# Data-driven Optimisation and Discovery For Engineering Systems

## Humans, Algorithms, & Applications

**Tom Savage**

Department of Chemical Engineering

Imperial College London

Submitted for the degree of

*PhD*

*There is yet one leverage point that is even higher than changing a paradigm. That is to keep oneself unattached in the arena of paradigms, to stay flexible, to realize that no paradigm is "true", that every one, including the one that sweetly shapes your own worldview, is a tremendously limited understanding of an immense and amazing universe that is far beyond human comprehension.* - Donella Meadows.

# Statement of Originality

Any work that has appeared outside of this thesis has been referenced and attributed appropriately. Otherwise, all work presented within this thesis is my own.

Tom Savage

# Copyright Declaration

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC). Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose. When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes. Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Tom Savage

# Acknowledgements

I would first like to thank my Mum, Bev, my Dad, Dave and my sister, Ami for supporting me throughout all of my studies. I appreciate that for them it hasn't always been easy to understand what I do at 'work' (at no point in the last 3-4 years did doing a PhD ever feel like going to 'work'), and I'm not sure reading this will even help, but I would not have been able to achieve what I have without them. To Jaina, the great cosmic Mistry of my life, who has given me endless joy.

I'd also like to acknowledge the huge role that many others have played in forming who I am in both an academic and a personal sense. Of course, my supervisor Antonio who has built one of the friendliest and most productive research groups from scratch over the past 4-5 years. Since we were first introduced in 2019, Antonio has provided the consistent resources and support that I needed to succeed, including enabling me to obtain the Imperial College President's Scholarship.

Ed D'Souza and Sunil Manghani who have given me endless support and some of the most eclectic discussions I have had the pleasure of being a part of. I have not taken for granted the ability to compare and combine notes with two people who are so knowledgeable in a field that is completely orthogonal to the one in this thesis. I've gained an appreciation that the worlds of engineering, science, and art are more connected than we might believe (even beyond the 'engineering/science is an art' trope).

From the Centre for Environmental Policy, Mark Workman for providing hugely valuable nuggets of wisdom from his time in the Army that have given me clarity regarding my

personal values and mission. Niall MacDowell for similarly pushing me to aim high with regards to publications and for the regular check-ups at the CMCL Christmas dinner, and Gbemi Oluleye for the consistent support throughout.

An immeasurable thank you to Cameron for our regular free squash games on a Sunday morning. Being able to talk about life and work is not to be taken for granted, and the Hello Fresh employee discount went a long way to fuelling a large chunk of the work in this thesis for better or for worse. From Manchester: Theo, Sol, Ruby, Annemarie, and Tom, thank you for being the nicest housemates and friends I could ask for. From Home: everyone in 8kgs of Steak/Toast/etc.... We have something truly unique. Specific thanks to Pete for providing a never-ending source of entertaining stories and less entertaining smells, and to Bean and Steele for provoking the opening argument (*the thesis*) to this thesis on a very enjoyable trip to Prague and for allowing me to use the kitchen island so liberally to write it.

Lastly, I'd like to thank the friends that I've made throughout my PhD. Miguel, Akhil, Damien, Brad, Abdullah, Ilya, and others. We've had the pleasure of travelling the world as friends, and have never failed to enjoy ourselves, may we continue to do so. Being able to discuss, critique, and enjoy each others work is no small deal, but the work in this thesis would have suffered without. We've done well.

# Abstract

This thesis explores the integration of machine learning, optimisation, and human expertise for scientific discovery and the improvement of engineering systems. Each chapter applies deductive reasoning through physics-based models, inductive reasoning through data-driven approaches, or a combination of the two, demonstrating how together they drive scientific progress. First, a methodology is established for designing novel chemical flow reactors by combining multi-fidelity Bayesian optimisation with computational fluid dynamics. This approach enables automated discovery of reactor geometries and validation of induced flow characteristics, improving performance by 60% over conventional designs. Key features of optimal designs are identified and validated experimentally through 3D-printed reactors, demonstrating how mixing-enhancing vortical structures can be induced at previously unattainable conditions. Second, a human-algorithm collaborative Bayesian optimisation framework is presented that integrates domain expertise into data-driven decision-making. By exploiting the hypothesis that humans are more efficient at discrete rather than continuous choices, the framework enables experts to influence solution selection via Bayesian optimisation with minimal effort. Case studies across benchmark functions, reactor design, and bioprocess optimisation demonstrate that even partial expertise can accelerate convergence compared to standard methods, particularly for noisy, high-dimensional problems. Finally, in-silico improvements to Bayesian optimisation are explored using large language models (LLMs) as computational decision-makers. This approach maintains automation while demonstrating convergence improvements similar to those from expert intervention. By using

LLMs to select between discrete candidate solutions, the methodology identifies patterns in optimisation trajectories and makes informed selections. This thesis establishes data-driven optimisation as a powerful framework for scientific discovery in engineering systems beyond the paradigms of modelling and optimisation. The practical improvements ranging from more efficient physical reactor designs to accelerated optimisation processes demonstrate how the methodologies presented in this thesis can advance scientific and engineering discovery.

# Table of contents

**List of figures** **xix**

**List of tables** **xxxiii**

**1 Introduction** **1**

1.1 Context within Chemical Engineering . . . . . 2

1.2 Gaining Knowledge . . . . . 4

1.3 Discovery and Outlook . . . . . 6

1.4 Contributions and Structure . . . . . 8

**2 Background** **11**

2.1 Modelling . . . . . 12

2.1.1 Physical Models . . . . . 13

2.1.2 Data-driven Models . . . . . 16

2.1.3 Modelling Summary . . . . . 33

2.2 Optimisation . . . . . 33

2.2.1 Numerical Optimisation . . . . . 34

2.2.2 Data-driven (Zero Order) Optimisation . . . . . 41

2.2.3 Bayesian Optimisation . . . . . 49

2.2.4 Other Optimisation Paradigms . . . . . 57

2.2.5 Optimisation Summary . . . . . 59

2.3 Chapter Summary . . . . . 60

**3 Methods for Data-Driven Design and Analysis of Reactor and Tube Simulations 61**

3.1 Introduction . . . . . 65

3.2 Background . . . . . 69

3.2.1 Multi-fidelity Bayesian optimisation . . . . . 69

3.3 Method . . . . . 73

3.3.1 Parameterisation . . . . . 73

3.3.2 Fidelities . . . . . 74

3.3.3 Simulation . . . . . 76

3.3.4 Multi-fidelity Bayesian optimisation . . . . . 78

3.3.5 Stopping criteria . . . . . 79

3.4 Results . . . . . 81

3.4.1 Parameterisation . . . . . 83

3.4.2 Model Validation . . . . . 84

3.4.3 Solution Validation . . . . . 91

3.5 Chapter Conclusions and Future Work . . . . . 93

**4 Machine Learning-assisted Discovery of Flow Reactor Designs 95**

4.1 Introduction . . . . . 99

4.2 Methods . . . . . 101

4.2.1 Parameterisations . . . . . 101

4.2.2 Simulation . . . . . 106

4.2.3 Optimisation . . . . . 107

4.2.4 Experimental Validation . . . . . 108

4.3 Results . . . . . 109

4.3.1 Effect of Design Features on Flow Structures . . . . . 110

4.3.2 Convergence Analysis . . . . . 114
4.3.3 Optimal Designs & Comparison with 3D-Printed Reactors . . . . . 118
4.4 Discussion . . . . . 121

**5 Human-algorithm Collaborative Bayesian Optimisation for Engineering Systems 123**

5.1 Introduction . . . . . 128
5.1.1 Contributions . . . . . 131
5.2 Background . . . . . 132
5.3 Methodology . . . . . 132
5.3.1 Expert-augmented Design of Experiments . . . . . 133
5.3.2 Defining Alternate Solutions . . . . . 134
5.3.3 Algorithm Characteristics . . . . . 136
5.3.4 Selecting from a Set of Solutions . . . . . 139
5.4 Case Studies . . . . . 139
5.4.1 Benchmark Functions . . . . . 140
5.4.2 Pulsed-flow Coiled Tube Reactor optimisation . . . . . 142
5.4.3 Black-box Bioprocess optimisation . . . . . 143
5.4.4 Human-in-the-loop Experimental Protocol . . . . . 146
5.5 Results & Discussion . . . . . 147
5.5.1 Effect of Noise . . . . . 147
5.5.2 Effect of Dimensionality . . . . . 149
5.5.3 Effect of Number of Alternative Solutions . . . . . 151
5.5.4 Effect of Specific Function Properties . . . . . 152
5.5.5 Coiled Tube Reactor Design . . . . . 154
5.5.6 Fed-batch Bioprocess Control . . . . . 155
5.6 Conclusions . . . . . 157

**6 Accelerating Convergence with LLM-in-the-Loop Bayesian Optimisation 159**

6.1 Introduction . . . . . 164

6.1.1 Contributions . . . . . 166

6.2 Background . . . . . 167

6.2.1 Human-in-the-Loop Optimisation . . . . . 167

6.2.2 Large Language Models for Scientific Tasks . . . . . 167

6.3 Methodology . . . . . 168

6.3.1 Generating Candidate Solutions . . . . . 169

6.3.2 Parsing Candidate Solutions for LLMs . . . . . 170

6.3.3 LLM Decision-Making Procedure . . . . . 171

6.3.4 Output Format and Selection Decision . . . . . 176

6.4 Results . . . . . 176

6.4.1 Performance on Classical Test Functions . . . . . 176

6.4.2 Comparison of Prompting Strategies . . . . . 179

6.4.3 Performance on Random Functions . . . . . 181

6.4.4 Chemical Optimisation Case Study . . . . . 184

6.5 Conclusions . . . . . 187

**7 Conclusion 191**

**References 199**

**Appendix A Methods for Data-Driven Design and Analysis of Reactor and Tube Simulations 215**

A.1 Pulsed-flow Helical-tube Reactors . . . . . 215

A.2 Effect of hyperparameters . . . . . 216

**Appendix B Machine Learning-assisted Discovery of Flow Reactor Designs 221**

B.1 Parameterisation & Meshing . . . . . 221
B.1.1 Cross Section . . . . . 222
B.1.2 Coil Path . . . . . 225
B.2 Simulation . . . . . 226
B.3 Optimisation Progress . . . . . 227
B.3.1 Coil Path Parameterisation . . . . . 227
B.3.2 Joint Parameterisation . . . . . 229
B.4 Experimental Uncertainty Quantification . . . . . 230

**Appendix C Human-algorithm Collaborative Bayesian Optimisation for Engineering Systems 233**

C.1 Optimisation Benchmark Functions . . . . . 233
C.1.1 Rastrigin . . . . . 233
C.1.2 Rosenbrock . . . . . 233
C.1.3 Ackley . . . . . 234
C.1.4 Schewefel . . . . . 234
C.2 Sampled Benchmark Functions . . . . . 234
C.3 Effect of acquisition function . . . . . 236
C.4 Effect of using an evolutionary approach to optimise acquisition function . 238

**Appendix D In-silico Improvements to Bayesian Optimisation using Large Language Models 241**

D.1 LLM Prompts for Decision Making . . . . . 241
D.1.1 Basic One-Shot Prompt . . . . . 241
D.1.2 Historian Prompt . . . . . 243
D.1.3 Differentiator Prompt . . . . . 246
D.1.4 Decision-maker Prompt . . . . . 248

# List of figures


1.1 Prague Astronomical Clock (Orloj). Image by Tom Savage, 2024. . . . . 5

2.1 One-dimensional example dataset used for the remainder of this section. . . 16

2.2 The values of the linear, quadratic, and Huber loss functions with respect to the error between a prediction and an observation. . . . . 18

2.3 Different degree one-dimensional polynomial models fit to the same dataset. As the degree increases accuracy may diminish due to overfitting. Here, $p$ represents the degree of the polynomial. . . . . 19

2.4 Increasing the neurons within a single layer neural network generally results in more smooth functions. Overfitting may still occur in low-data regimes, but with large amounts of data, neural networks tend to produce smoother, more generalisable functions. This behavior contrasts with traditional parametric models that may overfit as complexity increases. . . . . 21

2.5 Equation 2.10 used to model the one-dimensional example data. Here, $k$ refers to the number of observations used to define a local model. . . . . 23

2.6 Different values of $\sigma$ result in stronger or weaker correlations with points at a fixed distance. . . . . 24

2.7 Different values of $\sigma$ produce more, or less smooth approximations to an underlying function. . . . . 25

2.8 The value of each dimension of a single sample from a 20 dimensional multivariate Gaussian distribution with identity covariance matrix. . . . . 26
2.9 The value of each dimension of a single sample from a 20 dimensional multivariate Gaussian distribution with covariance matrix as defined by a kernel function. . . . . 26
2.10 The value of each dimension of a single sample from a 20 dimensional multivariate Gaussian distribution with covariance matrix as defined by a kernel function. . . . . 27
2.11 The value of each dimension of a single sample from a 20 dimensional multivariate Gaussian distribution with covariance matrix as defined by a kernel function. . . . . 27
2.12 Functional samples from a Gaussian process and the continuous covariance matrix that results from a kernel function. . . . . 28
2.13 A conditioned two-dimensional (joint) Gaussian distribution. The resulting distribution is always Gaussian. . . . . 29
2.14 Samples from a conditioned Gaussian process and resulting covariance matrix, the diagonal of which corresponds to the variance at each location. . 31
2.15 Two-dimensional test function used to illustrate approaches to numerical optimisation. . . . . 35
2.16 Different first order methods applied for 50 iterations to optimise the test function. Here, $\alpha$ represents step size and $\beta$ represents momentum. . . . . 39
2.17 Newton's method and standard gradient descent applied at the same starting point. Individual iterations are marked with crosses. . . . . 41
2.18 The random search algorithm, connections are drawn between successive solutions. . . . . 42
2.19 Nelder-Mead simplex operations . . . . . 43

