Hybrid Bayesian Optimisation / Evolution Strategy applied to the design of uni-axially driven ICF targets

29th Nov 2023

Thomas Edwards, James Allison and Rafel Bordas

tom.edwards@firstlightfusion.com

Contents

Introduction to First Light Fusion

First Light Fusion

- Spun out from Oxford University in 2011 to research projectile fusion
- Today there are > 100 employees
- We have raised £77m in private equity funding to date
- Two large experimental platforms
- Large numerical and simulation capability – Including data science & ML

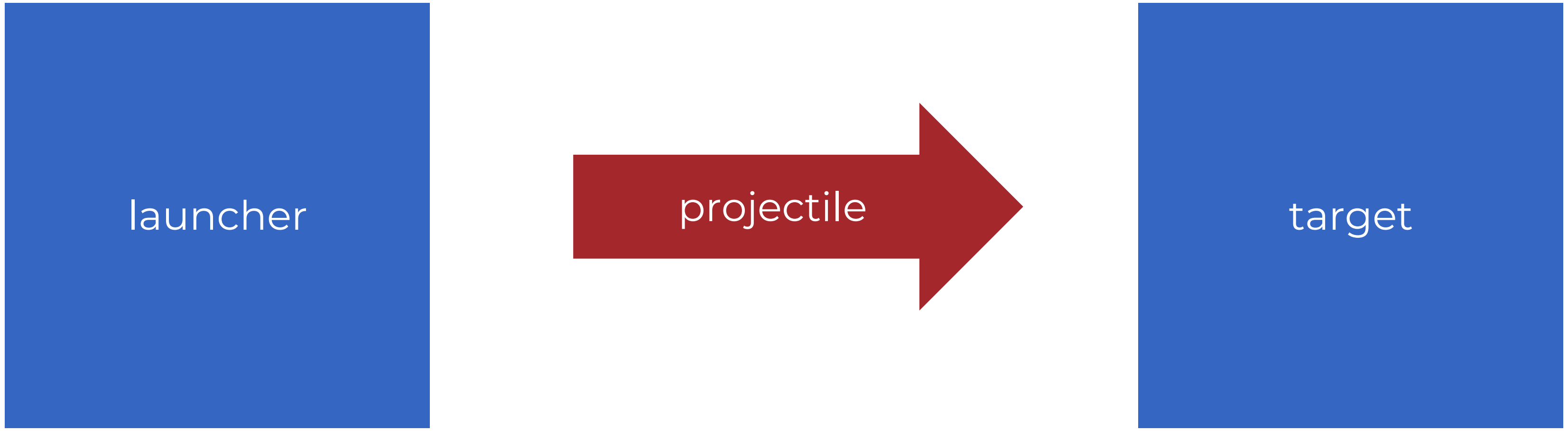
Departments

Experimental	Pulsed Power	Computational Science & Eng	Target Design	Power Plant

We use a projectile driver, which is low cost and high energy, but low power; the target design compensates

projectile

velocity and energy most important



electromagnetic launch

large but simple machine

target impact

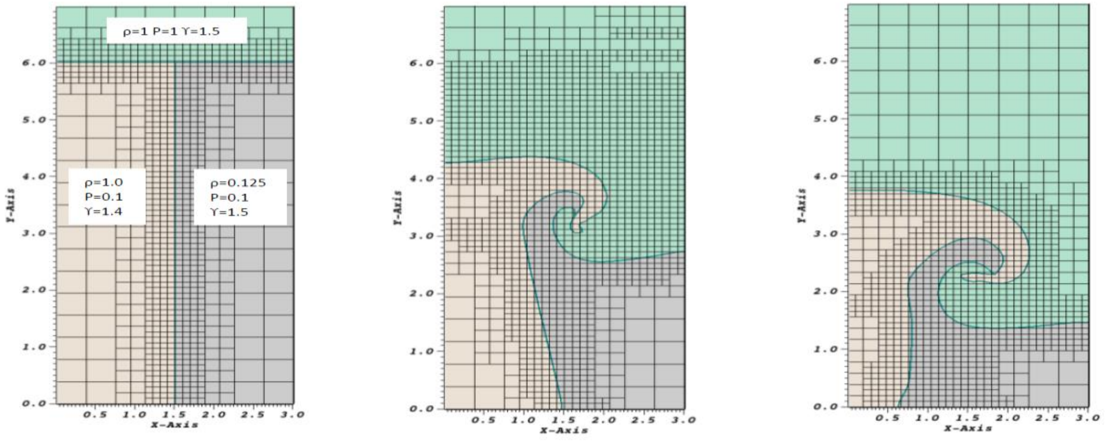
able to rapidly iterate design


HYPER VISION
SHIMADZU

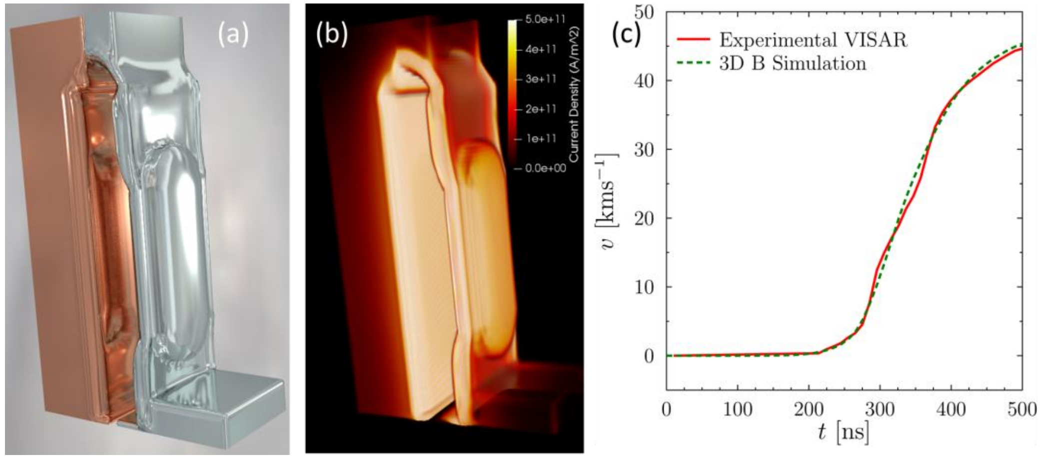
Core Simulation Capability

	Hytrac	B2
Hydro.	Eulerian 2T	Eulerian 2T
Transport	Thermal conduction Viscosity	Thermal conduction
Radiation	Moment closure.	Source term
EM	-	Resistive MHD
Hydro	Godunov	Lagrangian-remap
Geometry	2D axial/planar	+ 3D planar
Mesh	Cell based AMR	Structured grid
Interfaces	Front Tracking	SLIC-VoF
Parallelism	HPX	MPI

- Two in-house, multi-physics hydrodynamics codes.
- EoS (Equation of State) and microphysics models
- Strong software engineering principles
- Verification and validation.



(a) t = 0 (b) t = 3.5 (c) t = 5

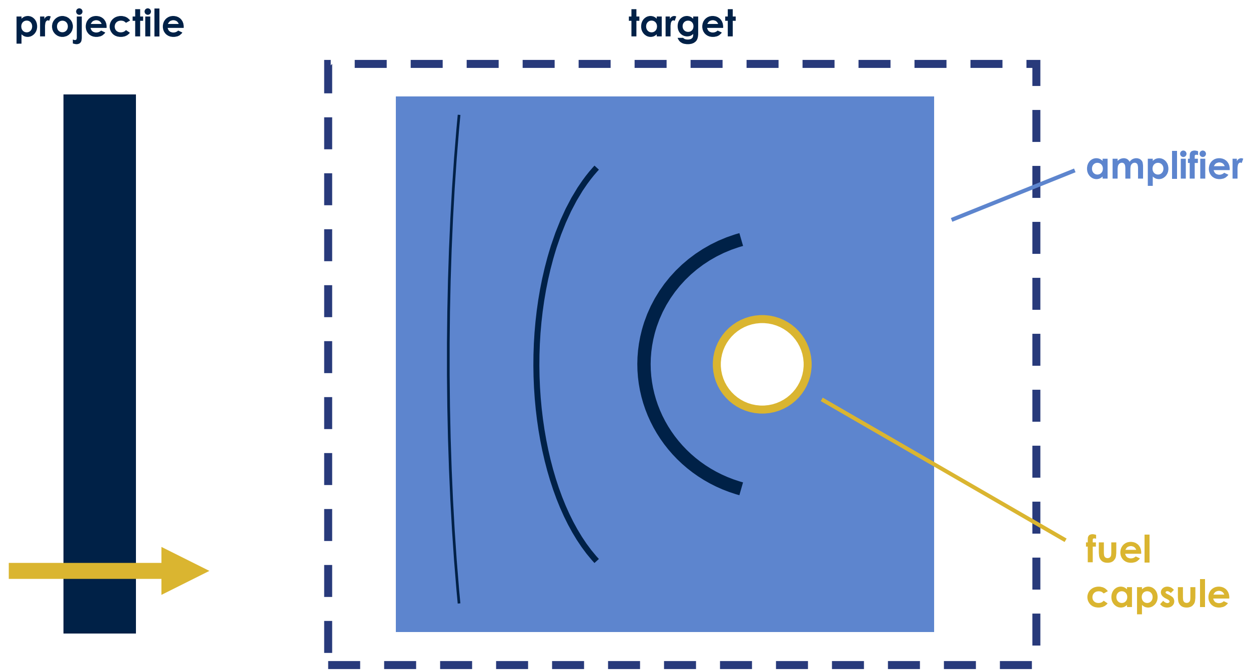


first light

In early 2022 we produced fusion in the lab in Oxfordshire, demonstrating projectile fusion works (validated by UK Atomic Energy Authority) – see www.firstlightfusion.com for media articles and our own whitepapers.

