PROBABILISTIC INTELLIGENCE

Universal Machine Learning based on Probabilistic Intelligence

Authors: Dr. Kevin Cremanns*
Date: 03.05.2024

*Co-founder & responsible for research
PI Probaligence GmbH

CADFEM GROUP

Introduction

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

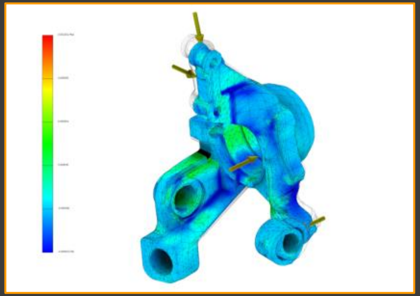
```python
from stochos.bayesian_optimization import bayesian_opt
from stochos.plot import plot_scatter_mat, plot_sensitivity
from stochos.dimgp import dimgp_regr, pam_regr
from stochos.sensitivity import sobol_indices
import numpy as np
import pandas as pd

"""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.

Excerpt of our industries



Simulation

Automotive

Materials science

Turbomachinery

Healthcare

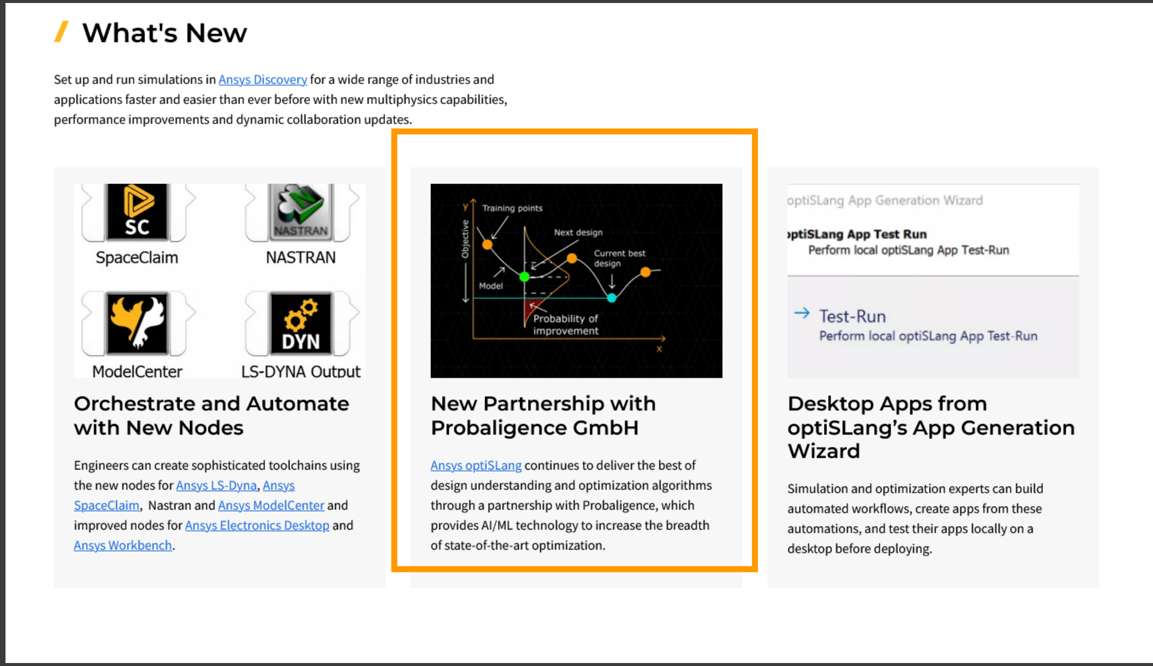
Chemicals

Sports medicine

Textile industry

Since 2024 part of CADFEM Group

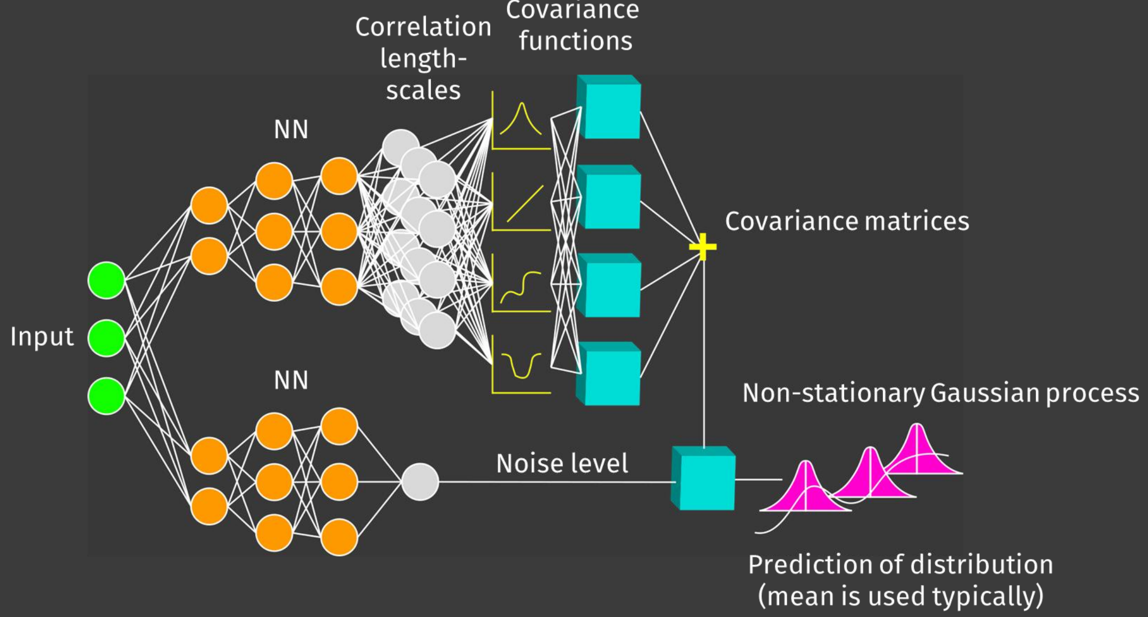
Since 2022 our software STOCHOS is partially integrated in the Ansys OptiSLang (AI+ license required)



Since 2024 PI Probaligence became part of the CADFEM Group as partner for AI / ML Solutions

Deep infinite mixture of Gaussian Processes (DIM-GP)

Can be applied to wide range of machine learning task with only one algorithm and no settings



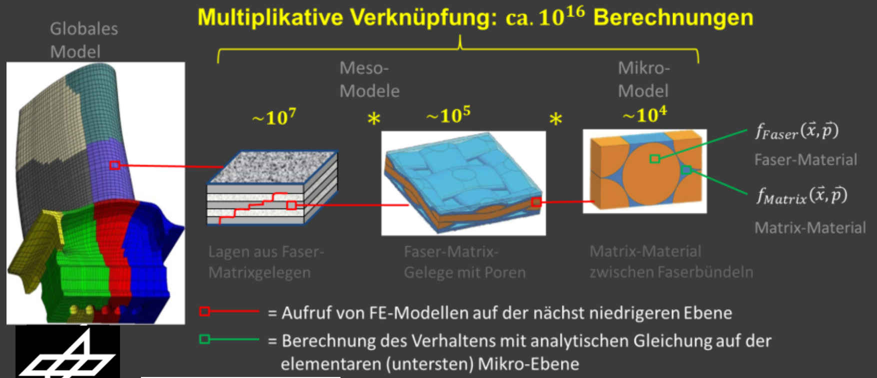
- **Non-stationary probabilistic model**
- **No settings (no expert knowledge)**
- **Can be used for various forms of data**
- **Requires little data for good results**
- **Automatic noise handling**
- **Low hardware requirements (no cloud, data remains with the customer)**

Unique combination of neural networks + Gaussian process

Usable data with DIM-GP

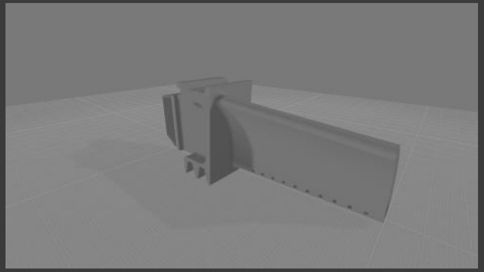
- Scalars, signals, fields, tensors, images, meshes can be used as input / partially as output:

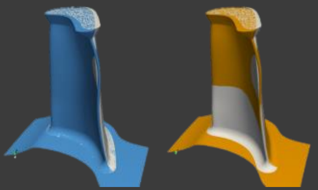
x1	x2	x3	x4	x5	x6	x7	y1	y2	y3
27	90	0.5	13.5	51	2	3.5	21.3277356	20759.878	29198.3713
27.6775	123.975	0.895	13.37	49.52	1.855	3.143	23.671277	26584.1086	44739.0519
31.7725	121.275	0.865	12.27	49.76	1.645	3.675	23.9304892	27406.0836	29920.9292
28.0025	115.425	0.255	12.93	48.32	1.505	4.095	27.7219304	21129.948	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



DLR

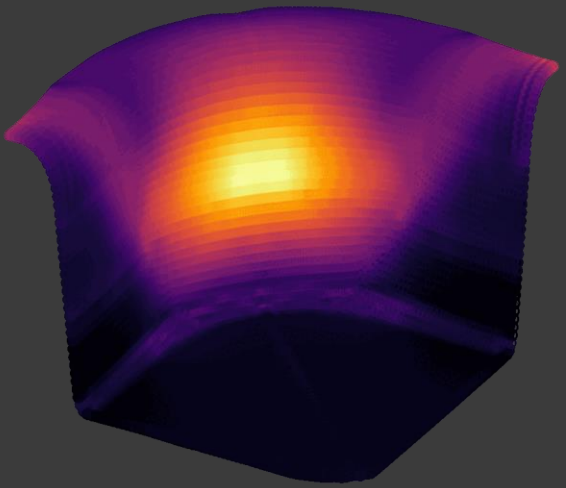
sgl carbon

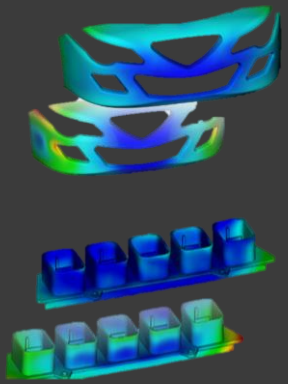




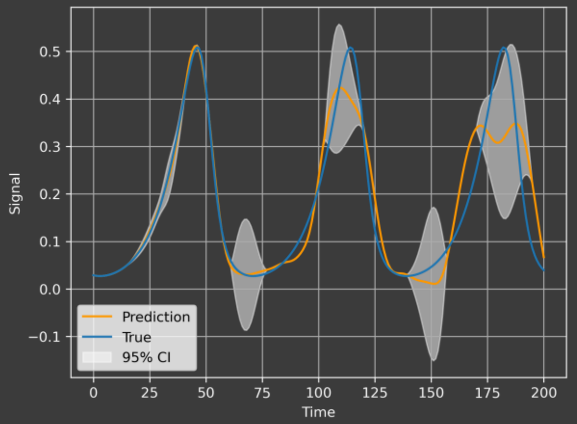
ROLLS ROYCE

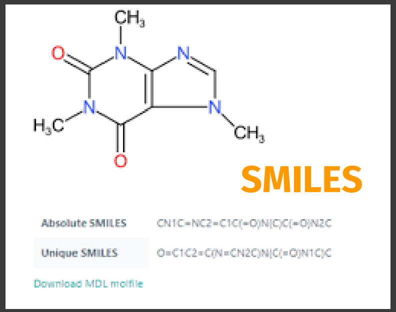
Live FEM & CFD



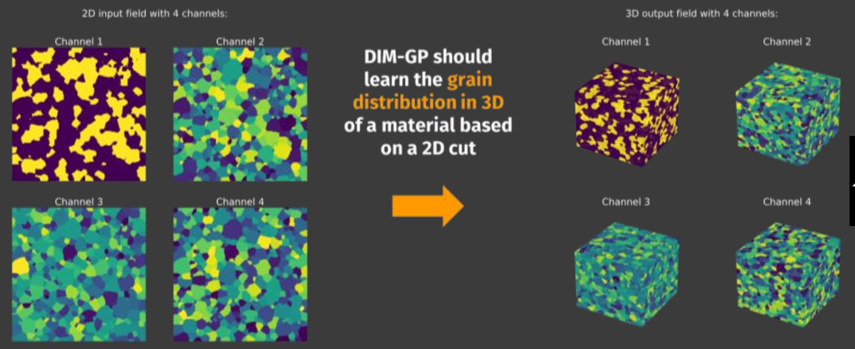


M.TEC ENGINEERING





Molecule information

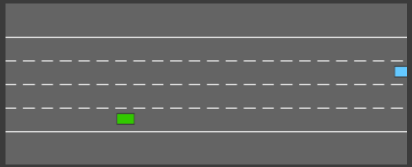


DLR

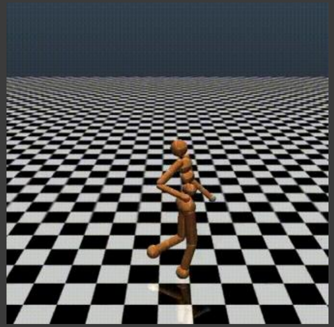
Covering the three main ML areas with one algorithm

Reinforcement learning

Production line



Autonomous driving



CPU
- Small models
- Small datasets
- Useful for design space exploration

GPU
- Medium-to-large models, datasets
- Image, video processing
- Application on CUDA or OpenCL

TPU
- Matrix computations
- Dense vector processing
- No custom TensorFlow operations

Control

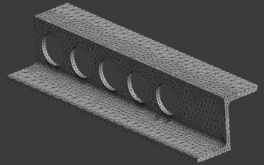
Predict

DIM-GP

Observe

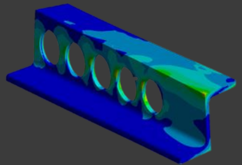
Supervised learning

Input



ML - Model

Output



Predict properties

Unsupervised learning

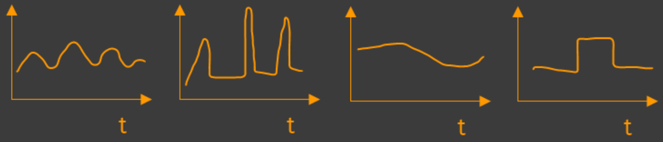
Predictive maintenance

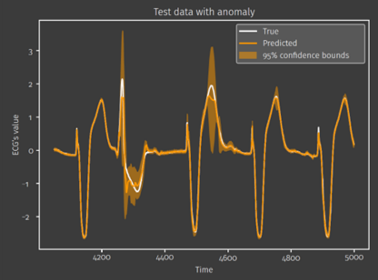
Sensor 1

Sensor 2

Sensor 3

Sensor ...



