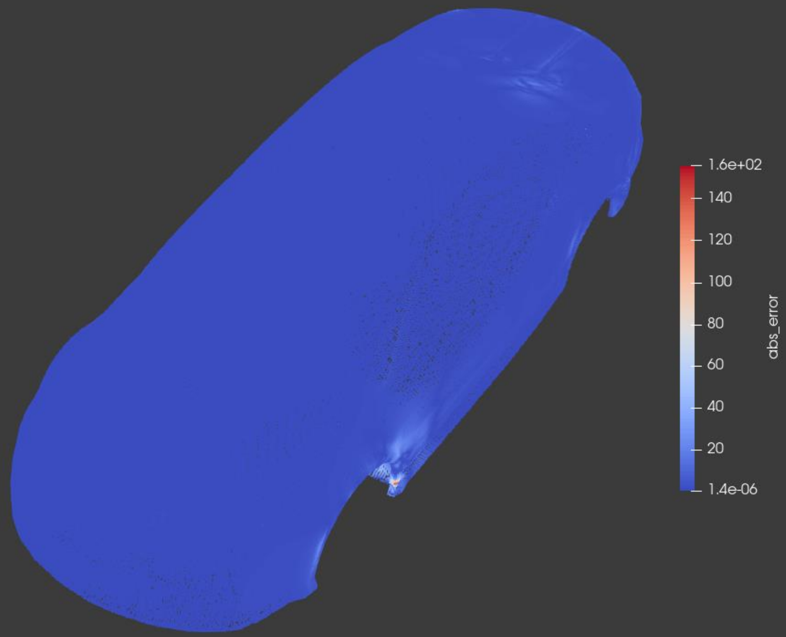
Overview prediction vs. true for a test design

- Training time: 10 seconds (advantage of constant mesh via mesh-morphing)