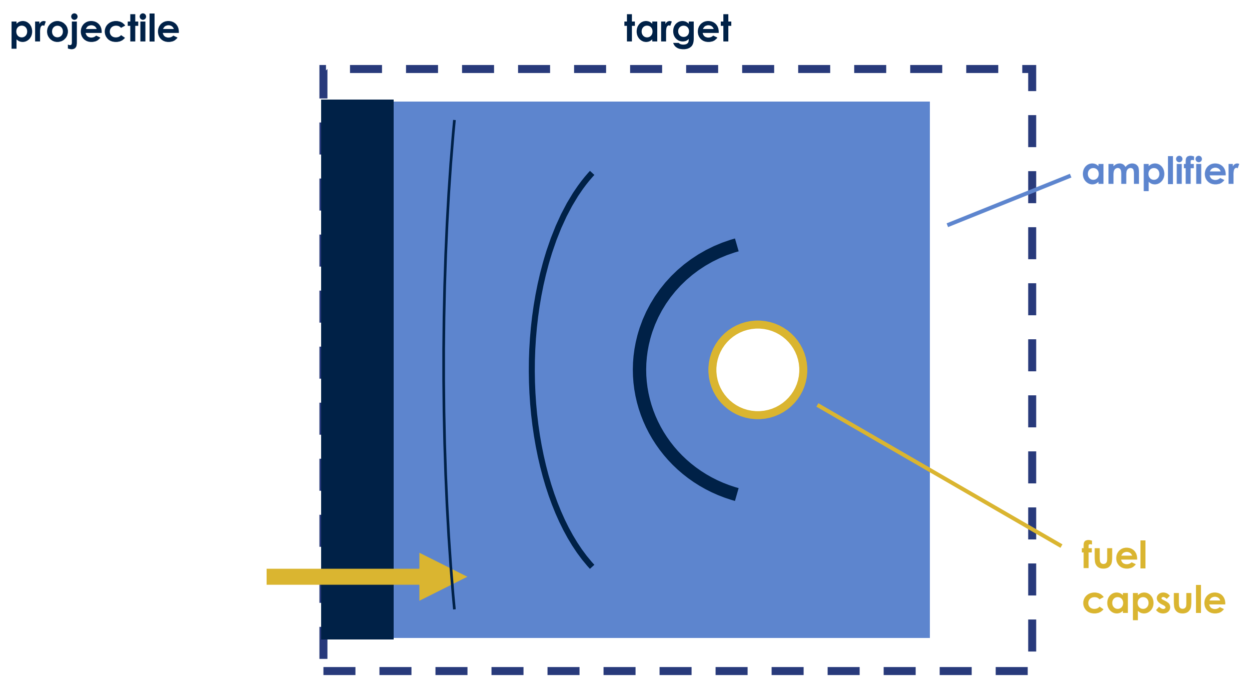
There is a key technology, the amplifier, which shapes and focuses the original shockwave

- Amplifiers **boost the pressure** and create spherical shaping
- Ignition demo will be based on fuel capsule with high-Z shell called "Revolver"
    - Gold shell, DT fuel
    - Molvig et al 2016, PRL



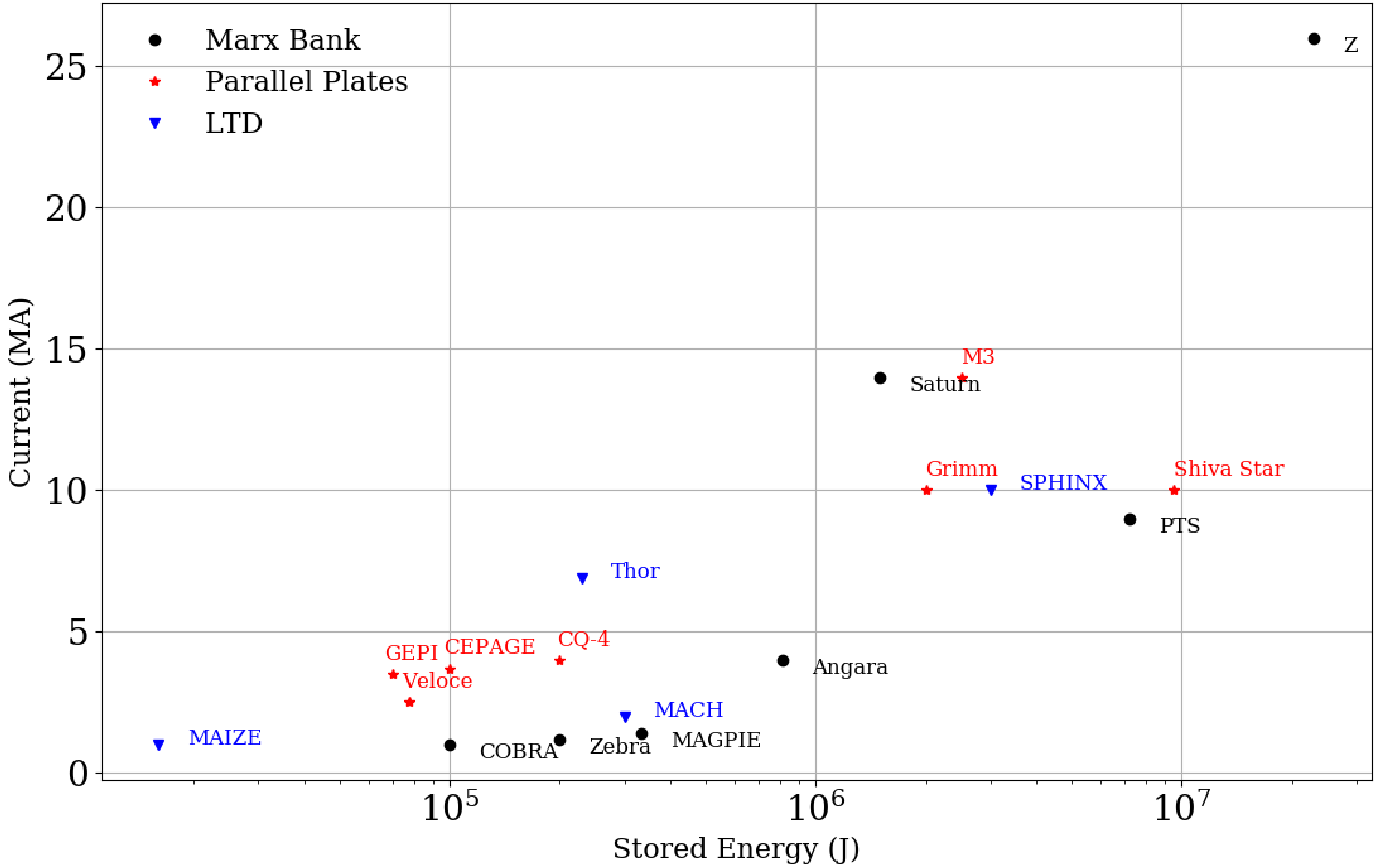
The Endor amplifier is a planar-to-planar variant

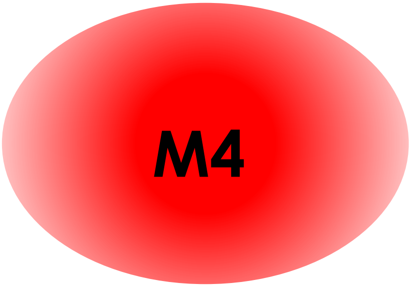
- The "Endor" amplifier is a planar-to-planar variant
- The impact pressure on the BFG is ~80 GPa, Endor boosts this to ~1200 GPa…
- … reducing the size by a factor of 10
- A 6.5 km/s impact gives a release velocity of ~80 km/s



We have proven this works, showing fusion for the first time, validated by UK Atomic Energy Authority