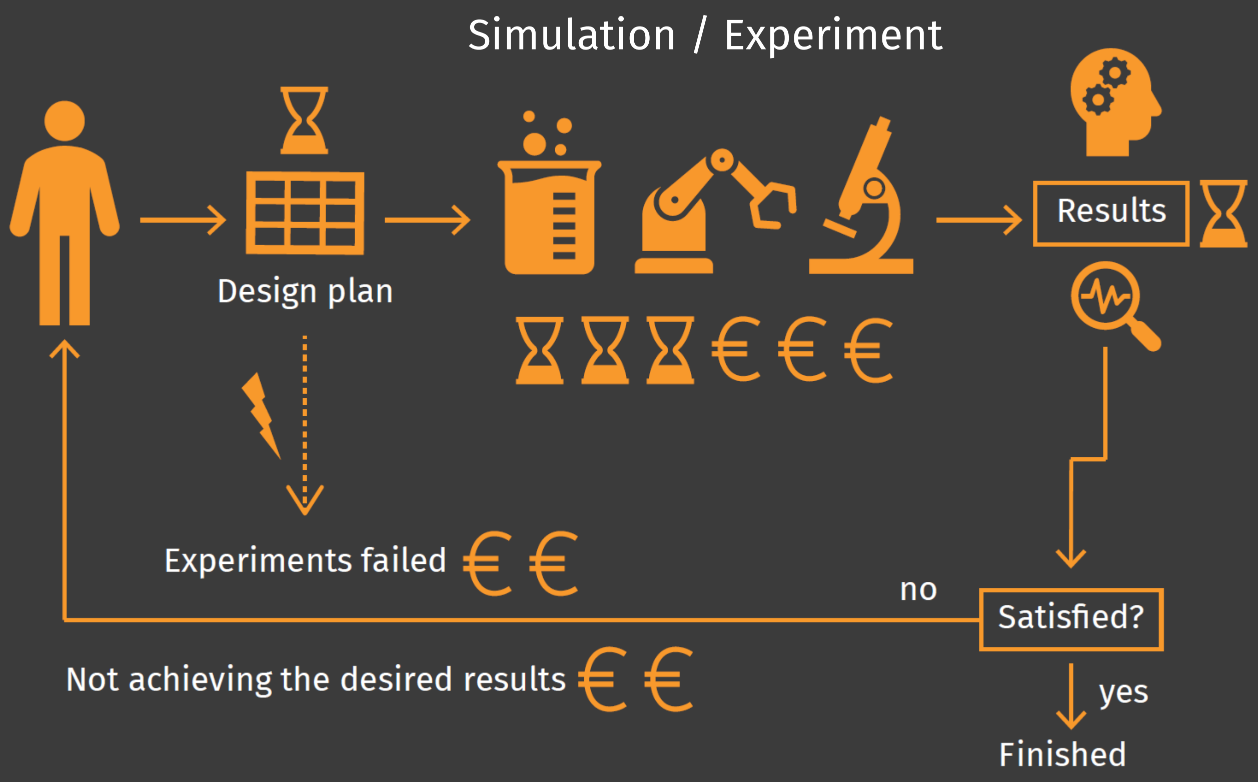


How to optimize products and processes with the help of ML most efficiently

Classical data generation process

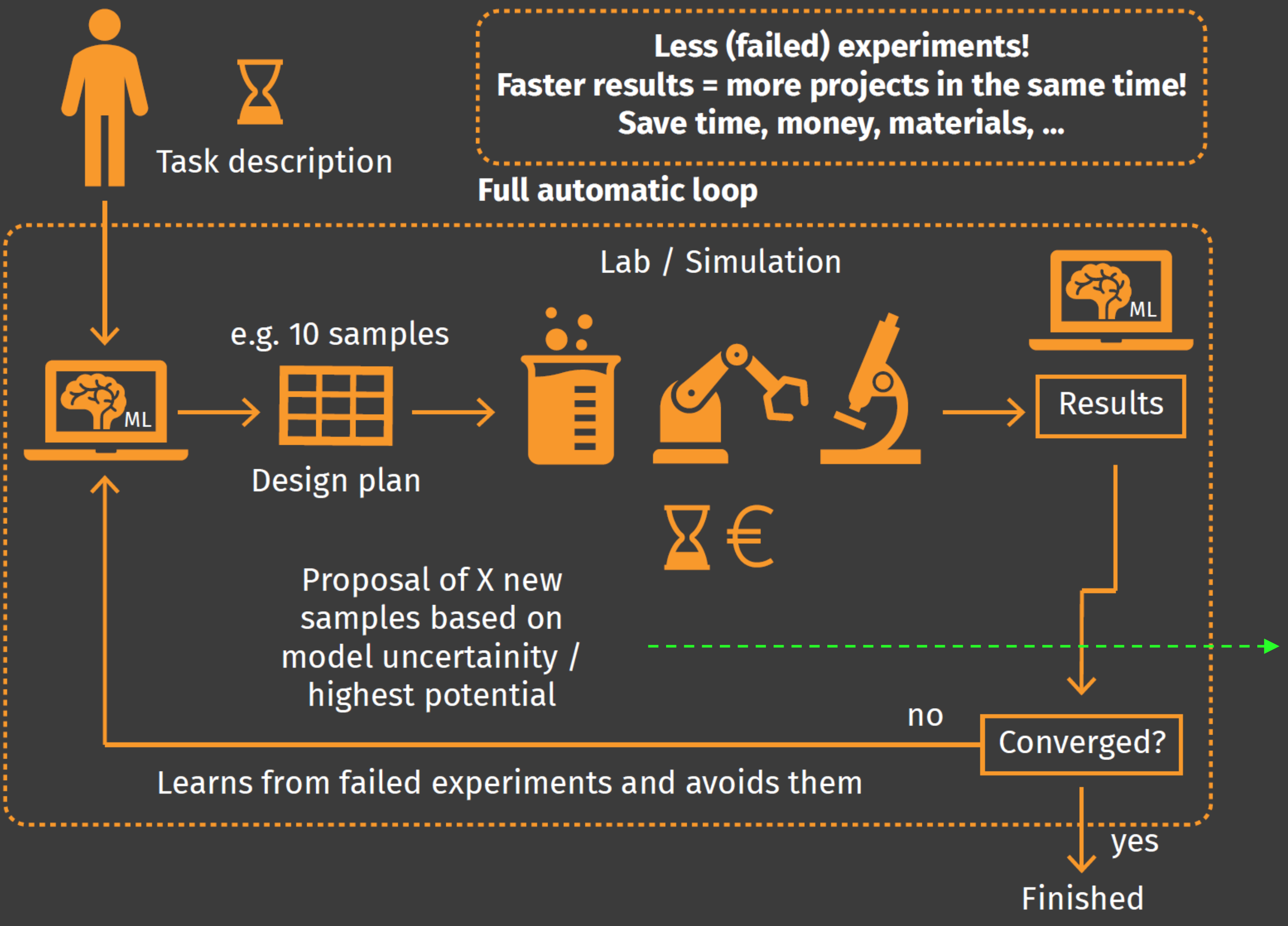
Classical way data generation:

1. Plan all designs in advance
2. Do simulation / experiments
3. Human analyze results (model training)
4. If not satisfied repeat (1-3)
5. If model is good enough, use it in production (web app, optimization, sensitivity studies, …)

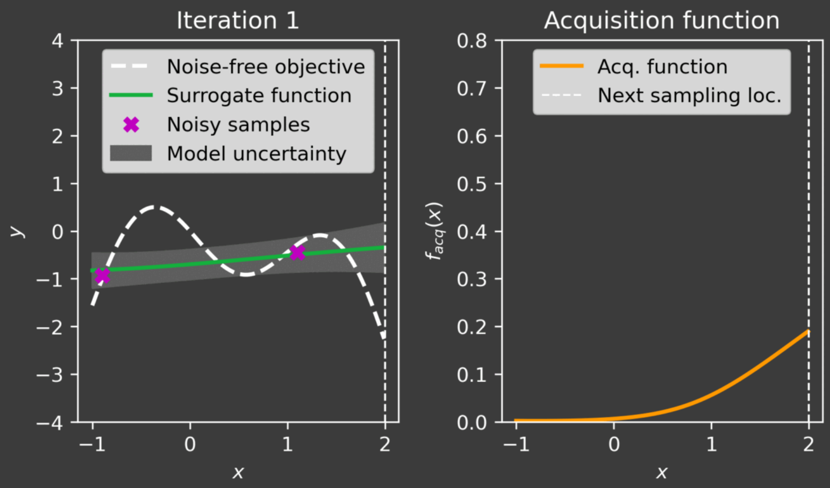


Efficient adaptive optimization / design of experiment

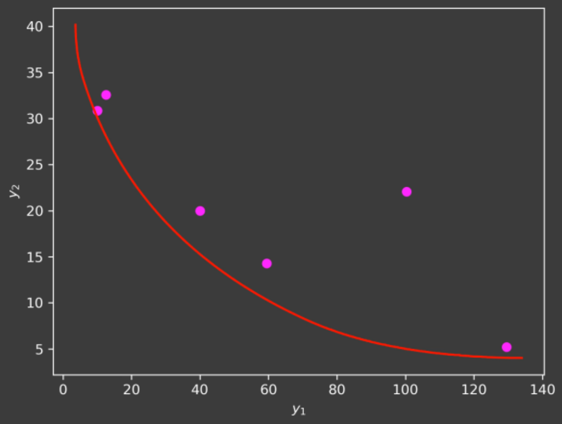
Adaptive search of next optimum



Single objective: search maximum of y



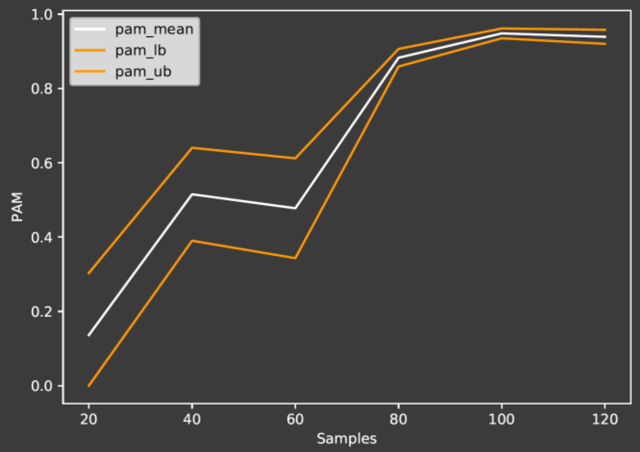
Multi-objective: search Pareto-frontier

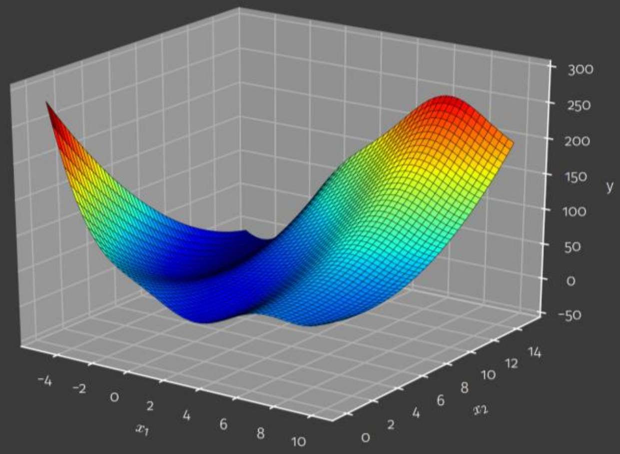


Efficient simulation / experiment replacement:

Adaptive model improvement

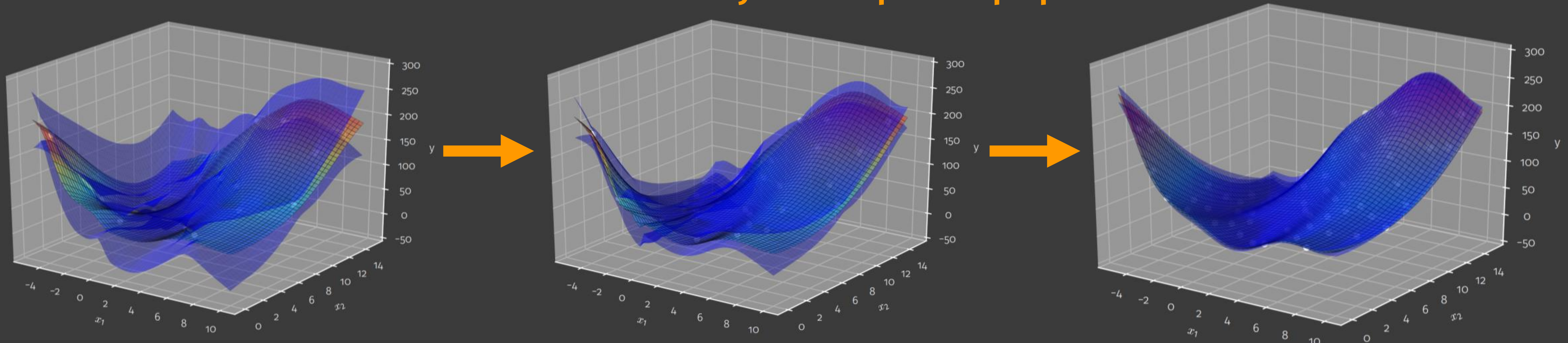
Convergence check of model prognosis





True function

Based on model uncertainty new samples are proposed



Customer Benchmark from Bosch

Roland Schirrmacher,
Robert Bosch GmbH

Keynote: Process and results of the One Click Optimizer benchmark at 2023 WOST Conference

Selection of Optimization Algorithms
Algorithms inside optiSLang and from external sources

- Nature Inspired Optimization Algorithms
  - Evolutionary Algorithm (EA)
  - Particle Swarm Optimization Algorithms (PSO)
- Adaptive Optimization Algorithms
  - Adaptive Single-Objective Optimization Algorithm (ASO)
  - Adaptive Multi-Objective Optimization Algorithm (AMO)
  - Adaptive Metamodel of Optimal Prognosis (AMOP)
  - Bayesian Optimization (BO)
- Hybrid Optimization Algorithms
  - One Click Optimization Algorithm (OCO)

- SIGOPT (SIGOPT)
  - Mixture from global and Bayesian optimization from the company Intel
- Black Box Optimization from Bosch (BCAI)
  - Space Filling by Sobol-Sequences
  - MBORE: Multi-objective Bayesian Optimization by Density-Ratio Estimation
- CR optimizer from Bosch (CROPT)
  - NSGA II algorithm
  - Special features, not suitable for that benchmark

5 CR/AME3 | 2023-06-22

BOSCH

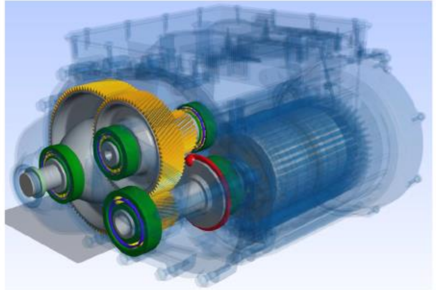
Published: https://www.ansys.com/events/wost-conference/wost-conference-presentations

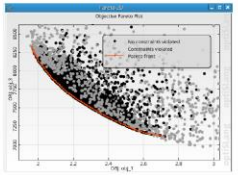
Customer Benchmark from Bosch

Overall 6 different applications have been benchmarked

Examples for Multi-Objective-Optimization
Acoustic Properties in Gear Simulation

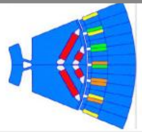
- The NVH behaviour of gear systems also depends on the micro geometry.
- 16 design variables of the micro geometry were used to calculate 32 response variables based on different loading conditions.
- The 32 response variables were used to define
  - 2 objective functions to minimize
  - 4 constraints





$Hypervolume: v = 1134.6$
$Hypervolumereferenz: rf1 = 3$
$rf2 = 8500$

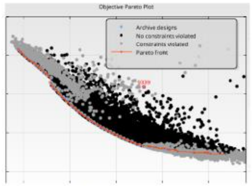
Examples for Multi-Objective-Optimization
Performance of an eMachine



- The geometry of the eMachine has 27 parameters. Two parameters have discrete values
  - p : number of pole pairs
  - q: slots per pole per phase
- No geometry check was taken into account.
- The original optimization problem consists of two objective functions and 2 constraints.
- Because the calculation of the hypervolume does not allow negative values for the objective function which comes from the maximization of the maximum power, an offset of 220 was selected and a minimization of the difference to 220.
- 30 optimization runs were performed.

Criteria

Name	Type	Expression	Criterion	Limit	
obj_mat_Cost	Objective	mat_Cost	MIN		125.599
obj_P_max	Objective	220-P_max	MIN		95.6283
constr_M_max	Constraint	M_max-200	≥	0	140.508
constr_I_AKS	Constraint	400-I_AKS	≥	0	176.87



$Hypervolume: v = 12807.0$
$Hypervolumereferenz: mat\_Cost = 120$
$P\_max = 220$

Example for Single-Objective-Optimization
RC15: Speed reducer

- The speed reducer is an official benchmark example for real-world applications.
- The optimization is a problem with 7 continuous design variables and 11 constraints.
- The optimum is
  - f = 2999.17063
  - x7 = 5.28636135
  - x6 = 3.34996302
  - x5 = 7.73038815
  - x4 = 7.30199647
  - x3 = 17.0183333
  - x2 = 0.700166667
  - x1 = 3.50246144

https://www.sciencedirect.com/science/article/pii/S2210650219308946


Swarm and Evolutionary Computation

A test-suite of non-convex constrained optimization problems from the real-world and some baseline results

Abhishek Kumar, Guohua Wu, Mostafa Z. Ali, Rammohan Mallipeddi, Ponnuthurai Nagaratnam Suganthan, Swagatam Das

Minimize:

$$f(\bar{x}) = 0.7854x_2^2x_1(14.9334x_3 - 43.0934 + 3.3333x_3^2) + 0.7854(x_5x_7^2 + x_4x_6^2) - 1.508x_1(x_7^2 + x_6^2) + 7.477(x_7^3 + x_6^3)$$

subject to:

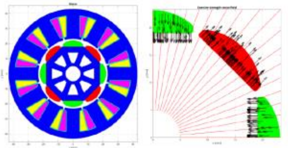
$$g_1(\bar{x}) = -x_1x_2^2x_3 + 27 \le 0,$$
$$g_2(\bar{x}) = -x_1x_2^2x_3^2 + 397.5 \le 0,$$
$$g_3(\bar{x}) = -x_2x_6^4x_3x_4^{-3} + 1.93 \le 0,$$
$$g_4(\bar{x}) = -x_2x_7^4x_3x_5^{-3} + 1.93 \le 0,$$
$$g_5(\bar{x}) = 10x_6^{-3}\sqrt{16.91 \times 10^6 + (745x_4x_2^{-1}x_3^{-1})^2} - 1100 \le 0,$$
$$g_6(\bar{x}) = 10x_7^{-3}\sqrt{157.5 \times 10^6 + (745x_5x_2^{-1}x_3^{-1})^2} - 850 \le 0,$$
$$g_7(\bar{x}) = x_2x_3 - 40 \le 0,$$
$$g_8(\bar{x}) = -x_1x_2^{-1} + 5 \le 0,$$
$$g_9(\bar{x}) = x_1x_2^{-1} - 12 \le 0,$$
$$g_{10}(\bar{x}) = 1.5x_6 - x_4 + 1.9 \le 0,$$
$$g_{11}(\bar{x}) = 1.1x_7 - x_5 + 1.9 \le 0,$$

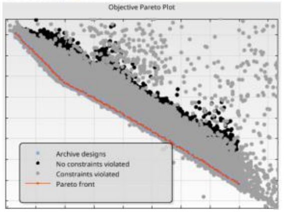
with bounds:

$$0.7 \le x_2 \le 0.8, 17 \le x_3 \le 28, 2.6 \le x_1 \le 3.6,$$
$$5 \le x_7 \le 5.5, 7.3 \le x_5, x_4 \le 8.3, 2.9 \le x_6 \le 3.9.$$

Examples for Multi-Objective-Optimization
Performance for an eDrive

- The geometry of the eDrive has 21 parameters. Four parameters have discrete values
  - Magnet_Material
  - Wire_Selector_X1
  - Do_Skew
  - Wire_Selector_X2
- The original optimization problem consists of 15 objective functions and 11 constraints, which was modified to 2 objective functions and 17 constraints.
- Because the calculation of the hypervolume does not allow negative values for the objective function which comes from the maximization of the torque (Trq_WP1), an offset of 7 was selected and a minimization of the difference to 7.
- 40 optimization runs were performed.