2.20 The Nelder-Mead algorithm, connections are drawn between successive evaluations. . . . . 44

2.21 The population of solutions within 100 iterations of an evolutionary algorithm. The transparency of a solution is proportional to the iteration of its population. 45

2.22 Successive quadratic models used either globally or within a trust region. Trust regions specify smaller local areas that behave closer to the quadratic approximation, resulting in convergence. . . . . 49

2.23 A Gaussian process and respective expected improvement function. . . . . 51

2.24 A Gaussian process with assumed noisy observations fit to a one-dimensional data set. . . . . 53

2.25 One-dimensional lines representing potential improvements over the current best observation as a function of standardised observations $z$, for a single evaluation of the utility function at $x = 5.0$. The upper envelope of these lines is used to analytically solve the integral in the noisy expected improvement acquisition function. . . . . 55

2.26 Noisy expected improvement acquisition function for the Gaussian process model shown in Figure 2.24. Each value represents the expected improvement in the 'signal' of the model, accounting for normally distributed noisy measurements. . . . . 56

3.1 The progression of multi-fidelity Bayesian optimisation using DGPs as a multi-fidelity model. Multiple different fidelity levels are selected for evaluation throughout optimisation. . . . . 71

3.2 *Left*: A side-view of how coil radius, pitch, and the inversion parameter affects a given coil, with an additional horizontal inlet and outlet. *Right*: When parameterising a coil with horizontal inlet and outlet, a smooth transition is included based on a quadratic interpolation of points. . . . . 73

3.3 A side-view of the meshing procedure of a coil based on axial fidelity. A central path is first defined based on geometric parameters. Subsequently, circles (tube cross-sections) are defined along this path the number of which depends on the axial fidelity. Finally, the mesh density near the inlet and outlet walls is increased by raising the number of circles to track the tracer concentration with higher accuracy. . . . . 75

3.4 A cross-section of the mesh radially. First, a coarse finite element topology is created. Subsequently, the number of cells near the tube wall are increased in order to better capture the effects of fluid flow near the boundary layer. Finally the mesh is further subdivided based on the radial fidelity value. . . 75

3.5 Experimental residence time distribution against the residence time distribution modelled using the ideal tank-in-series relation, given by Eq. 3.3, with a value of equivalent tanks in series derived from Eq. 4.4. Configurations that perform worse, resulting in lower values of N (right), result in a broader residence time distribution, the maximum concentration occurring earlier, and with a lower value. . . . . 78

3.6 The effect of inversion parameter $\delta$ for a helical coil tube with fixed length, top view. From left to right: $\delta$ is evaluated at 0, 0.15 0.3, 0.6, and 0.75 with coil radius and pitch remaining constant. . . . . 83

3.7 The effect of pitch $\phi$ for a helical coil tube with a fixed length, side view. From left to right: $\phi$ is evaluated from 7.5mm to 15mm with coil radius and inversion location remaining constant. . . . . 83

3.8 The effect of coil radius $\rho$ for a helical coil tube with a fixed length, side view. From left to right: $\rho$ is evaluated from 3mm to 12.5mm with inversion location and pitch remaining constant. . . . . 84

3.9 An instance of helical-tube reactor geometry as affected by axial and radial fidelity. . . . . 84

3.10 Validation of five discrete mesh fidelities corresponding to different cell counts, across two sets of experimental data under different conditions.[1] . . 85

3.11 The number of equivalent tanks-in-series evaluated labelled by the respective cost of simulation. The upper half of the figure shows these quantities against iteration and the lower half shows these quantities against wall-clock time, highlighting the importance of lower-cost simulations. . . . . 86

3.12 The fidelities selected throughout optimisation within $\mathbb{Z}$. The upper plot demonstrates the average simulation cost at each fidelity. As discrete fidelities are applied which are continuously approximated, the lower plot is presented demonstrating the number of times each discrete fidelity is evaluated. Simulation cost is confirmed to be not only a function of fidelity, but $\mathbf{x}$ as well, as there is no clear distribution of simulation costs across $\mathbb{Z}$. . . . . 88

3.13 The maximum predicted time to complete a 'standard' iteration compared with the maximum predicted time to complete both a standard iteration as well as a greedy iteration, the actual simulation time, and the remaining time. As the blue line passes above the black line, a greedy iteration must be performed as there is not enough remaining budget to perform both a greedy and standard evaluation . . . . . 88

3.14 Optimal reactor geometry viewed from different perspectives. The optimal geometry contains an inversion. . . . . 89

3.15 Dean vortices visualised throughout the length of the optimal reactor with the vortices near the outlet of the reactor seen to be stronger than for those associated with a standard, sub-optimal coil configuration. . . . . 90

3.16 Streamlines indicating tracer concentration within the optimal reactor throughout a single oscillatory pulse, lasting 0.5 seconds. As the oscillatory velocity changes from positive to negative, Dean's vortices form providing mixing throughout the radial direction. The inversion within the coil also contributes to these vortices shifting across the radial direction, providing stronger mixing characteristics. Tracer concentration ranges from 0 (purple) to 1 (red). . 91

3.17 Experimental validation of optimal reactor configuration via additive manufacturing and operation under pulsed-flow operating conditions. . . . . 92

3.18 The residence-time distribution predicted via CFD simulation of the optimal, high-fidelity solution returned from the framework, alongside 3 sets of experimental data obtained via the 3D printed reactor. . . . . 92

4.1 **Left**: Polar Gaussian process parameterisation of coil cross section, demonstrating samples from a Gaussian process prior with a polar kernel. **Right**: The posterior distribution after the polar Prior distribution has been conditioned on data. . . . . 103

4.2 Examples of sets of inducing points at given locations along the length of a coil. A polar Gaussian process is used to interpolate between points, where each data point's radial value is a parameter. . . . . 104

4.3 A flow-diagram of the main aspects of the methodology of this chapter. . . 110

4.4 The conventional coiled-tube reactor (i) alongside optimal coiled-tube reactors generated by parameterising the cross-section (ii), the coil path (iii), and a joint parameterisation (iv). . . . . 111

4.5 Velocity streamlines coloured with velocity magnitude within a standard coil (i) and within optimal joint parameterisation (ii). . . . . 112

4.6 Secondary flow streamlines at various cross-sections of the coil. . . . . 112

4.7 Cross-sectional plane across the coil demonstrating streamwise velocities. . 113

4.8 The presence of induced Dean vortices within the optimal joint parameterisation coil (ii) compared with a standard coil (i). . . . . 114

4.9 The optimisation objective against iteration (i) and wall-clock time (ii). The objective has been normalised. The initial data set generated via design-of-experiments is denoted as negative iterations and wall-clock time. Hence, optimisation begins at the $0^{\text{th}}$ iteration and at $t = 0$. . . . . 115

4.10 Gaussian process dimension lengthscales throughout Bayesian optimisation iterations (top) alongside histograms demonstrating the distribution of GP lengthscales changing throughout optimisation (bottom). . . . . 116

4.11 t-SNE analysis of the data generated throughout optimisation in design parameter space ($\mathfrak{X}$), reducing the dimensionality of design parameters to two dimensions, and labelled with different respective quantities. . . . . 117

4.12 Parameter variability, defined as a function of the GP lengthscale corresponding to each parameter, plotted for each inducing parameter on a nominal coil. . . . . 118

4.13 **Comparisons with nominal coiled-tube designs.** The nominal coiled-tube (**left**, R1) alongside extrapolated steady-state flow coil designs containing aspects from the optimal cross-section (**centre**, R2) and both cross-section and coil path (**right**, R3). . . . . 119

4.14 Data generated across three experiments for each reactor configuration, designed to maximise equivalent tanks-in-series and minimise non-symmetric RTDs. A tanks-in-series model is used to estimate the performance of each reactor across the experiments performed. Dimensionless concentration at the outlet of the reactor ($E(\theta)$) is plotted against dimensionless time ($\theta$). **a**, the control reactor (R1). **b**, the reactor with variable cross-section (R2). **c**, the reactor with variable cross-section and path (R3). . . . . 119

5.1 Methodology overview for proposing alternative solutions to a human expert using a high-throughput formulation solved with multi-objective optimisation. The trade-off between the sum of utility function values and the determinant of their covariance matrix is optimised to obtain a set of high-utility and distinct solutions, from which the expert selects the most promising one for evaluation. . . . . 131

5.2 The expert-augmented design of experiments procedure outlined using a two dimensional example. **Left:** the expert defined set of solutions. **Center:** the initial random set of solutions defined as $\mathbf{X}_{\text{design}}$. **Right:** the optimal set of solutions comprising of both $\mathbf{X}_{\text{design}}$ alongside $\mathbf{X}_{\text{expert}}$, resulting in the final design of experiments. . . . . 134

5.3 A standard iteration of our algorithm on a one-dimensional case-study. . . . 138

5.4 A reactor configuration visualised in order to aid the expert in making discrete choices. In doing so, the expert had a reduced need to interpret higher-dimensional decision vectors. . . . . 143

5.5 A reactor configuration visualised in order to aid the expert in making discrete choices. In doing so, the expert had a reduced need to interpret higher-dimensional decision vectors. . . . . 144

5.6 Four alternative control solutions, as made available to the expert in order to facilitate easier reasoning particularly in higher dimensions. . . . . 145

5.7 An instance of the textual information made available to the expert at each iteration. . . . . 146

5.8 The effect of the level of observation noise on the difference in convergence for various expert behaviours across general 2D and 3D functions of a given length scale. . . . . 148

5.9 The effect of the dimensionality of the problem on the difference in convergence for various expert behaviours. . . . . 150
5.10 The effect of the number of alternative solutions made available on convergence for various expert behaviours. . . . . 151
5.11 Convergence of collaborative Bayesian optimisation of the 2D Rastrigin function with no observation noise. . . . . 152
5.12 Convergence of collaborative Bayesian optimisation of the 2D Ackley function with no observation noise. . . . . 153
5.13 Convergence of collaborative Bayesian optimisation of the 5D Schewefel function with 5% observation noise. . . . . 153
5.14 Simple and average reward, corresponding to the number of equivalent tanks-in-series for a given reactor geometry and operating conditions, presented alongside the reward obtained from standard Bayesian optimisation. Filled sections represent one standard deviation and solid lines represent the mean reward. . . . . 154
5.15 Simple and average reward presented alongside the reward obtained from standard Bayesian optimisation. Filled sections represent one standard deviation and solid lines represent the mean reward. . . . . 155

6.1 The solution in JSON format `{"x1":1.4056,"x2":2.3033,"x3":5.690,"x4":23.349}` tokenised using OpenAI's tokeniser `tiktoken`, pre-normalisation and rounding (above) and post-normalisation and rounding (below). Fewer tokens are used post-processing resulting in a single token for each variable value. . . 171
6.2 Convergence comparison of optimisation strategies across test functions when presented with two alternative solutions, highlighting the difference in performance between the basic one-shot prompt and the multi-agent strategy. 180

6.3 Example random functions sampled from Gaussian process priors at different lengthscales. As the lengthscale decreases from 0.5 to 0.05, the functions transition from smoother to more complex behavior. . . . . 181

6.4 Convergence comparison on random functions with lengthscale 0.5, using two alternative solutions across various dimensions. Performance of LLMBO (Deepseek One-Shot and MoA) is compared against Expert, standard BO, and Adversarial baselines. . . . . 182

6.5 Convergence comparison on random functions with lengthscale 0.05, using two alternative solutions across various dimensions. Performance of LLMBO (Deepseek One-Shot and MoA) is compared against Expert, standard BO, and Adversarial baselines. . . . . 183

6.6 Convergence comparison of different LLM models and prompting strategies on random functions with lengthscale 0.5, using two alternative solutions across dimensions $d = 2,3,4,5$. Performance of LLMBO (Deepseek V3 with one-shot and multi-agent prompts and Anthropic Haiku with one-shot prompt) is compared. . . . . 184

6.7 Convergence comparison for the simulated 10-variable photocatalysis optimisation task. Performance of standard Bayesian Optimisation (BO) is compared against LLMBO using the Deepseek model with both a basic One-Shot prompt and a Chain-of-Thought (CoT) reasoning prompt. Results show the mean normalised performance over multiple runs, with shaded areas representing confidence intervals. . . . . 186

A.1 The number of equivalent tanks-in-series evaluated coloured by the respective cost of simulation. optimisation is performed with hyper-parameters $\gamma = 1/2$ and $\beta = 1/2$. The upper half of the figure shows these quantities against iteration and the lower half shows these quantities against wall-clock time, highlighting the importance of lower-cost simulations. . . . . 216

A.2 The fidelities selected throughout optimisation within $\mathbb{Z}$. The upper plot demonstrates the average simulation cost at each fidelity. optimisation is performed with hyper-parameters $\gamma = 1/2$ and $\beta = 1/2$. . . . . 217

A.3 The optimal solution resulting from optimisation with hyper-parameters $\gamma = 1/2$ and $\beta = 1/2$, the solution visually is similar to the solution presented in the main part of this article. The number of equivalent tanks-in-series of this configuration and operating conditions is 34.5. . . . . 217