World Wide Pulsed Power Drivers





Overview

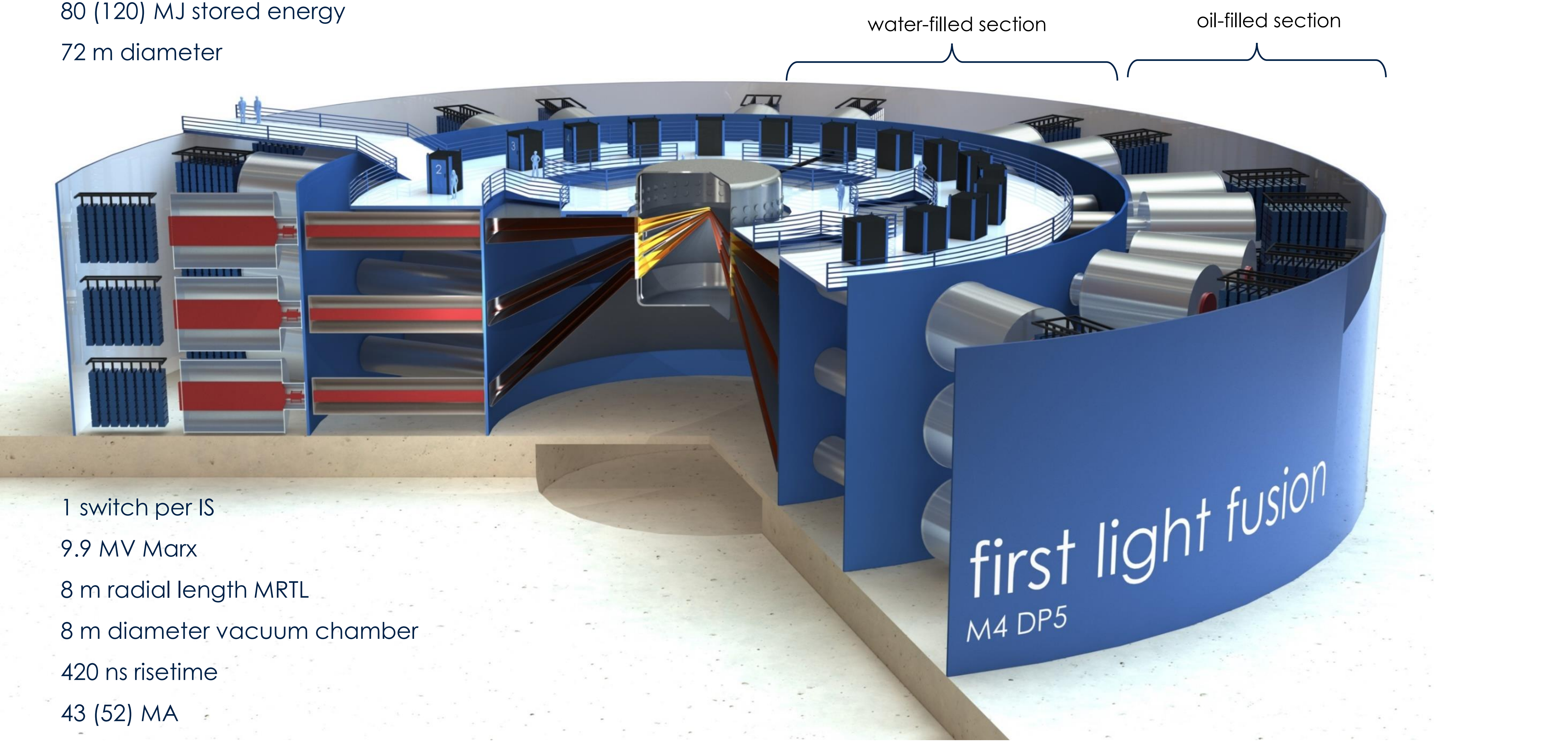
3 layers

20 (30) modules per layer

80 (120) MJ stored energy

72 m diameter

Machine 4 (M4) will be our gain demonstrator, aiming for a fuel gain of 100



1 switch per IS

9.9 MV Marx

8 m radial length MRTL

8 m diameter vacuum chamber

420 ns risetime

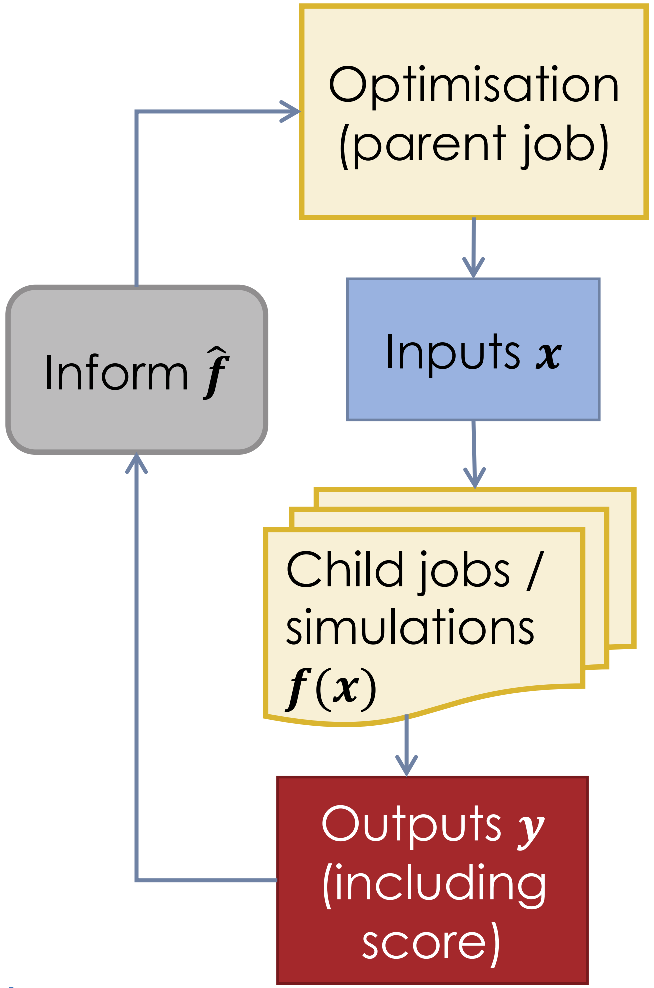
43 (52) MA

Optimisation & Job Marshalling Toolkit

For machine and target design

Optimisation Workflow

- Optimisation executed on HPC using ML methods
  - Setup
    - Simulation template $f(x)$
    - Input parameter ranges $\{x\}$
    - Output objective to be maximised y
    - Algorithm choice and meta-model $\hat{f}$
  - Run optimisation loop
    - Generate candidate solutions $x_i = \mathrm{argmax}(\hat{f}_i)$
    - Child job fills in template $f(x_i)$
    - Child job executes simulation $y_i = f(x_i)$
    - Output returned — inform meta-model $\hat{f}_{i+1} \leftarrow (x_i, y_i)$
- Algorithms: CMA-ES, Bayesian Optimisation and others
- Fault tolerant asynchronous job distribution using Slurm/Celery



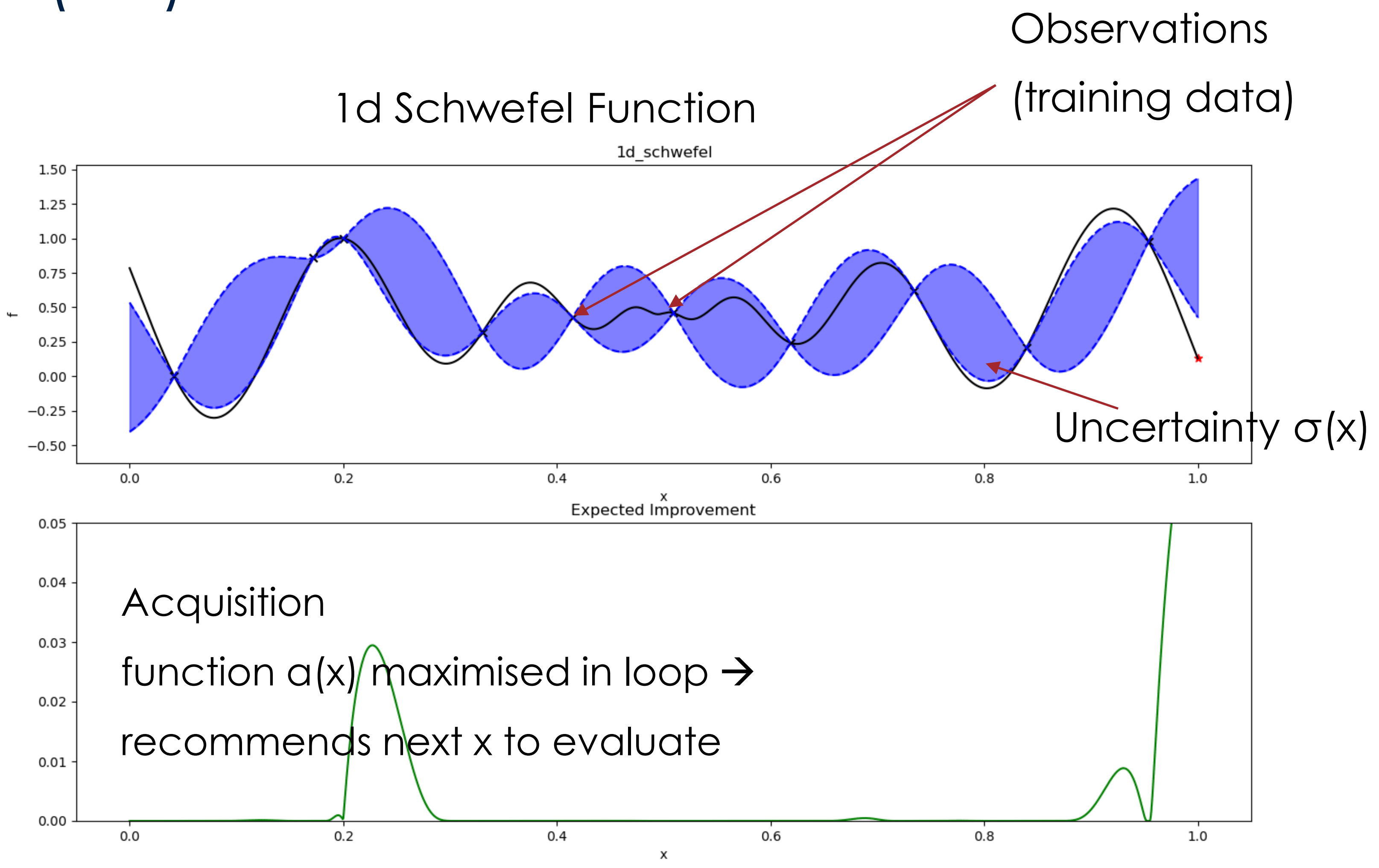
Covariance Matrix Adaptation – Evolution Strategy (CMA-ES)

- Derivative Free, Stochastic, Quasi-Hyperparameter Free optimisation for noisy, non-linear, non-convex functions
- Evolutionary strategy produces generations of N candidate solutions **x =** {x1, x2, …, xN}
  - **x** produced stochastically from parent generation by sampling from multivariate gaussian distribution
  - 'Fitness' of points based on **ranking** of **f(x)**
  - **Covariance matrix** is adapted from learning the 2nd order model of **f(x)**

- Robust, easy to use, great for rugged local optimisation
- However, does not learn entire response surface → can fall in local optima
- Requires large population size N to find global optima → expensive

Bayesian Optimisation (BO)

- Used to optimise black box expensive-to-evaluate high-dimensional objective functions
- No derivative information
- Machine learned emulator is optimised in place of f(x)
- Emulator updated (trained) as observations are made
- ‘Decision theory’ approach to recommend next point
- Good for finding global optima
- We have implemented BO using BoTorch



