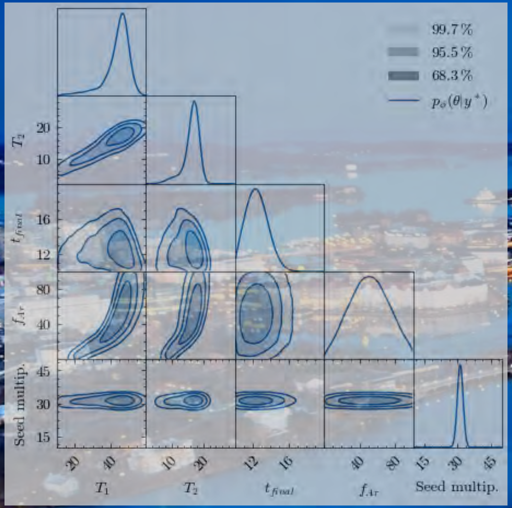
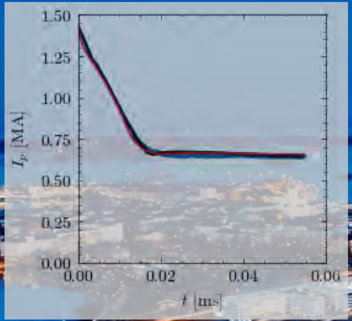




finnfusion

BUSINESS FINLAND

VTT

Machine learning acceleration of fusion model validation

Presented by Aaro Järvinen
Many contributors from VTT, EUROfusion ACH-05, TSVV-09, TSVV-11, FCAI, PPPL & LLNL
FinnFusion annual seminar 2024

Aalto University

23/05/2024 **VTT – beyond the obvious**

CSC

EUROfusion

FCAI

Research Council of Finland

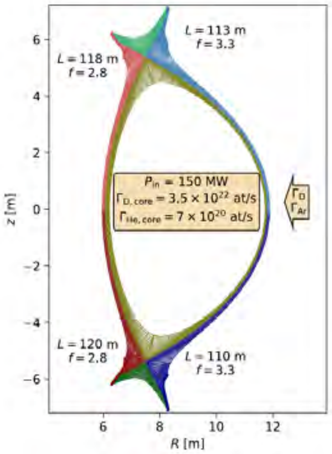
This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

Computationally expensive models with uncertain code parameters are ubiquitous in magnetic confinement fusion energy research

Fusion performance
~ plasma turbulence

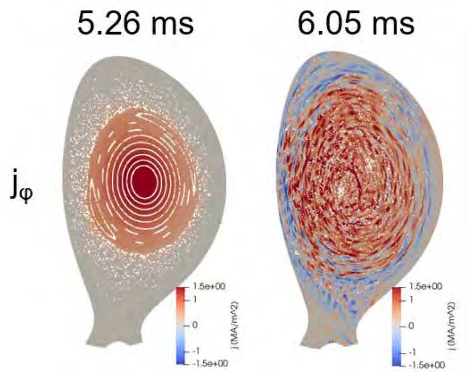
https://genecode.org/
F. Jenko, et al. PoP 2000

Power exhaust and heat load control ~ *plasma-neutral-materials interactions*



L. Aho-Mantila, et al. Nucl. Mat. Ene. 2021
https://doi.org/10.1016/j.nme.2020.100886
SOLPS-ITER: S. Wiesen, et al. J. Nucl. Mat. 2015
https://doi.org/10.1016/j.jnucmat.2014.10.012

Prediction of off-normal events and disruptions ~ *rapidly evolving thermal and relativistic populations & fields*



E. Nardon, et al. Nucl. Fusion 2023
https://doi.org/10.1088/1741-4326/acc417
JOREK: Hoeltzl, et al. Nucl. Fusion 2021
https://doi.org/10.1088/1741-4326/abf99f

Due to the uncertain code parameters, a single forward simulation is rarely sufficient to actually quantify the prediction uncertainty

Common problem statement

Given limited resources (CPUh / time), how to optimally quantify the uncertainties?

Due to the uncertain code parameters, a single forward simulation is rarely sufficient to actually quantify the prediction uncertainty

Common problem statement

Given limited resources (CPUh / time), how to optimally quantify the uncertainties?

Forward UQ ~ Uncertainty propagation

Input

Distributions of code parameters *θ*

Model

Computationally costly y = f(θ, Φ, ε)

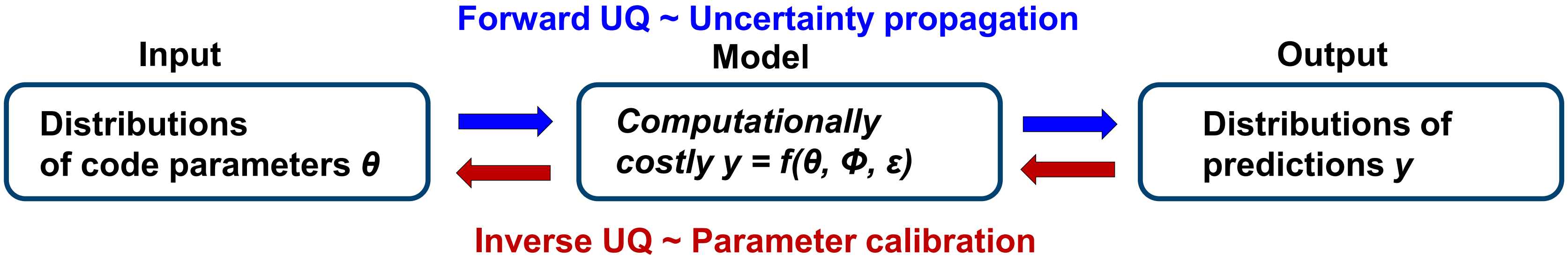
Output

Distributions of predictions *y*

Due to the uncertain code parameters, a single forward simulation is rarely sufficient to actually quantify the prediction uncertainty

Common problem statement

Given limited resources (CPUh / time), how to optimally quantify the uncertainties?



Forward UQ ~ Uncertainty propagation

Input

Distributions of code parameters θ

Model

Computationally costly $y = f(\theta, \Phi, \varepsilon)$

Output

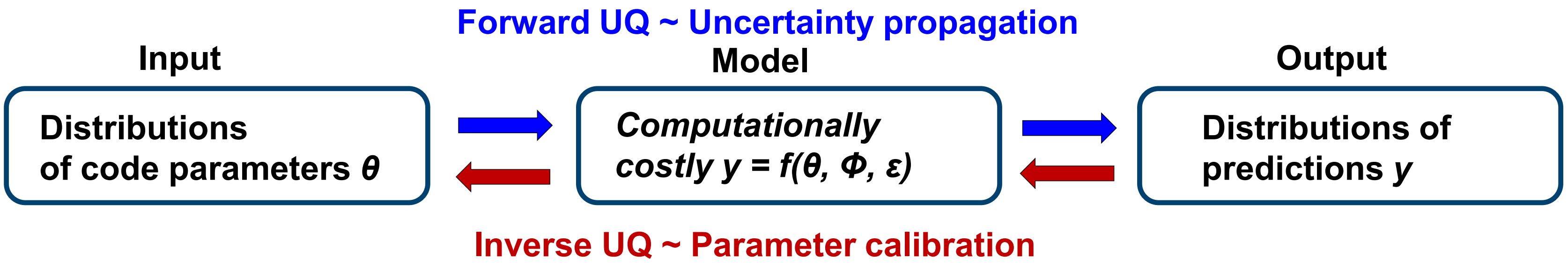
Distributions of predictions y

Inverse UQ ~ Parameter calibration

Due to the uncertain code parameters, a single forward simulation is rarely sufficient to actually quantify the prediction uncertainty

Common problem statement

Given limited resources (CPUh / time), how to optimally quantify the uncertainties?