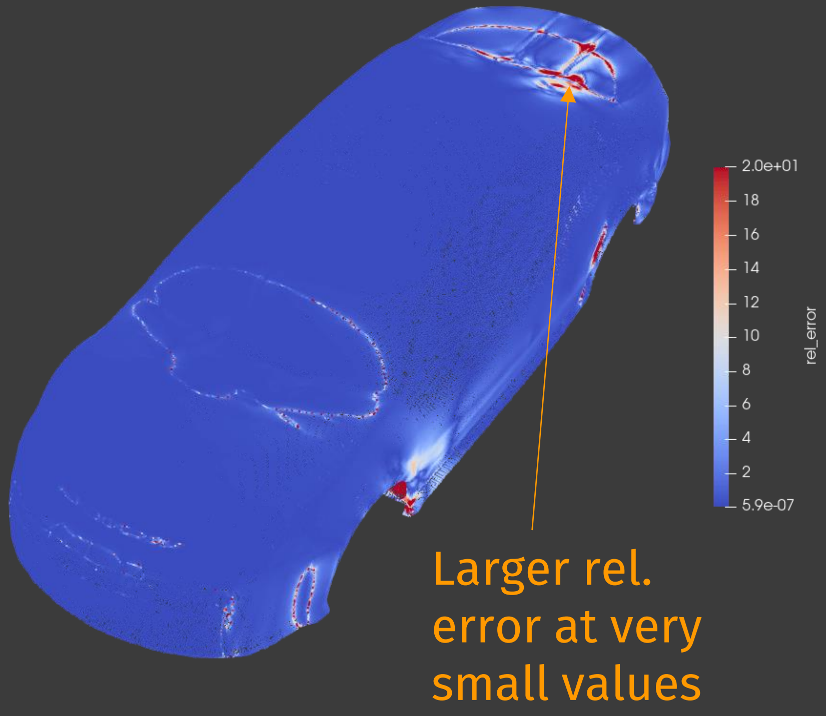


Overview prediction vs. true for a test design

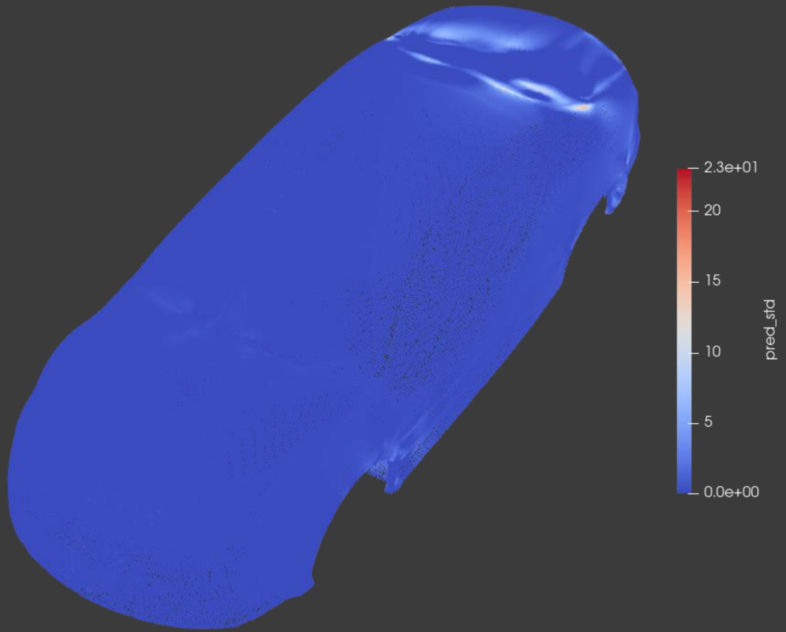
Absolute error



Precentage error capped at 20%



Predicted standard deviation



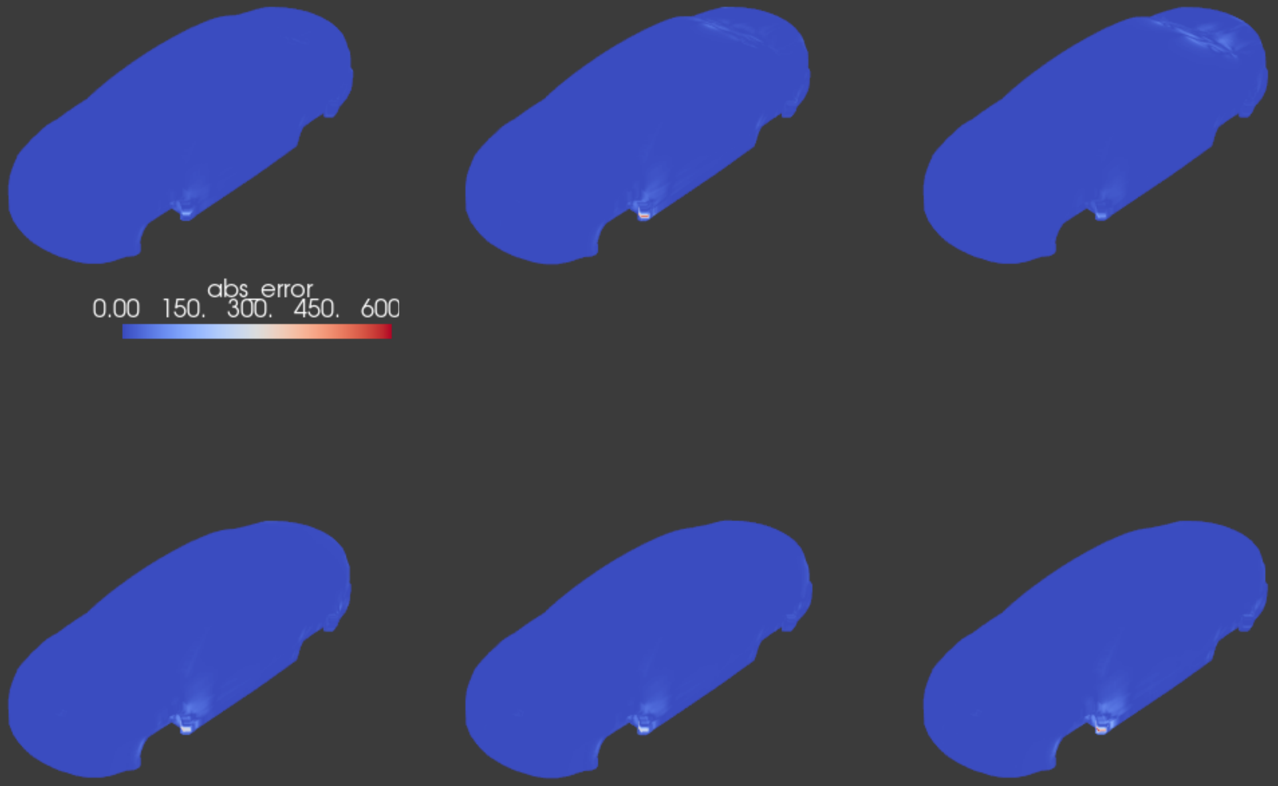
Mean absolute error over all nodes: 1.07