Gaussian Process

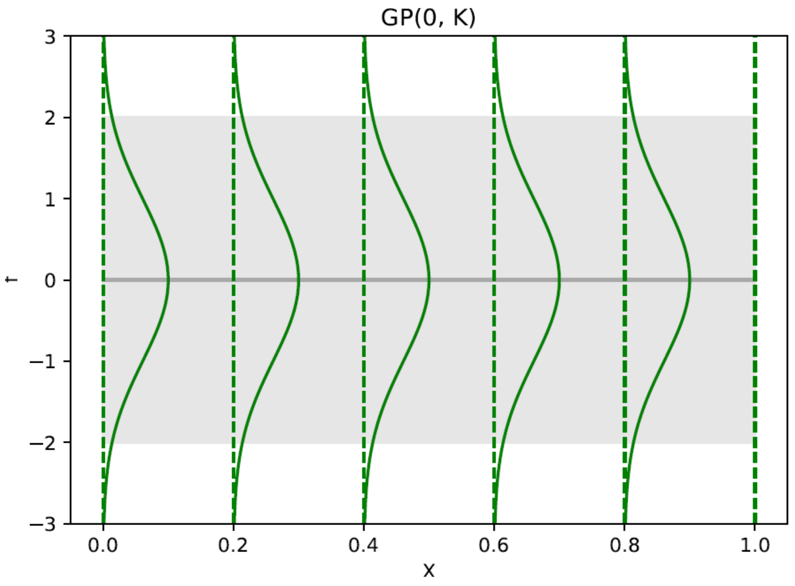
- Collection of random variables (indexed by time or space) with a joint Gaussian distribution
- Characterised by ***mean*** function and ***covariance*** function (or ***kernel***)

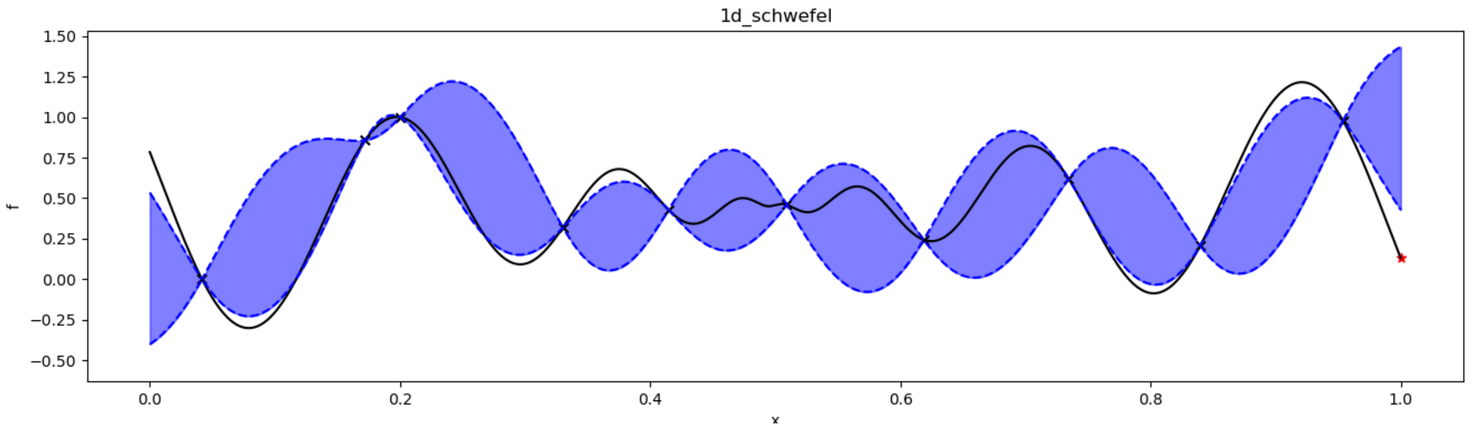
$$f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$

Mean $$\mu(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$$

Covariance $$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}') - \mu(\mathbf{x}'))]$$

- ***Covariance (or kernel)*** encodes smoothness/similarity between any 2 points as well as uncertainty
- ***Uncertainty*** pinched at observation points **X** in posterior (GP is 'conditioned') → O(n^3) computation, O(n^2) memory
- ***Hyperparameters*** (scale lengths, etc) of GP optimised in-loop





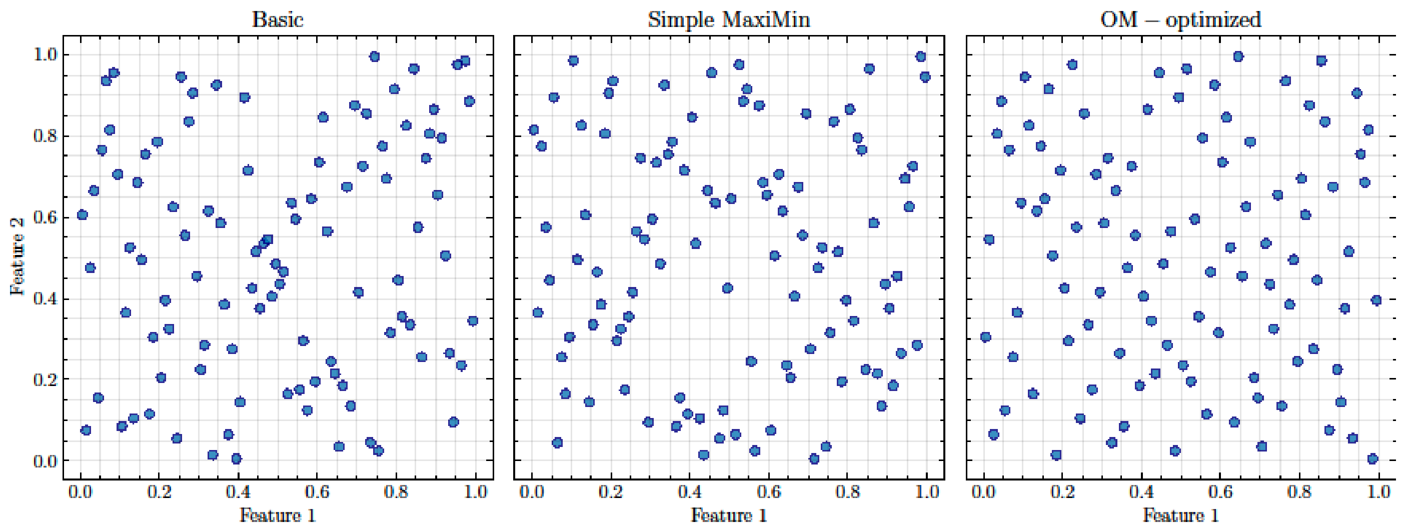
Space filling parameter scans – Latin Hypercubes

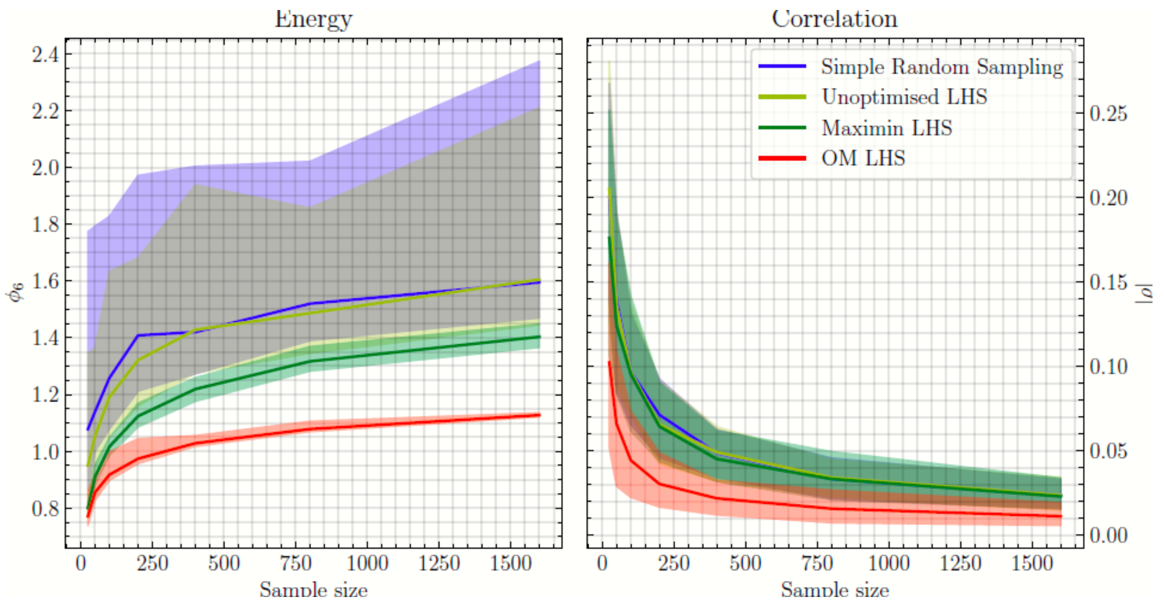
- We initialise the Gaussian Process with a space-filling exploratory scan
- Latin Hypercube Sampling
  - Space filling properties poor in high dimensions
  - Correlations between inputs.
- LHS row/column swap optimiser based on Joseph & Hung 2018
  - Energy (distance) metric

$$\phi_\lambda = \sum_{i \neq j} \left(\frac{1}{d(\mathbf{x}_i, \mathbf{x}_j)^\lambda} \right)^{\frac{1}{\lambda}}$$

  - Correlation (orthogonality) metric

$$\rho^2 = \frac{\sum_{i=2}^{k} \sum_{j=1}^{i=1} \rho_{ij}^2}{k(k-1)/2)}$$





Acquisition Functions

- Decision theory approach for selecting next point to evaluate
- Acquisition function a(x) encodes preference over which values of x to recommend
  - Exploration vs exploitation
- a(x) maximised using **gradient-based optimisers** within the BO loop
- Many different a(x) → Standard is 'Expected Improvement'

$$\alpha_{EI}(x;\mathcal{D}) = \int \max(y - y^*, 0)\, p(y|x,\mathcal{D})\, \mathrm{d}y$$