Forward UQ ~ Uncertainty propagation

Input

Distributions of code parameters *θ*

Model

Computationally costly $y = f(\theta, \Phi, \varepsilon)$

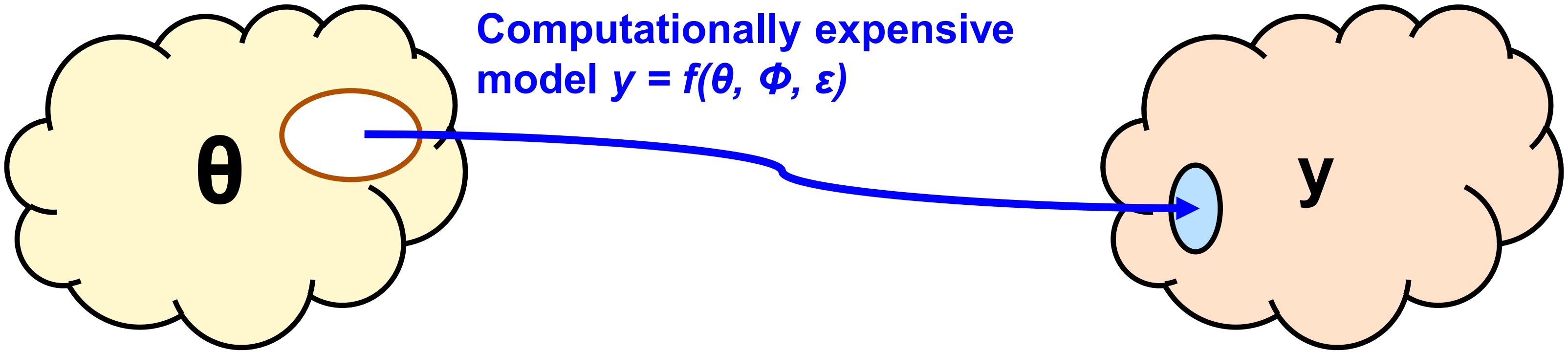
Output

Distributions of predictions *y*

Inverse UQ ~ Parameter calibration

- **Forward UQ → confidence interval for a model prediction:**
  - **Given the input & code uncertainties, how likely it is that the prediction falls within the tolerance of the system, such as heat load limit on divertor plate?**
- **Inverse UQ → model validation:**
  - **Is the model able to reproduce experimental observations with physically valid input parameters?**

Inverse mapping is defined only implicitly through the forward model



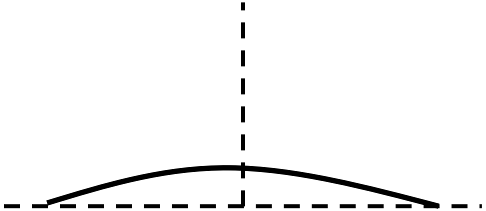
Computationally expensive model $y = f(\theta, \Phi, \varepsilon)$

- **With computationally expensive numerical models:**
  - **Given θ, y can be computed with the forward model**
  - **Given y, there is no direct computational model to determine θ**
  - **Bayesian inference to establish probability distributions for θ, given samples of (θ, y)**

Bayesian inference (BI) algorithms provide a principled approach to quantify the uncertainty for the state of the investigated system

Prior belief

$\pi(\theta)$

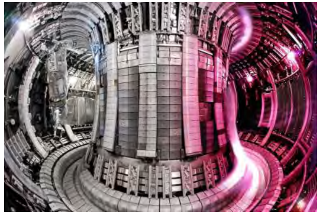


State of high uncertainty

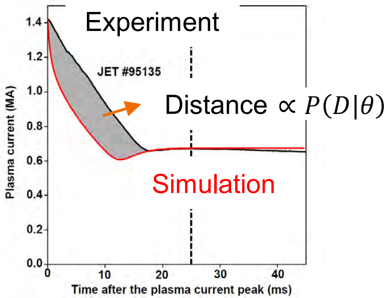
Likelihood

$P(D|\theta)$

Measured data D

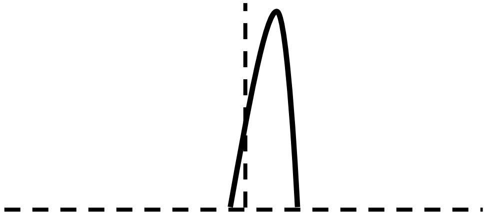


Simulation, input θ



Posterior belief

$\pi(\theta|D) \propto P(D|\theta)\pi(\theta)$



State of low uncertainty

Bayesian inference (BI) algorithms provide a principled approach to quantify the uncertainty for the state of the investigated system

Likelihood

$P(D|\theta)$

Measured data D

Prior belief

$\pi(\theta)$

Posterior belief

$\pi(\theta|D) \propto P(D|\theta)\pi(\theta)$

- **Likelihood typically not available in closed-form with complicated models – but can be sampled**
  - ➢ **Approximate Bayesian Computation (ABC) [see e.g. J.M. Marin, et al. (2012) https://doi.org/10.1007/s11222-011-9288-2]**
- **Sampling the likelihood can be computationally prohibitively expensive**
  - ➢ **e.g. 100 000 samples with 10 CPUh per sample → ~ 1M CPUh**

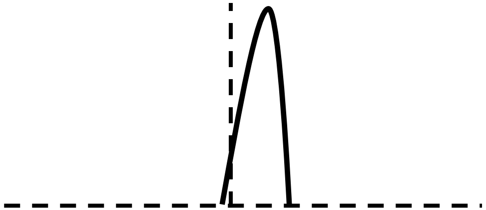
➢ **Data-efficiency is the key for BI with most models in fusion research**

Bayesian optimization is a class of optimization methods focused on finding a global optimum of a forward model within a search space

Bayesian Inference

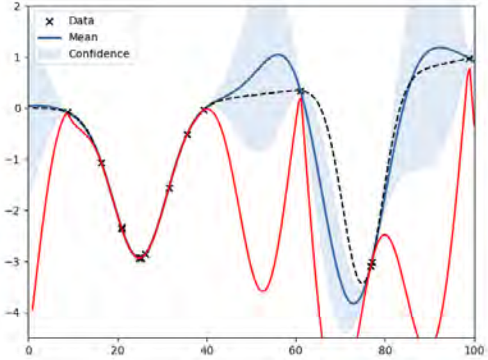
Establish a posterior distribution for uncertain parameters, given observations *π(θ|D)*

$$\pi(\theta|D) \propto P(D|\theta)\pi(\theta)$$



Bayesian Optimization (BO)