A.4 The number of equivalent tanks-in-series evaluated coloured by the respective cost of simulation. optimisation is performed with hyper-parameters $\gamma = 1/2$ and $\beta = 1$. The upper half of the figure shows these quantities against iteration and the lower half shows these quantities against wall-clock time, highlighting the importance of lower-cost simulations. . . . . 218

A.5 The fidelities selected throughout optimisation within $\mathbb{Z}$. The upper plot demonstrates the average simulation cost at each fidelity. optimisation is performed with hyper-parameters $\gamma = 1/2$ and $\beta = 1$. . . . . 219

A.6 The optimal solution resulting from optimisation with hyper-parameters $\gamma = 1/2$ and $\beta = 1$, the solution differs in that it does not contain an inversion, however the coil radius and operating conditions are consistent with the other optimal solutions. The number of equivalent tanks-in-series of this configuration and operating conditions is 26.8. . . . . 219

B.1 The radial variables that define the cross section of the reactor throughout the length, alongside appended cross-sections of constant radius at the start and end of the reactor. . . . . 222
B.2 Polar GP interpolation of the 6 inducing points for each distinct cross-section. 64 points are returned which represent the highest-fidelity representation of coil cross-section. . . . . 223
B.3 An example of cross-section inducing points and Polar GP interpolations. . 223
B.4 The resulting data after GP interpolation of each cross-section, and appropriate spacing in 3D cartesian coordinates. . . . . 224
B.5 Different perspectives of interpolated cross-sections distributed along coil path. . . . . 224
B.6 Final data made available as a result of cross-section parameterisation. . . . 225
B.7 The optimisation objective against iteration and wall-clock time for the optimisation of coil path. . . . . 228
B.8 t-SNE analysis of the data generated throughout optimisation in design parameter space, reducing the dimensionality of design parameters to two dimensions, and labelled with different respective quantities. . . . . 228
B.9 The optimisation objective against iteration and wall-clock time for the optimisation of coil path, given the fixed coil path geometry. . . . . 229
B.10 t-SNE analysis of the data generated throughout optimisation of coil cross-section given a fixed optimal coil path in design parameter space, reducing the dimensionality of design parameters to two dimensions, and labelled with different respective quantities. . . . . 230
B.11 Experimental data and associated ‘optimal’ tank-in-series model for reactor 1 (control). . . . . 231

B.12 Experimental data and associated 'optimal' tank-in-series model for reactor 2 (varying cross-section). . . . . 231

B.13 Experimental data and associated 'optimal' tank-in-series model for reactor 3 (varying cross-section and path). . . . . 231

B.14 Samples from the estimated distribution of N values for each reactor. . . . . 232

B.15 All experimental data across all three reactors, alongside uncertainty estimates of 'optimal' tanks-in-series models. . . . . 232

C.1 Functions sampled from a Gaussian process prior used for 1D benchmarking $f \sim \mathcal{GP}(\mu \equiv 0, K_M(d, \nu = 0.05))$ where $K_M$ is the Matérn 5/2 kernel function. 235

C.2 Functions sampled from a Gaussian process prior used for 2D benchmarking $f \sim \mathcal{GP}(\mu \equiv 0, K_M(d, \nu = 0.04))$ where $K_M$ is the Matérn 5/2 kernel function. 236

C.3 The effect of the acquisition function on the difference in convergence for sampled 2D functions. . . . . 237

C.4 The effect of the acquisition function on the difference in convergence for sampled 3D functions. . . . . 238

C.5 The effect of the dimensionality of the problem on the difference in convergence for various expert behaviours when using an evolutionary approach to locate the single optima of the acquisition function. . . . . 239

# List of tables

4.1 Absorbance values at 353 nm at the outlet of each reactor following a Villermaux–Dushman reaction. . . . . 121

5.1 Human behaviours Applied for Benchmarking . . . . . 140
5.2 The parameters and respective bounds for the optimisation of pulsed-flow reactor geometry and operating conditions. . . . . 142
5.3 The parameters and respective bounds for the optimisation of a fed batch bioprocess. . . . . 145

6.1 Benchmark results for the Ackley function using LLM-BO with Sonnet 3.5 and a multi-agent approach . . . . . 178
6.2 Benchmark results for the Rastrigin function using LLM-BO with Sonnet 3.5 and a multi-agent approach . . . . . 178
6.3 Benchmark results for the Schwefel function using LLM-BO with Sonnet 3.5 and a multi-agent approach . . . . . 179

# Nomenclature

| Symbol | Typical meaning |
|---|---|
| $a, b, c, \alpha, \beta, \gamma$ | Scalars are lowercase |
| $\mathbf{x}, \mathbf{y}, \mathbf{z}$ | Vectors are bold lowercase |
| $\mathbf{A}, \mathbf{B}, \mathbf{C}$ | Matrices are bold uppercase |
| $\mathbf{x}^\top, \mathbf{A}^\top$ | Transpose of a vector or matrix |
| $\mathbf{A}^{-1}$ | Inverse of a matrix |
| $(\mathbf{x}, \mathbf{y})$ | Inner product of $\mathbf{x}$ and $\mathbf{y}$ |
| $\mathbf{x}^\top \mathbf{y}$ | Dot product of $\mathbf{x}$ and $\mathbf{y}$ |
| $B = (b_1, b_2, b_3)$ | (Ordered) tuple |
| $B = [b_1, b_2, b_3]$ | Matrix of column vectors stacked horizontally |
| $B = \{b_1, b_2, b_3\}$ | Set of vectors (unordered) |
| $\mathbb{Z}, \mathbb{N}$ | Integers and natural numbers, respectively |
| $\mathbb{R}, \mathbb{C}$ | Real and complex numbers, respectively |
| $\mathbb{R}^n$ | $n$-dimensional vector space of real numbers |
| $\forall x$ | Universal quantifier: for all $x$ |
| $\exists x$ | Existential quantifier: there exists $x$ |
| $a := b$ | $a$ is defined as $b$ |
| $a =: b$ | $b$ is defined as $a$ |
| $\circ f$ | Function composition: "$\circ$ after $f$" |
| $\iff$ | If and only if |
| $\implies$ | Implies |
| $a \in A$ | $a$ is an element of set $A$ |
| $\emptyset$ | Empty set |
| $D$ | Number of dimensions; indexed by $d = 1, \ldots, D$ |
| $N$ | Number of data points; indexed by $n = 1, \ldots, N$ |
| $\mathbf{I}_m$ | Identity matrix of size $m \times m$ |
| $\mathbf{0}_{m,n}$ | Matrix of zeros of size $m \times n$ |
| $\mathbf{1}_{m,n}$ | Matrix of ones of size $m \times n$ |
| $\mathbf{e}_i$ | Standard/canonical vector (where $i$ is the component that is 1) |
| $\dim$ | Dimensionality of vector space |
| $\operatorname{span}[b_i]$ | Span (generating set) of $b_i$ |
| $\operatorname{tr}(\mathbf{A})$ | Trace of $\mathbf{A}$ |
| $\det(\mathbf{A})$ | Determinant of $\mathbf{A}$ |
| $\lvert \cdot \rvert$ | Absolute value or determinant (depending on context) |
| $\lVert \cdot \rVert$ | Norm; Euclidean, unless specified |

# Chapter 1

# Introduction

## Contents


1.1 Context within Chemical Engineering . . . . . 2

1.2 Gaining Knowledge . . . . . 4

1.3 Discovery and Outlook . . . . . 6

1.4 Contributions and Structure . . . . . 8

This thesis explores the roles of machine learning, optimisation, and human expertise for discovery in science and engineering, focusing on applications within chemical engineering. The work is dictated by the relationship between two fundamental paradigms prevalent throughout science and engineering: deductive reasoning, which manifests in physics-based models derived from first principles, and inductive reasoning, which involves learning from experiments or observations and underpins data-driven methods. Inspired by Oden [109], I examine knowledge acquisition and discovery through the lens of both deductive (theory-driven) and inductive (observation-driven) methods. This thesis highlights how the interaction, rather than opposition, between these approaches can drive discovery, particularly in a field as rich in experiments, physics and data as chemical engineering.

## 1.1 Context within Chemical Engineering

The tension between theory-driven understanding and data-driven exploration is prevalent across many modern chemical engineering settings. For example, consider the design, optimisation, and control of chemical reactors, where performance relies on the understanding of fluid dynamics, heat transfer, and mass transfer phenomena.

Traditionally, reactor design has heavily relied on deductive, physics-based approaches. First-principles models, such as the Navier-Stokes equations used in Computational Fluid Dynamics (CFD) simulations, enable engineers to simulate and predict reactor behaviour. These simulations offer insights into flow patterns, mixing, and temperature distributions. However, high-fidelity simulations can be computationally expensive, particularly when exploring large design spaces or complex geometries. Furthermore, validating these theoretical models requires experimental data, reintroducing inductive observation into the cycle.

At the same time, the ‘fourth industrial revolution’ has brought in large amounts of sensor data, high-throughput experimentation, and accurate simulations, fuelling the adoption of inductive, data-driven techniques. Machine learning algorithms, including probabilistic models

like Gaussian Processes, excel at capturing input-output relationships directly from experimental measurements or simulation outputs, even when underlying physical mechanisms are incompletely understood or are computationally intractable. These models are finding increasing importance in process modelling and optimisation, uncertainty quantification, and the identification of non-obvious correlations that may lead to discoveries.

Consider the development of a flow reactor as an example. Initial real-world experiments might provide observations on a baseline system. This empirical data can then be used to inform the validation or refinement of a deductive, physics-based computational model (e.g., CFD). Subsequently, this model facilitates extensive in silico experimentation (simulations), generating data used to train inductive, data-driven surrogate models. These surrogates might accelerate exploration, reveal unexpected flow dynamics, offer new physical insights, or be used to suggest novel reactor designs. Ultimately, a promising design identified through this process would be physically manufactured and experimentally validated, closing the loop with real-world observation.

This example, which will play out over the course of this thesis, demonstrates how discovery emerges from combining these approaches. This thesis specifically applies such integrated strategies to tackle optimisation and discovery problems in Chemical Engineering systems, with a strong emphasis on reactor design. I will demonstrate how multi-fidelity modelling bridges the gap between costly high-fidelity simulations and faster approximations for efficient design space exploration. Furthermore, I will highlight how data-driven optimisation techniques, particularly Bayesian optimisation, including both human and non-human-in-the-loop variants, can reveal novel reactor geometries with superior performance, subsequently validated via simulation and physical experiments. The developed methodologies are showcased in case studies involving the design of flow reactors enabled by additive manufacturing, the optimisation of bioprocess control strategies, and various computational benchmarks, illustrating the practical integration of deductive and inductive paradigms.

The subsequent discussion broadens the perspective, examining the fundamental act of gaining knowledge to provide a deeper context for the modelling and optimisation methods detailed in the Background Chapter 3.2.

## 1.2 Gaining Knowledge

Defining the 'act of gaining knowledge' provides a deeper fundamental framework for understanding different modelling and optimisation methodologies. Within process engineering, knowledge gain often involves learning about existing complex systems to improve their monitoring, control, or operation, or identifying new processes for enhanced efficiency or sustainability. Whether or not they concern chemical engineering specifically, these pursuits of understanding existing systems and discovering the new ones, appear throughout science and engineering.

Philosophically, these pursuits echo a long-standing dialogue. Plato, student of Socrates, emphasised knowledge derived from reason and ideal forms, suggesting fundamental truths could be combined deductively to yield larger insights [13]. This mindset underpins much of modern mathematics, theoretical physics, and mechanistic modelling like CFD. Conversely, Plato's student Aristotle championed empirical observation, arguing that knowledge stems from systematically studying the existing physical world, laying groundwork for inductive reasoning. The relationship between deductive and inductive thought continues to shape how people approach tasks in science and engineering [73].

A historical example of this can be found in the city of Prague. The astronomical clock, or *Orloj*, completed in 1410 by Mikuláš of Kadaň, was at the time a feat of engineering based on geocentric observations, accurately modelling the observed motions of the sun and moon (Figure 1.1) with the earth at the center. Legend suggests a later clockmaker involved in its maintenance was blinded to prevent him replicating it.



Fig. 1.1 Prague Astronomical Clock (Orloj). Image by Tom Savage, 2024.

Centuries later, Johannes Kepler, while living in Prague and interested in astronomical timekeeping [91], studied the extensive observational data collected by his mentor Tycho Brahe, alongside existing models potentially including the *Orloj* itself. The observational reasoning that led to the original clock, combined with deductive mathematical reasoning led Kepler to formulate his laws of planetary motion and elliptical orbits. This superseded the inductively-derived geocentric models such as the one that drove the original mechanism of the *Orloj*. Kepler's breakthrough demonstrates how deductive reasoning, fuelled, in effect by data-driven models, can challenge, refine, and ultimately transform understanding derived purely from induction. Innovation came about from the interaction between these paradigms, both driving and provoking the other towards progress.

This thesis similarly employs both perspectives, applying modern 'observational' tools like machine learning alongside 'mathematical' methods such as CFD simulations in a bid to drive discovery in chemical engineering.

## 1.3 Discovery and Outlook

This final section examines the nature of scientific discovery, the role of paradigms, human-machine collaboration, and the future outlook, particularly concerning the emerging area of AI for Science before highlighting the contributions of each chapter.