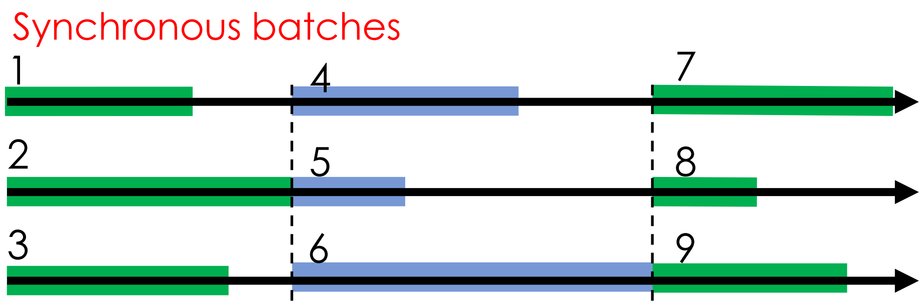
where y^* = Maximum observation so far

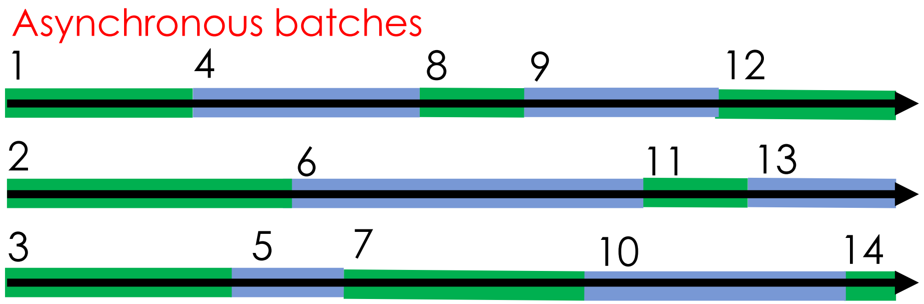
$$\mathbf{x} = \arg\max_{\mathbf{x}' \in \mathcal{X}} \alpha(\mathbf{x}';\mathcal{D})$$

Where D = {**x**, **y**}, the set of input points and observations made so far

Advanced Bayesian Optimisation Features

Parallel Batch Acquisition





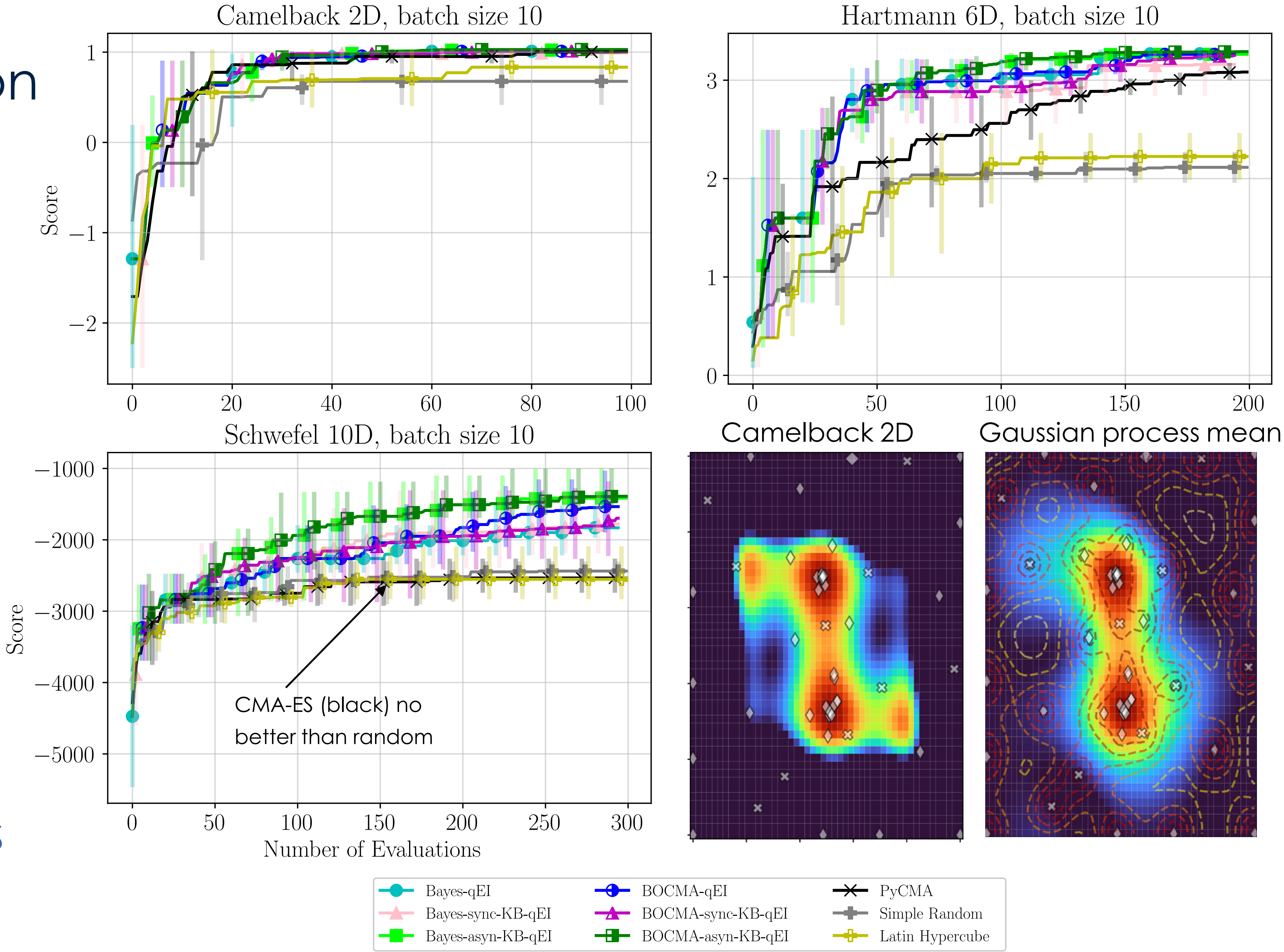
- Parallel batch acquisition for BO is non-trivial → because of nested integral for a(x1, …, xN)
  - Batch EI calculated approximately using Monte Carlo schemes → q-EI (q-batch Expected Improvement)
- Maximise use of resources → asynchronous batching
  - Batches decomposed into series of serial acquisitions
  - Gaussian Process trained on 'fantasy' observations until real observations finished
    - 'Kriging Believer' heuristic [Ginsbourger et al, 2010]
- Alternative parallel batch schemes
  - Local penalisation [Gonzalez et al, 2016]
  - Thompson Sampling – For massive parallel batches [W. R. Thompson, 1933], [Kandasamy et al, 2018]

BOCMA – Hybrid BO/CMA-ES scheme

- Combines global optimisation (BO) with local optimisation (CMA-ES)
  - Bayesian Optimisation → Covariance Matrix Adaptation - Evolution Strategy
- Runs Bayesian Optimisation then switches to CMA-ES under certain criteria
  - ‘Patience factor’ heuristic determines when to switchover
  - CMA-ES triggered if errors in BO occur
- Advantages
  - Fault tolerant
  - Bayes opt becomes computationally expensive → CMA-ES is lightweight

Synthetic function Benchmarks

- 5 optimisation runs per scheme
  - Solid line - Mean of cumulative max
  - Error bars – range of cumulative max
- **Bayes opt consistently out-performs CMA-ES**
- Schwefel 10D is ‘multi-funnel’; particularly challenging for CMA-ES