Mean relative error over all nodes: 1.29%

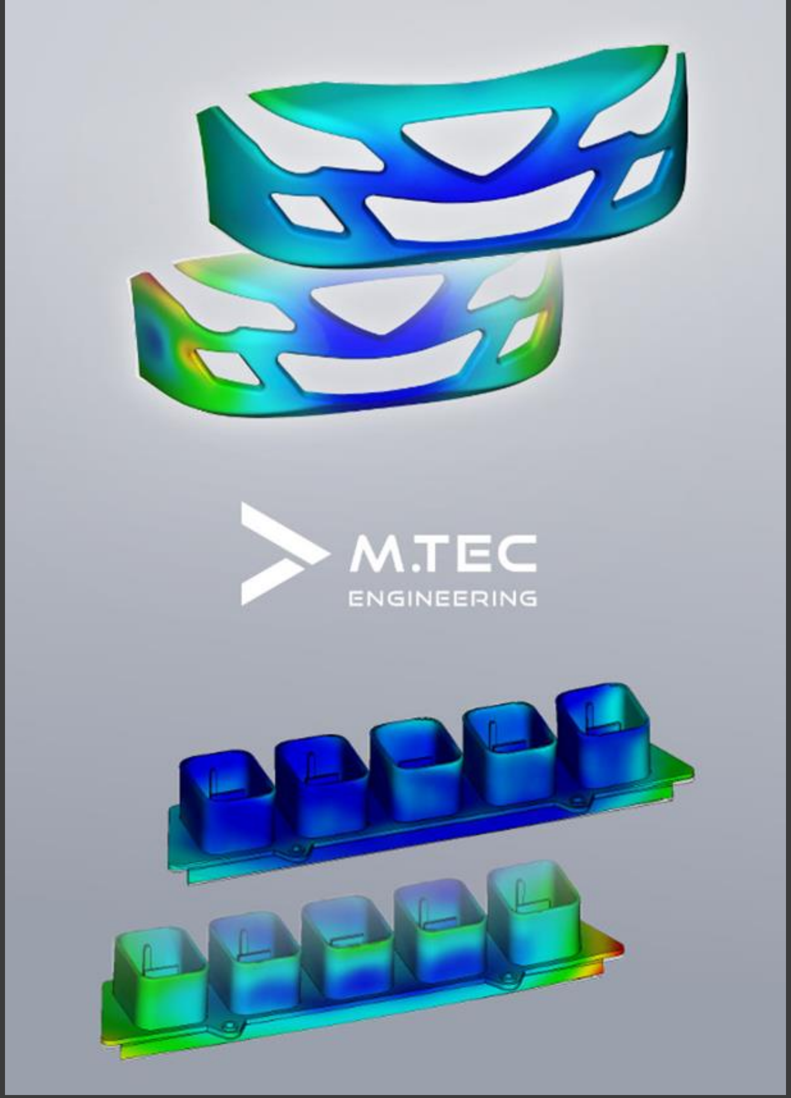
Overview prediction vs. true for 6 test designs

- Mean absolute error: 1.75
- Mean relative error: 3.1 %

- Error estimated over all 6 test designs over all nodes



AI-based warpage optimization of plastic components



Automotive Front Bumper

- overall warpage reduced from 8.0 mm to 4.4 mm (45% improvement)