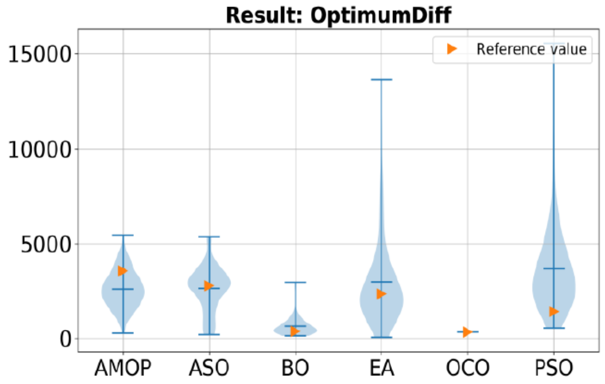
$Hypervolume: v = 13.497$
$Hypervolumereferenz:$
$obj\_cost\_indicator = 10$
$obj\_Trq\_WP1 = 7$

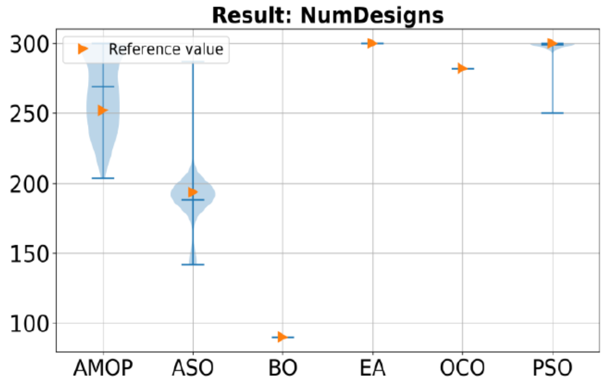
Customer Benchmark from Bosch

Example for Single-Objective-Optimization

RC18: Results – setting_300_90_300_400

- The best optimizer is again BO.
- The OCO performs much better with the double number of designs (200 → 400).
- All other algorithms do not find the optimum well.





We find the optimum in most repetitions

We need much less designs than other algorithms

17 CR/AME3 | 2023-06-22

BOSCH

Customer Benchmark from Bosch

In 5 out of 6 benchmarks our Bayesian optimization algorithm performed best considering:

- **Needed simulation runs**
- **Reproducibility**
- **Optimal result**

Benchmark of One-Click-Optimizer

Summary

- An automatic workflow could be established to benchmark different optimization algorithms.
- The integration of optiSLang-external algorithms is quite difficult. Several interfaces in Python were necessary to create the required files. Sometimes the OutputSlots like Ocriteria were used and sometimes the export of parameters/criteria via .csv format. It could be clarified whether a custom integration is a better approach.
- The adaptive and hybrid optimization algorithms showed the best performance. Often, the PI-BO showed the best results, but requires a long computation time. Perhaps the integration of PI-BO in optiSLang could be improved e.g. parallel training of criteria.
- The nature inspired optimization algorithms EA & PSO showed similar results, but they need much more designs for a good solution.
- The One Click Optimizer OCO does not show the best solution for all applications, but the OCO belongs to the better optimization algorithms.