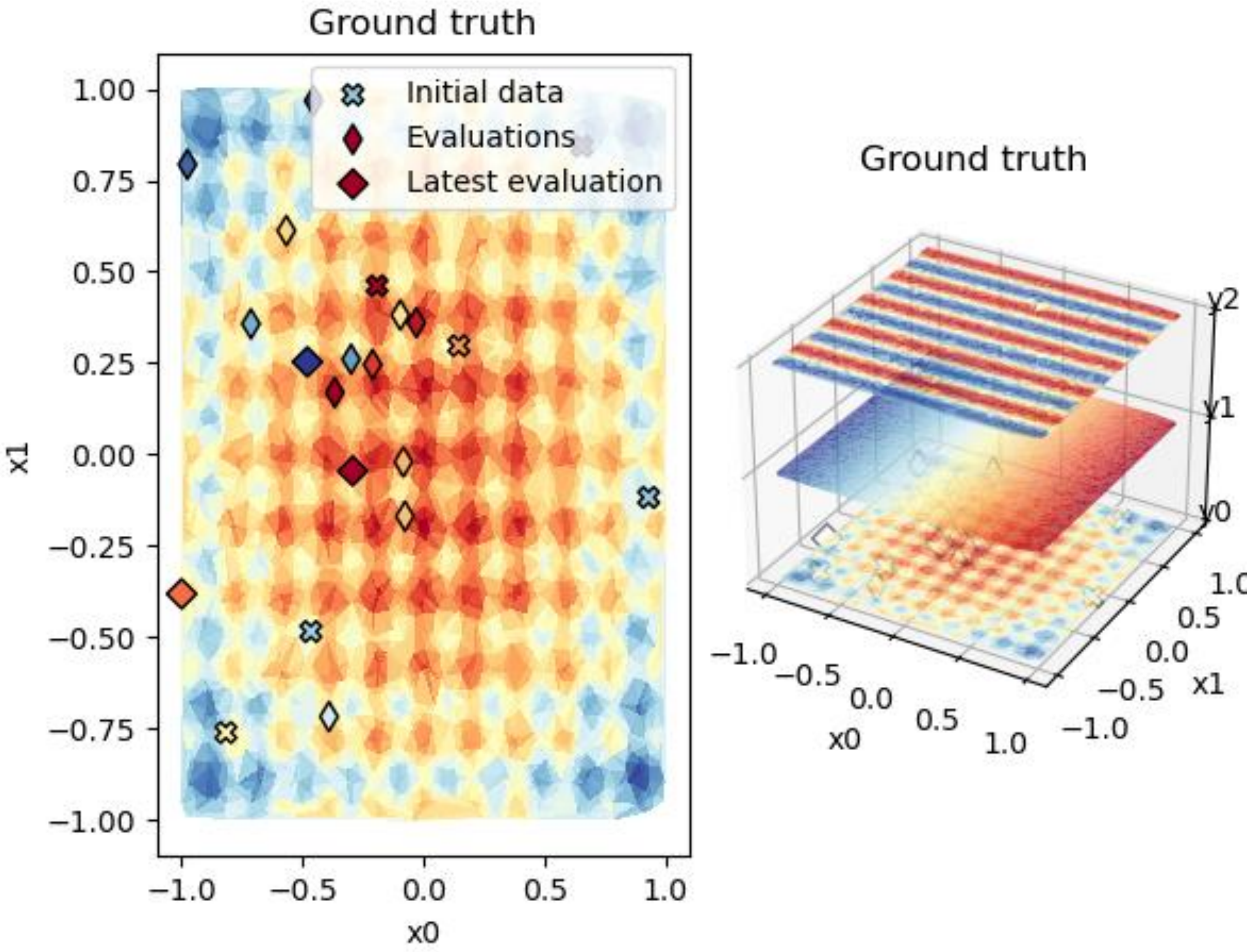


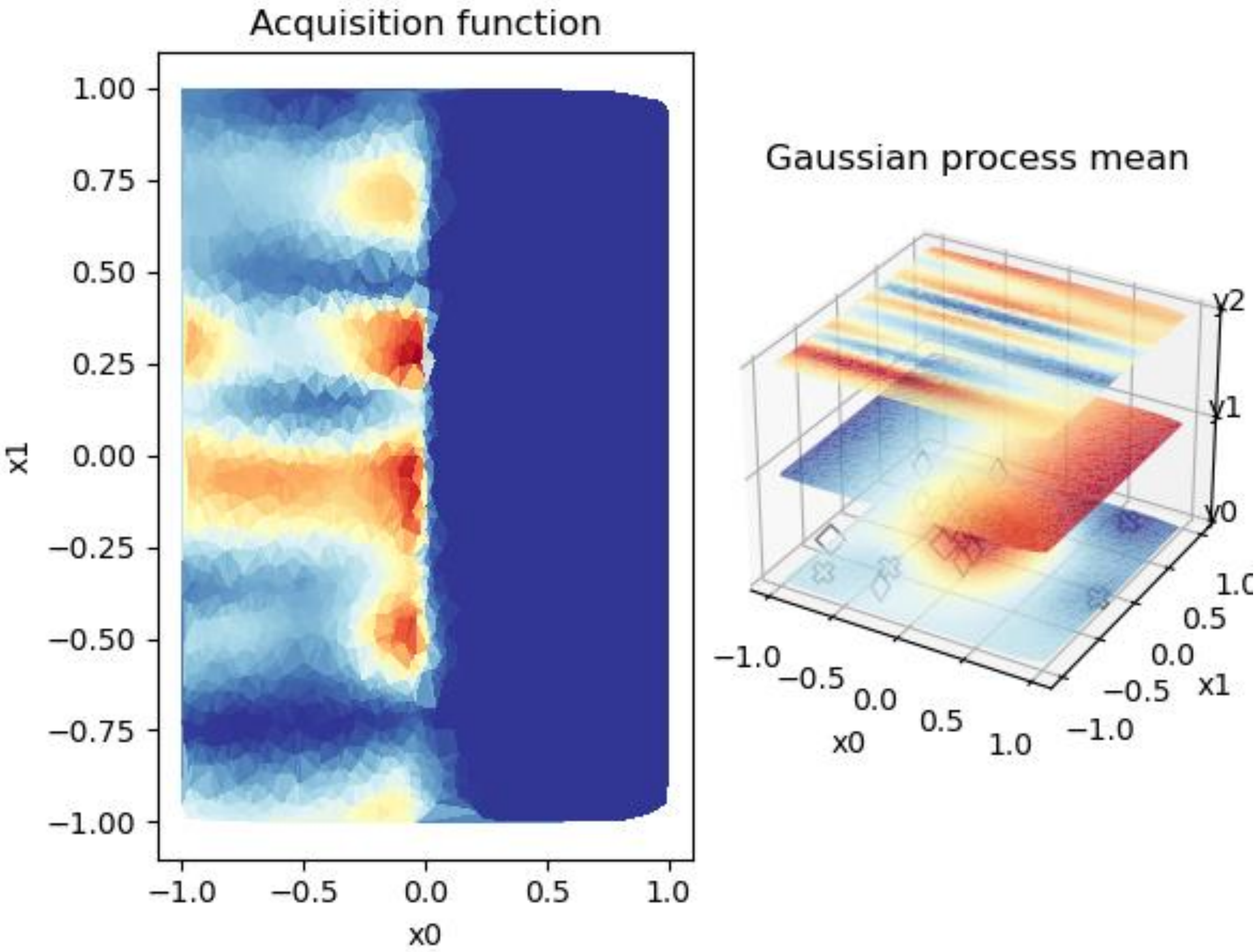
Bayesian Optimisation – Black Box Constraints

ND Rastrigin Function: $$f(\mathbf{x}) = An + \sum_{i=1}^{n} \left[x_i^2 - A\cos(2\pi x_i)\right]$$

Constraints: $x_0 < 0$

$\sin(3x_1) < 0$



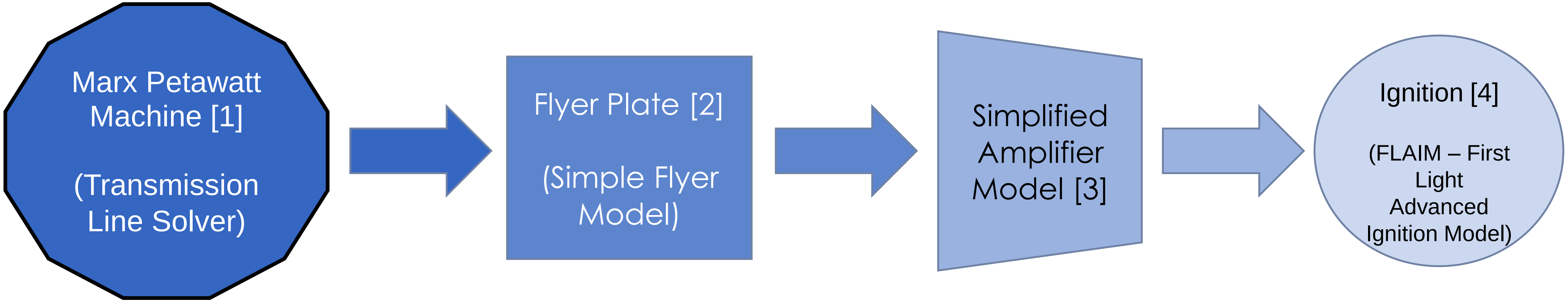


Case Study – Machine System Design

FuSE Gain Model

FuSE (Fusion System Evaluator) – System of Reduced Physics Models

Optimise end-to-end fusion system model with these components:



Components have been validated/calibrated by high-fidelity 1D to 3D simulations
12 Input parameters are optimised for.

[1] Peta-Watt-Class Z-Pinch Accelerator based on *W. A Stygar et. al. Physical Review Special Topics: Accelerators and Beams 10, 030401 (2007)*

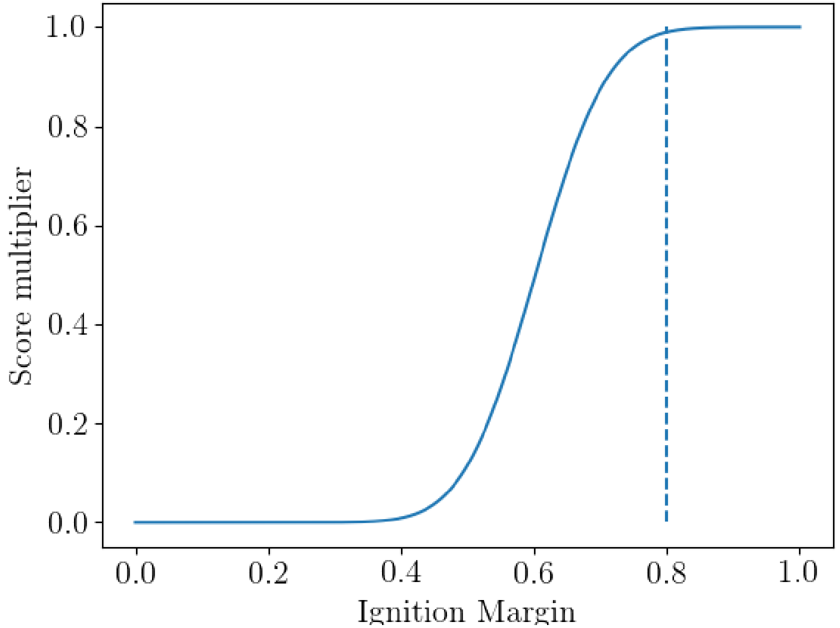
[2] Lumped parameter model calibrated on data from Lemke et al and 3D B2 simulations, using LRC machine model

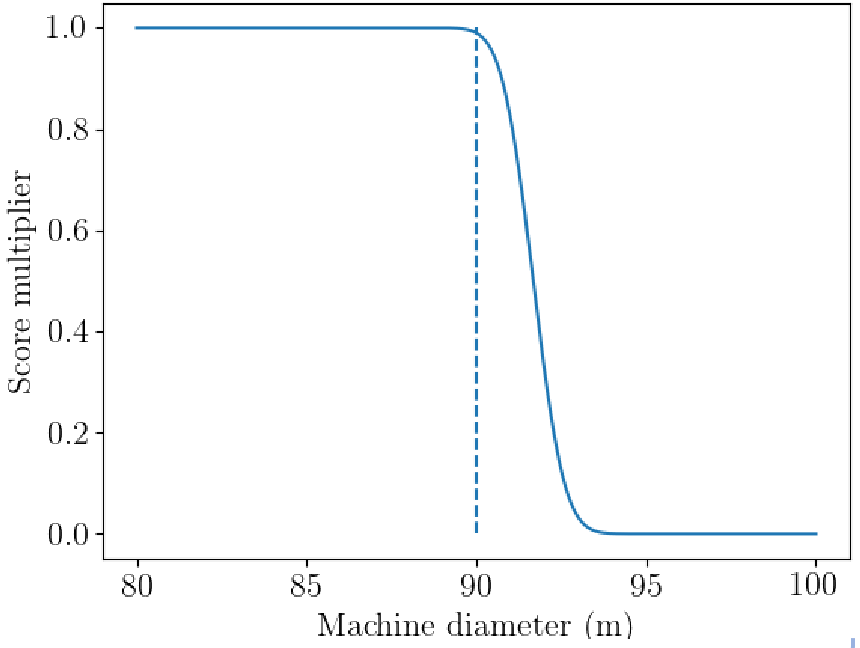
[3] Converts flyer KE to pusher KE, assuming exit is a spherical shell (pusher) surrounding the fuel