Thomas Kuhn, in *The Structure of Scientific Revolutions*, suggested that "Discovery commences with the awareness of an anomaly, i.e. with the recognition that nature has somehow violated the paradigm-induced expectations that govern normal science" [72]. Mapping this onto optimisation within a defined solution space $\mathcal{X}$ (the set of all possible designs or parameters), a discovery can be considered to occur when an identified solution is 'surprising'. This surprise, or anomaly, might manifest in the solution's structure itself (a novel design configuration within $\mathcal{X}$) or in its performance (an objective function value previously thought unattainable). Therefore, this thesis adopts the view that optimisation facilitates discovery when it yields solutions that significantly deviate from typical, known solutions or initial starting points, especially within sufficiently broad solution spaces. The objective then becomes using modelling and optimisation tools to induce such potentially valuable anomalies more frequently. This thesis aims not only to present discoveries achieved in this manner but also to introduce specific tools that support this objective.

The subsequent Background Chapter will detail various modelling and optimisation approaches, often framed within distinct paradigms that sometimes arose in response to perceived limitations of others[1]. For instance, the deductive, physics-based paradigm of CFD contrasts with the inductive, data-driven paradigm of machine learning. Similarly, the statistically principled, sample-efficient nature of Bayesian optimisation contrasts with the population-based, often data-intensive exploration of evolutionary algorithms. Kuhn noted that proponents of new paradigms often claim they can resolve crises faced by the old [72]. Max Planck offered a more pragmatic observation on paradigm shifts: "...a new scientific

[1]The term 'paradigm' gained widespread use following Kuhn's work [72], though Kuhn himself later noted its overuse [74].

truth does not triumph by convincing its opponents..., but rather because its opponents eventually die, and a new generation grows up that is familiar with it" [122].

However, machine learning and optimisation, used to formulate and solve complex problems, do not merely represent another competing domain-specific paradigm. Instead, they function more as an enabling methodology, analogous to how systems thinking is applied across diverse fields like biology, engineering, and social sciences. They provide a powerful toolkit that supports discovery within existing domains by integrating information from simulation, experiment, and theory.

This suggests that future shifts in scientific methodology may increasingly relate to the power and generality of such enabling techniques and the inherent 'discoverability' within a field, rather than solely domain-specific theoretical revolutions. This can already be observed: identical generative models find application in molecular design, protein folding, and image creation. The longevity and impact of these general methods will depend on their success in driving tangible discoveries within specific fields and their transferability across disciplines. The potential impact of 'AI for Science' as a unifying field, integrating data sources and reasoning paradigms, is therefore significant. As Donella Meadows advised regarding paradigms in *Thinking in Systems*: "There is yet one leverage point that is even higher than changing a paradigm. That is to keep oneself unattached in the arena of paradigms, to stay flexible, to realize that no paradigm is 'true', that every one, including the one that sweetly shapes your own worldview, is a tremendously limited understanding of an immense and amazing universe that is far beyond human comprehension.". Machine learning and optimisation, by their general nature and ability to identify anomalies irrespective of the underlying domain paradigm, align well with this perspective.

Consequently, the role of the human in discovery is evolving. Discussions about human-computer collaboration are not new; Michie and Johnston [99] highlighted the need for human creativity alongside computers back in 1984, citing the Apollo 13 intervention. As

Mindell [100] notes, control systems historically integrated people and machines long before the current AI era, arguing against viewing computers solely as logic machines detached from their engineering context. What is changing now is the use of ML and optimisation into domains like chemistry, where human intuition and expertise have traditionally dominated, creating new links between simulation, experiment, and physical knowledge. This prompts discussions about optimal human intervention and expert knowledge integration, leading to developments like human-in-the-loop methods [150], which will be explored in this thesis. Building upon this introductory section, the remainder of this thesis details the specific methodologies developed and applied to address the challenges of integrating deductive and inductive approaches for discovery in chemical engineering. The contributions are structured across the following chapters:

## 1.4 Contributions and Structure

**Chapter 2** establishes the prerequisites for this thesis. It discusses modelling paradigms, contrasting physics-based approaches like CFD with data-driven methods including parametric models (polynomials, neural networks) and non-parametric models GPs. It then covers various optimisation methodologies, from numerical first and second-order methods to zero-order techniques, with a focus on Bayesian optimisation. The chapter emphasises the links between modelling, optimisation, and scientific discovery, setting the theoretical stage for the methods developed and applied later.

**Chapter 3** introduces a framework for the **D**esign and **A**nalysis of **R**eactor and **T**ube **S**imulations (DARTS), applying Multi-fidelity Bayesian optimisation for the optimisation of simulated pulsed-flow helical-tube reactors. This chapter details the reactor parameterisation, the use of continuous axial and radial mesh fidelities, a cost-adjusted acquisition function for balancing information gain and simulation cost, and a novel stopping criterion to ensure

a high-fidelity solution is returned within budget. The framework successfully identifies an optimal geometry featuring a flow inversion, which is experimentally validated via 3D printing to confirm enhanced mixing performance.
This research is published in [140]: T. Savage, N. Basha, J. McDonough, O. Matar, E. A. del Rio Chanona, "Multi-fidelity data-driven design and analysis of reactor and tube simulations," Computers Chemical Engineering, vol. 179, p. 108410, 2023.

**Chapter 4** extends the reactor design problem by employing high-dimensional parameterisations for both the reactor cross-section and path, using polar GPs. The focus shifts towards *discovery*, aiming to induce beneficial flow structures (Dean vortices) under steady-state, low Reynolds number conditions without requiring pulsed flow. A composite objective function balancing plug flow performance and residence time distribution ideality is optimised using multi-fidelity Bayesian optimisation. Key design features leading to enhanced mixing are identified, resulting in novel reactor designs that demonstrate experimentally validated performance improvements of 60% over conventional coils.
This research is published in [141]: T. Savage, N. Basha, J. McDonough, J. Krassowski, O. Matar, E. A. del Rio Chanona, "Machine learning-assisted discovery of flow reactor designs," Nature Chemical Engineering, vol. 1, no. 8, pp. 522–531, 2024.

**Chapter 5** explores the integration of human expertise into the BO loop. A collaborative framework is presented where experts guide the optimisation by selecting from a set of diverse, high-utility candidate solutions generated via multi-objective batch BO. The chapter introduces an expert-augmented design of experiments strategy and benchmarks the approach using simulated expert behaviours (ranging from perfect expertise to adversarial) and real human participants on chemical engineering case studies (reactor design, bioprocess control). Results show that human collaboration can accelerate convergence compared to standard Bayesian optimisation.

This research is published in [139]: T. Savage E. A. del Rio Chanona, "Human-algorithm collaborative Bayesian optimization for engineering systems," Computers Chemical Engineering, vol. 189, p. 108810, 2024.

**Chapter 6** builds upon the proceeding chapter by addressing the need for full automation in applications like self-driving labs by replacing the human expert with a large-language model (LLM). The LLM makes discrete choices between candidate solutions based on structured historical data and solution features, aiming to mimic expert pattern recognition. Various LLMs and prompting strategies (one-shot, multi-agent, chain-of-thought) are tested. The framework demonstrates convergence improvements over standard Bayesian optimisation for structured problems and is proposed as a novel, leakage-proof benchmark for evaluating LLM reasoning in scientific discovery contexts.

**Chapter 7** concludes the thesis by summarising the key findings and contributions across all chapters. It reflects on the effectiveness of combining data-driven optimisation, simulation, and expert knowledge (human or artificial) for chemical engineering discovery and outlines potential directions for future research.

In conclusion, this chapter has established the central theme of integrating deductive and inductive approaches for discovery in chemical engineering. It has provided context through specific examples, historical parallels, and philosophical underpinnings related to knowledge gain and paradigms. It has positioned machine learning and optimisation as powerful enabling tools within this framework and touched upon the evolving human role. Throughout the remainder of this thesis, the described integration of methodologies, coupled with human expertise, will be applied to advance scientific discovery and innovation in chemical engineering and beyond.

# Chapter 2

# Background

## Contents


**2.1 Modelling . . . . . 12**

2.1.1 Physical Models . . . . . 13

2.1.2 Data-driven Models . . . . . 16

2.1.3 Modelling Summary . . . . . 33

**2.2 Optimisation . . . . . 33**

2.2.1 Numerical Optimisation . . . . . 34

2.2.2 Data-driven (Zero Order) Optimisation . . . . . 41

2.2.3 Bayesian Optimisation . . . . . 49

2.2.4 Other Optimisation Paradigms . . . . . 57

2.2.5 Optimisation Summary . . . . . 59

**2.3 Chapter Summary . . . . . 60**

This chapter provides a background for the technical content used throughout this thesis, demonstrating different fundamental approaches to modelling and optimisation and how they contribute to scientific discovery. I begin with physical models, with an emphasis on computational fluid dynamics, before outlining the complementary paradigm of data-driven models, both parametric and non-parametric. Following this, I provide a motivation and discussion regarding various optimisation methodologies, from first and second-order numerical methods to zero-order data-driven approaches, with an emphasis on understanding how each serves to support model understanding in different ways. Throughout, I focus on the practical implications and trade-offs of each methodology when applied to engineering applications, particularly in the context of discovery. Where appropriate, I reference historical developments that motivate the state-of-the-art, while maintaining emphasis on how these methods support scientific discovery extending the traditional duality of modelling and optimisation. The chapter concludes by examining extensions to these paradigms and their broader implications.

## 2.1 Modelling

Models are how scientists and engineers represent knowledge about the physical world. Depending on who is asked, models either make useful assumptions that approximate a deeper, nondeterministic, and inherently unknowable reality, or represent this underlying reality itself. Deductive and inductive reasoning primarily inform two broad categories of models: physics-based models, often derived from first principles, and data-driven models, inferred largely from observations. In this section I will outline relevant examples of both paradigms with a view towards scientific and engineering discovery.

### 2.1.1 Physical Models

Physical models can be defined as the composition of elementary operations or laws, each of which is known or assumed to be true. One such example is the simulation of fluid flows via the Navier-Stokes equations. Despite consisting of individual, discrete molecules, a guiding assumption for the simulation of fluids is that they are *continuous*. In other words, the fluid is treated as a continuous medium where macroscopic properties like density and velocity are well-defined, continuous functions of position, allowing the use of differential calculus, even at very small scales (down to the limit where the hypothesis applies) [9]. This is known as the *continuum hypothesis*, and is applied in the face of the clear intractability and lack of need to model the motion of an almost uncountable number of molecules. When modelling fluids the continuum hypothesis is assumed to be true in the majority of cases. Following from this, and making no further assumptions, the basic laws of mass conservation can be applied to a single control volume. In Cartesian coordinates[1], and assuming an incompressible fluid, where the density does not change with respect to time or otherwise[2], this can be written as follows

$$\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z} = \nabla \cdot \mathbf{u} = 0 \tag{2.1}$$

where $x$, $y$, and $z$ are Cartesian coordinates and $\mathbf{u}$ is the overall velocity vector. This equation states that the net flow of fluid into or out of any control volume within the fluid is zero. Following mass conservation is the conservation of *momentum*, which will remain constant in the absence of external forces. To do so all forces acting upon a control volume are considered, both those acting on the total volume such as gravity, and those acting on the surfaces such as fluid pressure and shear stresses. A derivation of these contributions is omitted for conciseness but can be found within Bird et al. [12].

[1] Note that Batchelor [9] provides a description for why any control volume can be equivalently represented in Cartesian coordinates, as the following rules are independent of coordinate system.

[2] This thesis will only deal with incompressible fluids.

The rate-of-change of momentum of a fluid particle, $\rho \frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t}$, where D is the material derivative, and the sum of these three components

$$\rho \frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t} = \rho \mathbf{g} - \nabla P + \mu \nabla^2 \mathbf{u}. \tag{2.2}$$

Expanding this derivative and dividing by $\rho$ provides

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = \mathbf{g} - \frac{1}{\rho}\nabla P + \nu \nabla^2 \mathbf{u}, \tag{2.3}$$

effectively stating that the joint convective and diffusive changes in momentum are equivalent to the forces acting on a fluid. When combined with the continuity equation, these form the Navier-Stokes equations for incompressible fluid flow. While the Navier-Stokes equations are physically realistic, due to nonlinearity, they are intractable to solve for velocity or pressure fields in the general case. Here, is a limitation of physical models is encountered. Despite being accurate and well motivated, they are not always *useful* [14]. In the case of the Navier-Stokes equations, to find solutions the equations must be discretised, and approximate discrete solutions found using numerical methods, giving rise to the field of Computational Fluid Dynamics (CFD).

The partial derivatives presented in Eqs. 2.1 and 2.3 are approximated on a computational mesh using finite difference methods (such as forward, backward, or central differencing), finite volume methods, or finite element methods. This process converts the partial differential equations into a large system of algebraic equations linking values at discrete points in space and time. For example, applying a simple Forward Time, Central Space (FTCS) finite difference scheme to the one-dimensional linear convection-diffusion equation ($\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}$) yields:

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} + c\frac{u_{i+1}^k - u_{i-1}^k}{2\Delta x} = \nu \frac{u_{i+1}^k - 2u_i^k + u_{i-1}^k}{(\Delta x)^2} \tag{2.4}$$

where $u_i^k$ represents the velocity at spatial point $i$ and time step $k$, $\Delta t$ is the time step size, $\Delta x$ is the spatial step size, $c$ is the constant convection velocity, and $\nu$ is the kinematic viscosity. This algebraic equation explicitly relates the unknown value at the next time step ($u_i^{k+1}$) to known values at the current time step ($k$). While this example is linear, discretising the full Navier-Stokes equations results in a coupled system of *nonlinear* algebraic equations due to terms like $(\mathbf{u} \cdot \nabla)\mathbf{u}$.