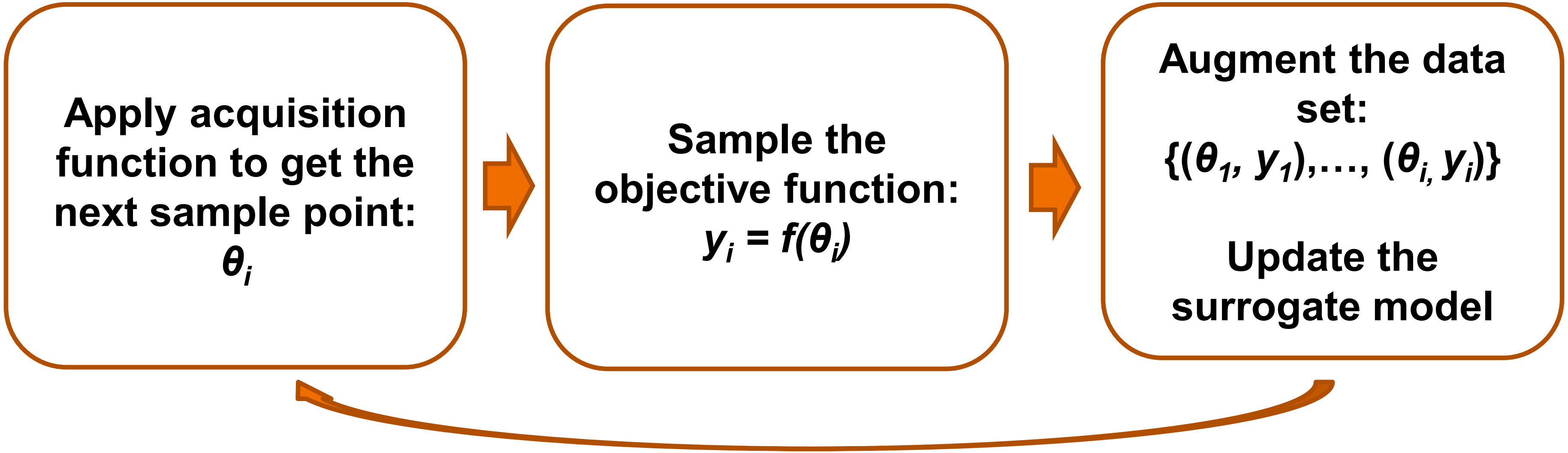
Conduct Bayesian Inference in the space of objective functions to data-efficiently find the global optimum



The ability to optimize expensive ”black-box” functions without access to derivatives makes BO very powerful.

A Bayesian optimization algorithm has two main components

1. **A probabilistic (surrogate) model of the objective function**
2. **An acquisition function**



Apply acquisition function to get the next sample point: $\boldsymbol{\theta_i}$

Sample the objective function: $y_i = f(\theta_i)$

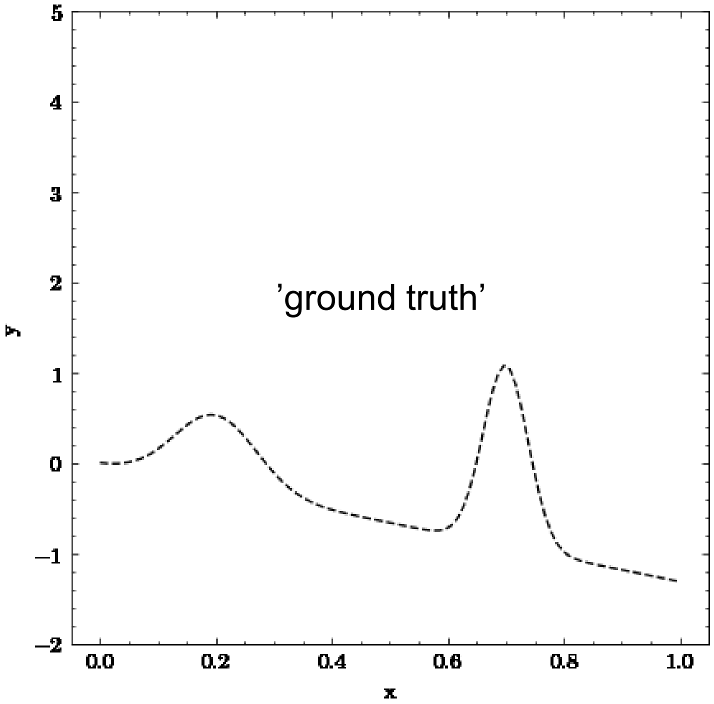
Augment the data set: $\{(\theta_1, y_1),\ldots, (\theta_i, y_i)\}$

Update the surrogate model

See, e.g. [E. Brochu et al. arXiv:1012.2599 https://arxiv.org/abs//1012.2599]

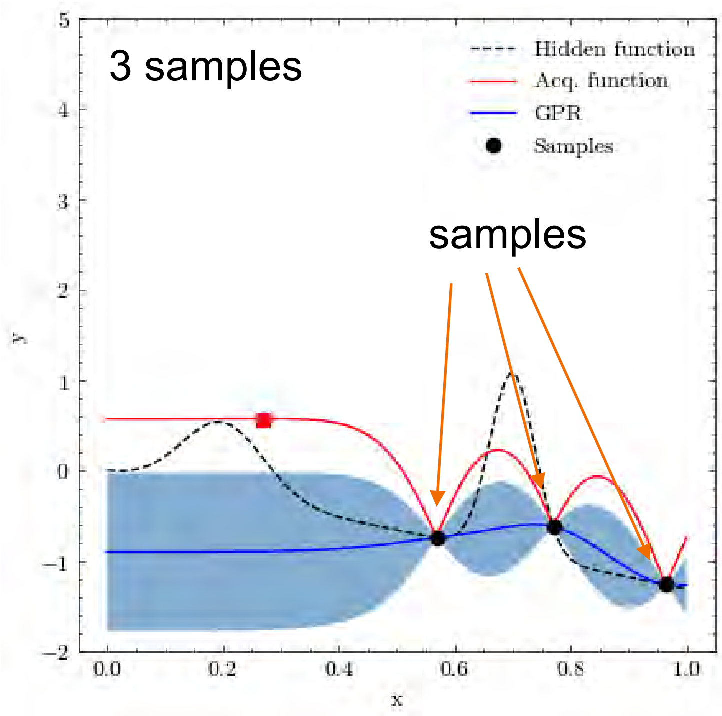
1D Bayesian optimization example – Find the maximum