- There are ideas to couple several methods sequentially to get better optimization results.

BOSCH

Example from chemistry – metalic coating development

In total 17 parameters

- 32 possible raw materials to choose from raw materials types (binder, addtives, flow Additives, …)
- Concentrations
- Process parameters (spray parameter, speed mixer, …)

Total 5 adaptation with 3 formulations = 15 formulations

Final adaptation

Is it possible to be even more ressource efficient?

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**

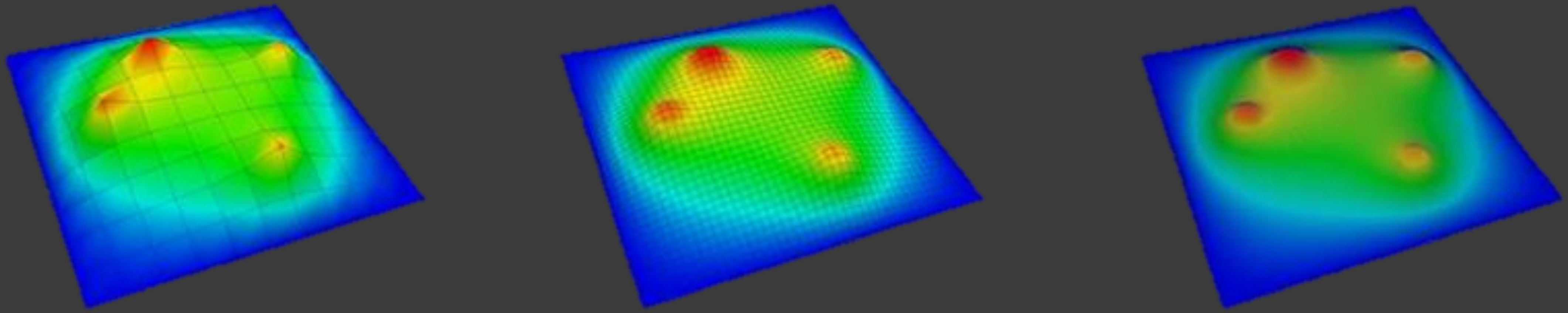
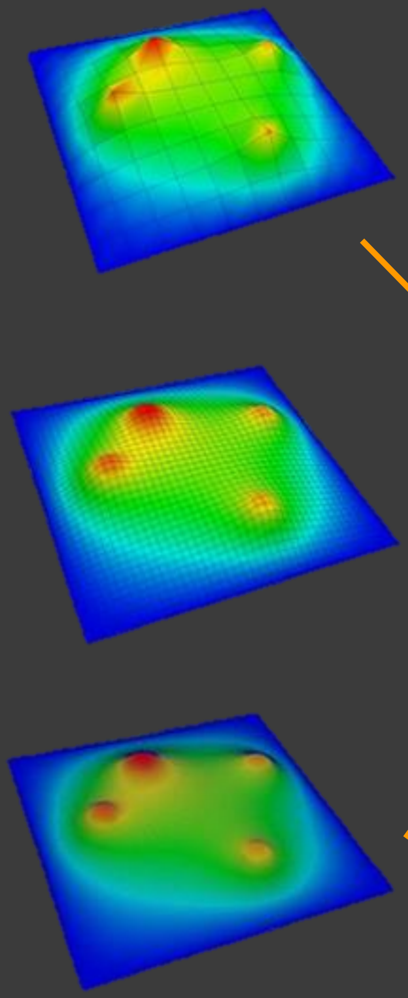
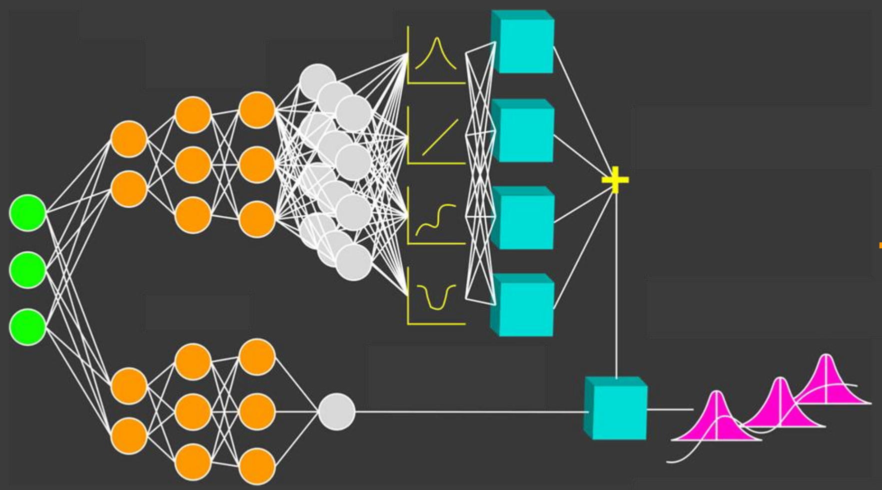


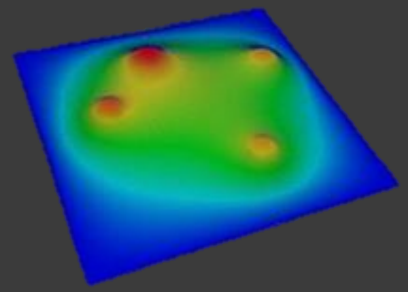
Image source: Aydin, Roland Can, Fabian Albert Braeu, and Christian Johannes Cyron. "General multi-fidelity framework for training artificial neural networks with computational models." *Frontiers in Materials* 6 (2019): 61.

What is multi-fidelity modling?



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





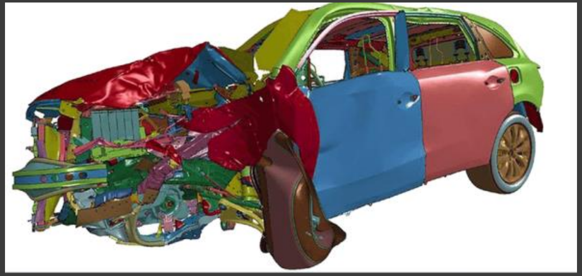
Advantages:

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

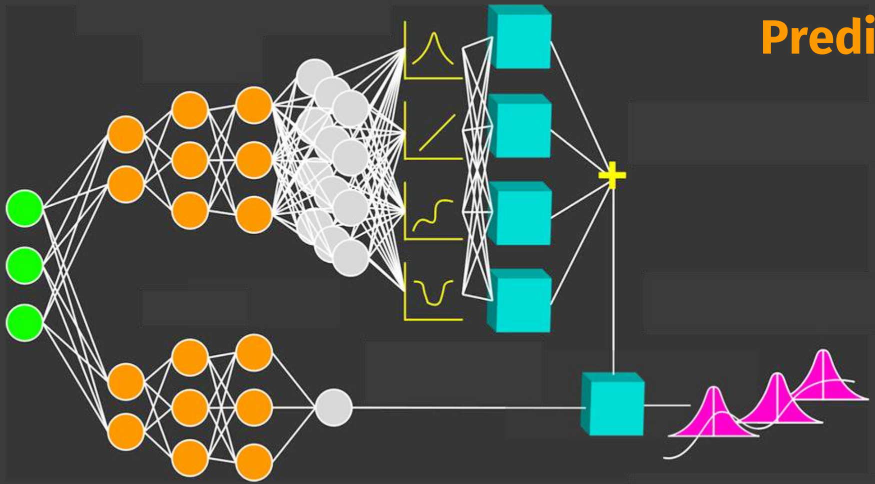
What is multi-fidelity modling?

Real experimental data

Simulation data



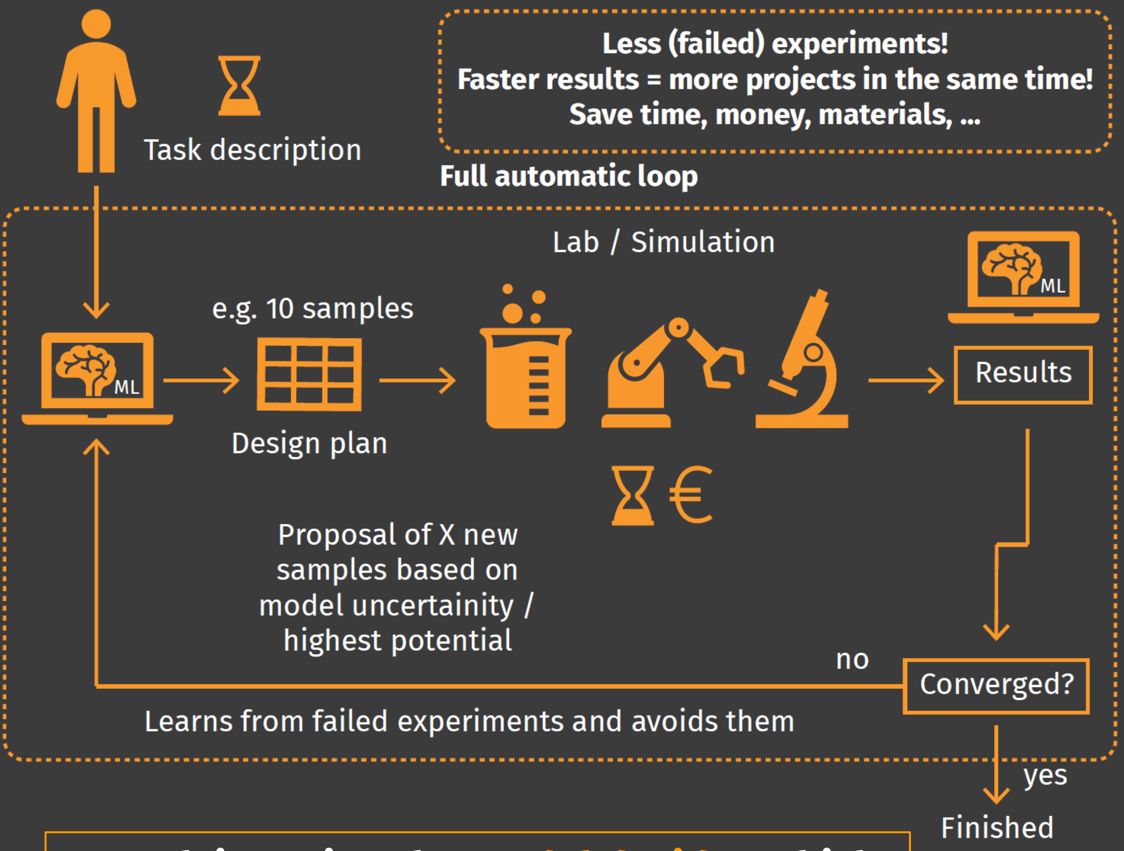
Model utilizes information from all fidelity sources and predicts in the highest fidelity quality



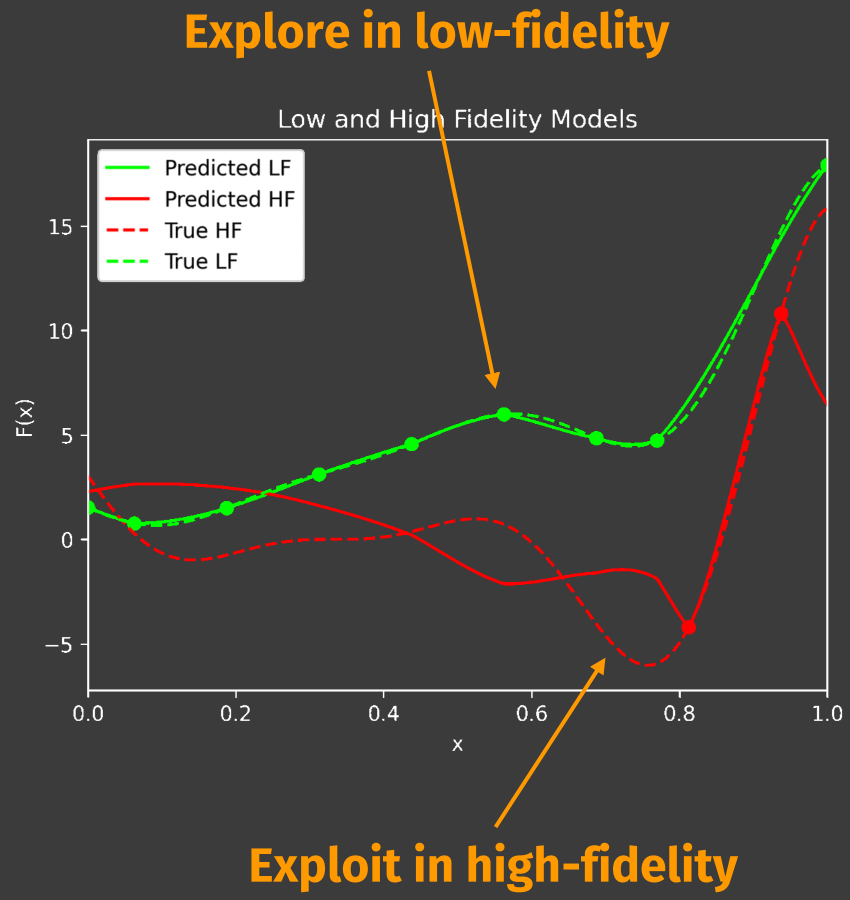
Prediction in quality of real experiment

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

Multi-fidelity optimization



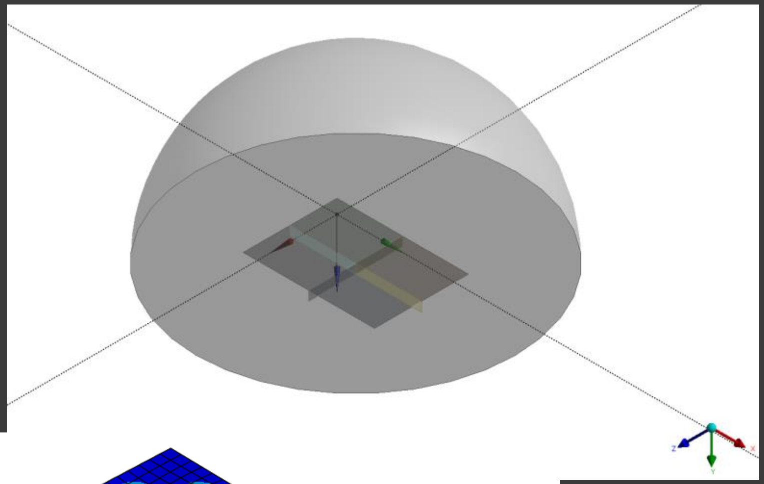
In each iteration the model decides which fidelity level is needed based on user specified costs

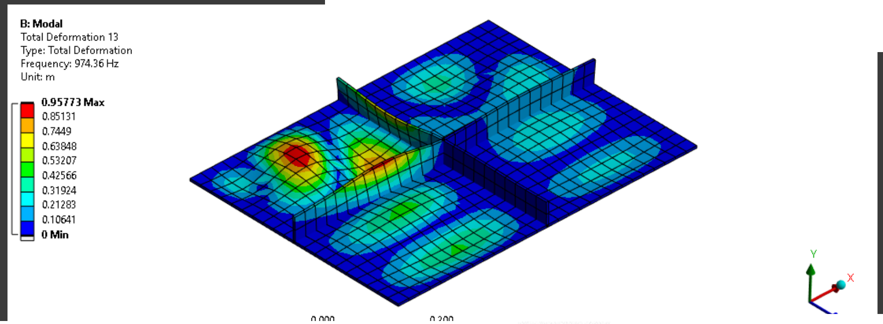


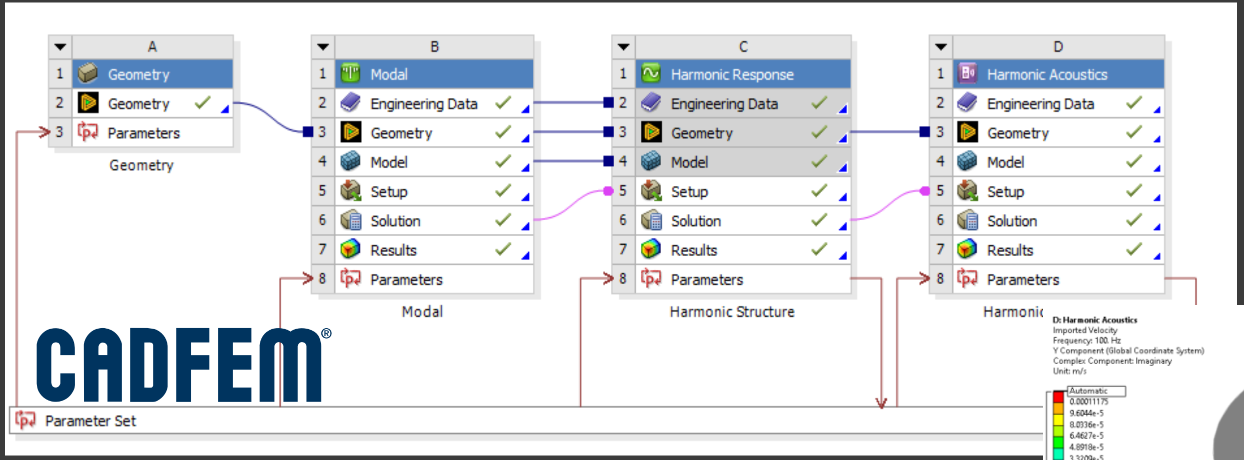
Example: sound radiation of a stiffened plate

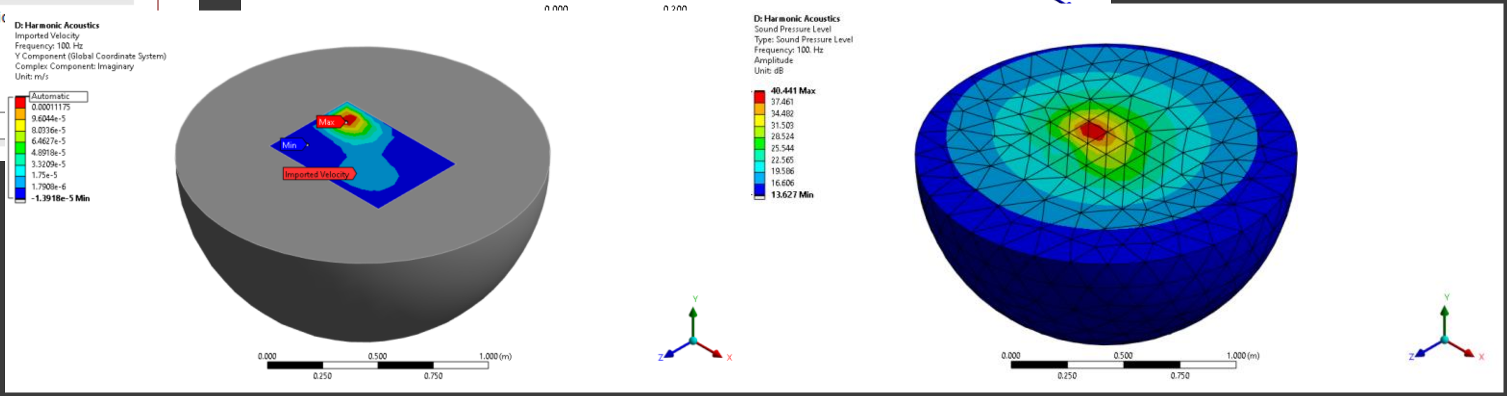