Due to the large number of variables (corresponding to the mesh size) and the nonlinearity, these systems are typically solved iteratively using specialised numerical algorithms, often involving linear approximations and adaptive time-stepping schemes. While the continuous equations were assumed to represent the truth, discretisation breaks the assumption of continuity. By relaxing the continuity requirement in favour of gaining tractability the validity of the model must be brought about in other ways. To ensure numerical accuracy, the solution method must be verified (e.g., through mesh independence studies demonstrating convergence). Subsequently, the physical realism of the model must be validated by performing real-world experiments and measuring representative quantities (such as output concentration in the case of a chemical reactor), comparing these converged simulation results against the experimental data.

The Navier-Stokes equations, CFD, and experimental mesh validation are an example of how both deductive reasoning in the form of physics based models *and* inductive reasoning in the form of experimental validation can coexist to result in both accurate and useful models, and in turn knowledge. In many scenarios, there are no laws or assumed truths, only experimental observations. In this case, inductive reasoning must be used giving rise to *data-driven* modelling.

### 2.1.2 Data-driven Models

In contrast to physics-based models, data-driven models rely on observational data to inform model structure and resulting predictions. This is either done by using the data to directly influence the models prediction (non-parametric models), or using data to inform the parameters of a model, which in turn influence the prediction (parametric models). Some models that would be considered non-parametric contain a small number of parameters such as lengthscales and noise levels, that are denoted as hyper-parameters. To aid in the motivation of parametric models, a one-dimensional example dataset will be used presented in Figure 2.1.

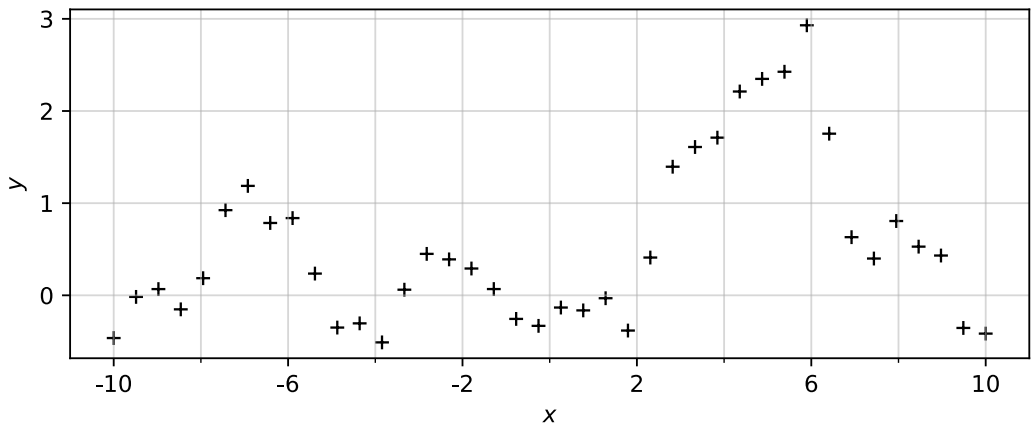



Fig. 2.1 One-dimensional example dataset used for the remainder of this section.

#### Parametric Models

For a parametric model $m(\boldsymbol{\theta}, \mathbf{x})$, where $m : \mathbb{R}^{\vartheta} \times \mathbb{R}^{d} \to \mathbb{R}$, $\boldsymbol{\theta} \in \mathbb{R}^{\vartheta}$, and $\mathbf{x} \in \mathbb{R}^{d}$ predictive capacity relies on the ability to identify the appropriate model parameters to fit a set of data. *Loss functions* quantify the performance of a model, and can depend on the model structure, parameters, and observational data. These three aspects represent the three unknowns when applying data-driven models [109].

For a given set of input data $\mathbf{X} \in \mathbb{R}^{n \times d}$, respective scalar-valued output data $\mathbf{y} \in \mathbb{R}^n$, and a fixed model structure $m$, common loss functions include the summation of the absolute differences between a prediction at each input data location and respective observed output, i.e.

$$\mathcal{L}_{\text{abs}}(\theta) = \sum_{i=0}^{n} |y_i - m(\theta, \mathbf{x}_i)|, \tag{2.5}$$

a quadratic loss, also known as the squared error, that more greatly penalises predictions that are far away from the given observations:

$$\mathcal{L}_{\text{quadratic}}(\theta) = \sum_{i=0}^{n} (y_i - m(\theta, \mathbf{x}_i))^2, \tag{2.6}$$

or a composite of the two, for example the Huber loss function, which maintains the numerical attractiveness of quadratic behaviour when observations are close to predictions, but remains robust to outliers by penalising large errors linearly [56].

$$\mathcal{L}_{\text{Huber}}(\theta) = \sum_{i=1}^{n} \begin{cases} \frac{1}{2}(y_i - m(\theta, \mathbf{x}_i))^2 & \text{if } |y_i - m(\theta, \mathbf{x}_i)| \leq \delta \\ \delta(|y_i - m(\theta, \mathbf{x}_i)| - \frac{1}{2}\delta) & \text{otherwise} \end{cases} \tag{2.7}$$

Figure 2.2 demonstrates the characteristics of each of these loss functions.

To identify the parameters which provide the most accurate model predictions, the loss function must be minimised using optimisation, in a process generally known as parameter estimation. Examples of basic parametric models include polynomial equations, for example in the one-dimensional case of the form

$$y = \theta_1 + \theta_2 x + \cdots + \theta_p x^{p-1} + \theta_{p+1} x^p \tag{2.8}$$

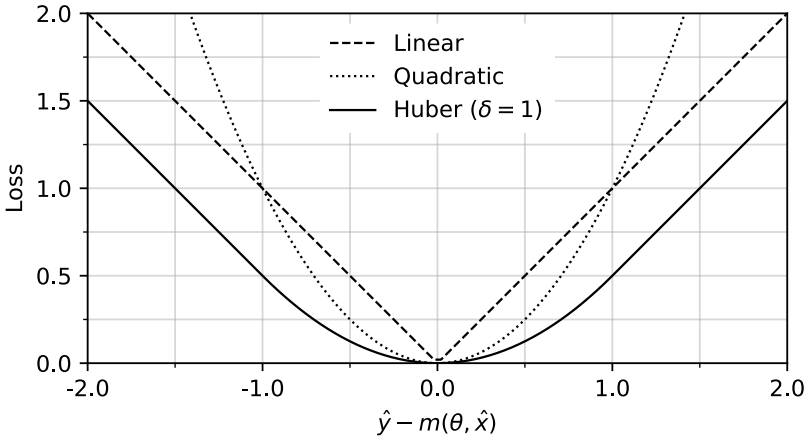



Fig. 2.2 The values of the linear, quadratic, and Huber loss functions with respect to the error between a prediction and an observation.

where each parameter $\theta$ is associated to a respective monomial of degree $p-1$. A polynomial model has the advantage of being arbitrarily complex, from convex (i.e. linear and quadratic) to nonconvex in the input space, but importantly linear in the parameter space.

Simply minimising loss does not lead to models that perform well in practice. As the complexity of parametric models such as the degree of a polynomial increases, the model will more closely fit observed data, potentially at the expense of poorly predicting unseen observations (known as the generalisability of a model), in a phenomena known as overfitting. Balancing the size, structure, and predictive capacity to avoid overfitting is one of the key challenges of statistical parametric modelling.

Figure 2.3 demonstrates how overfitting can occur when the size of a model increases, in this case the degree of a polynomial. A subset of the dataset has been used to more greatly demonstrate the impact of overfitting.

To overcome overfitting, observational data can be reserved from the loss function when performing parameter estimation, resulting in a test data set and training data set. The model structure that minimises the resulting loss with respect to this unseen data is then deemed most useful, known as cross-validation. Alternatively, criteria as the Akaike information

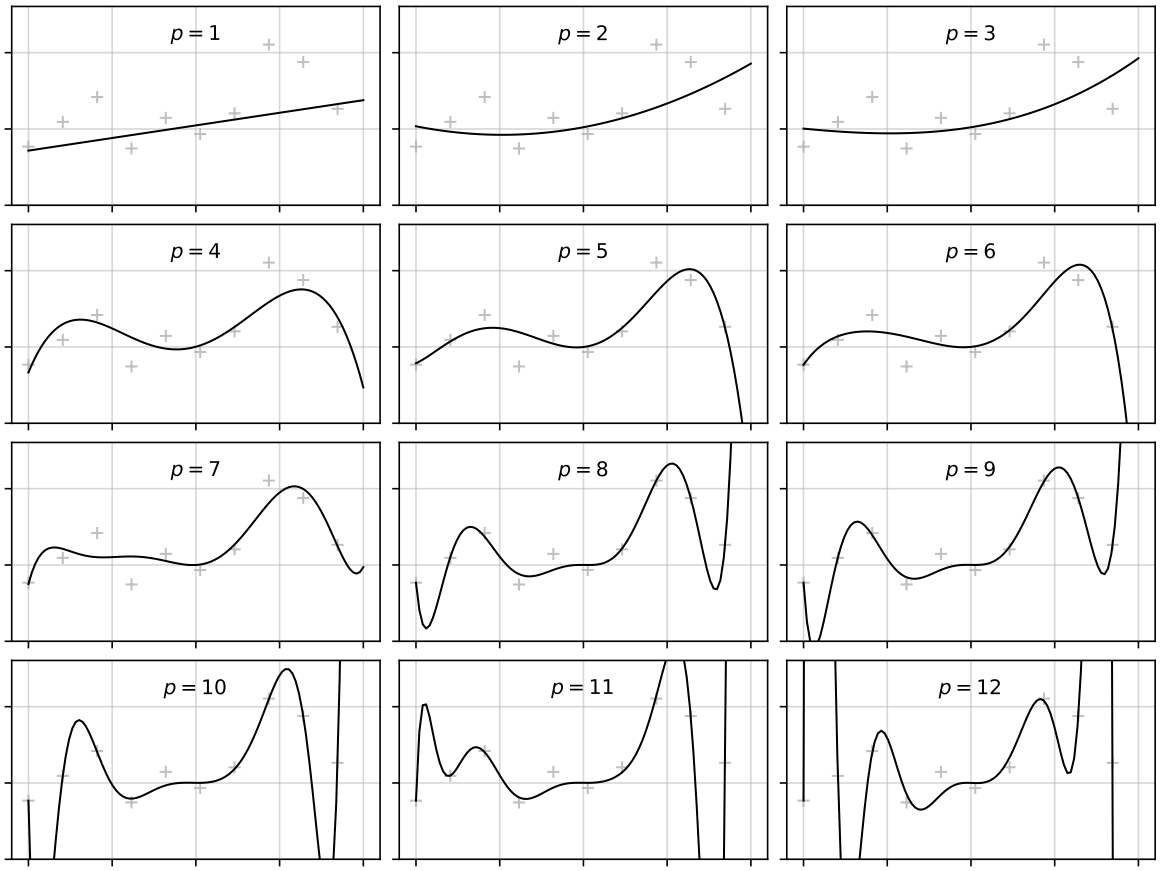



Fig. 2.3 Different degree one-dimensional polynomial models fit to the same dataset. As the degree increases accuracy may diminish due to overfitting. Here, $p$ represents the degree of the polynomial.

criteria (AIC) [3] or Bayesian information criteria (BIC) [143] can be used, that under some statistical assumptions balance the number of model parameters and predictive capacity.

Another approach to balance overfitting and predictive performance (also known as bias and variance) of a parametric model is through *regularisation*. A regularised loss function includes a penalisation term that limits the magnitude of the parameters themselves, typically a norm of a given degree. The weight of the magnitude of the parameters is varied, and the parameters with the best predictive performance on a test dataset is then used. Different norms result in parameters with different distributions. For example, the squared norm results in all parameters jointly taking on smaller values, whilst the absolute norm (sometimes called the Manhattan or taxicab norm) favours some parameters taking the value of 0, inducing

sparsity. Hastie et al. [52] provides a geometric intuition for why different regularisation norms result in different distributions of parameters, and how the absolute norm can be used for feature selection.

Neural networks offer another class of parametric data-driven models. From a statistical standpoint, their hyperparameters (e.g., number of hidden layers, batch size) can be tuned using cross-validation, and overfitting can be managed with techniques such as early stopping during parameter optimisation.

A standard multi-layer feedforward network implements a function $m(\theta, \mathbf{x})$ that maps an input vector $\mathbf{x}$ to an output through a sequence of layered operations:

$$m(\theta, \mathbf{x}) = g \circ (\mathbf{W}^{(L+1)} \cdot + \mathbf{b}^{(L+1)}) \circ f \circ \cdots \circ f \circ (\mathbf{W}^{(1)} \cdot + \mathbf{b}^{(1)})(\mathbf{x}) \tag{2.9}$$

Here, $\theta$ represents the complete set of learnable parameters, encompassing all weight matrices $\mathbf{W}^{(l)}$ and bias vectors $\mathbf{b}^{(l)}$ for the network's $L$ hidden layers and output layer ($l = 1, \ldots, L+1$). Thus, the parameters subject to optimisation are $\theta = \{\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \ldots, \mathbf{W}^{(L+1)}, \mathbf{b}^{(L+1)}\}$. Each transformation involves a linear operation defined by $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$, followed by an element-wise non-linear activation function $f$ in the hidden layers (e.g., ReLU), or a task-specific output activation function $g$ in the final layer (e.g., identity for regression, softmax for classification). The symbol '$\circ$' denotes this function composition, as shown in Equation (2.9). Theoretically, under specific conditions, even networks with a single hidden layer can act as universal function approximators [54].