- The code is available at https://github.com/aejarvin/BO_tutorial



1D Bayesian optimization example – Find the maximum

- The code is available at https://github.com/aejarvin/BO_tutorial



Gaussian process regression as probabilistic surrogate model

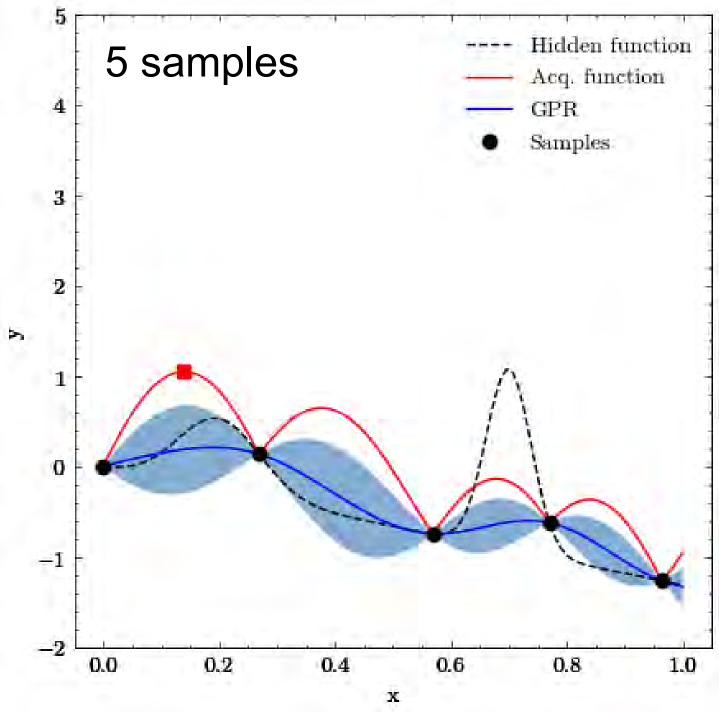
- Convenient non-parametric probabilistic regression when the amount of data is limited
- ’Learn’ model hyperparameters from data

Acquisition function uses the GPR mean and confidence to recommend new query points

More about GPR: Rasmussen, MIT Press 2006
Gaussian Processes for Machine Learning: Book webpage

1D Bayesian optimization example – Find the maximum

- The code is available at https://github.com/aejarvin/BO_tutorial



Gaussian process regression as probabilistic surrogate model

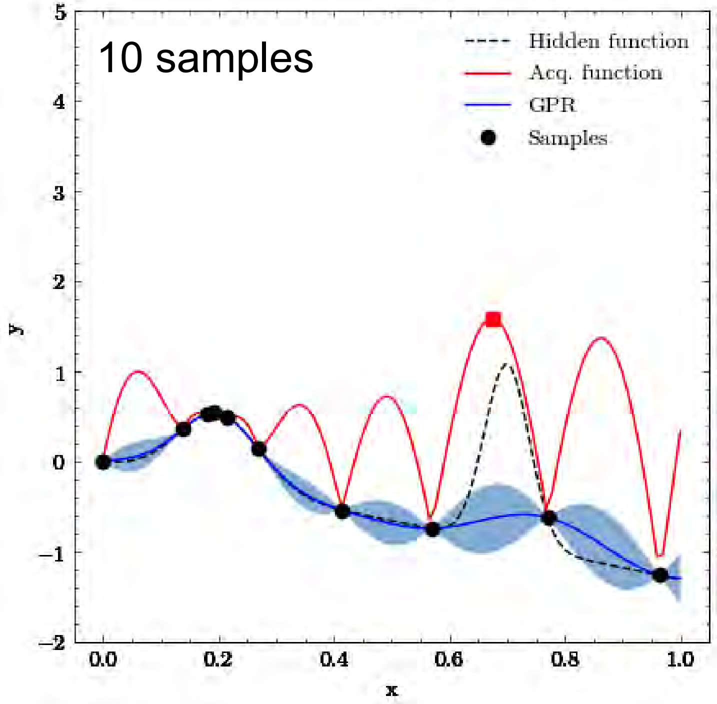
- Convenient non-parametric probabilistic regression when the amount of data is limited
- ’Learn’ model hyperparameters from data

Acquisition function uses the GPR mean and confidence to recommend new query points

More about GPR: Rasmussen, MIT Press 2006
[Gaussian Processes for Machine Learning: Book webpage]()

1D Bayesian optimization example – Find the maximum

- The code is available at https://github.com/aejarvin/BO_tutorial



Gaussian process regression as probabilistic surrogate model

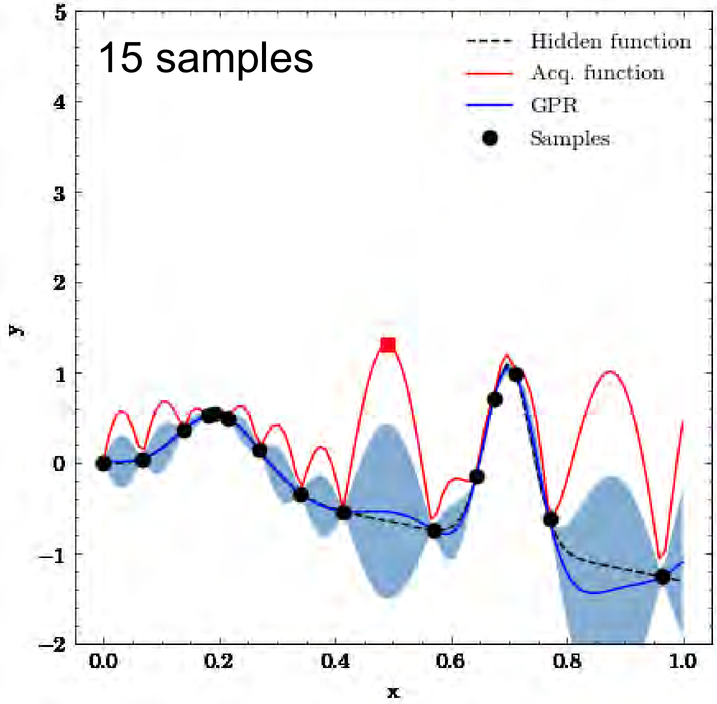
- Convenient non-parametric probabilistic regression when the amount of data is limited
- ’Learn’ model hyperparameters from data

Acquisition function uses the GPR mean and confidence to recommend new query points

More about GPR: Rasmussen, MIT Press 2006
[Gaussian Processes for Machine Learning: Book webpage]()

1D Bayesian optimization example – Find the maximum

- The code is available at https://github.com/aejarvin/BO_tutorial



Gaussian process regression as probabilistic surrogate model

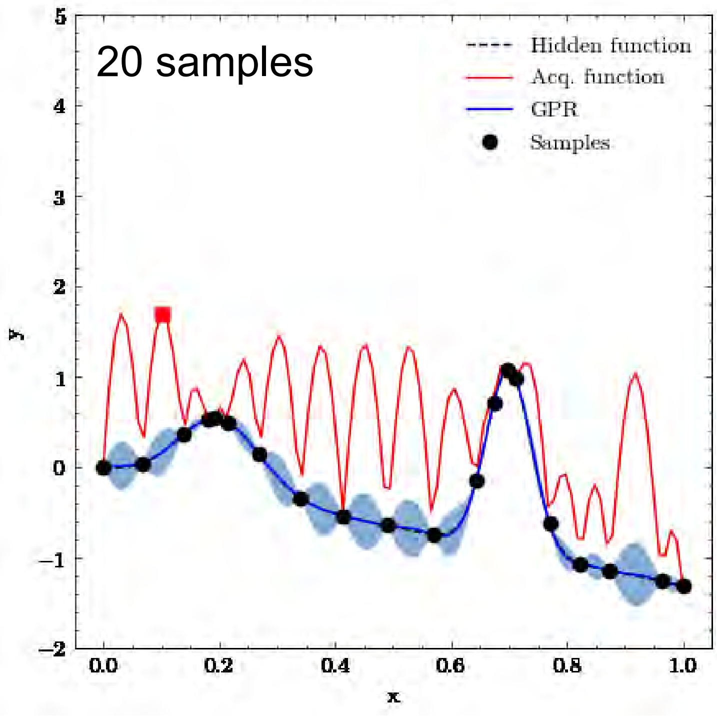
- Convenient non-parametric probabilistic regression when the amount of data is limited
- ’Learn’ model hyperparameters from data