[4] Lagrangian 1D hydro code capturing fusion burn physics of a spherical pusher into a cavity of DT fuel. Fuel capsule design based on Revolver (Molvig et al 2016)

Built-in Constraints

- Optimise target gain w.r.t constraints
  - Sigmoid penalisers
- Constraints are on outputs, not inputs
- Resulting score value is transformed to log values if too small





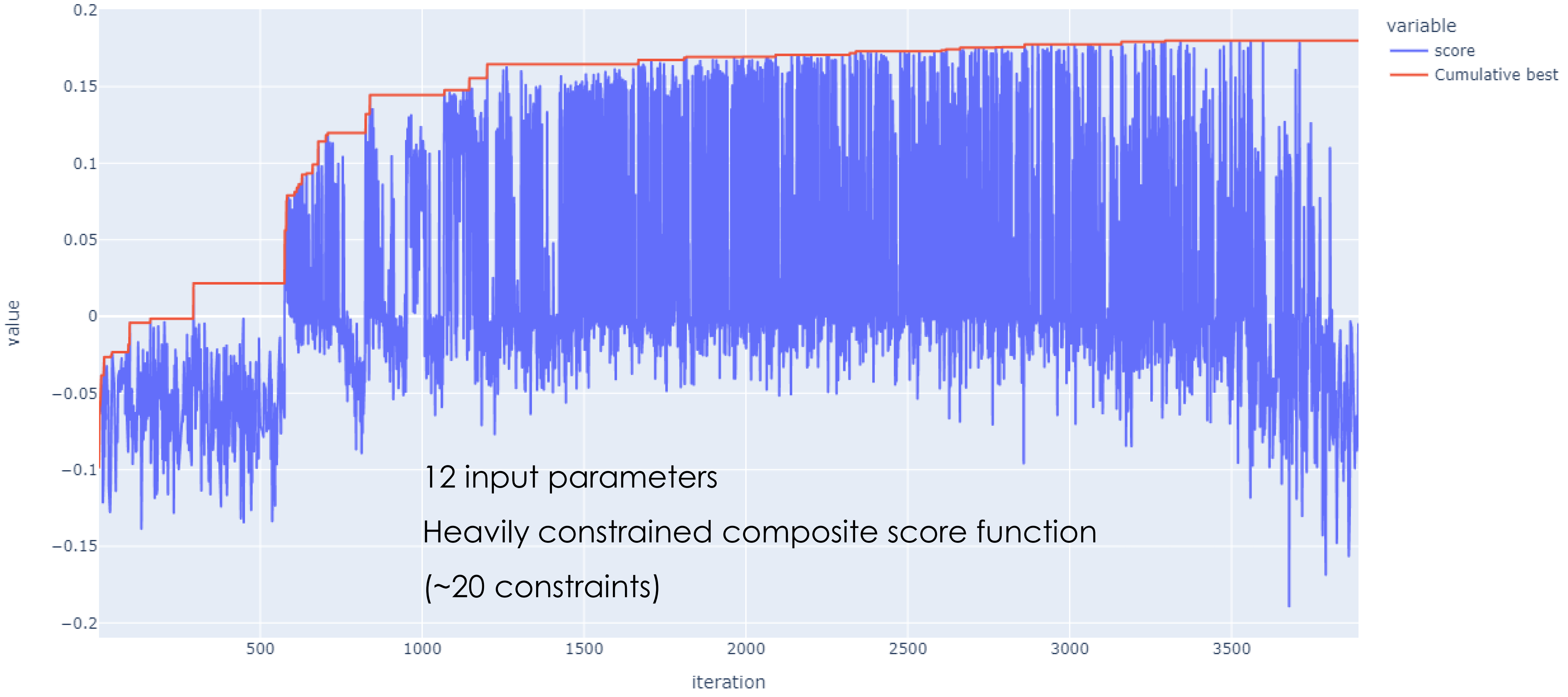
$$\text{score}_0 = \text{gain} \times \prod_i \text{penaliser}_i$$

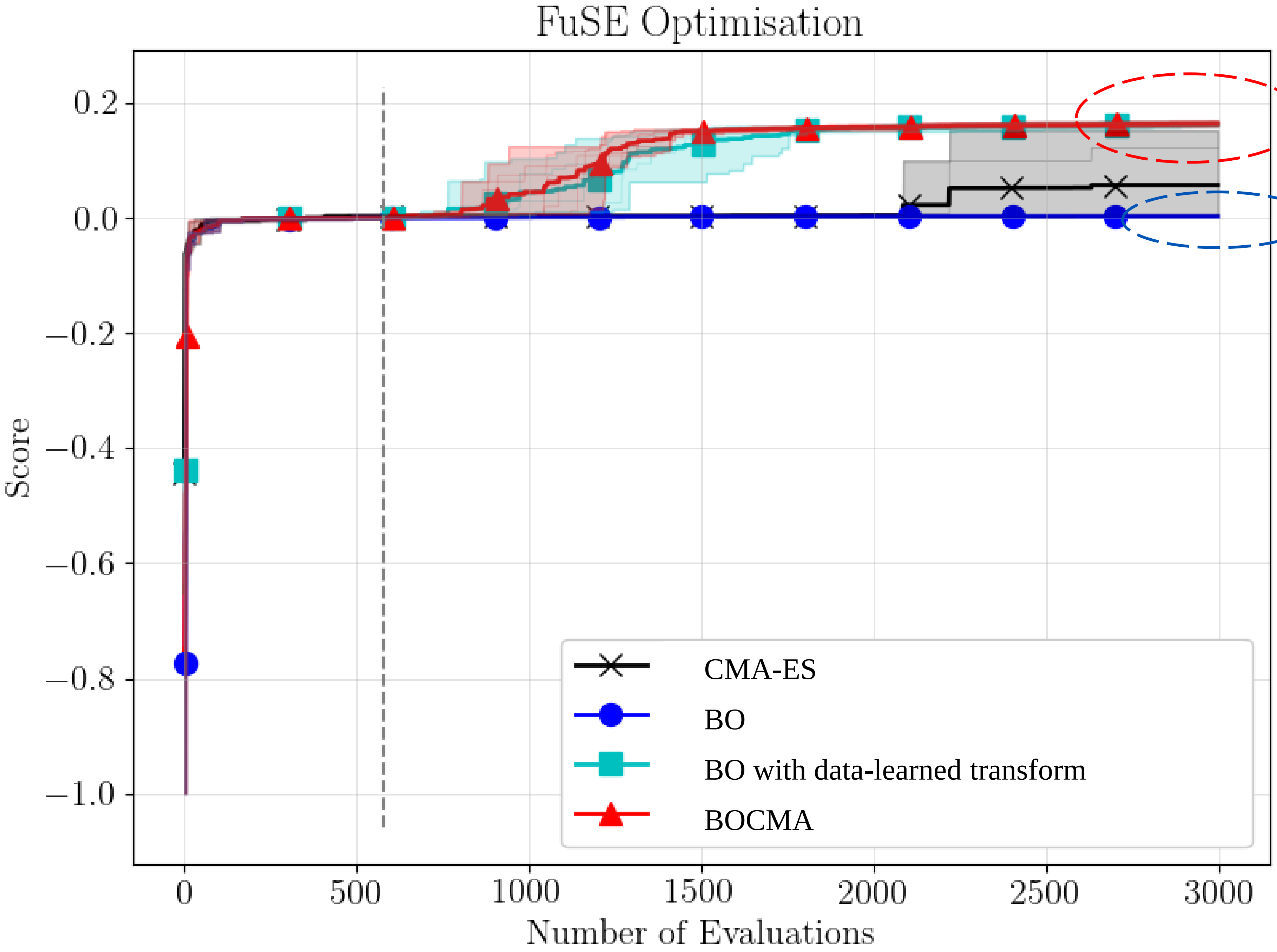
$$\text{penaliser}_i = 1 + \text{sign}_i \times \text{erf}\left(\beta\left(\frac{x_i - x_{i,c}}{x_{i,c}}\right)\right)$$

$$\text{score} = \begin{cases} \text{score}_0 & \text{if score}_0 \geq a \\ a\left(\log\left(\frac{\text{score}_0}{a}\right) + 1\right) & \text{if score}_0 < a \end{cases}$$