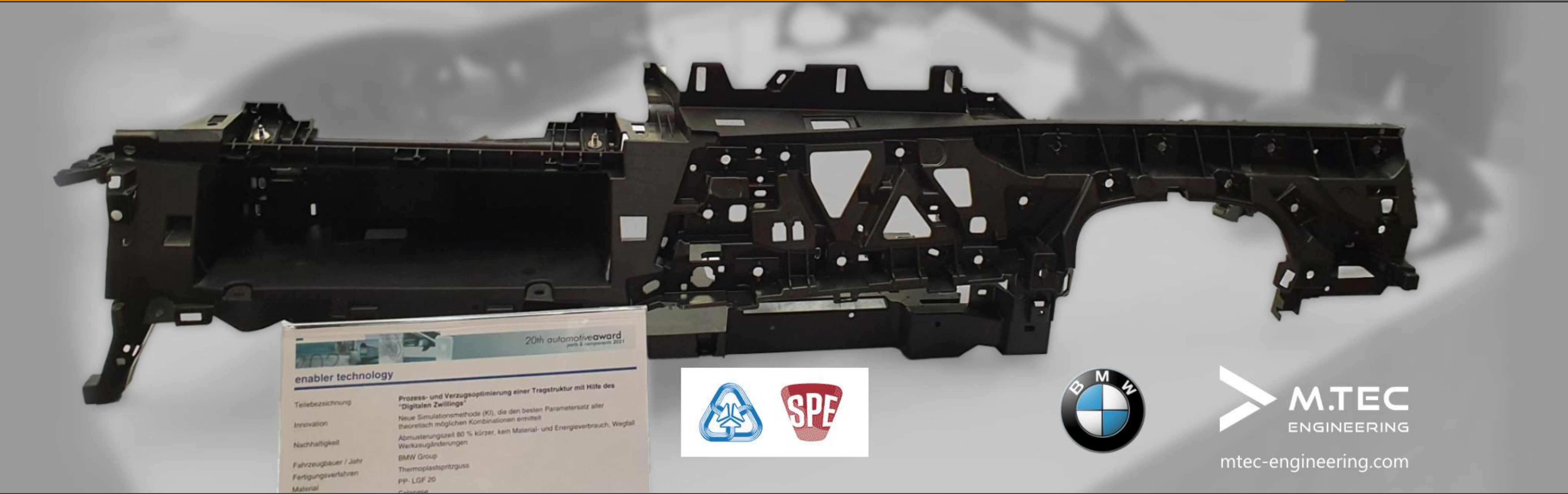
High Precision Connector

- overall warpage reduced from 0.46 mm to 0.29 mm (36% improvement)

"For over 30 years, our customers have been asking for the perfect parameter set for their plastic component. Thanks to new mathematical methods, we have now created a development tool that calculates this parameter set.“