Acquisition function uses the GPR mean and confidence to recommend new query points

More about GPR: Rasmussen, MIT Press 2006
[Gaussian Processes for Machine Learning: Book webpage]()

1D Bayesian optimization example – Find the maximum

- The code is available at https://github.com/aejarvin/BO_tutorial



Gaussian process regression as probabilistic surrogate model

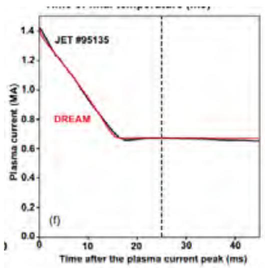
- Convenient non-parametric probabilistic regression when the amount of data is limited
- ’Learn’ model hyperparameters from data

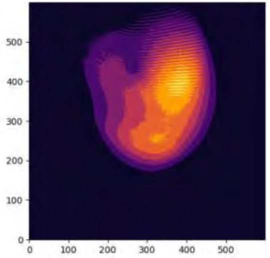
Acquisition function uses the GPR mean and confidence to recommend new query points

More about GPR: Rasmussen, MIT Press 2006
Gaussian Processes for Machine Learning: Book webpage

A portfolio of BI and BO tasks pursued in close connection to the EUROfusion Advanced Computing Hub (05) in Finland

DREAM runaway electron simulations





A.E. Järvinen, et al. J. Plasma Phys. 2022
https://doi.org/10.1017/S0022377822001210

Validation of integrared modelling

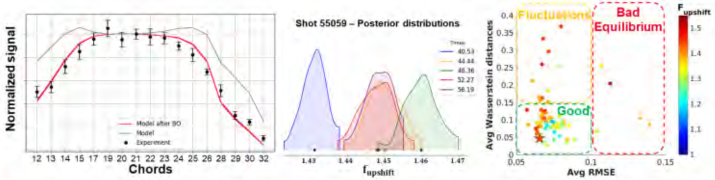
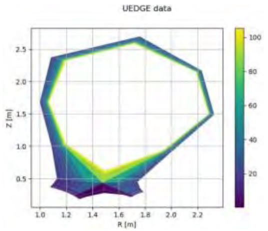
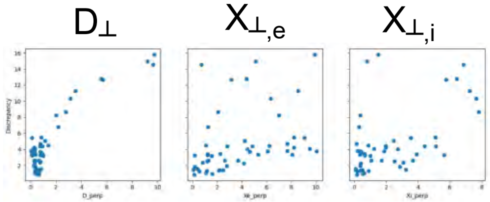


Fig 10. Single time optimization : (a) Forward UQ comparing with default parameters based on posterior distribution showed in (b). (c) Plateau-averaged performance of BO algorithm on the whole hot branch database and resulting fupshift as a color scale

T. Fonghetti, E. Amnell, et al.

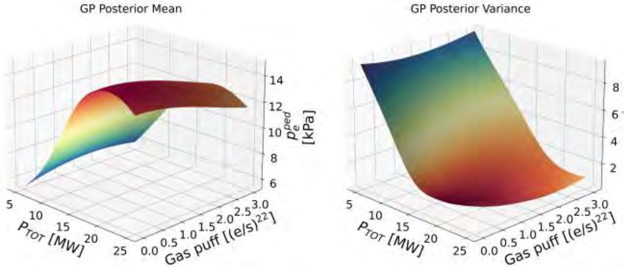
UEDGE SOL model calibration





same UEDGE case as in C.S. Furia, R.M. Churchill 2022 *Plasma Phys. Control. Fusion* **64** 104003
See PiAI seminar October 24, 2022

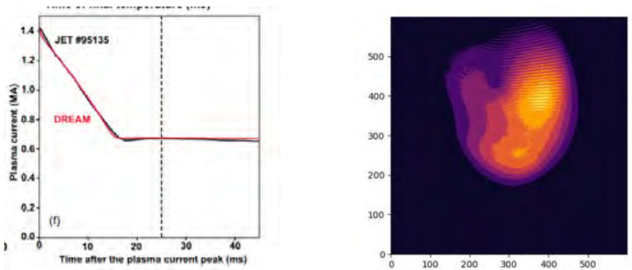
BO for experiment design



A. Kit,
G. Clarte, et al. ICDDPS-4

A portfolio of BI and BO tasks pursued in close connection to the EUROfusion Advanced Computing Hub (05) in Finland

DREAM runaway electron simulations



A.E. Järvinen, et al. J. Plasma Phys. 2022
https://doi.org/10.1017/S0022377822001210

Validation of integrared modelling

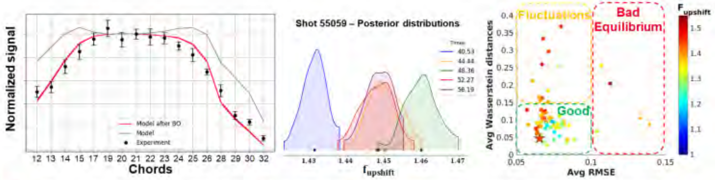
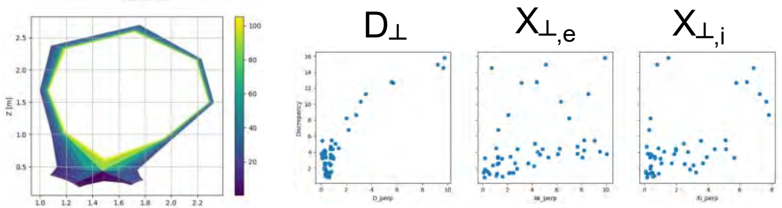


Fig 10. Single time optimization : (a) Forward UQ comparing with default parameters based on posterior distribution showed in (b). (c) Plateau-averaged performance of BO algorithm on the whole hot branch database and resulting fupshift as a color scale

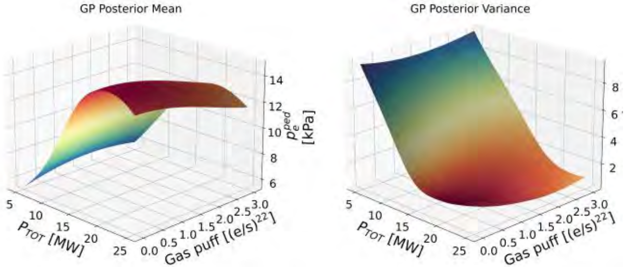
T. Fonghetti, E. Amnell, et al.