We used **a = 0.001** as the crossover point for log-transforming the score

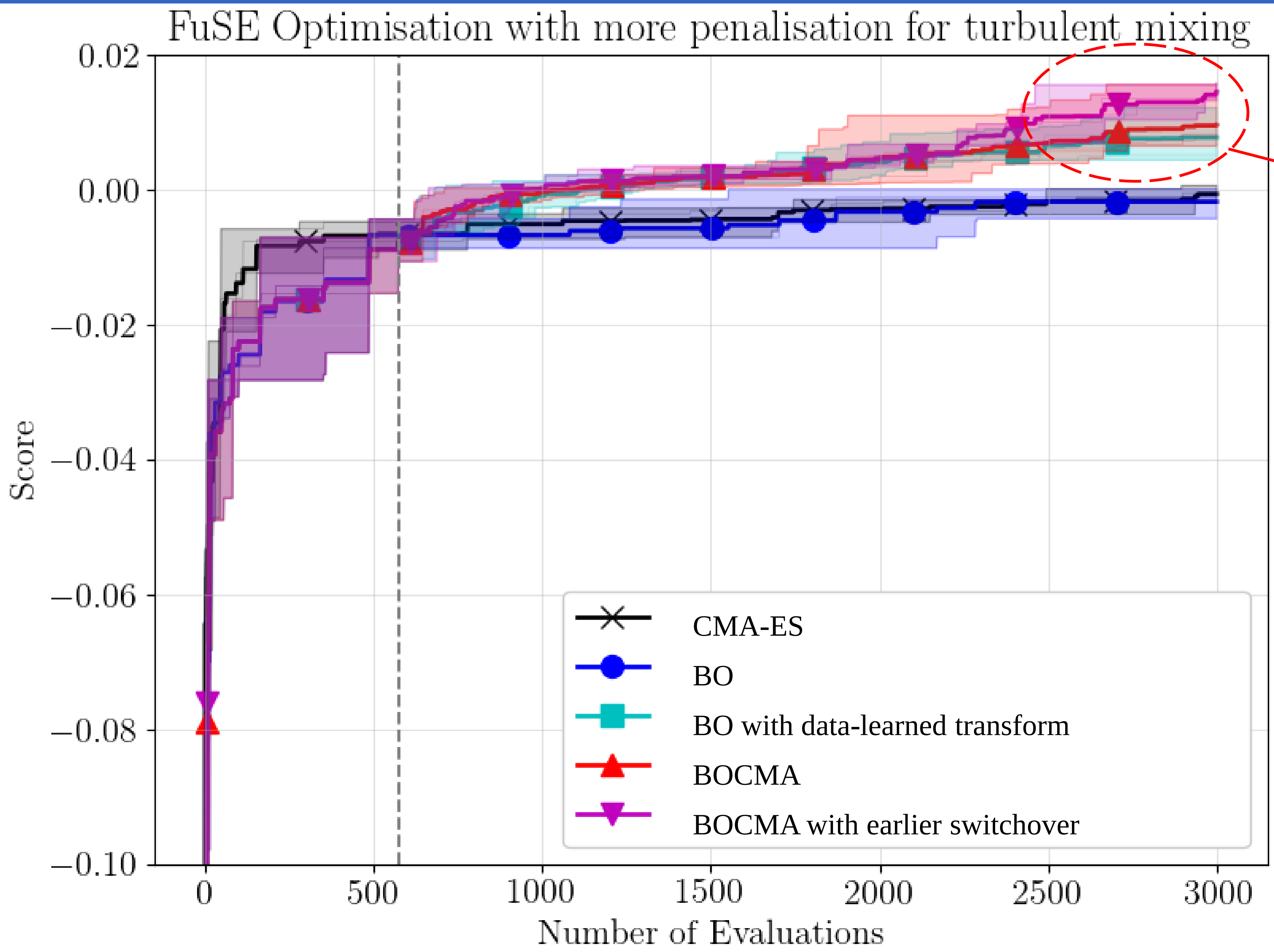
FuSE Optimisation Score vs Number of Evals





Bayes/BOCMA with data-learned transform is best

Standard Bayes opt on it's own performed worse



Clear improvement using BOCMA vs Bayes or CMA-ES alone

Summary

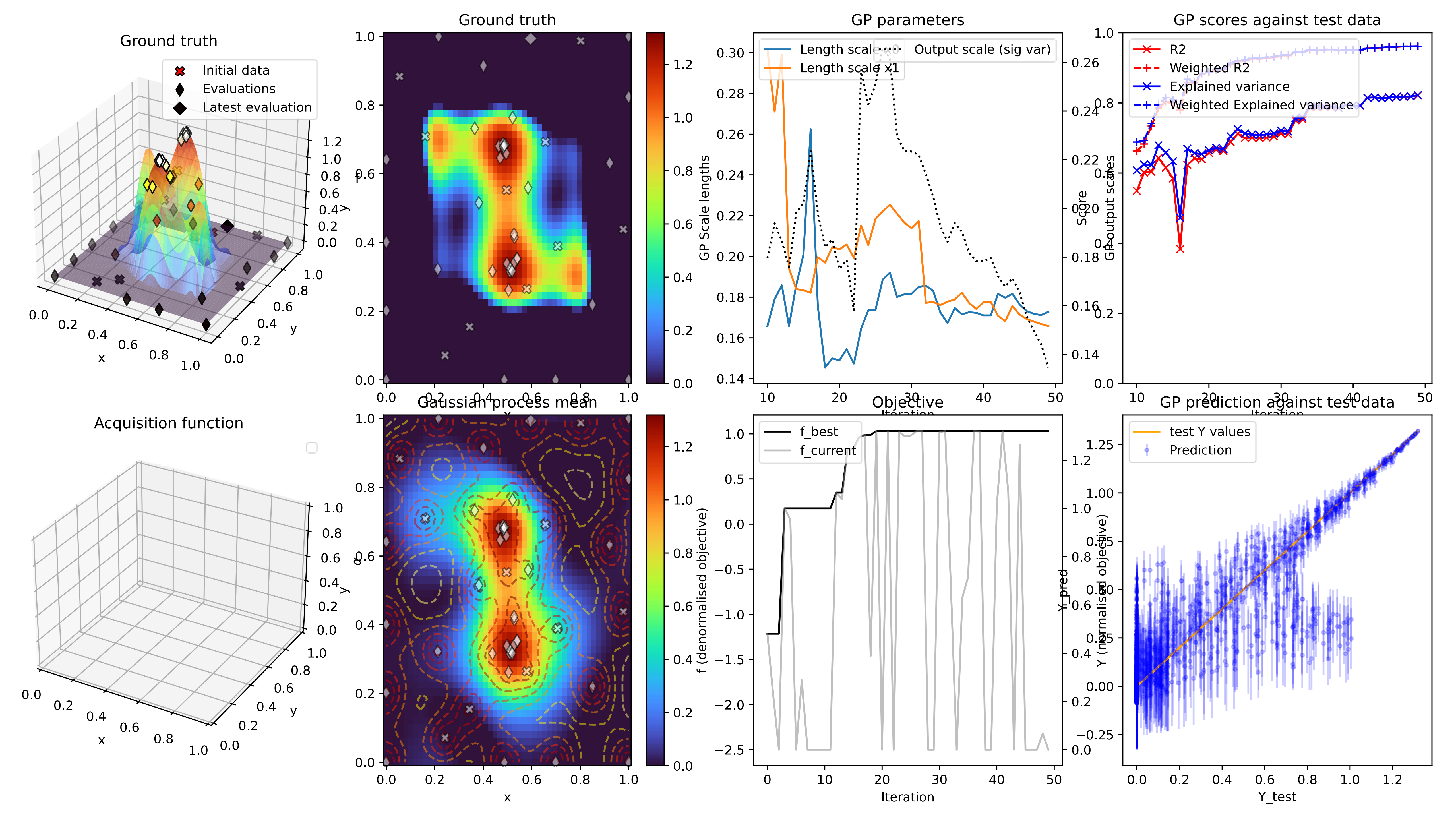
- A fault-tolerant, asynchronous optimisation and parameter sweeping toolkit has been developed
  - Scalable to large number of parallel evaluations
- Implements various algorithms out of the box: CMA-ES, Bayes opt (via BoTorch), Latin Hypercube sampling
  - Modular so other algorithms can be easily added on
- BO good for global optimisation but can **struggle with complex objectives** encountered in ICF design
  - Using data-learned transform helps with this
- CMA-ES is robust and easy to use but can fall into local optima
- Hybrid CMA-ES / Bayesian Optimisation algorithm (BOCMA)
  - Combines global optimisation of BO with robust local optimisation of CMA-ES
  - Fault tolerant (falls back to CMA-ES upon error)
- Future Roadmap: **Multi-objective**, **multi-fidelity** and **Constrained Optimisation under Uncertainty**
- We're looking for good benchmark problems and collaboration

first light

Thank you for your attention
Please get in touch

tom.edwards@firstlightfusion.com

APPENDIX



Ground truth
Initial data
Evaluations
Latest evaluation
Ground truth
GP parameters
Length scale x0
Length scale x1
Output scale (sig var)
GP Scale lengths
GP output scales
Iteration
GP scores against test data
R2
Weighted R2
Explained variance
Weighted Explained variance
Score
Iteration
Acquisition function
Gaussian process mean
f (denormalised objective)
Objective
f_best
f_current
Iteration
Y (normalised objective)
GP prediction against test data
test Y values
Prediction
Y_pred
Y_test

Posterior Prediction / Conditioning of Gaussian Process

Observations / Training Data

$$\mathcal{D} = (X, Y) = [(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N))]$$

Test points for evaluation

$$X_*$$

Prior mean and covariance

$$\boldsymbol{\mu} = \mu(X)$$
$$K = k(X, X)$$
$$\boldsymbol{\mu}_* = \mu(X_*)$$
$$K_* = k(X_*, X_*)$$
$$k_* = k(X, X_*)$$

Posterior mean and covariance

$$\tilde{\boldsymbol{\mu}} = \boldsymbol{\mu}_* + k_*^\top (K + \sigma_n I)^{-1} (Y - \boldsymbol{\mu})$$
$$\tilde{K} = K_* - k_*^\top (K + \sigma_n I)^{-1} k_*$$

Gaussian process model is
'conditioned' on observations Y

Covariance Functions

- Squared exponential (A.K.A Gaussian or Radial Basis Function)

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\sum_{i=1}^{d} \frac{1}{2l_i^2}(x_i - x_i')^2\right)$$

- Matern Kernel

$$k_{M_{1/2}}(x, x') = \exp\left(-|x - x'|\right)$$

$$k_{M_{3/2}}(x, x') = (1 + \sqrt{3}|x - x'|) \exp\left(-\sqrt{3}|x - x'|\right)$$

$$k_{M_{5/2}}(x, x') = (1 + \sqrt{5}|x - x'| + \frac{5}{3}|x - x'|^2) \exp\left(-\sqrt{5}|x - x'|\right)$$

- Linear, quadratic, periodic and combinations
- Many others

Hyper-parameter Estimation (tell)

Example Priors;

Constant mean and squared exponential kernel

$$\mu(x) = c$$

Output scale

$$k(x, x') = \sigma^2 \exp\left(\frac{1}{2l^2}(x - x')^2\right)$$

Length scale

Hyper-parameters

$$\boldsymbol{\theta} = [\sigma, l, c]$$

Maximise the log-likelihood using a **gradient based optimiser**

$$\log p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}^\top K_y^{-1}\mathbf{y} - \frac{1}{2}\log|K_y| - \frac{n}{2}\log 2\pi$$