The flexibility of neural networks allows them to approximate complex functions. Under certain conditions, particularly with large datasets and specific architectures, they can exhibit smooth behaviour which contributes to their generalisation capabilities, although overfitting remains a challenge, especially in low-data regimes. The benefits in being able to generalise arbitrary functions induces the double edged-sword of overfitting, particularly in low-data

regimes such as the modelling of expensive simulations. Figure 2.4 demonstrates how neural networks can produce generalised smooth functions, though in low data regimes may still overfit. A single layer neural network with ReLU activation function ($f(x) = \max(x, 0)$) is trained to completion with an increasing number of neurons, only considering training loss.

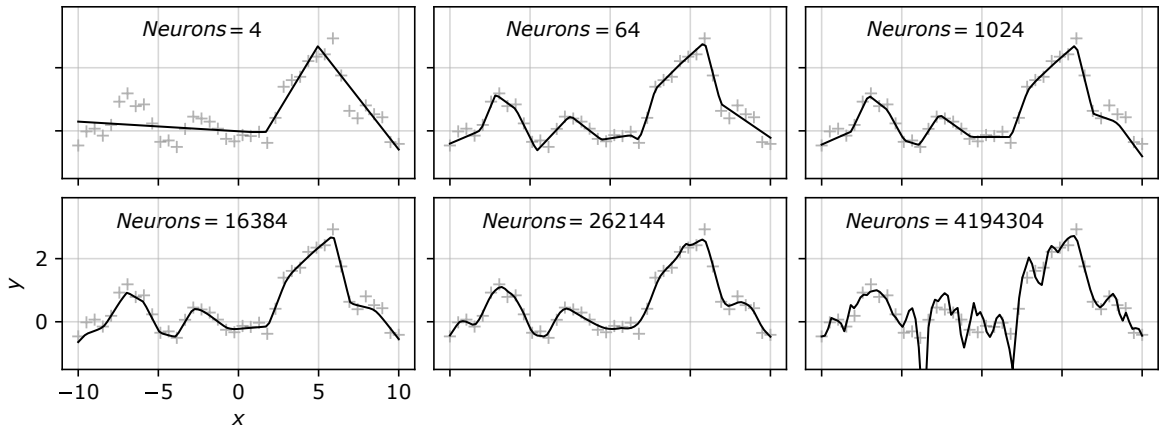



Fig. 2.4 Increasing the neurons within a single layer neural network generally results in more smooth functions. Overfitting may still occur in low-data regimes, but with large amounts of data, neural networks tend to produce smoother, more generalisable functions. This behavior contrasts with traditional parametric models that may overfit as complexity increases.

Common approaches to training neural networks while mitigating against overfitting include regularising parameters or mathematical equivalents such as *drop-out* [155].

In recent years, the statistical view of neural networks as function approximators has diminished in favour of an alternative perspective. Empirical observations such as double-descent [142], and grokking [125], have demonstrated that increasing scale, or training time results in neural networks behaving more akin to kernel-smoothing functions, despite being parametric. In large-data scenarios, instead of overfitting like a polynomial with increasing degree, increasing the neurons results in a smoother function at the expense of computational resource of training for longer.

Modern neural networks have therefore shifted structurally towards capturing complex patterns in large-data scenarios as opposed to adhering strictly to traditional statistical paradigms that have found use in interpretable low-data scenarios. This shift in perspective

has led to several developments in the field of machine learning including transformer architectures [167], graph-neural networks [16], and large-language models [163]. In light of this, the universal function approximation theorem in the case of neural networks has been considered ultimately distracting, as it indicates an emphasis on scaling existing basic architectures as opposed to alternative topologies, such as transformers.

In the context of this thesis it is important to understand the downsides of applying neural networks, in order to motivate where and when they should be applied in the context of scientific discovery. Neural networks require large amounts of high-quality data and compute to reach the smoothing-regime (or grokking). Within engineering, observational data is often scarce, and nowhere near as available than text, audio, or video datasets common in machine learning tasks. In low data regimes, non-parametric models are more widely used. Intuitively, in non-parametric models, the information contained within observational data directly influences predictions without passing through model parameters, analogous to minimising the number of times energy is converted from one form to another in order to reduce thermodynamic inefficiencies.

### Non-parametric Models

Non-parametric models directly use observational data to influence the output of a model. The most intuitive non-parametric model would be to simply return the value of the closest observation within the dataset, in a Euclidean sense. Alternatively, the average of the closest $k$-observations could be taken. The type of model is called a $k$-nearest neighbours model, where $k \in \mathbb{Z}$ represents a hyper-parameter that may be decided using cross-validation or otherwise. The $k$-nearest neighbours model is as follows:

$$m(\hat{\mathbf{x}}, \mathbf{X}, \mathbf{y}) = \frac{1}{k} \sum_{i=1}^{k} y_{(i)} \tag{2.10}$$

where $y_{(i)}$ are the output values corresponding to the $k$ nearest neighbours of $\hat{\mathbf{x}}$ in $\mathbf{X}$.

Figure 2.5 demonstrates this form of non-parametric model with differing values of $k$.

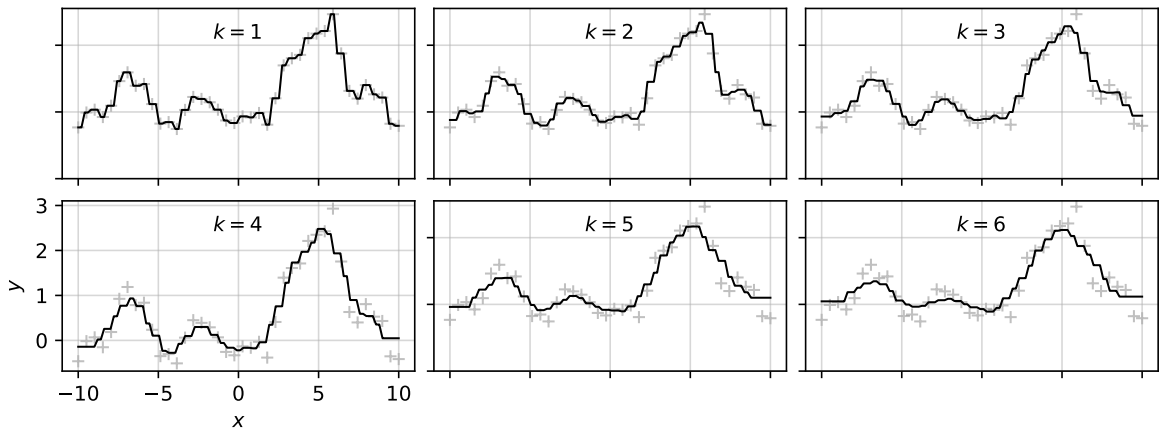



Fig. 2.5 Equation 2.10 used to model the one-dimensional example data. Here, $k$ refers to the number of observations used to define a local model.

While this model exactly interpolates the data in the case of $k = 1$, it does not capture the true form of the underlying function well, and the resulting function is discontinuous. Increasing the neighbourhood value results in smoother-like functions, however they remain discontinuous. To mitigate against the inaccuracy of this discontinuity, *all* data points could be used to influence a prediction, weighted according to their Euclidean distance. By replacing the integer $k$ with a a weighting, or covariance function, the resulting model is continuous.

A common covariance function, also known as a *kernel*, is the Gaussian function, indicating that the importance of an observation diminishes exponentially the further it is from the desired input. The Gaussian kernel is as follows (note that $k$ is now overloaded, representing both the number of nearest neighbours as well as the kernel function to maintain consistency with the literature[3]):

$$k(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{2\sigma^2}\right) \tag{2.11}$$

[3]It will later also represent the current iteration.

where $\sigma$ is a hyperparameter representing the standard deviation of the weighting Gaussian. The resulting function, also known as a Nadaraya-Watson estimator [103, 172], is therefore

$$m(\hat{\mathbf{x}}, \mathbf{X}, \mathbf{y}) = \frac{\sum_{i=1}^{n} k(\hat{\mathbf{x}}, \mathbf{x}_i) y_i}{\sum_{i=1}^{n} k(\hat{\mathbf{x}}, \mathbf{x}_i)} \tag{2.12}$$

where the contribution of each observation is weighted proportional to the kernel function at that observations location. Figure 2.6 demonstrates the value of the kernel function for different distances between inputs, note that it is always positive.

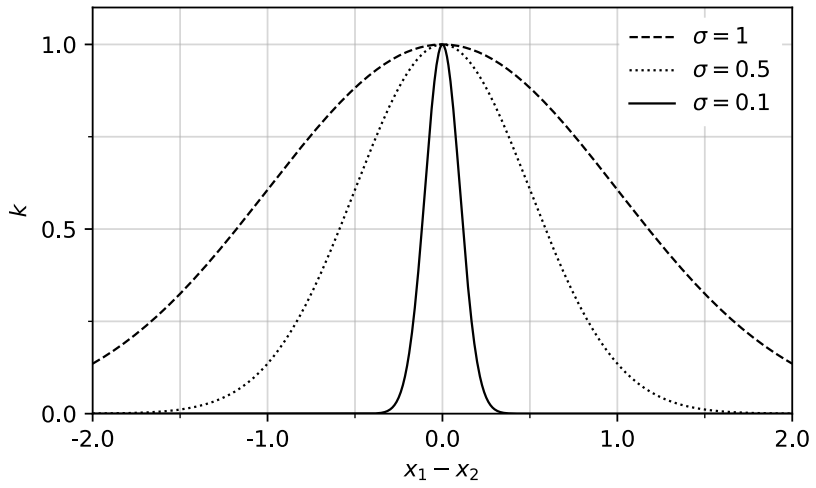



Fig. 2.6 Different values of $\sigma$ result in stronger or weaker correlations with points at a fixed distance.

Analogously to the number of nearest neighbours, the value of $\sigma$ can be varied to produce smooth functions of different lengthscales (i.e. contributions from observations further away are weighted more strongly or weakly). Figure 2.7 demonstrates how larger values of $\sigma$ result in smoother functions, as observations further away contribute more towards the model output.

Nadaraya-Watson estimators provide smooth data-driven models, with hyper-parameters $\sigma$ that may be tuned to provide optimal predictive performance. However, in low-data scenarios it becomes desirable to use all the data available, limiting the utility of splitting data into training and test sets. Therefore, it becomes appropriate to model functions statistically,

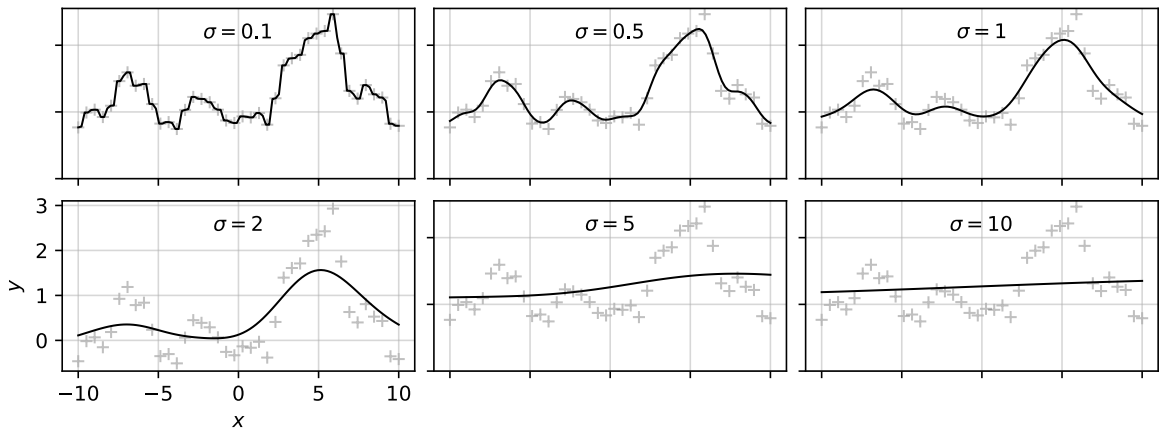



Fig. 2.7 Different values of $\sigma$ produce more, or less smooth approximations to an underlying function.

enabling the ability to select the *most likely* model analogous to information criteria for parametric models. This statistical viewpoint will motivate *Gaussian process* models.

To begin with it becomes important to motivate the notion of a distribution *over functions*. To do so, the notion of sampling from a finite dimensional multi-variate Gaussian distribution with identity covariance matrix is introduced, i.e.

$$\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \tag{2.13}$$

Figure 2.8 demonstrates a single sample from a 20-dimensional multivariate Gaussian distribution, displaying the sampled value for each index, or dimension, of the multivariate distribution.

Each value is independent from the others as the covariance matrix contains no off-diagonal terms. By extension, non-diagonal terms can be introduced to the covariance matrix to indicate that neighbouring indices should be correlated. The value of these terms are determined by a kernel function as previously presented, resulting in a symmetric covariance matrix where indices further away are less correlated. Figure 2.9 demonstrates a new sample from this multivariate Gaussian.

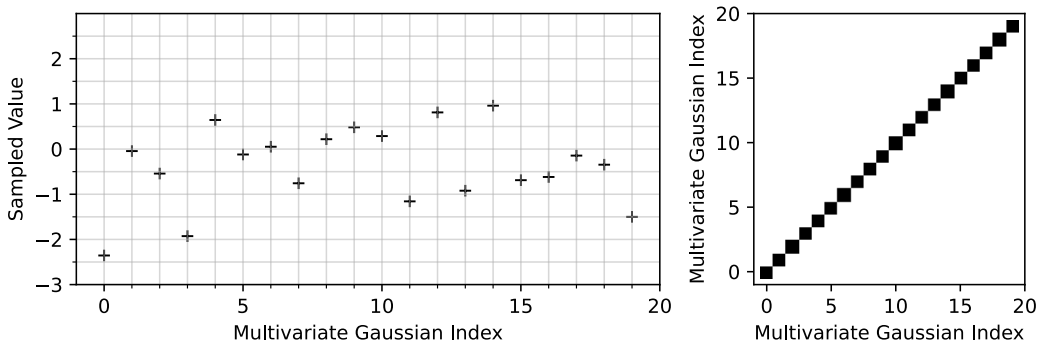



Fig. 2.8 The value of each dimension of a single sample from a 20 dimensional multivariate Gaussian distribution with identity covariance matrix.