Example - sound radiation of a stiffened plate

- **Two use cases:**
  - **1 minimize maximum of the far-field sound power level (scalar output)**
  - **Build global accurate model to replace simulation (signal output)**









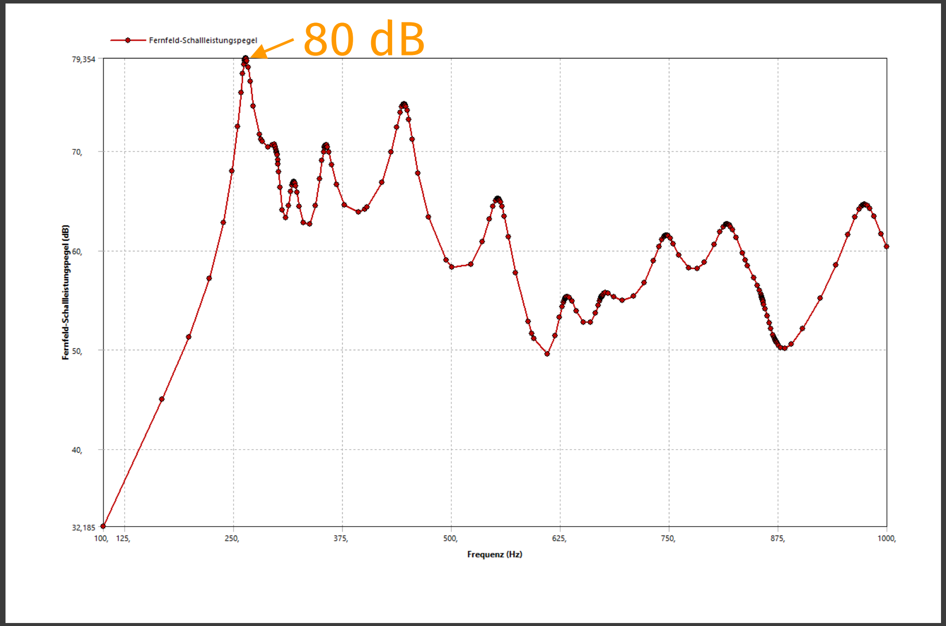
Example - sound radiation of a stiffened plate

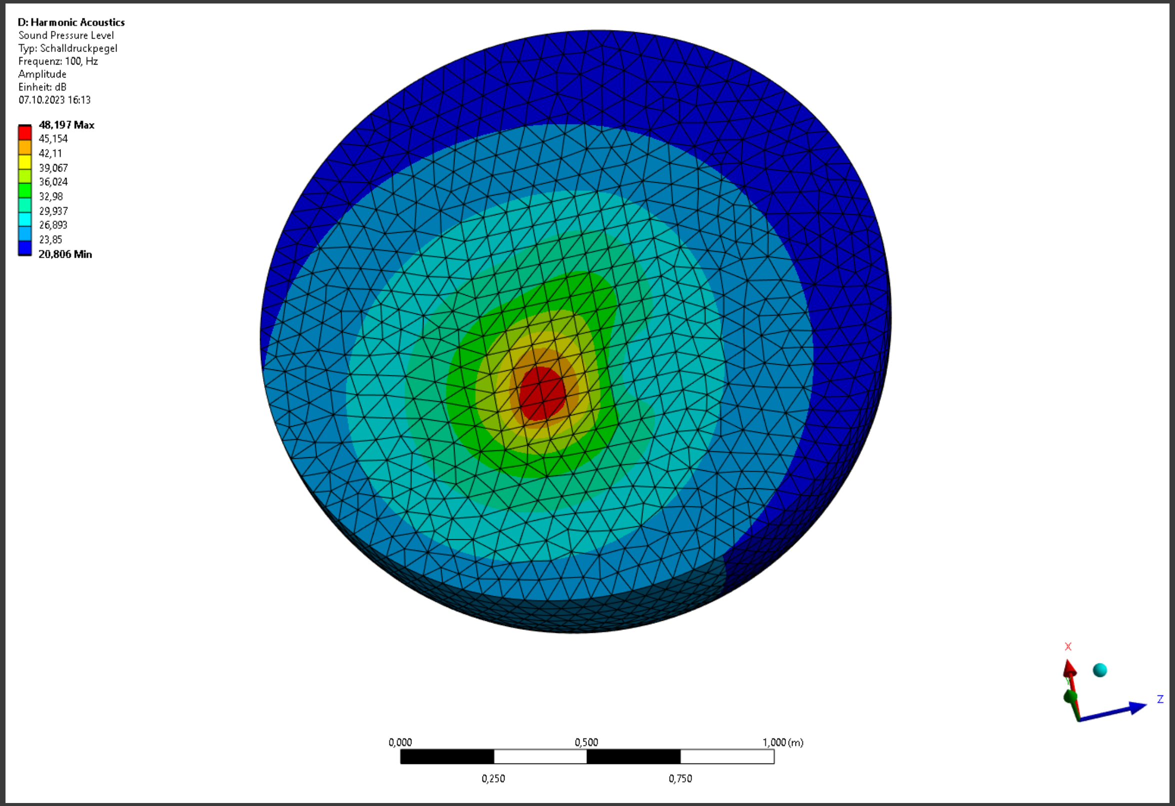
Input parameter	Comment
R_frac	Fraction of largest acoustic wavelength
EPW	Elements per wavelength
h_stiff	Height of the stiffeners
t	Thickness of plate and stiffeners
w	Width of the plate
l	Length of plate
Point mass	Point mass at force excitation node
OffsetNodeX	Coordinate of force excitation node
OffsetNodeZ	Coordinate of force excitation node

Parameters which controll the accuracy of the simulation (In this example only EPW was used)

Example - sound radiation of a stiffened plate

- Reference accuracy with EPW = 6 & R_frac = 0.25 -> Simulation time 45 minutes

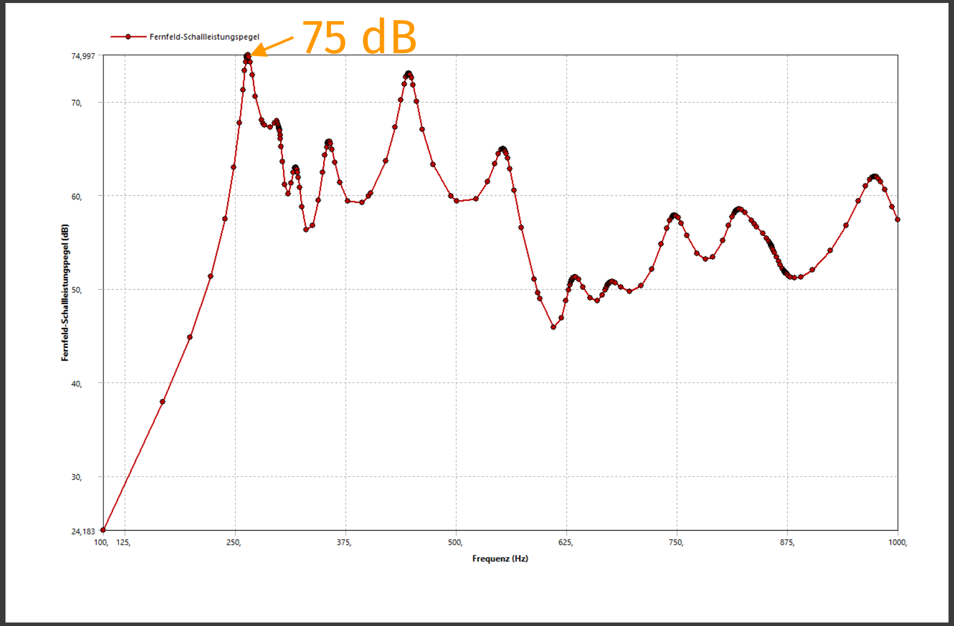


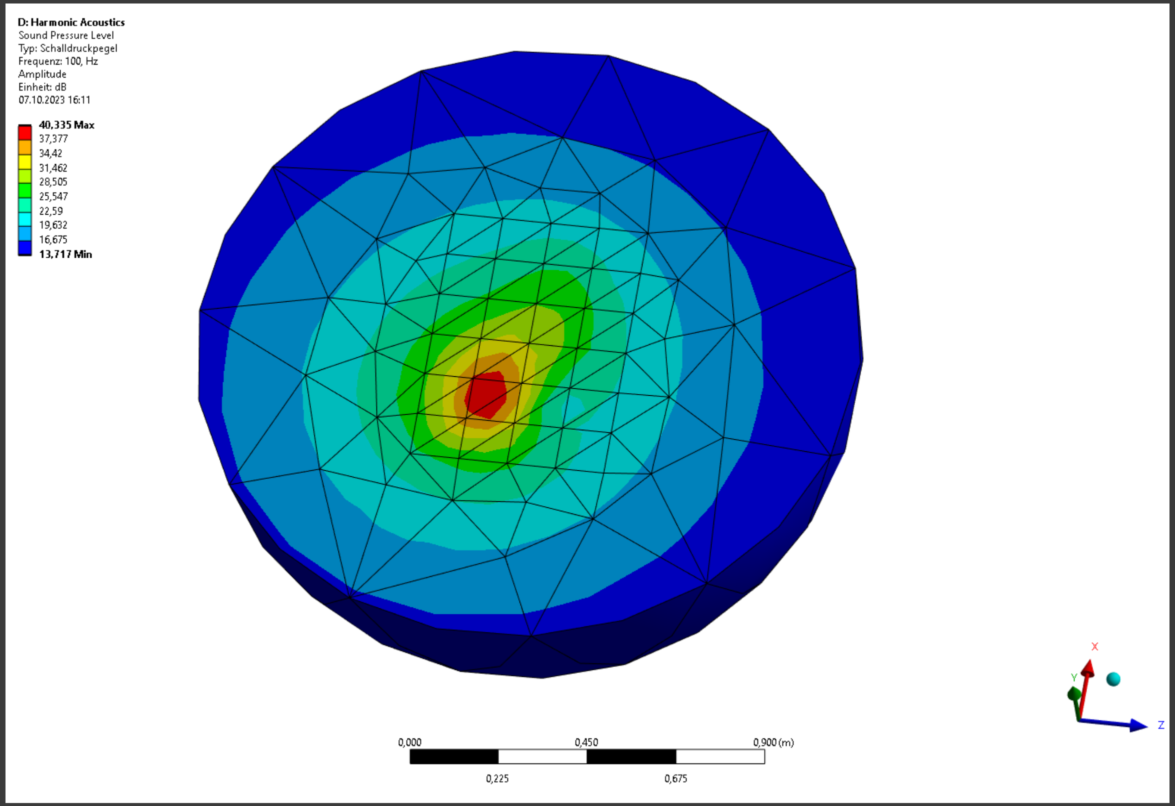


Output to be learned up to 400 discretization points

Example - sound radiation of a stiffened plate

- Reference accuracy with EPW = 0.5 & R_frac = 0.25 -> Simulation time 1 minutes

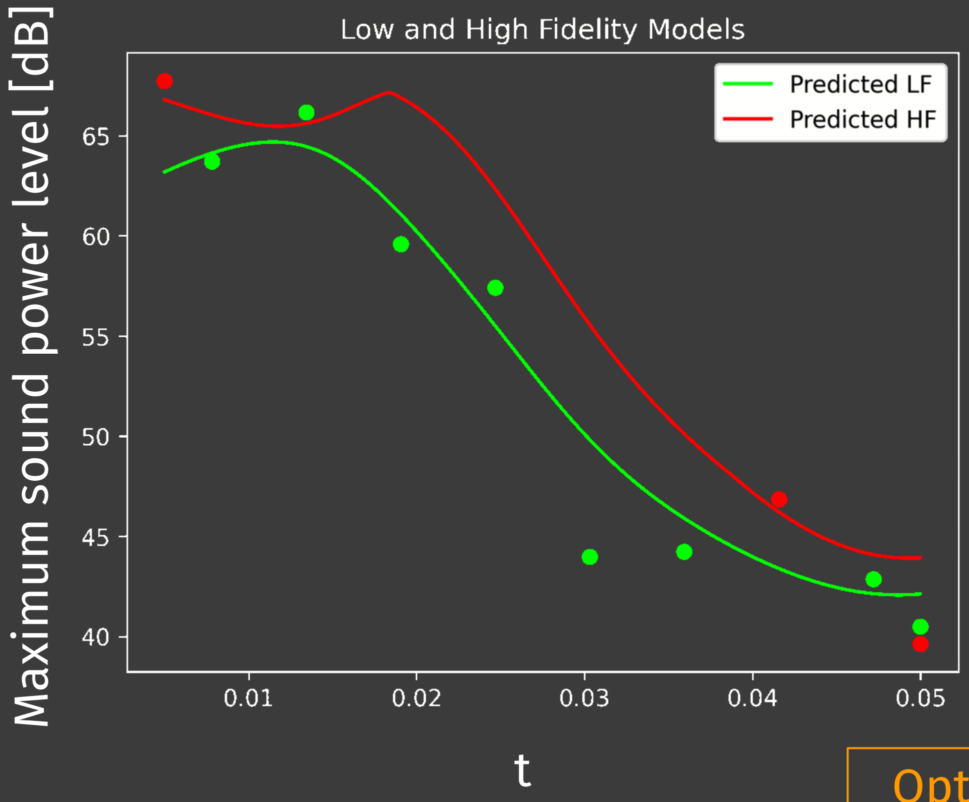




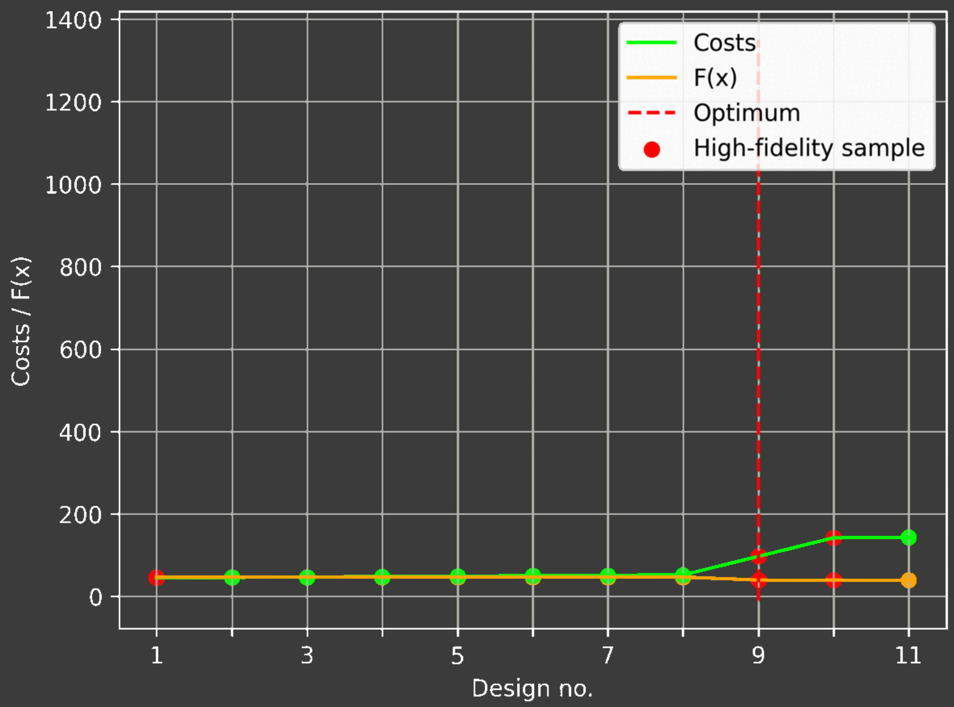
Output to be learned up to 400 discretization points

Example - sound radiation of a stiffened plate

- The used costs for the multi-fidelity optimization are the simulation times 1 minute & 45 minutes



First 10 designs are start samples

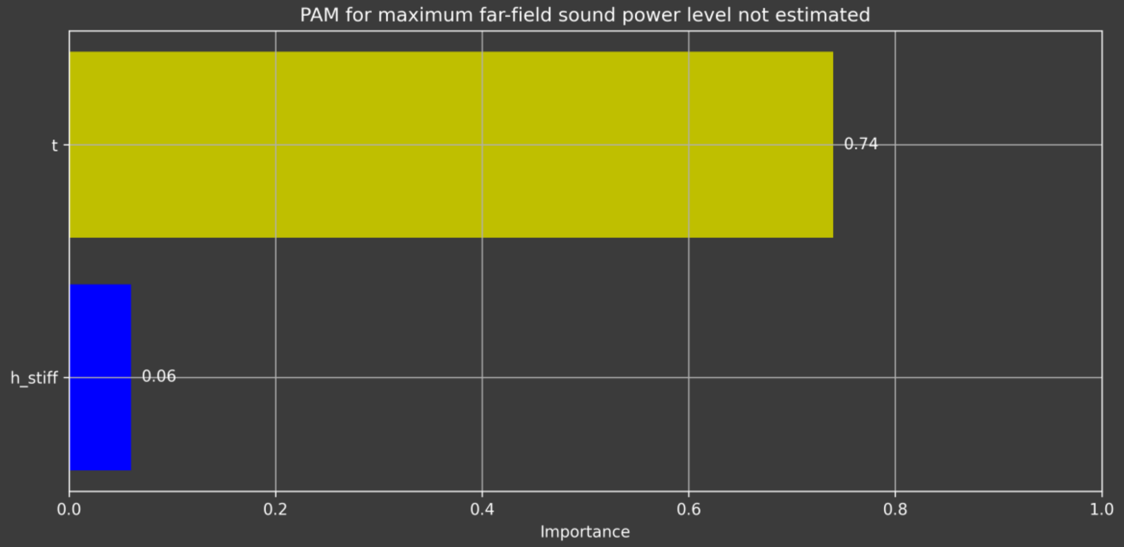


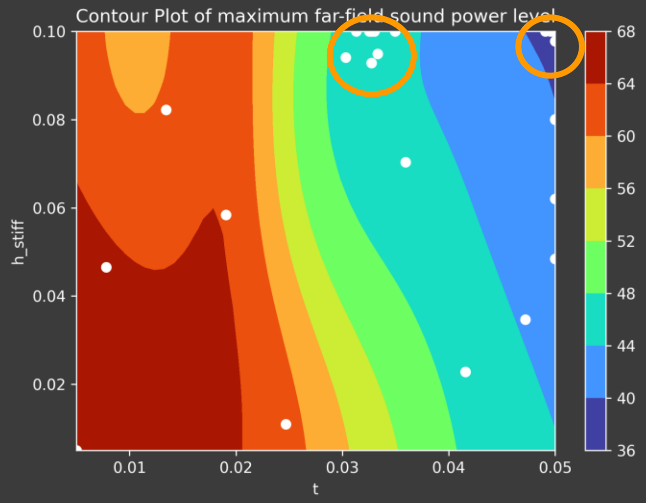
Optimization use-case

Low-fidelity simulation
High-fidelity simulation

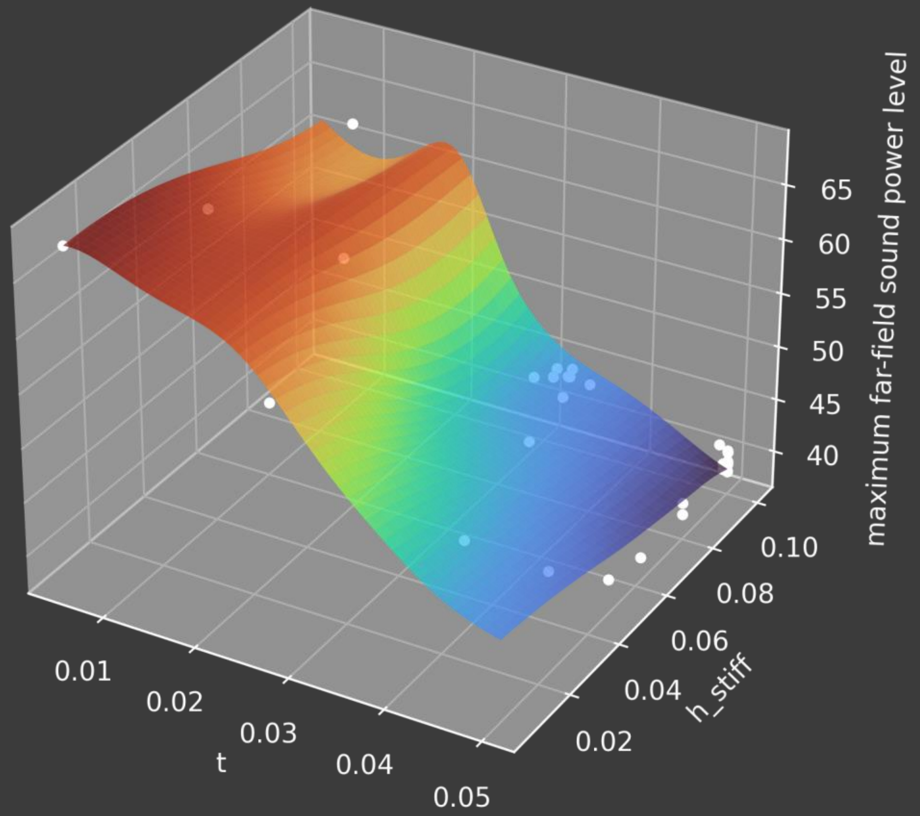
Example - sound radiation of a stiffened plate

Sensitivity analysis



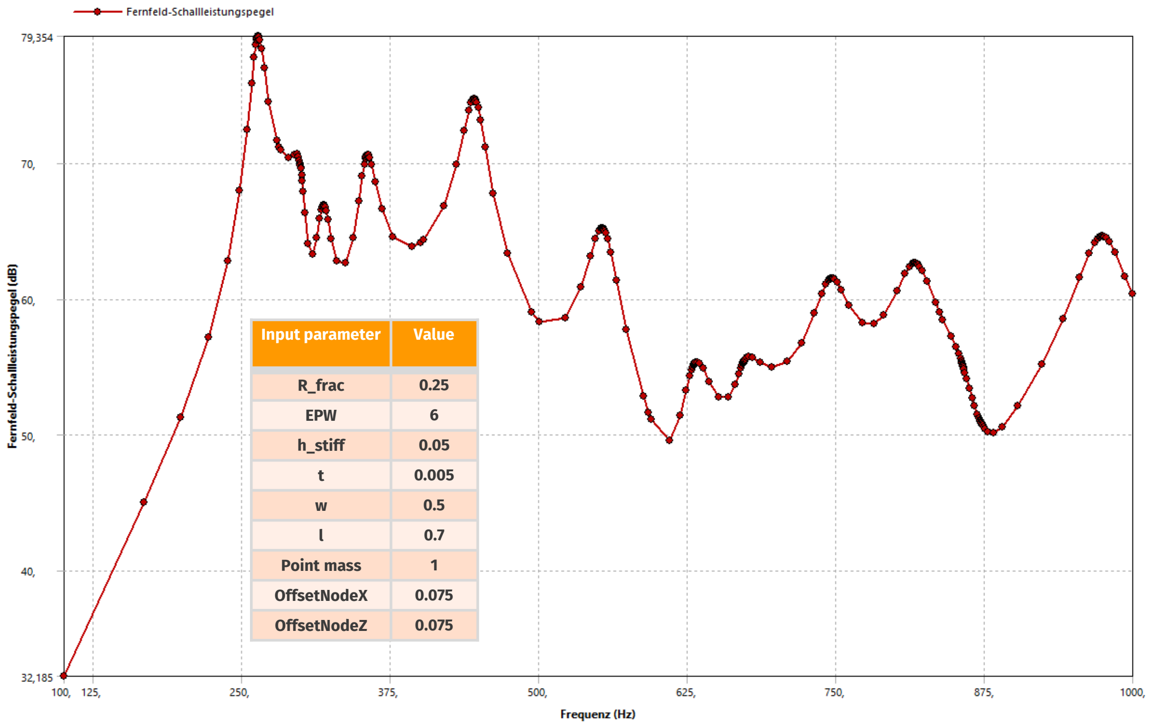