Stefan Vogler
Team Manager Simulation & Calculation, M.TEC ENGINEERING GmbH

AI-based warpage & process optimization

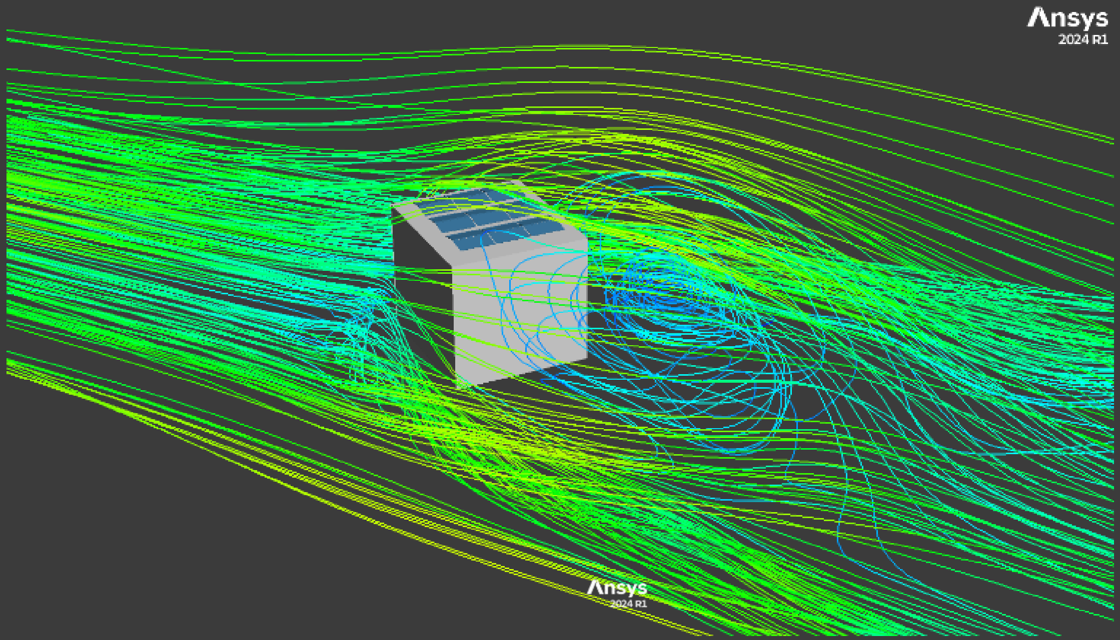


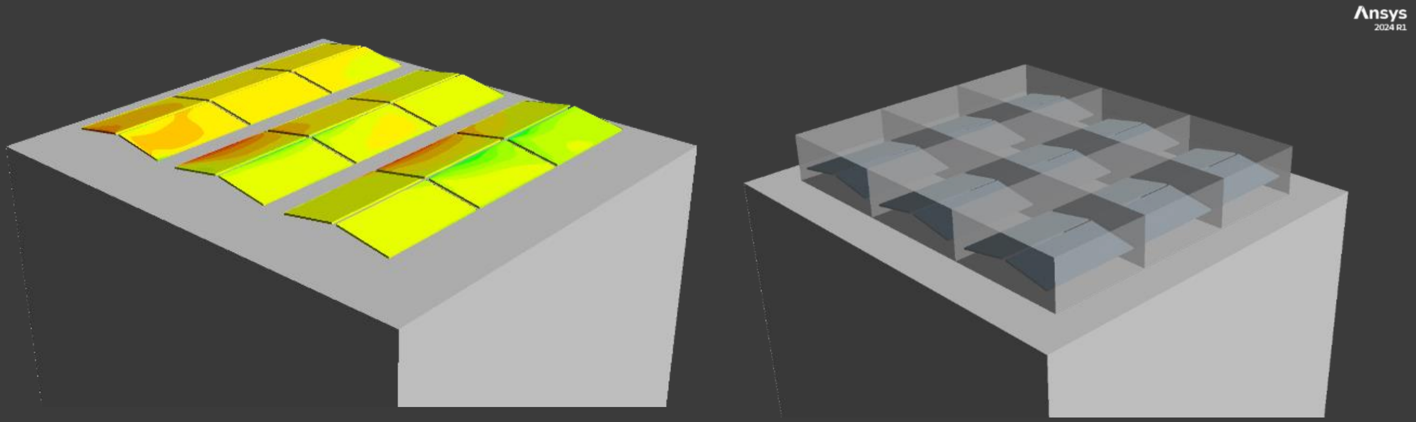
Support structure of the vehicle interior

- sample time 80% shorter
- greatly reduced material and energy consumption
- elimination of tool changes

Solar panel force prediction

- 1 design with 2000 time steps
- 4 different flow angles 0°, 30°, 60°, 90°
- Total 6000 time steps (samples)
- Input parameters:
  - Inlet velocities u, v, w
- Output parameters:
  - 9 x 3 forces x, y, z
- Output is averaged over 100 time steps and mean / std should be predicted