UEDGE SOL model calibration



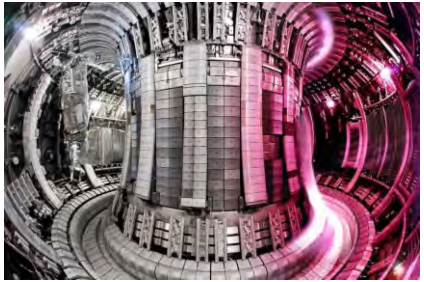
same UEDGE case as in C.S. Furia, R.M. Churchill 2022 *Plasma Phys. Control. Fusion* **64** 104003
See PiAI seminar October 24, 2022

BO for experiment design



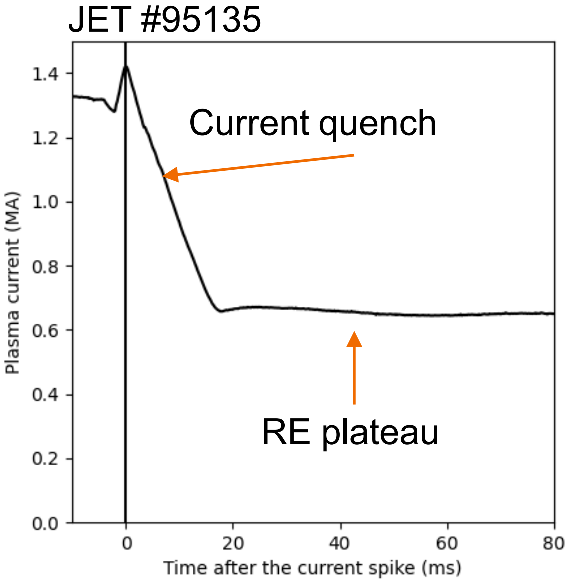
A. Kit,
G. Clarte, et al. ICDDPS-4

Inferring uncertain parameters for runaway electron simulations of disrupting tokamak plasmas



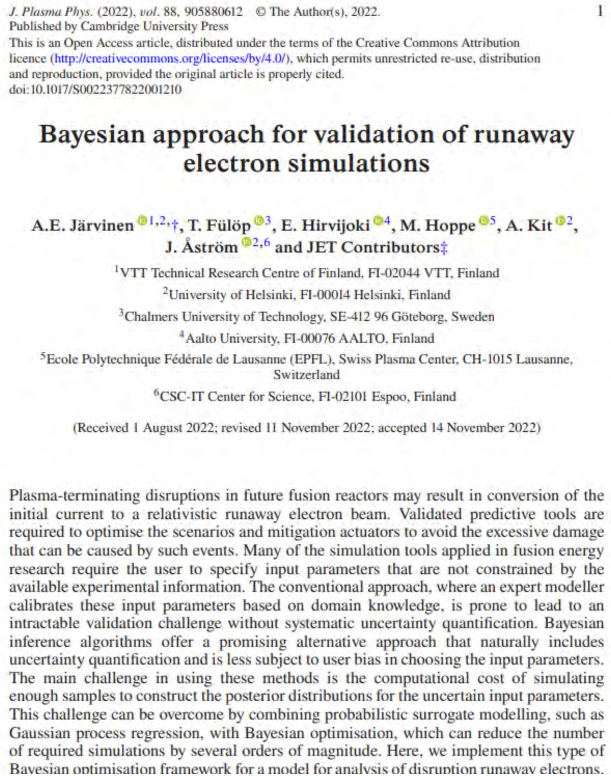
Ar massive gas injection to trigger a disruption.

Simulated with DREAM [Hoppe CPC 2021].



Uncertain input parameters:

Plasma temperature, Ar assimilation fraction, RE seed



J. Plasma Phys. (2022), *vol.* 88, 905880612 1

Published by Cambridge University Press

doi:10.1017/S0022377822001210

Bayesian approach for validation of runaway electron simulations

A.E. Järvinen[1,2,†], T. Fülöp[3], E. Hirvijoki[4], M. Hoppe[5], A. Kit[2], J. Åström[2,6] and JET Contributors[‡]

[1]VTT Technical Research Centre of Finland, FI-02044 VTT, Finland

[2]University of Helsinki, FI-00014 Helsinki, Finland

[3]Chalmers University of Technology, SE-412 96 Göteborg, Sweden

[4]Aalto University, FI-00076 AALTO, Finland

[5]Ecole Polytechnique Fédérale de Lausanne (EPFL), Swiss Plasma Center, CH-1015 Lausanne, Switzerland

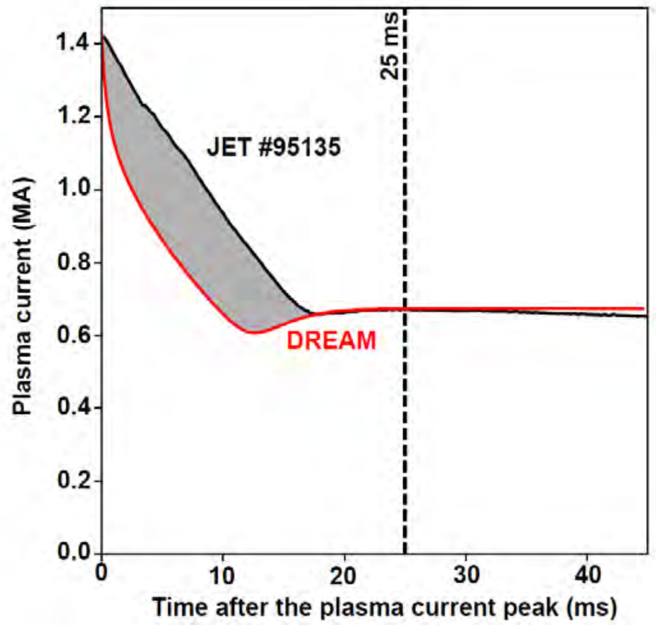
[6]CSC-IT Center for Science, FI-02101 Espoo, Finland

(Received 1 August 2022; revised 11 November 2022; accepted 14 November 2022)