Learned model for maximum far-field sound power level



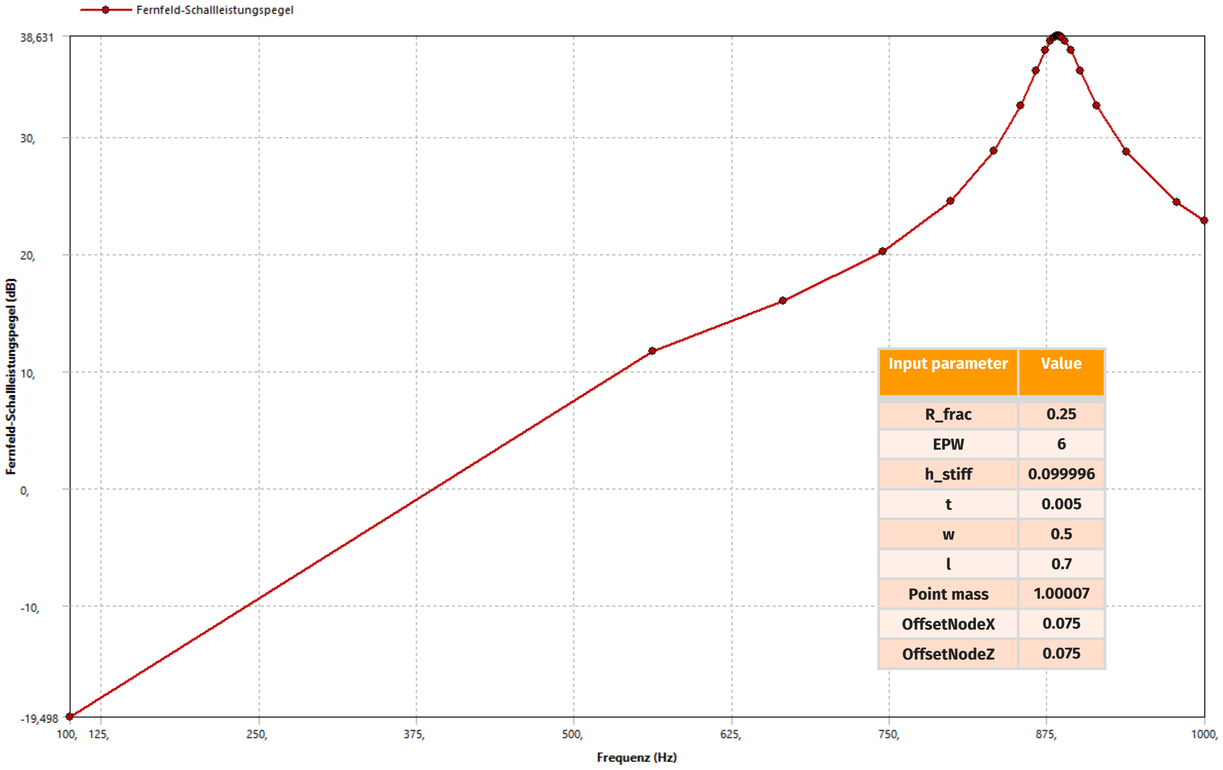
Example - sound radiation of a stiffened plate

Reference design maximum: 79 dB



Input parameter	Value
R_frac	0.25
EPW	6
h_stiff	0.05
t	0.005
w	0.5
l	0.7
Point mass	1
OffsetNodeX	0.075
OffsetNodeZ	0.075

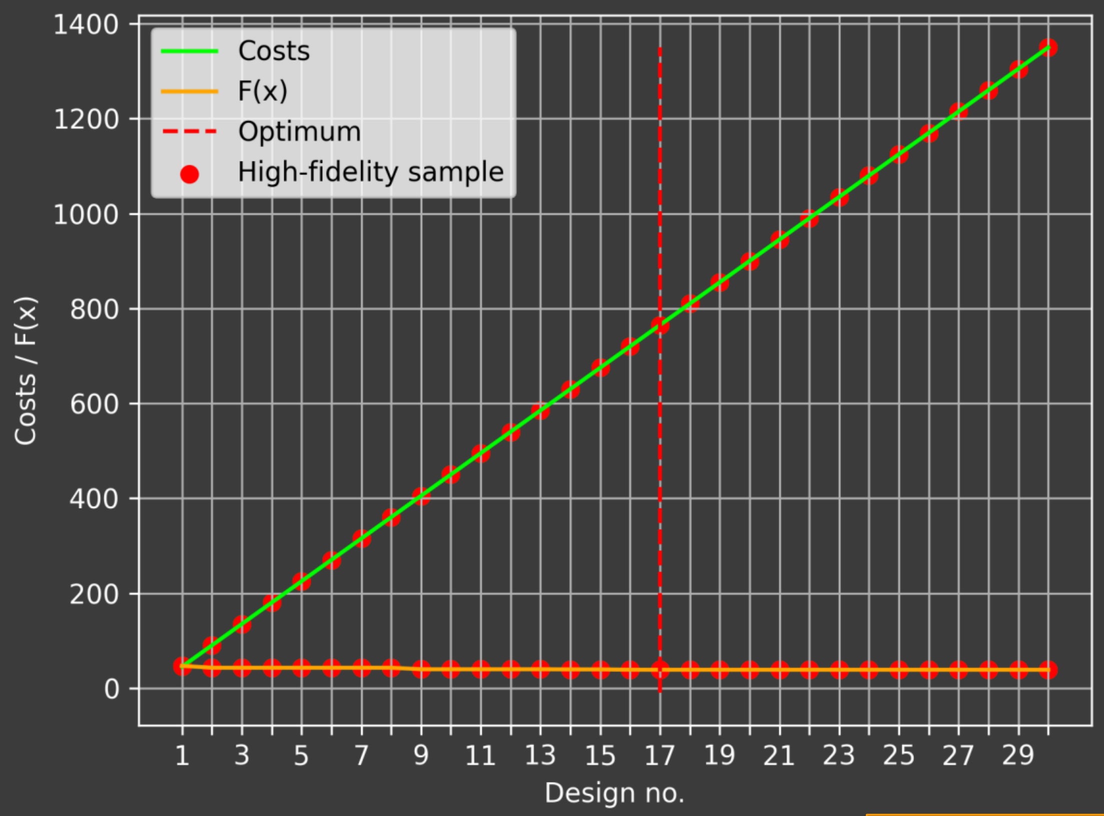
Optimized design maximum: 39 dB



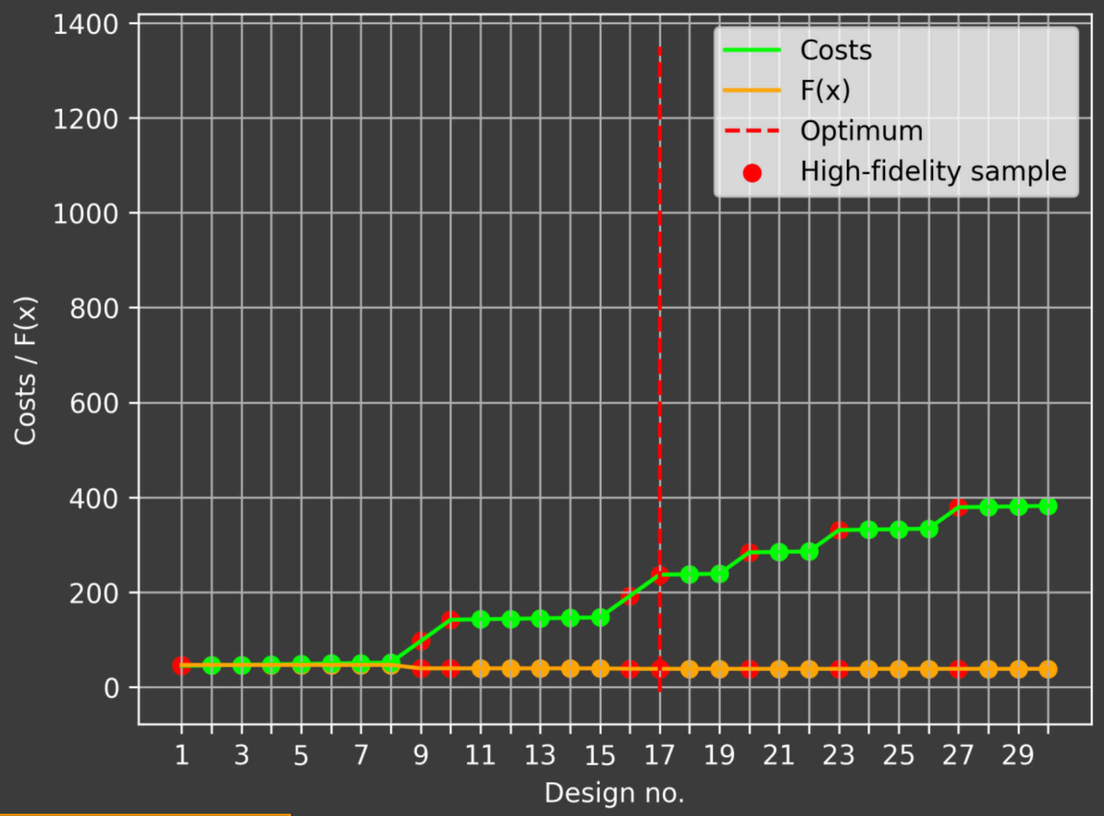
Input parameter	Value
R_frac	0.25
EPW	6
h_stiff	0.099996
t	0.005
w	0.5
l	0.7
Point mass	1.00007
OffsetNodeX	0.075
OffsetNodeZ	0.075

Example - sound radiation of a stiffened plate

Only high-fidelity optimization



Multi-fidelity optimization



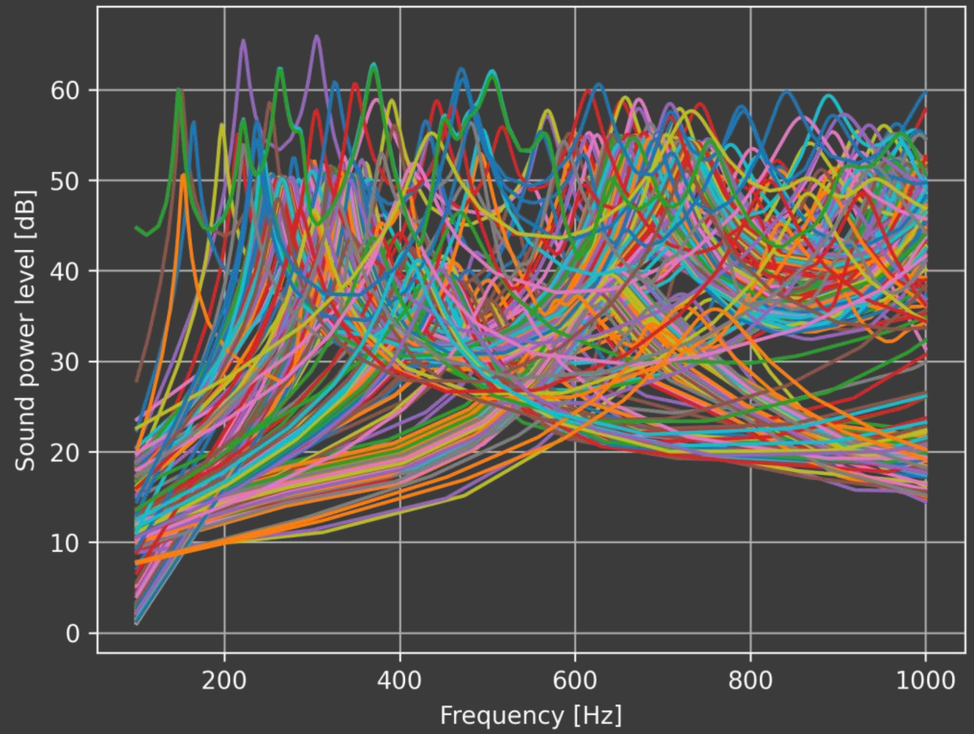
Simulationtime:
1350 minutes vs. 382 minutes
353% faster

Low-fidelity simulation
High-fidelity simulation

Example - sound radiation of a stiffened plate

- Global modeling took 310 low-fidelity and 39 high-fidelity calculations to obtain a good model
- Since the output consists of 400 discrete points a larger number of training samples is needed

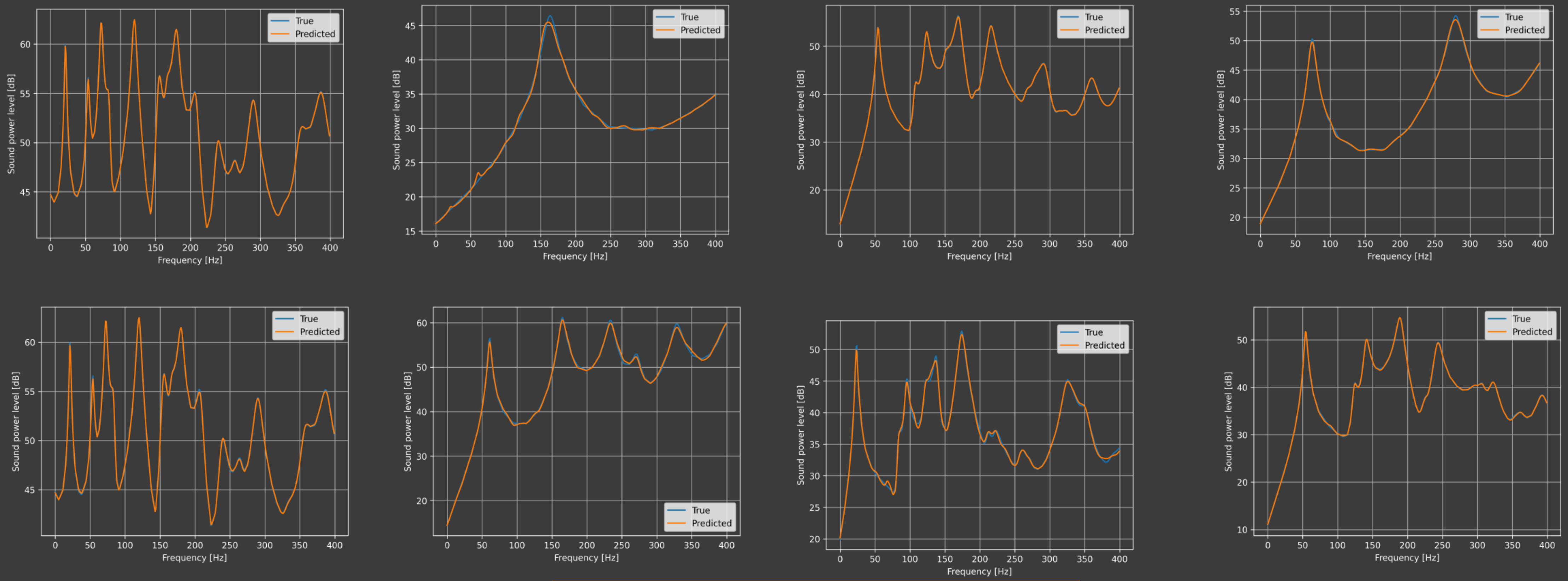
Global model use-case to replace simulation



Visualization of different training samples

Example - sound radiation of a stiffened plate

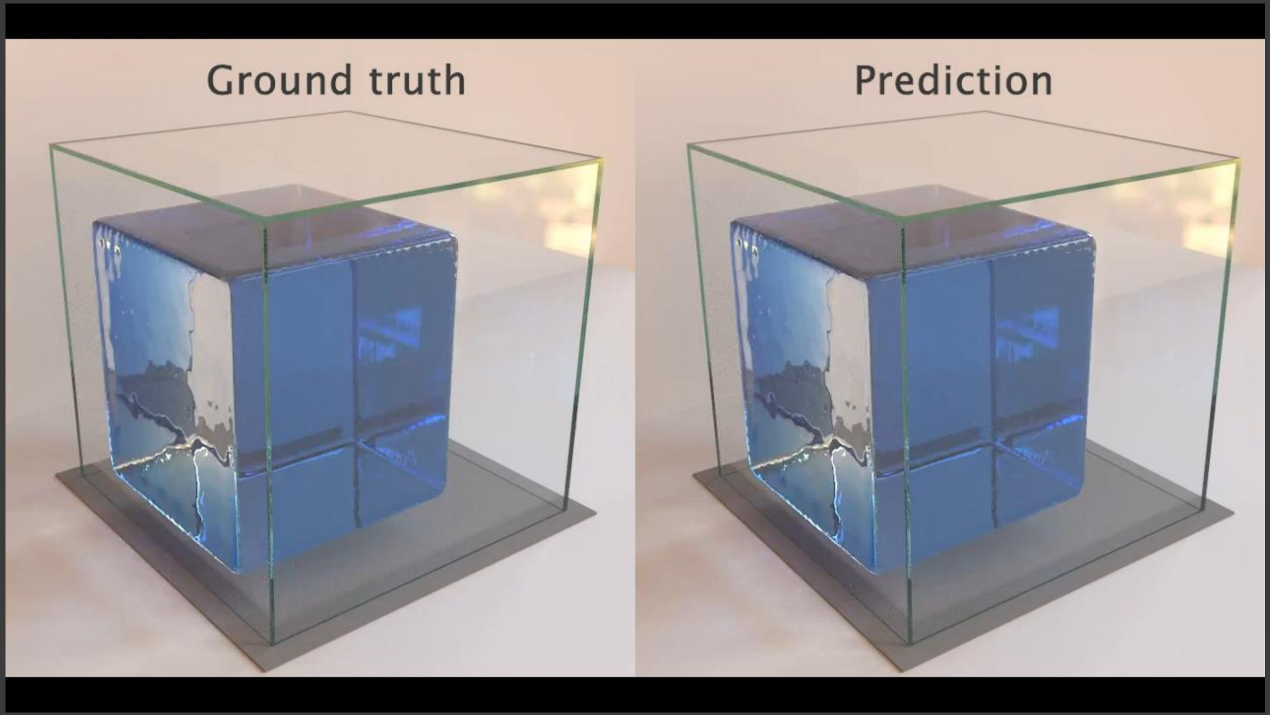
Results on test data

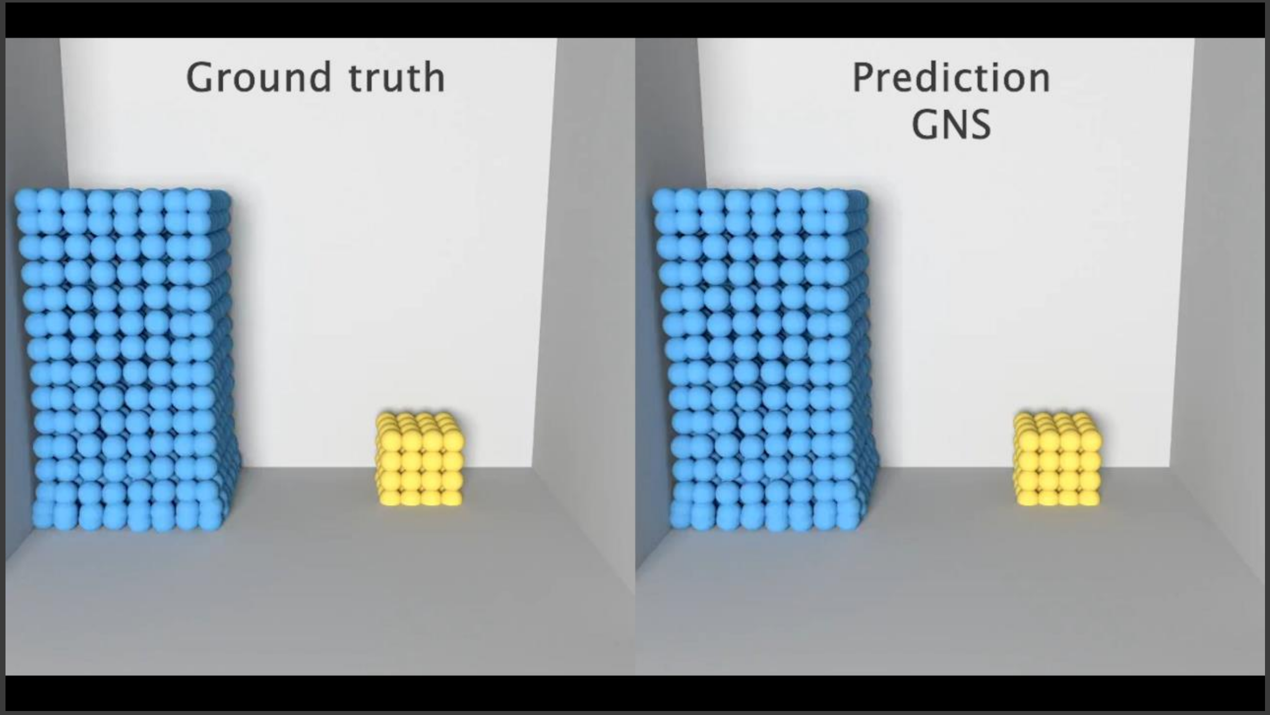


Simulationtime:
15975 minutes vs. 2065 minutes
773% faster

2D / 3D Simulation data of parametrized and non-parametrized geometries

Complex physics (Google Deepmind 2020)



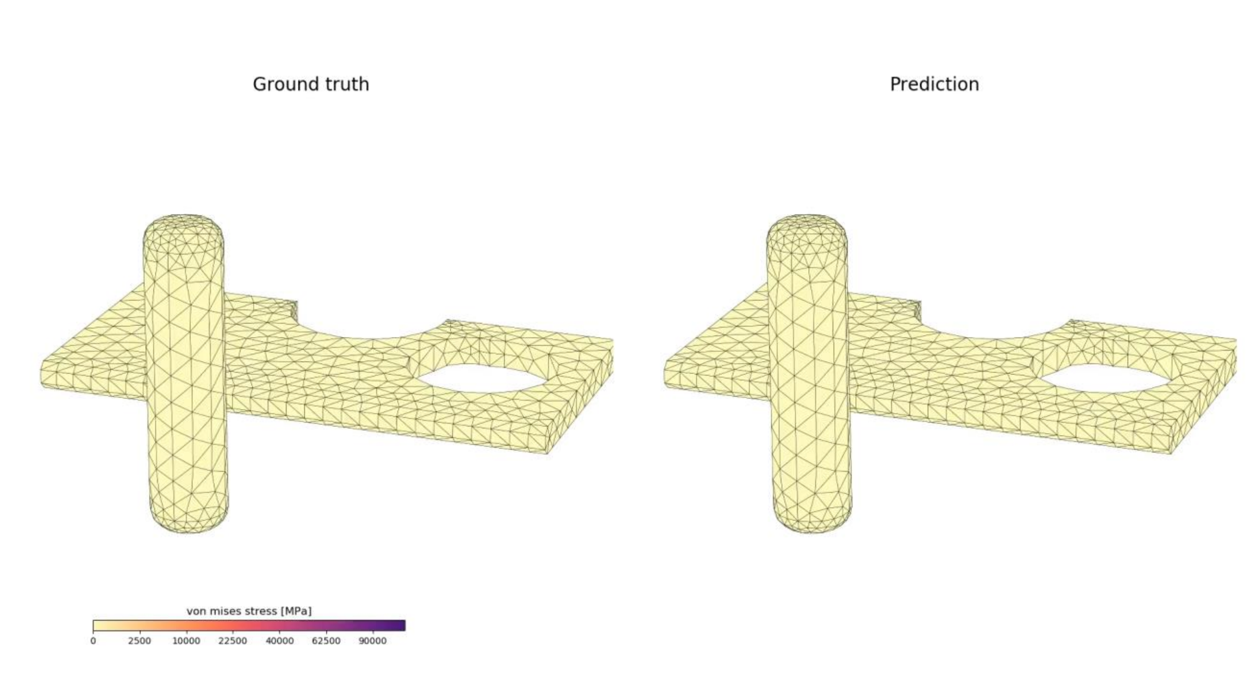


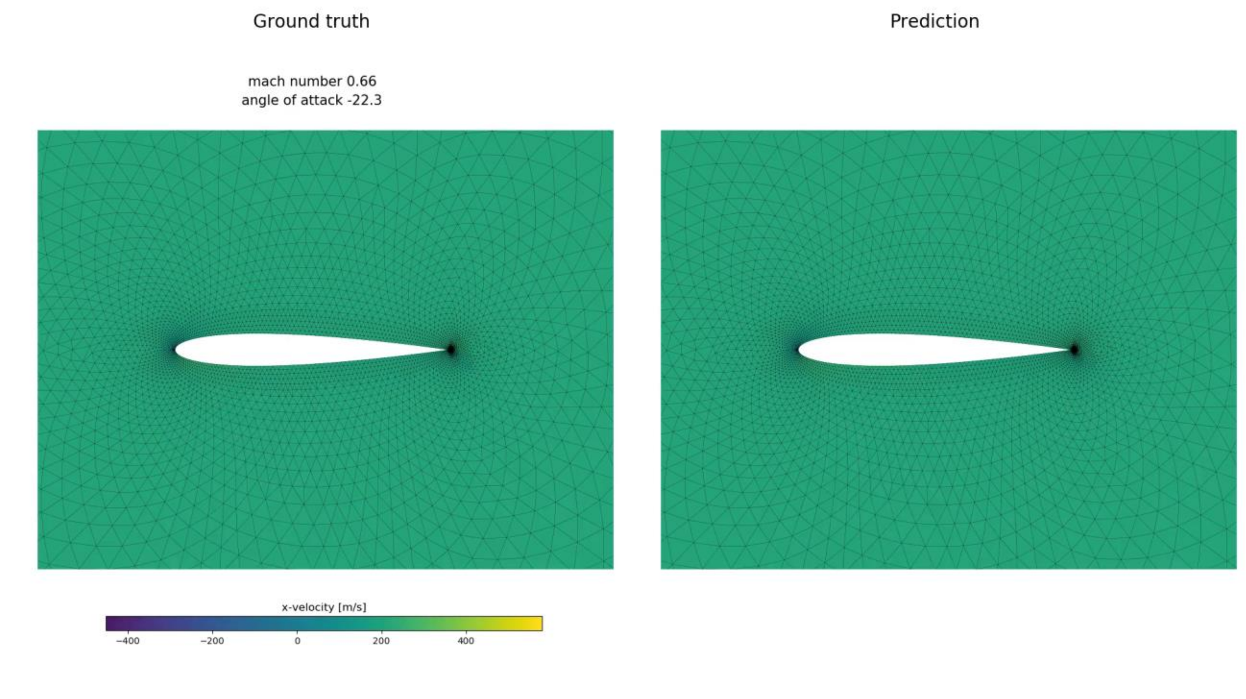
- 14k particles
- 800 steps

- 1k particles
- 150 steps

Sanchez-Gonzalez, Alvaro, et al. "Learning to simulate complex physics with graph networks." *International conference on machine learning*. PMLR, 2020.