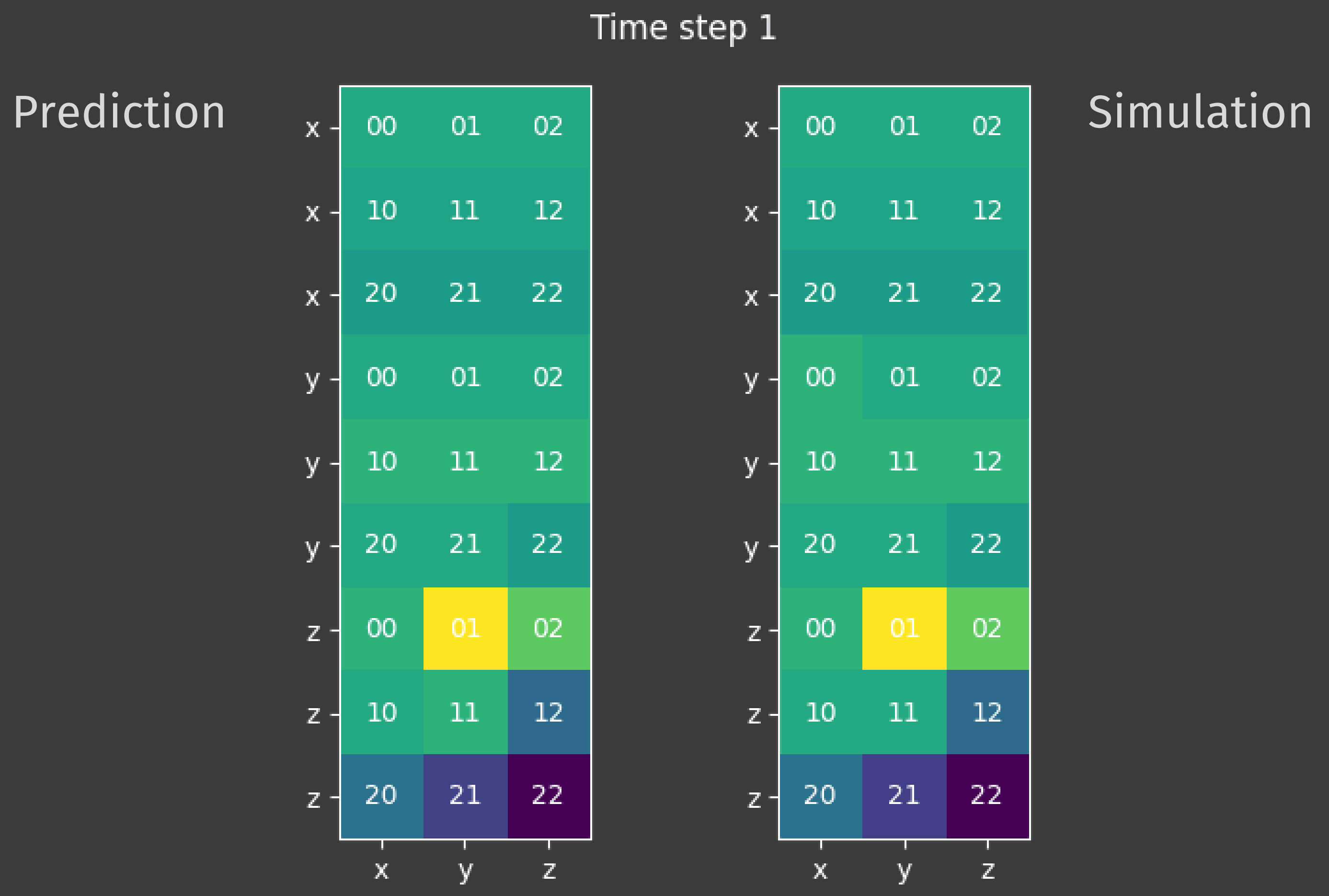
Solar panel force prediction

Force	Mean R^2	Std R^2
x_panel_00	0.97	0.97
x_panel_01	0.98	0.98
x_panel_02	0.99	0.97
x_panel_10	0.98	0.98
x_panel_11	0.98	0.97
x_panel_12	0.99	0.96
x_panel_20	0.98	0.98
x_panel_21	0.98	0.98
x_panel_22	0.98	0.96

Force	Mean R^2	Std R^2
y_panel_00	0.98	0.97
y_panel_01	0.98	0.97
y_panel_02	0.98	0.98
y_panel_10	0.98	0.97
y_panel_11	0.98	0.98
y_panel_12	0.97	0.97
y_panel_20	0.97	0.98
y_panel_21	0.98	0.95
y_panel_22	0.98	0.96

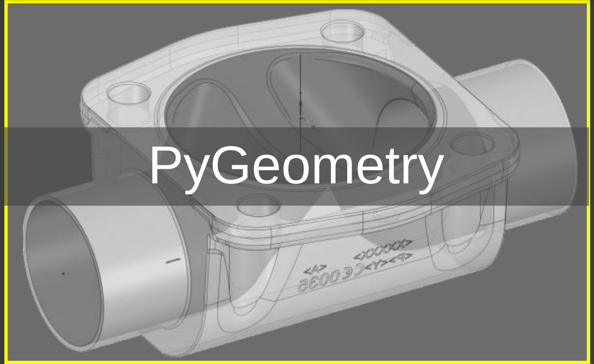
Force	Mean R^2	Std R^2
z_panel_00	0.98	0.98
z_panel_01	0.99	0.96
z_panel_02	0.99	0.94
z_panel_10	0.98	0.98
z_panel_11	0.98	0.98
z_panel_12	0.98	0.95
z_panel_20	0.98	0.97
z_panel_21	0.98	0.97
z_panel_22	0.97	0.97