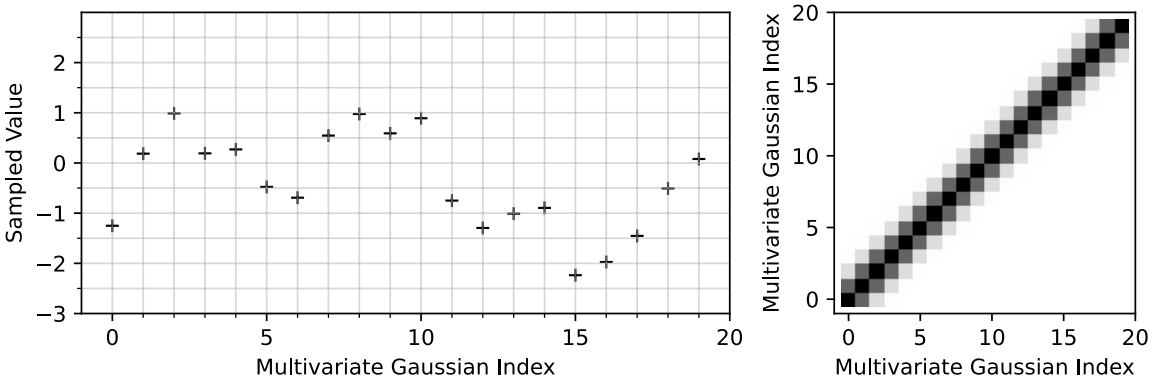



Fig. 2.9 The value of each dimension of a single sample from a 20 dimensional multivariate Gaussian distribution with covariance matrix as defined by a kernel function.

Neighbouring values have closer values, as the Gaussian distribution that is defined by the covariance matrix indicates that 'closer' values are more likely to have similar values. This behaviour is made more clear by increasing the value of $\sigma$ further still. Figure 2.10 demonstrates a sample from a multivariate Gaussian with covariance matrix indicating strong correlation between indices further away.

At this point the sample in Figure 2.10 looks strikingly similar to a continuous function. Joining the values of the indices and sampling a number of times makes this clear, as shown in Figure 2.11. The key intuition that will lead to a Gaussian distribution over functions is that functions are equivalent to infinite-dimensional vectors. By making the extension from sampling from a finite dimensional multivariate Gaussian distribution with mean vector and

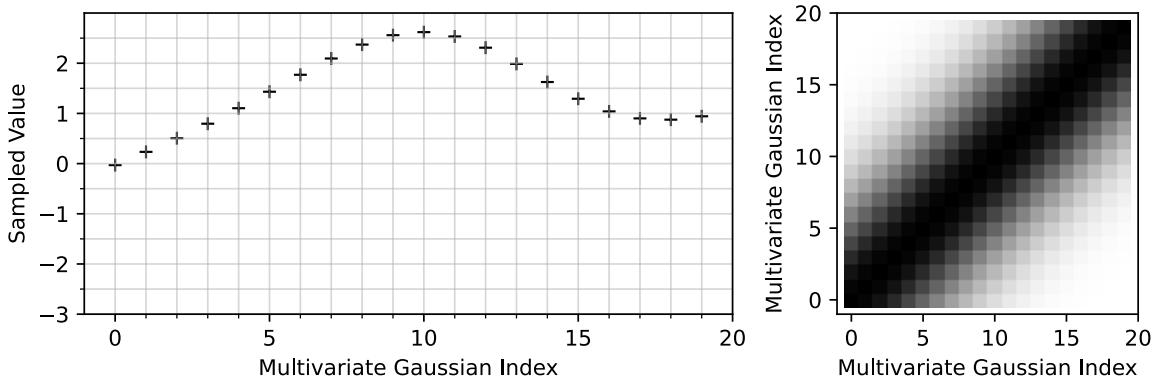



Fig. 2.10 The value of each dimension of a single sample from a 20 dimensional multivariate Gaussian distribution with covariance matrix as defined by a kernel function.

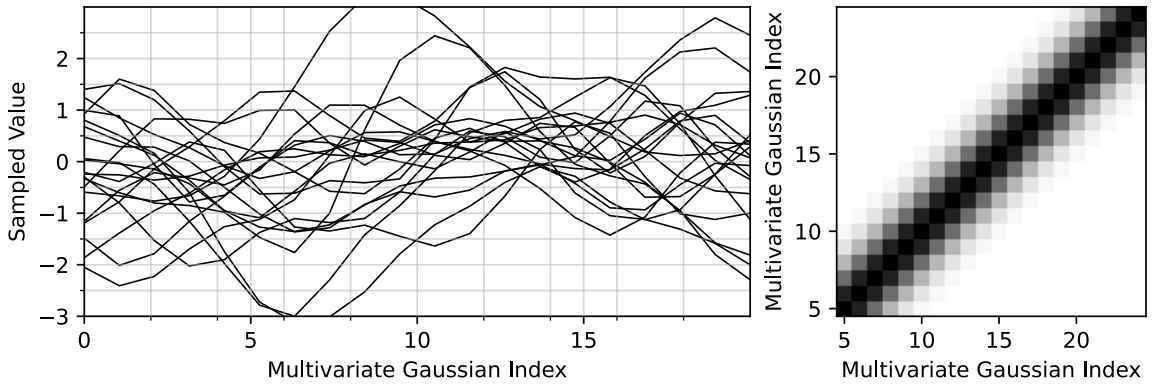



Fig. 2.11 The value of each dimension of a single sample from a 20 dimensional multivariate Gaussian distribution with covariance matrix as defined by a kernel function.

covariance matrix, to a Gaussian distribution with mean function and covariance function, continuous functions can be sampled. This infinite-dimensional Gaussian distribution is also called a *Gaussian process*, and acts as a distribution over functions. Equation 2.14 demonstrates this distribution

$$f \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{2.14}$$

where $m$ is the mean function, and $k$ is the covariance function between two input points $\mathbf{x}$ and $\mathbf{x}'$. In practice, there is no way of truly sampling a continuous function from a Gaussian process, only at finite 'locations'. In spite of this, the true continuous underlying nature of

Gaussian processes can be illustrated by sampling at a large enough number of locations. Figure 2.12 demonstrates an approximation of samples from a GP. The covariance matrix of a multivariate Gaussian distribution is defined by the kernel function between each of the input locations on the $x$ axis, and in this case the mean function is zero.

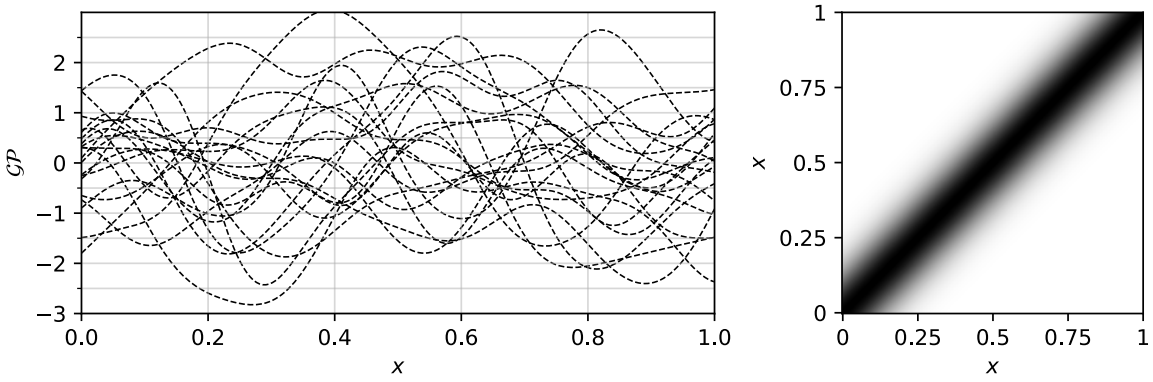



Fig. 2.12 Functional samples from a Gaussian process and the continuous covariance matrix that results from a kernel function.

On their own, unconditioned Gaussian processes are not entirely useful. However, despite being infinite-dimensional, the same mathematical features of Gaussian distributions still hold, notably the ability to condition them. Conditioning a Gaussian distribution, by specifying a value in one dimension, always results in a lower-dimensional Gaussian distribution. Consequently, conditioning a Gaussian process, by specifying known function values (observations) $\mathbf{y}$ at certain locations $\mathbf{X}$, yields a valid Gaussian process representing the distribution of the unknown function values $\mathbf{f}$ at other locations of interest $\mathbf{X}'$. This conditioning is analogous to the two-dimensional case shown in Figure 2.13. The joint multivariate Gaussian distribution over the unknown values $\mathbf{f}$ and the observed values $\mathbf{y}$ forms the basis for this conditioning. Let $\mu$ be the prior mean vector evaluated at the locations $\mathbf{X}'$ and $\mu_{\mathbf{y}}$ be the prior mean vector evaluated at the observation locations $\mathbf{X}$. Furthermore, let $\mathbf{K_{AB}}$ denote the covariance matrix computed using the kernel function between location sets $\mathbf{A}$ and $\mathbf{B}$. The joint distribution is

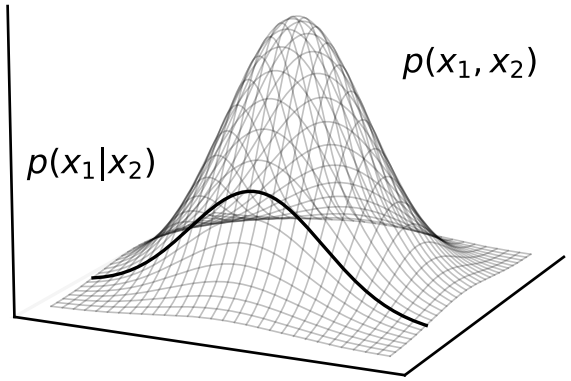



Fig. 2.13 A conditioned two-dimensional (joint) Gaussian distribution. The resulting distribution is always Gaussian.

then given by:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left( \begin{bmatrix} \mu \\ \mu_{\mathbf{y}} \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{\mathbf{X}'\mathbf{X}'} & \mathbf{K}_{\mathbf{X}'\mathbf{X}} \\ \mathbf{K}_{\mathbf{X}\mathbf{X}'} & \mathbf{K}_{\mathbf{X}\mathbf{X}} \end{bmatrix} \right) \tag{2.15}$$

Here, $\mathbf{K}_{\mathbf{X}'\mathbf{X}'}$ is the prior covariance matrix evaluated between all pairs of points in $\mathbf{X}'$, $\mathbf{K}_{\mathbf{X}\mathbf{X}}$ is the prior covariance matrix evaluated between all pairs of points in $\mathbf{X}$, and $\mathbf{K}_{\mathbf{X}'\mathbf{X}}$ (with its transpose $\mathbf{K}_{\mathbf{X}\mathbf{X}'}$) represents the prior cross-covariance between points in $\mathbf{X}'$ and $\mathbf{X}$. Using the standard rules for conditioning multivariate Gaussians, the resulting posterior distribution for $\mathbf{f}$ given the observations $\mathbf{y}$, denoted $p(\mathbf{f}|\mathbf{y})$, remains Gaussian with the following mean and

variance:

$$\mathbb{E}[\mathbf{f}|\mathbf{y}] = \mu + \mathbf{K}_{\mathbf{X}'\mathbf{X}}\mathbf{K}_{\mathbf{X}\mathbf{X}}^{-1}(\mathbf{y} - \mu_{\mathbf{y}}) \tag{2.16}$$

$$\text{Var}[\mathbf{f}|\mathbf{y}] = \mathbf{K}_{\mathbf{X}'\mathbf{X}'} - \mathbf{K}_{\mathbf{X}'\mathbf{X}}\mathbf{K}_{\mathbf{X}\mathbf{X}}^{-1}\mathbf{K}_{\mathbf{X}\mathbf{X}'} \tag{2.17}$$

As a result of conditioning on observational data, a finite dimensional Gaussian distribution can be obtained at the specified locations $\mathbf{X}'$, enabling the estimation of the uncertainty in $\mathbf{f}$, an important property of Gaussian process models. The ability to analytically calculate the posterior via Bayes' rule, makes Gaussian processes of particular importance with respect to other statistical processes. As an extension, it can be assumed that the mean function is zero, and that the value of a Gaussian process posterior is to be inferred at a single location $\mathbf{x}'$. The resulting mean value becomes

$$\mathbb{E}[\mathbf{f}(\mathbf{x}')|\mathbf{y}] = \mathbf{k}_{\mathbf{x}'\mathbf{X}}\mathbf{K}_{\mathbf{X}\mathbf{X}}^{-1}\mathbf{y} \tag{2.18}$$

where the term $\mathbf{k}_{\mathbf{x}'\mathbf{X}}\mathbf{K}_{\mathbf{X}\mathbf{X}}^{-1}$ can be interpreted as a weight vector with respect to each observational datapoint, where the weight of each observed value is proportional to how 'similar' $\mathbf{x}'$ is to the corresponding observed input (with respect to covariance), and inversely proportional to the similarity between observed inputs, therefore balancing the influence of nearby points with the overall structure of the data.