3D FEM / 2D CFD (Google Deepmind 2021)





- 10 Million training steps on 1,000 samples
- 1,271 nodes (avg.) / 400 time steps
- 1 NVIDIA v100 GPU training time >= 102 hours
- 1 CPU (8 cores) training time >= 1330 hours
- RMSE 1-step prediction disp.: 0.25 x 10e-3

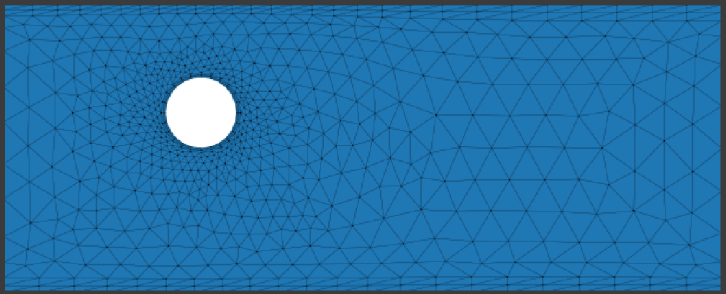
- 10 Million training steps on 1,000 samples
- 5,233 nodes / 600 time steps
- 1 NVIDIA v100 GPU training time >= 66 hours
- 1 CPU (8 cores) training time >= 478 hours
- RMSE 1-step prediction: 0.314

*T.Pfaff, M. Fortunato, A. Sanchez-Gonzalez, & P. Battaglia (2021). Learning Mesh-Based Simulation with Graph Networks. In *International Conference on Learning Representations*.

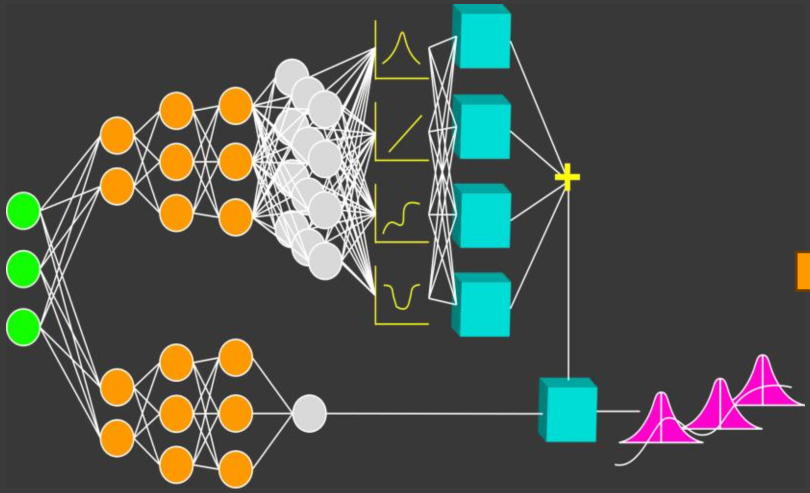
Geometric Deep Infinite Mixture of Gaussian Processes

Input

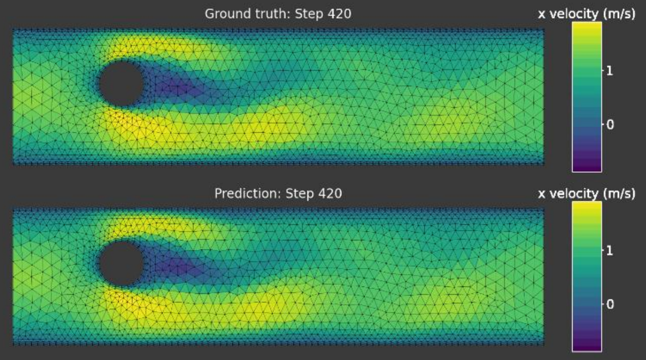
Mesh node positions + initial node features / boundary conditions (e.g. stress, velocity, ...) + optional global features



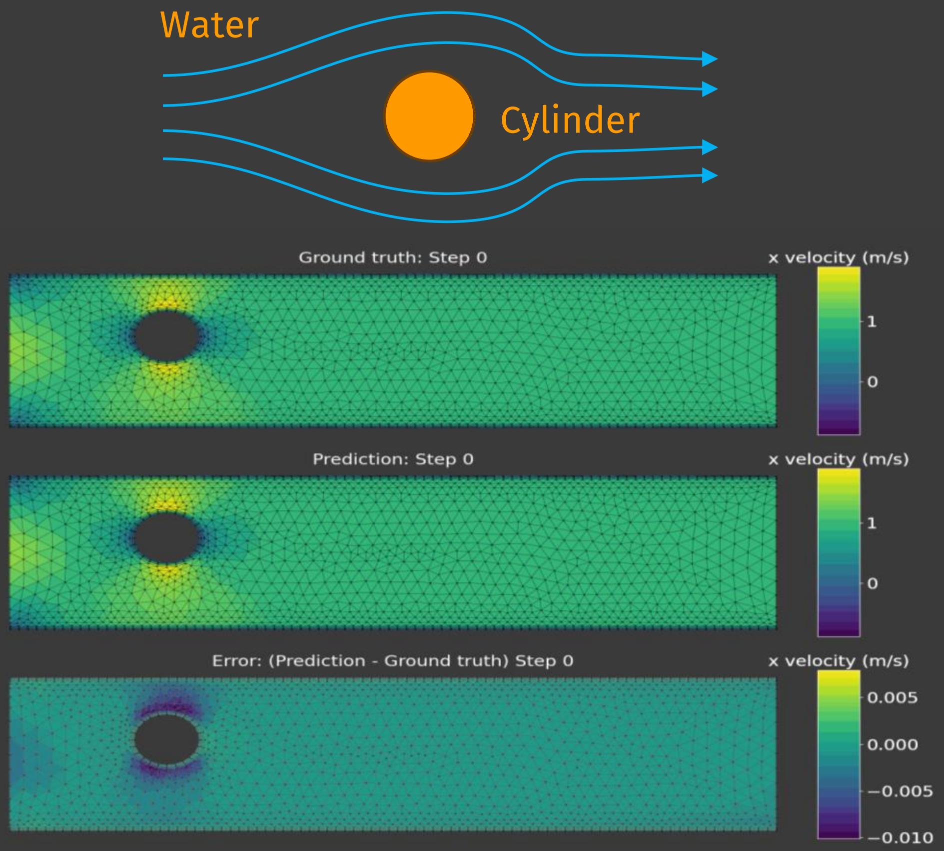
Geometric DIM-GP

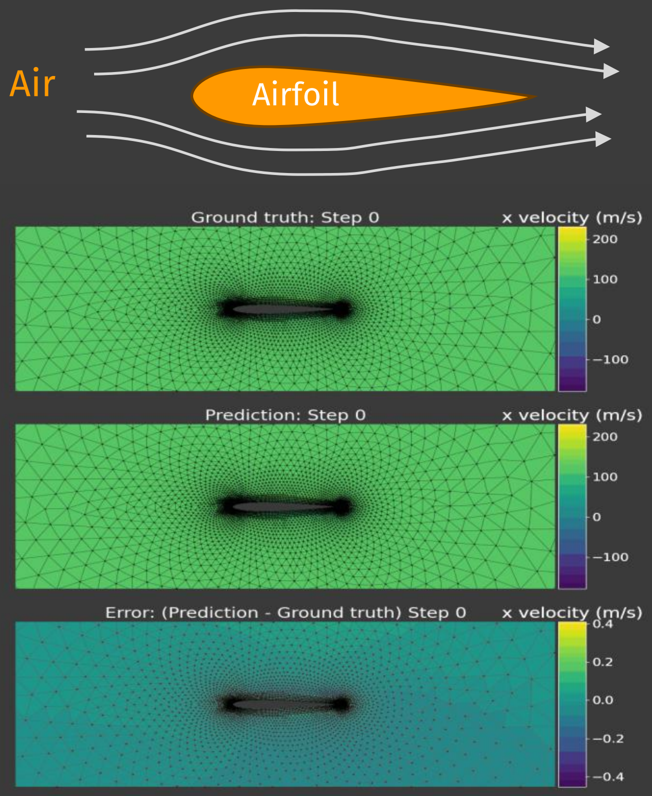


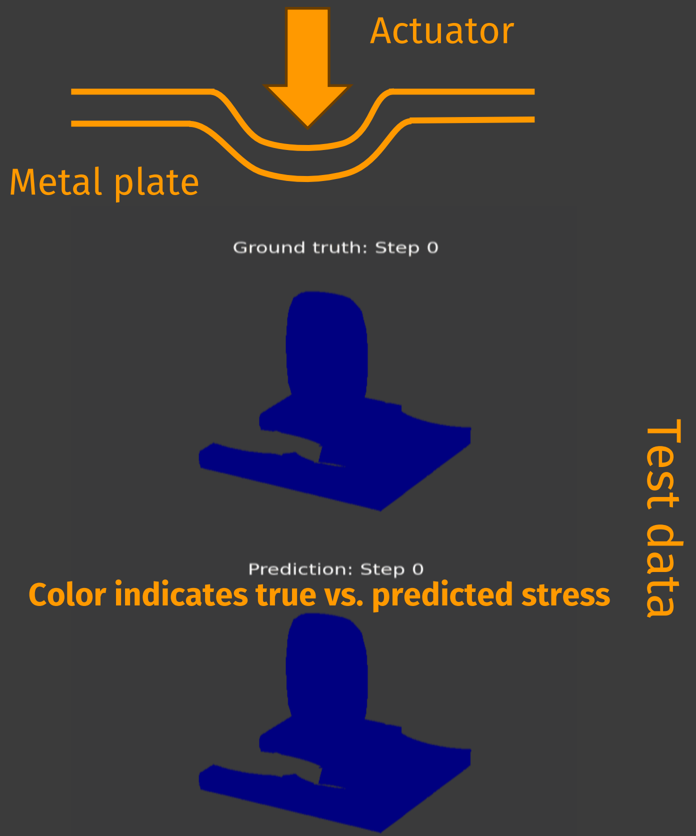
ML-based predictions of (transient) FEM / CFD results



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**

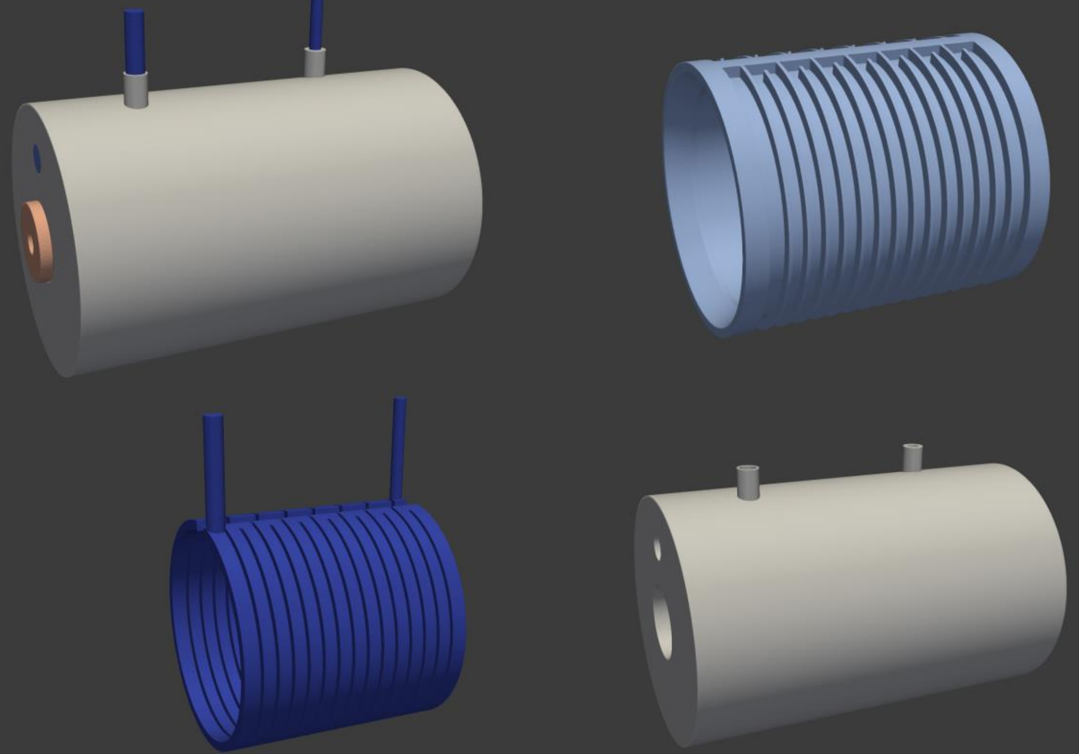
- **30,606 training steps = 1 Epoch on 5 samples**
- **1 NVIDIA 4090 GPU training time 14 minutes**
- **1 CPU (8 cores) training time 32 minutes**
- **RMSE 1-step prediction: 0.05**

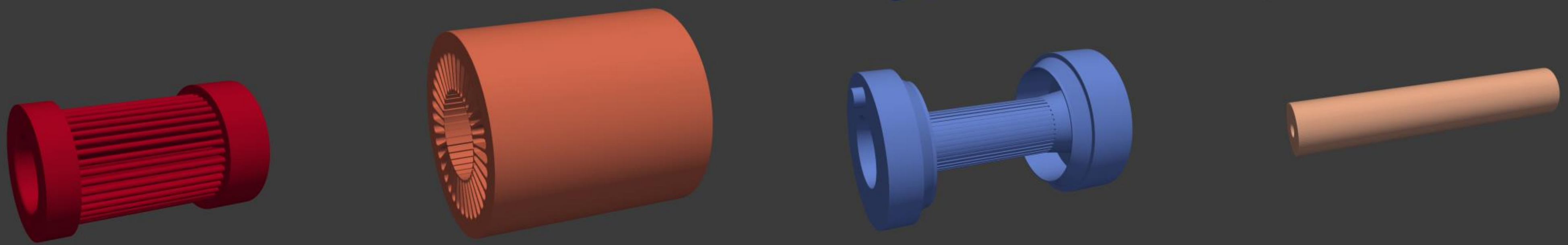
- **3,035 training steps = 1 Epoch on 5 samples**
- **1 NVIDIA 4090 GPU training time 2 minutes**
- **1 CPU (8 cores) training time 7 minutes**
- **RMSE 1-step prediction: 0.55 x 10e-4**

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

Complete

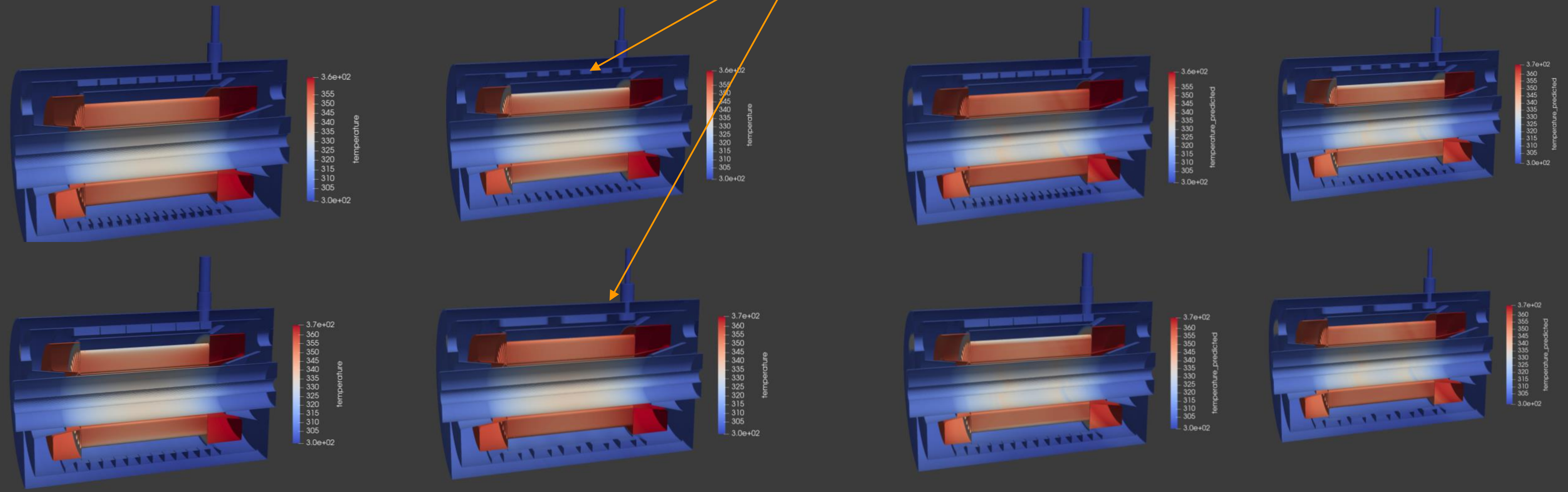




Supervised learning: E-Motor Cooling

Test designs (new cooling channel geometries)

Simulation (6-10h) | Prediction (30 sec)

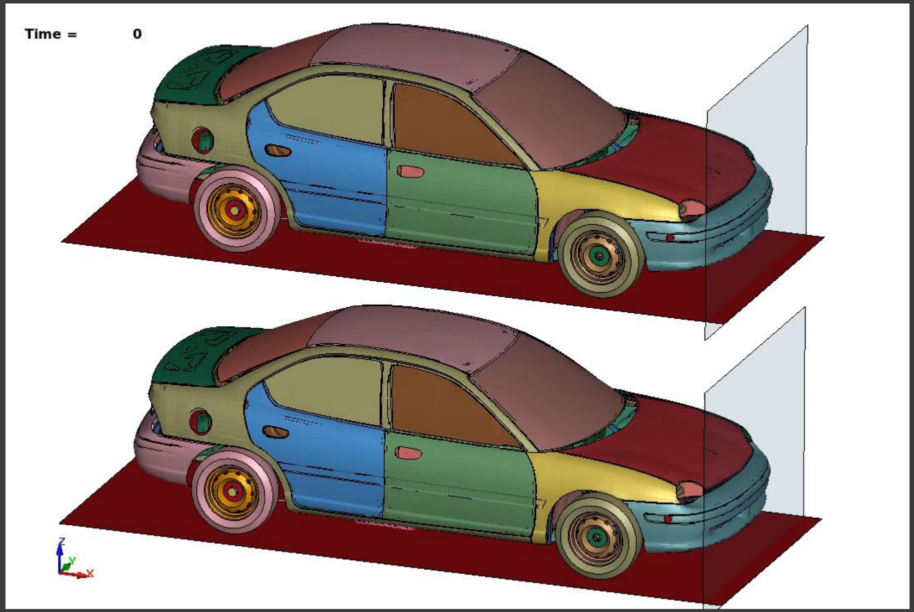


Field	Abs. Error over all nodes		Rel. Abs. Error over all nodes [%]		R2 over all nodes	
Temperature	Mean	Std	Mean	Std	Mean	Std
	1.395	0.312	0.419	0.09	0.993	0.002

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

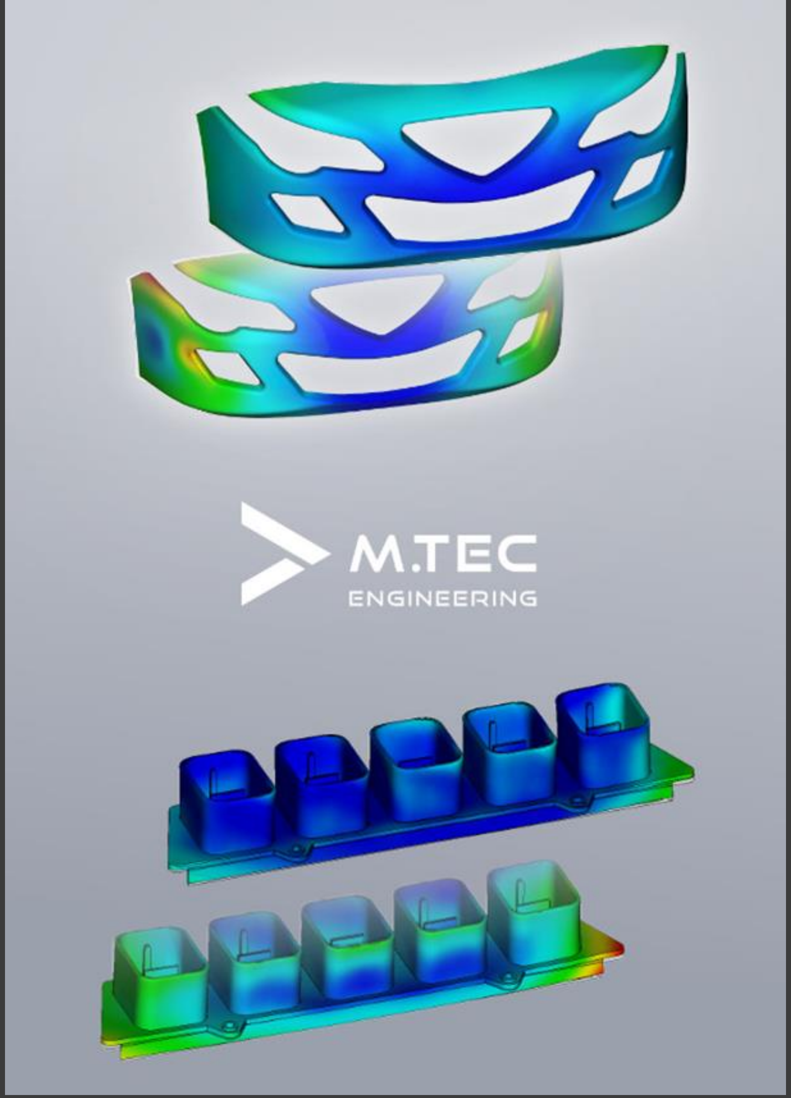
Crash simulation

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%

AI-based warpage optimization of plastic components



Automotive Front Bumper

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

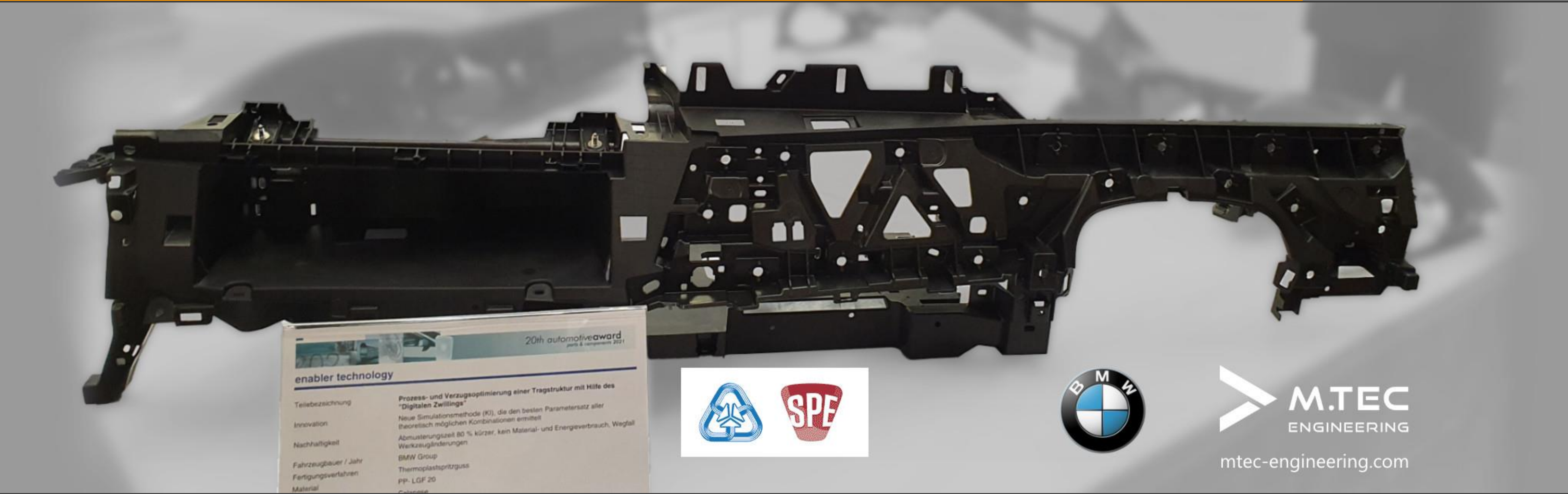
High Precision Connector

- 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

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

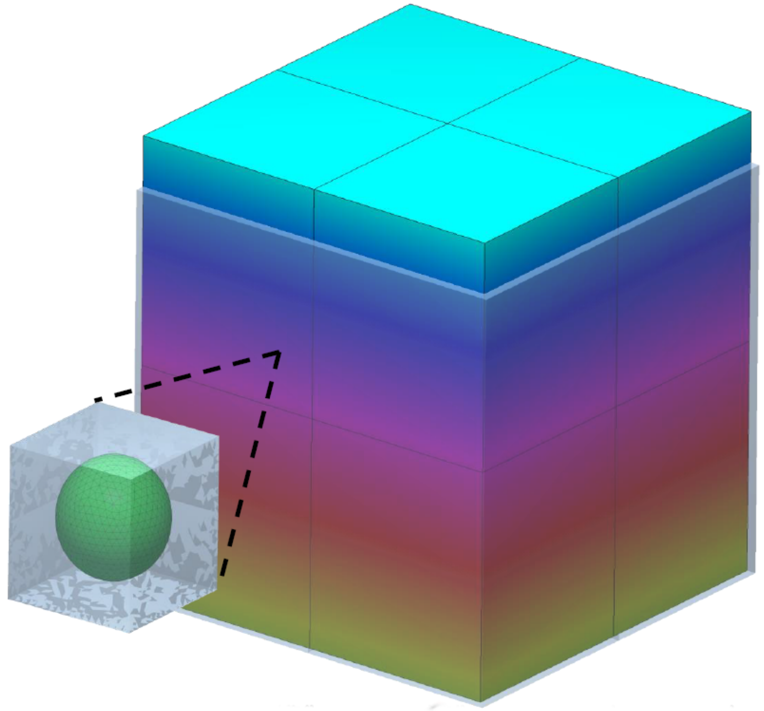
- Trained model on cube