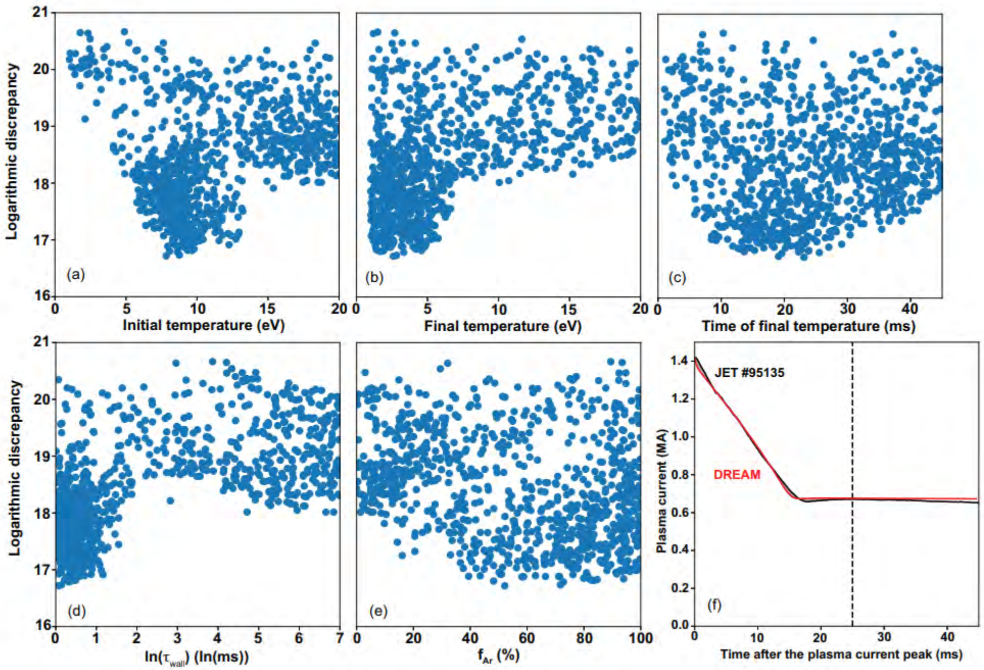
Plasma-terminating disruptions in future fusion reactors may result in conversion of the initial current to a relativistic runaway electron beam. Validated predictive tools are required to optimise the scenarios and mitigation actuators to avoid the excessive damage that can be caused by such events. Many of the simulation tools applied in fusion energy research require the user to specify input parameters that are not constrained by the available experimental information. The conventional approach, where an expert modeller calibrates these input parameters based on domain knowledge, is prone to lead to an intractable validation challenge without systematic uncertainty quantification. Bayesian inference algorithms offer a promising alternative approach that naturally includes uncertainty quantification and is less subject to user bias in choosing the input parameters. The main challenge in using these methods is the computational cost of simulating enough samples to construct the posterior distributions for the uncertain input parameters. This challenge can be overcome by combining probabilistic surrogate modelling, such as Gaussian process regression, with Bayesian optimisation, which can reduce the number of required simulations by several orders of magnitude. Here, we implement this type of Bayesian optimisation framework for a model for analysis of disruption runaway electrons.

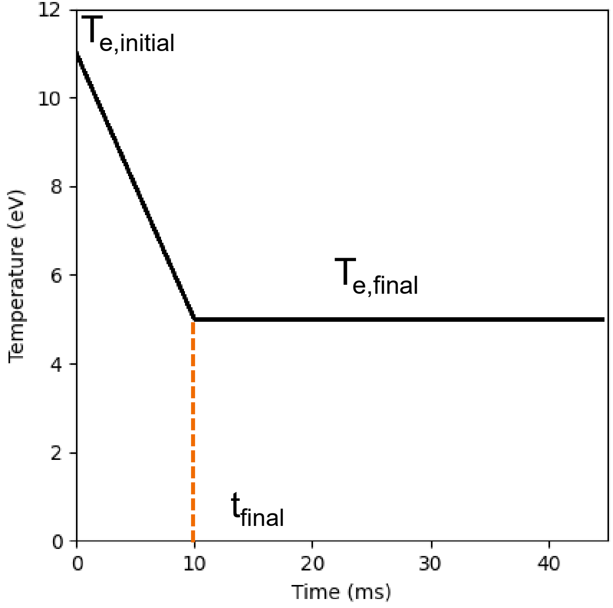
A.E. Järvinen, et al. Journal of Plasma Physics **88** (2022) 905880612, https://doi.org/10.1017/S0022377822001210

Objective is to reproduce the measured plasma current

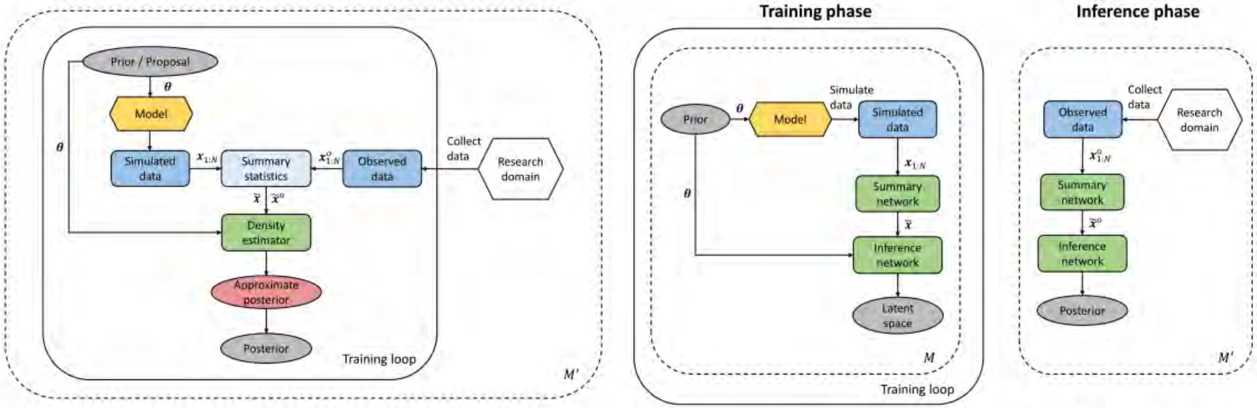


Parameterizing the plasma temperature evolution with 3 parameters, the algorithm is able to find a very good agreement