Solar panel force prediction

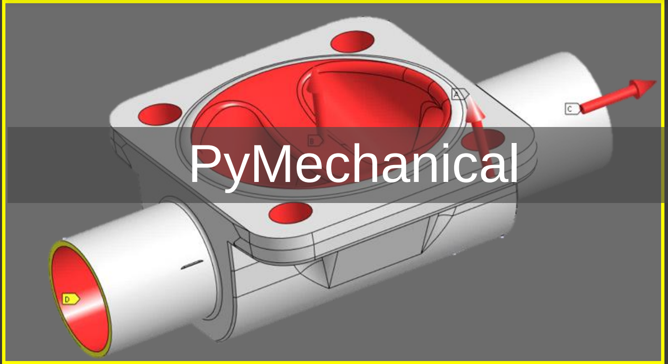


Simulation Automation

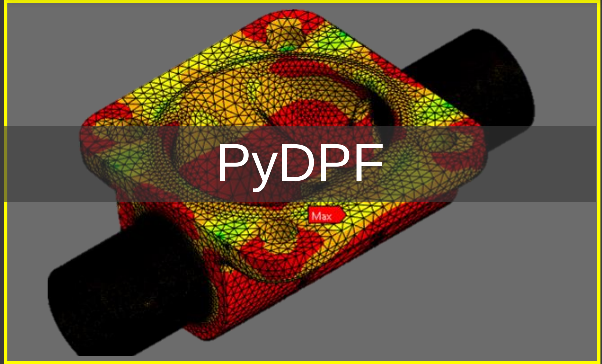
Geometry



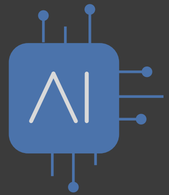
Setup



Report

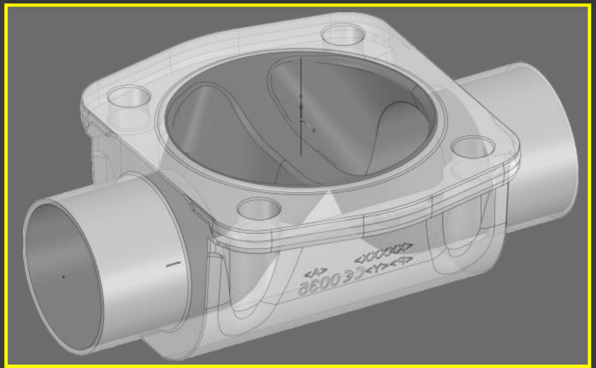
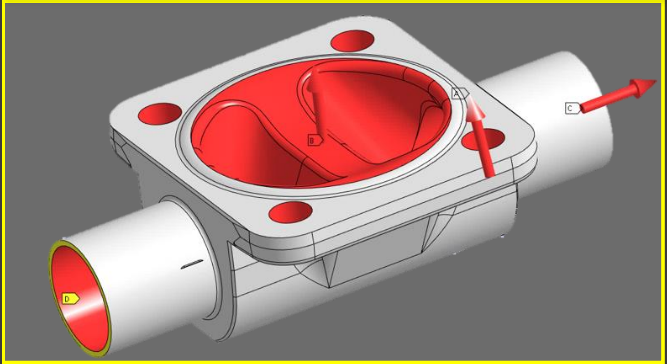
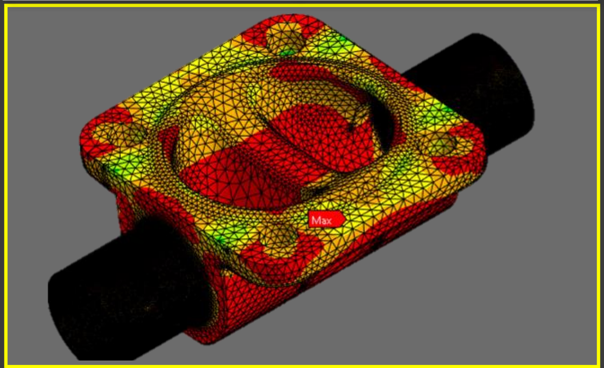


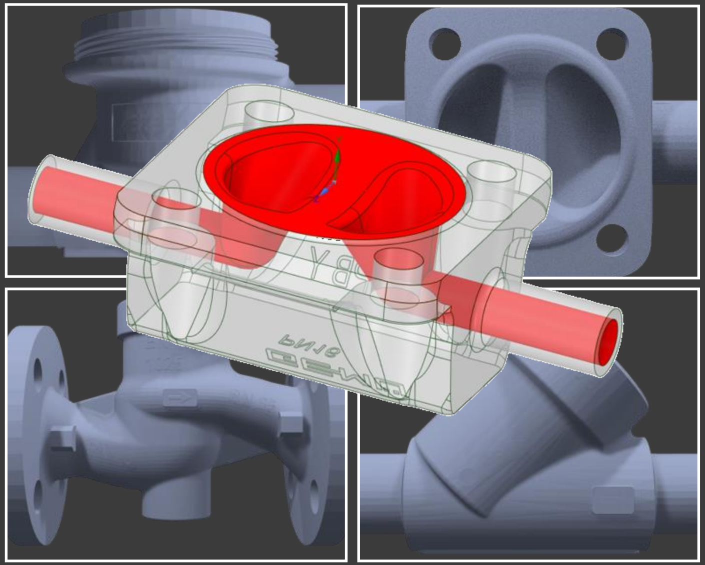
- ➢ Process integration → Interface between software
- ➢ Process automation → Full-scale data utilization



- Process automation → Full-scale data utilization → Geometry recognition

Geometry → Setup → Report



Challenges

- Over 2000 different valve parts
- „Dead“ geometry with no attributes like inside and outside faces
- How to detect and select the proper faces correctly in an automatic way?