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

Materialtensor

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

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



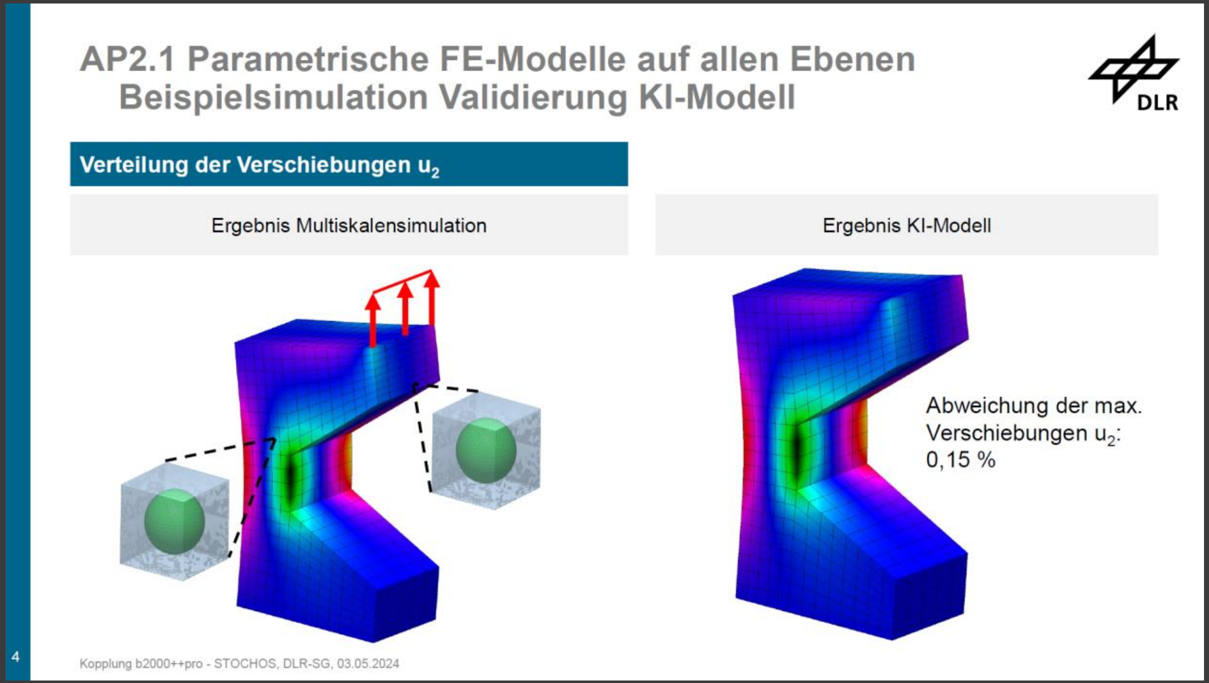
Simulationsergebnis Multiskalensimulation [DLR-SG]

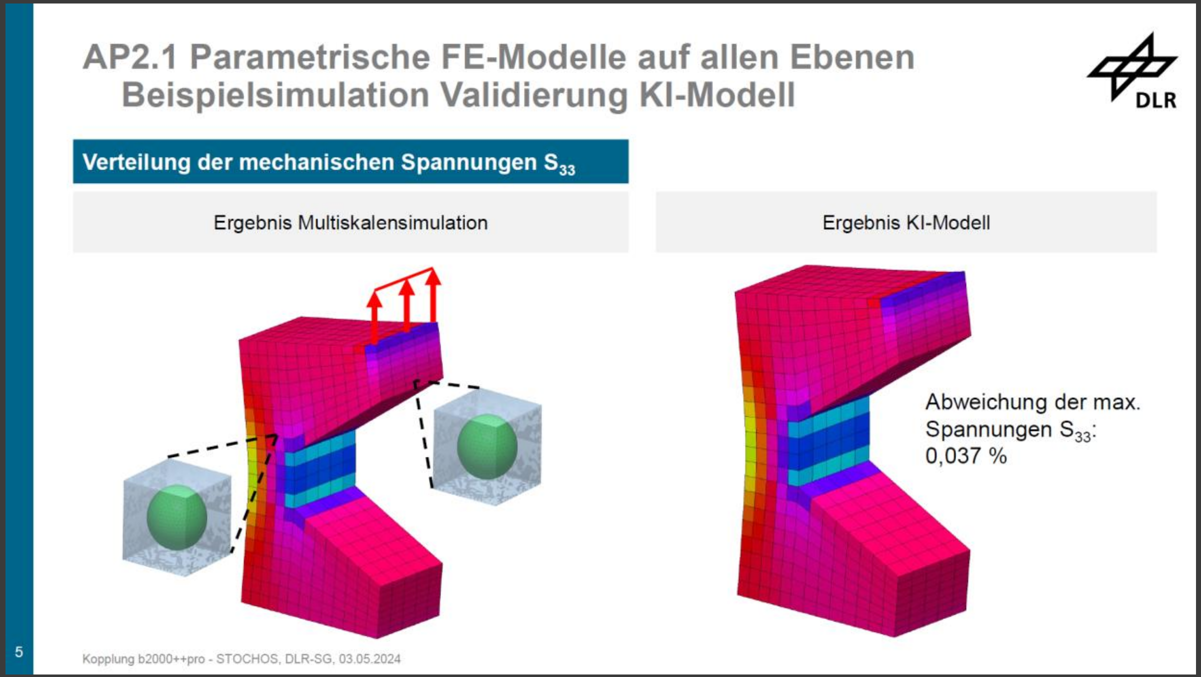
3

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

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

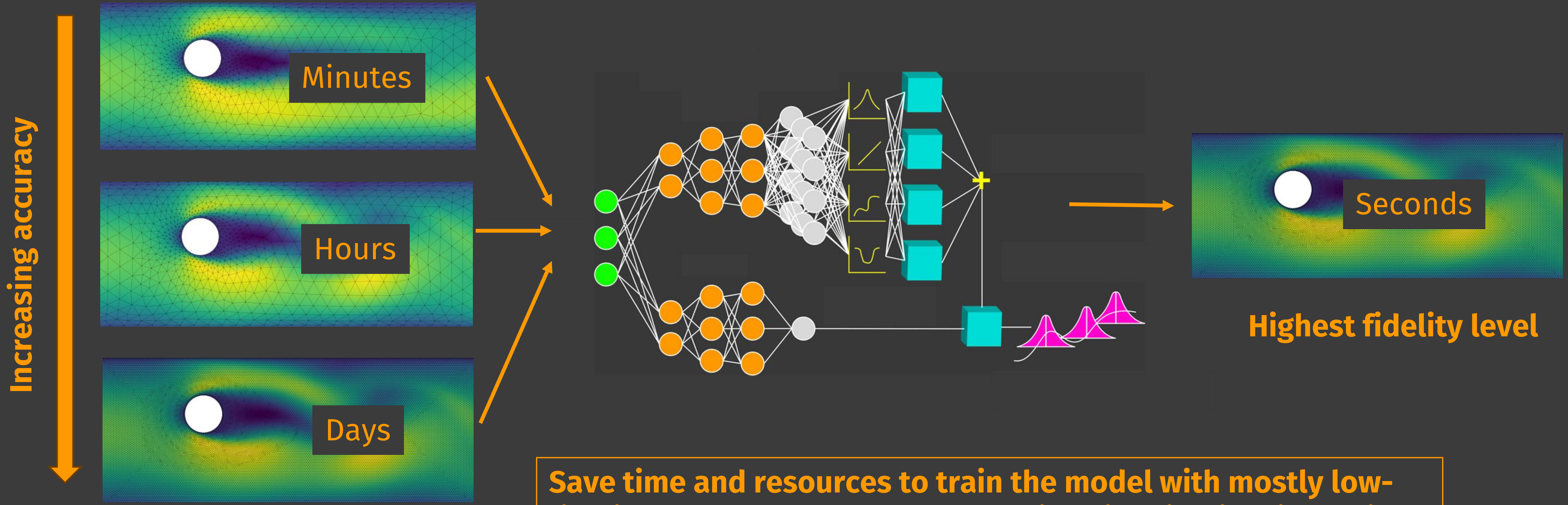
- Trained model on cube has been used on a c-beam geometry





Combination of multi-fidelity + 2D / 3D Simulation data

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



Save time and resources to train the model with mostly low-fidelity data and only a few expensive high fidelity simulations

Simple to use

- DIM-GP has no settings except a few options which might speed things up or for noisy data handling
- It is practically independent of the use case, simply enter data and start model training
- In Python it's as easy as this:

```python
from stochos.dimgp import dimgp_regr
import numpy as np

X = np.load("mesh_boundary.csv")
Y = np.load("node_results.csv")

model = dimgp_regr()
model.fit(X, Y, batch_size=500)
```

Placeholder for loading & preprocessing training data

Initial model and just pass training data no settings needed

```python
Y_pred = model.predict(X)
```

Predict new data

Key findings

- Adaptive data generation for training ML models is always preferable to a large Design of Experiment (DoE)
- If simulation / experiment is too complex / expensive, direct ML-driven optimization can be an alternative that already delivers very good results with less than 20 designs (even for high-dimensional problems)
- Multi-fidelity modeling / optimization allows to generate data even more efficiently because even fewer resources have to be used, such as simulation times, which can be extremely reduced
- It is also suitable for simulation calibration (combination of simulation and real experiment), especially if precise simulation calibration is not possible
- DIM-GP is a ML algorithm for domain experts of different fields to use this potential without the need of most of the ML-releated expert knowledge and it can be used in nearly all ML-related areas
- No cloud computing / no expensive hardware / no large amounts of data required for training
- Additional tools for data generation / optimization / sensitivity analysis all in one package everything perfectly matched
- Easy integration in existing workflow since it‘s a Python library

Contact

Dr.-Ing. Kevin Cremanns

kevin.cremanns@probaligence.de

www.probaligence.de