From case-based to amortized inference



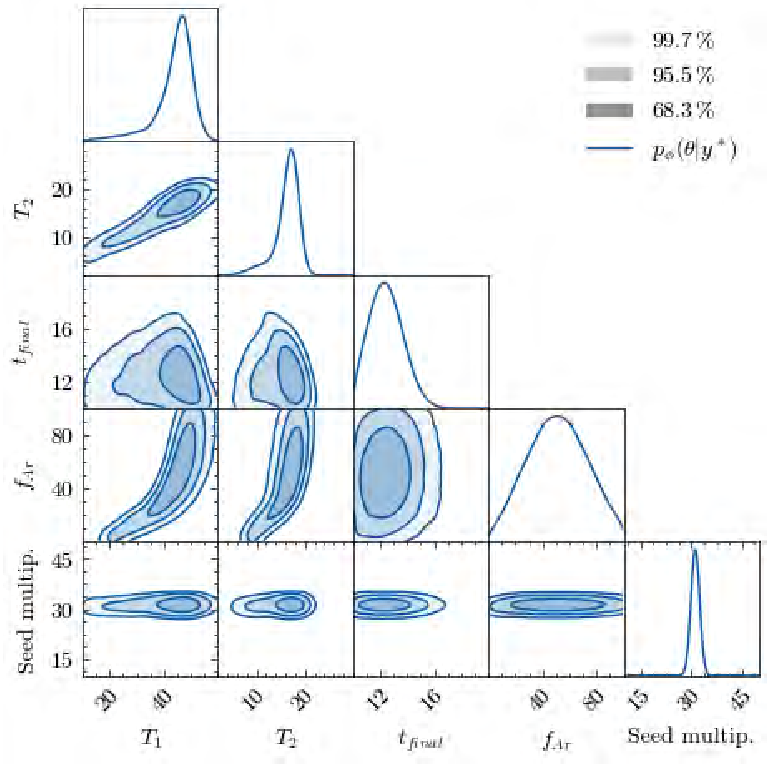
(a) Case-based inference

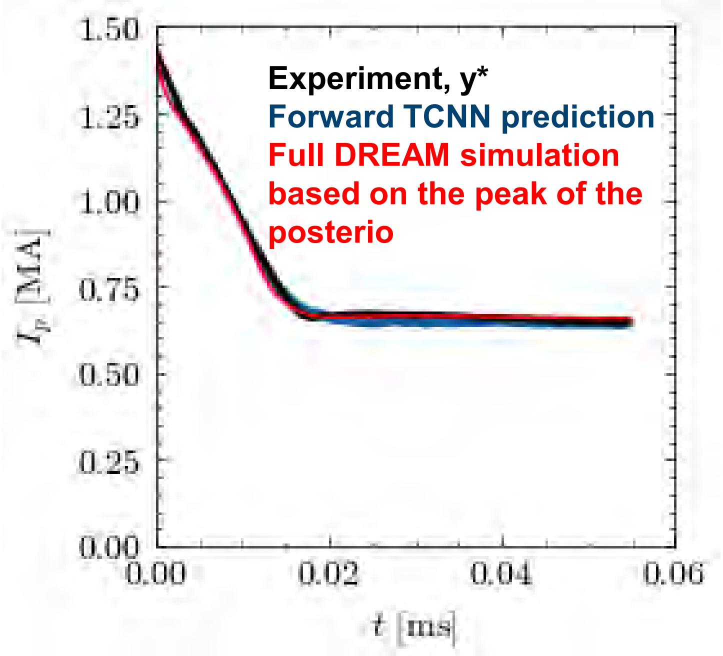
(b) Globally amortized inference with BayesFlow

Radev et al. IEEE 2022
https://ieeexplore.ieee.org/document/9298920

- The previous discussion has focused on conducting BO for a single input-simulation-output combination starting from scratch for each BO task – Case-based inference
- In practice, one ends up repeating similar simulations when inferring parameters for multiple cases
- A learning algorithm that is able to generalize and use previous experiences to guide searches would be very attractive → called amortized inference – this is much like the human brain is argued to operate [Gershman & Goodman, Cognitive Science 2014 https://api.semanticscholar.org/CorpusID:924780]

Neural posterior estimation can be used to amortize the inference task



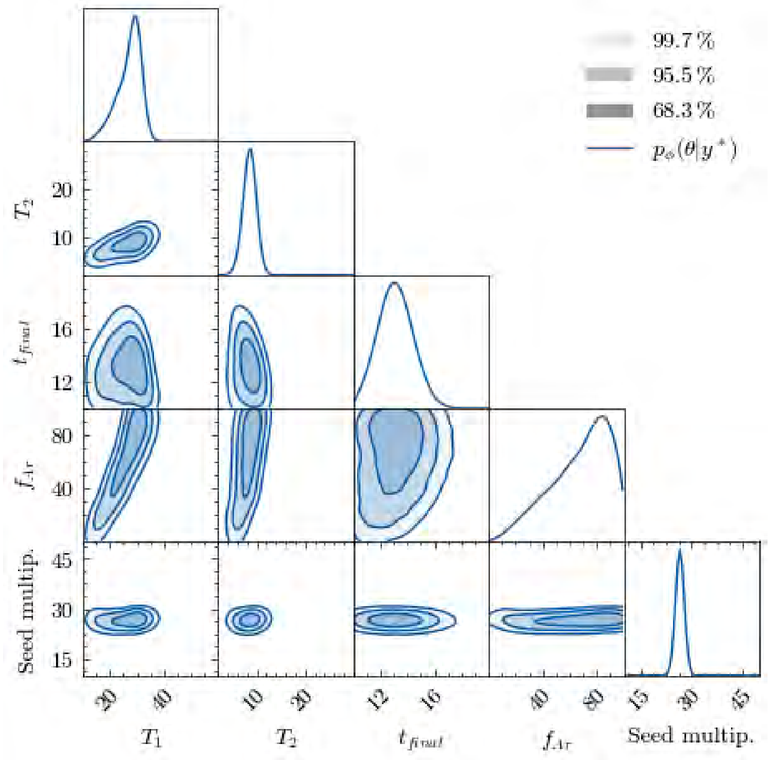


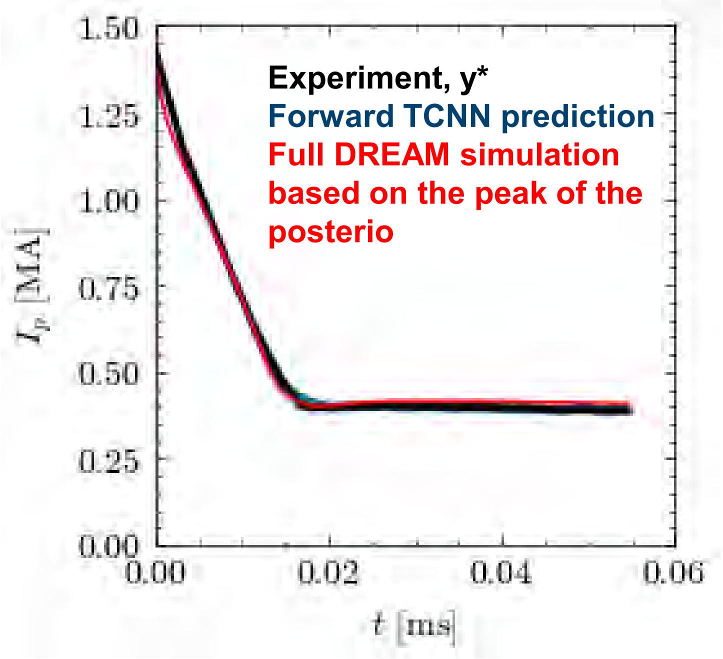
Experiment, y*
Forward TCNN prediction
Full DREAM simulation based on the peak of the posterio

- Database of ~ 25 000 simulations
- Normalizing flows based amortized posterior p(θ|x*)
- Transposed convolution neural network likelihood p(x|θ)

Using LAMPE: https://lampe.readthedocs.io/en/stable/index.html

The amortized model can quickly infer the parameters for a new target





Experiment, y*

Forward TCNN prediction

Full DREAM simulation based on the peak of the posterio

- Database of ~ 25 000 simulations
- Normalizing flows based amortized posterior p(θ|x*)
- Transposed convolution neural network likelihood p(x|θ)

Using LAMPE: https://lampe.readthedocs.io/en/stable/index.html

Outlook: Multi-fidelity amortized likelihood and posterior models can address multiple inference and prediction challenges in fusion

- **Machine learning surrogate modelling and amortized posterior inference represent two sides of the same coin:**
  - Surrogate models amortize p(y|θ) → c.f. amortized posterior p(θ|y)
- **Challenge is training data generation at high fidelity:**
  - Multi-fidelity and smart sampling / active learning c.f. Bayesian optimization – **New FCAI Virtual Laboratory**
- **Multiple use cases for such amortized models, including large scale validation and fast predictive workflows**
- **The Fusion AI team at VTT is developing a code framework, ’enchanted-surrogates’, for surrogate modelling of complex physics codes**
  - See posters by Adam Kit, Amanda Bruncrona & Daniel Jordan

→ GitHub:
https://github.com/DIGIfusion

VTT

beyond
the obvious

Thank you!

aaro.jarvinen@vtt.fi

vttresearch.com