Solution

From classic approach to AI approach

The images were provided by courtesy of GEMÜ Gebr. Müller Apparatebau GmbH & Co. KG

GEMÜ

CADFEM AI

→ Geometry recognition

AI approach enables full-scale simulation automation

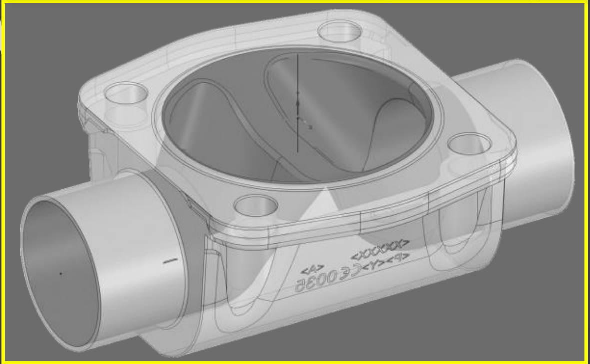
Classic approach

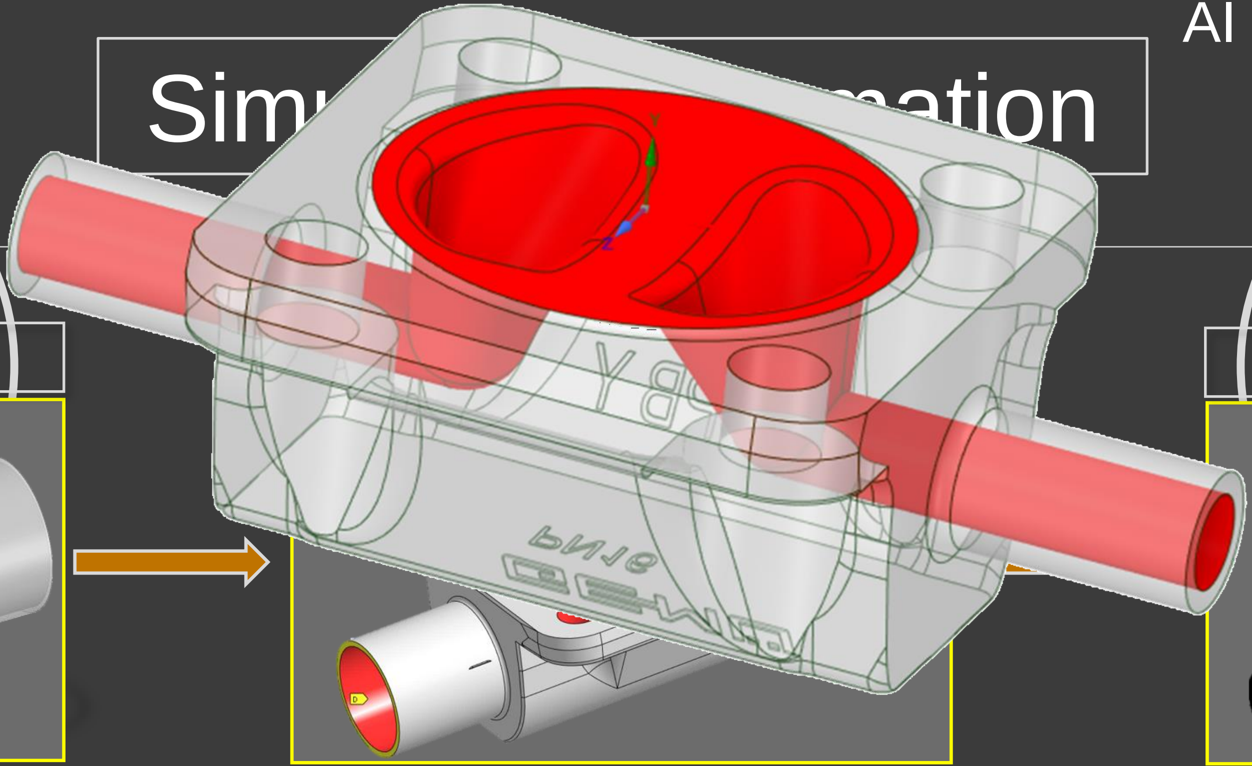
AI approach with DIM-GP

Sim... mation

60 – 70

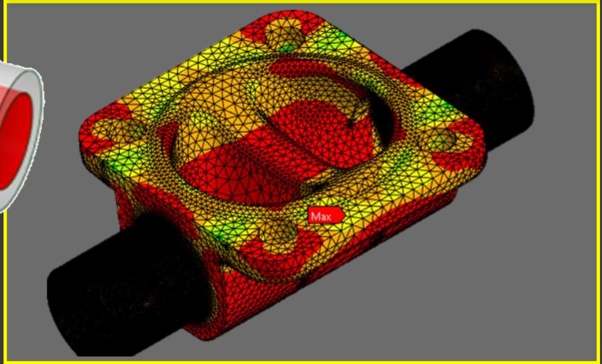
Geometry





99.7

Report



The images were provided by courtesy of GEMÜ Gebr. Müller Apparatebau GmbH & Co. KG

Contact

Dr.-Ing. Kevin Cremanns
kevin.cremanns@probaligence.de
www.probaligence.de

PROBABILISTIC INTELLIGENCE

CADFEM® GROUP