The GP posterior mean $\mathbb{E}[\mathbf{f}(\mathbf{x}')|\mathbf{y}] = \mathbf{k}_{\mathbf{x}'\mathbf{X}}\mathbf{K}_{\mathbf{X}\mathbf{X}}^{-1}\mathbf{y}$ acts as a linear smoother, where the effective weights on observations $\mathbf{y}$ depend on the covariance structure of the data via $\mathbf{K}_{\mathbf{X}\mathbf{X}}^{-1}$. While conceptually related to kernel smoothers like the Nadaraya-Watson estimator (both use kernels to weight nearby points), the GP framework provides these weights through Bayesian inference, offering advantages like principled uncertainty quantification.

Figure 2.14 demonstrates samples from a posterior Gaussian process conditioned on observational data, alongside the resulting covariance matrix of this posterior.

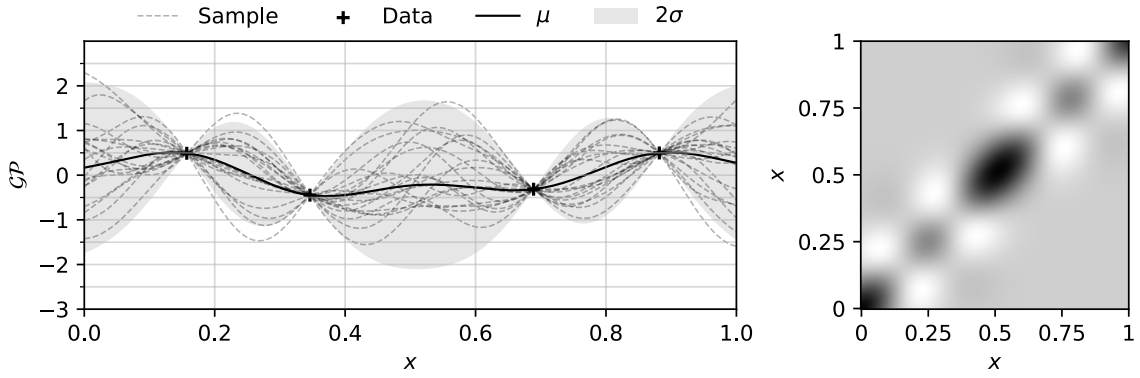



Fig. 2.14 Samples from a conditioned Gaussian process and resulting covariance matrix, the diagonal of which corresponds to the variance at each location.

To model noisy observational data, a constant value $\sigma_n$ is added to the covariance matrix, resulting in the term $\mathbf{K_{XX}}$ becoming $\mathbf{K_{XX}} + \sigma_n \mathbf{I}$. This can be added either inside or outside the covariance function and makes the extension to modelling noisy-observational data natural. Whilst Gaussian processes provide efficient ways of modelling small amounts of data with or without noise, they contain a number of important hyperparameters. Values such as length scale, magnitude, and noise level within the covariance, as well as any mean function all need to be decided. Some are considered prior-knowledge, such as the form of the covariance function. This enables the exploitation of prior information such as the knowledge that the underlying function is periodic[4]. As a Gaussian process is a statistical object, the tools of probability can be exploited to infer the optimal parameters. The likelihood of observing a set of data in a Gaussian process model is given by the marginal likelihood, which integrates over all possible latent functions consistent with the GP prior. In the case of a finite set of observational data, this integral has the following analytical form

$$p(\mathbf{y}|\mathbf{X},\theta) = \frac{1}{(2\pi)^{n/2}|\mathbf{K_{XX}}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{y}-\mu)^T \mathbf{K_{XX}^{-1}}(\mathbf{y}-\mu)\right) \tag{2.19}$$

[4]A true Bayesian would argue that to do so would be to include potentially false inductive knowledge within the paradigm, and that the form of the kernel function itself be considered uncertain. Though most proceed regardless. In a practical sense, this is an example of how gaining knowledge becomes an exercise in knowing where and when to exploit different forms of reasoning to reach a useful conclusion, as opposed to using one or the other, as was described in the discussion of the Navier-Stokes equations earlier.

where $|\cdot|$ is the determinant. The likelihood can be maximised with respect to hyperparameters to produce a Gaussian process that is most likely to have produced the observational data, in spite of interpolation in the case of no observational noise. This ability ensures that Gaussian processes rarely overfit to a dataset. Due to the exponential and multiplicative terms, which may induce in numerical instabilities and ill-conditioning, often the log-version is used. Additionally, the equation is made negative as optimisers generally deal with minimisation problems. The negative log-likelihood is as follows:

$$-\log p(\mathbf{y}|\mathbf{X},\theta) = \frac{1}{2}\log|\mathbf{K_{XX}}| + \frac{1}{2}(\mathbf{y}-\mu)^T\mathbf{K_{XX}^{-1}}(\mathbf{y}-\mu) + \frac{n}{2}\log(2\pi). \tag{2.20}$$

The negative log-likelihood can be optimised through zero, first, or second order optimisation as the gradient can be calculated analytically (in the case of the radial basis kernel, others may only have first order derivatives available).

Though this thesis is concerned with low-data scenarios, it is important to highlight a number of disadvantages of Gaussian process. The computational complexity of evaluating the negative log-likelihood, and performing inference is $\mathcal{O}(n^3)$ with respect to the number of data points due to the inversion of the covariance matrix $\mathbf{K_{XX}}$. Similarly, the memory complexity is $\mathcal{O}(n^2)$ due to the matrices involved. A number of approximations including low-rank approximations of the covariance matrix [174], inducing-point based latent variables [160] or pseudo-inputs [151], and neural kernels [78] have all been proposed. Quiñonero-Candela and Rasmussen [127] presents an early unifying view of these approximations. Another feature of the fact that a Gaussian process is a distribution, is that different 'types' of outputs from the same input space can be conditioned jointly. Common examples include jointly conditioning function value and gradient observations, observations from multiple fidelities, or from functions or experiments with more than one output [42]. The cross-correlation between multiple classes of observation provides a way of relating the trends across outputs.

### 2.1.3 Modelling Summary

Modelling is how the knowledge about an underlying system is represented. Models can either be deduced through physical laws which are assumed to be true, or inferred through observed data from the system. The Navier-Stokes equations are an example of a physics-based model that is only made useful in the presence of observational data, used to validate the model when assumptions such as discontinuity are made to make the equations tractable to solve. When only data is available, parametric or non-parametric models can be used where model outputs are influenced through a set of learned parameters or by directly informing model outputs respectively. Examples of parametric models include polynomials or neural networks, while non-parametric approaches like k-nearest neighbours and Gaussian processes directly use observational data to make predictions. By motivating a smooth approximation to a k-nearest neighbour model, the Nadaraya–Watson estimator can be derived, which have conceptual links to Gaussian processes. Gaussian processes are particularly valuable in low-data scenarios, providing uncertainty estimates and avoiding overfitting through a principled statistical approach to hyperparameter selection. Each modelling paradigm has strengths and limitations, highlighting how both deductive and inductive reasoning each play an often complimentary role in scientific and engineering discovery.

## 2.2 Optimisation

Optimisation concerns locating the minima or maxima of a given function or model. For the purposes of this section, as is convention, optimisation will refer to minimisation. Whether this minima should be a local or global optima is down to the problem setting, and the end goal. A local minima can be defined as a solution $\mathbf{x}^*$ where

$$f(\mathbf{x}^*) < f(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{N} \tag{2.21}$$

where $\mathcal{N}$ is a neighbourhood around $\mathbf{x}^*$ [107]. A global minima can similarly be defined as a solution that has the lowest value throughout the entire domain $\mathcal{X}$:

$$f(\mathbf{x}^*) < f(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{X}. \tag{2.22}$$

There are a large number of numerical approaches to optimise functions, each of which is highly dependent on the function evaluation cost, non-linearity, convexity, and continuity. These can be broadly categorised as zero-order, first-order and second-order approaches, with respect to the approximation that they use to inform the solutions that are evaluated, i.e. function value, gradient, or Hessian based[5].

### 2.2.1 Numerical Optimisation

Numerical optimisation describes iterative methods to locate local or global minima of a given function. Figure 2.15 demonstrates a two-dimensional test function that will be used to illustrate methods for numerical optimisation. This function is sampled from a two-dimensional Gaussian process prior.

Numerical optimisation can be used to both find the optimal parameters of parametric models such as neural networks, or optimise models themselves. Different scenarios such as dimensionality and convexity will dictate the most appropriate method.

#### First Order Optimisation

First order methods rely on function values and their gradients. The most basic first order algorithm is gradient descent. At a given iteration $t$, and with knowledge of function value

[5]Note that whilst third-order methods and higher are feasible, the cost of evaluating higher-order derivatives outweighs the improved approximation, and analytical solutions become difficult or impossible to locate.

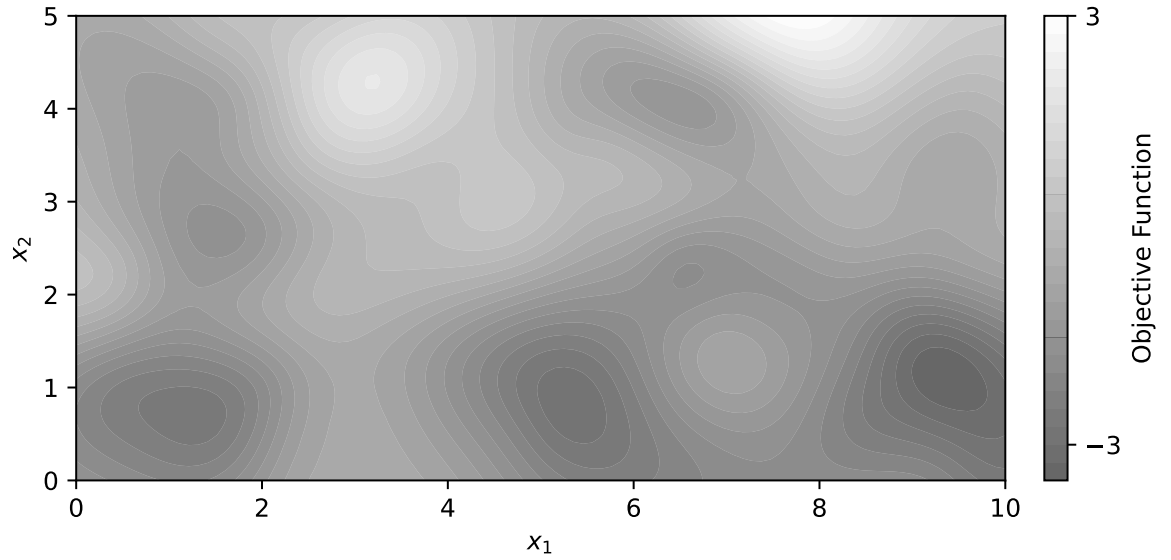



Fig. 2.15 Two-dimensional test function used to illustrate approaches to numerical optimisation.

$f(\mathbf{x}_t)$ and gradient $\nabla f(\mathbf{x}_t)$, steps are performed iteratively in the direction of the gradient

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \alpha \nabla f(\mathbf{x}_t) \tag{2.23}$$

where $\alpha \in \mathbb{R}$ is a step-size parameter that controls how far a solution is moved 'down' the gradient of the function (i.e. the magnitude of the gradient vector that is added to the solution vector). If the step-size is too small, the algorithm will slow to converge to a minima. Alternatively, if the step-size is too large the algorithm will *overshoot* minima, and enter an oscillatory regime. Methods to select appropriate step sizes include solving a one-dimensional line search problem [176], using a fixed regime such as decreasing, repeating, sinusoidal campaigns [115, 48], or adaptive approaches [6]. A key trade-off across all first-order methods is balancing the computational effort spent on identifying an 'optimal' step-size with the frequency of solution updates and subsequent gradient evaluations.

An approach for mitigating against the importance of selecting an appropriate step size and accelerating convergence in certain scenarios involves incorporating information from previous updates [157]. This helps the optimisation process gain 'inertia' in a consistent direction and dampen oscillations. In doing so, the iterations can become more robust against noisy or sparse gradients and potentially navigate ravines in the loss landscape more effectively.

One of the foundational methods is the *Heavy Ball method* introduced by Polyak [123]. It incorporates a term based on the previous update step, making the iterates behave somewhat like a heavy ball rolling down a hill. The update rule can be written as:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \alpha \nabla f(\mathbf{x}_t) + \beta(\mathbf{x}_t - \mathbf{x}_{t-1}) \tag{2.24}$$

Here, $\alpha$ is the learning rate, and $\beta \in [0,1)$ is the momentum coefficient, controlling the contribution of the previous step $(\mathbf{x}_t - \mathbf{x}_{t-1})$. Alternatively, using a momentum vector $\mathbf{m}_t = \mathbf{x}_t - \mathbf{x}_{t-1}$ representing the previous step, the update you originally presented can be derived:

$$\mathbf{m}_{t+1} = \beta \mathbf{m}_t - \alpha \nabla f(\mathbf{x}_t) \tag{2.25}$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{m}_{t+1} \tag{2.26}$$

where $\mathbf{m}_t$ now accumulates scaled gradients and momentum. Note the direct link between $\mathbf{m}_{t+1}$ and the step $\mathbf{x}_{t+1} - \mathbf{x}_t$.

However, a more common formulation for momentum in modern machine learning contexts, often referred to simply as 'SGD with Momentum', maintains an exponentially decaying moving average of *past gradients*. This momentum term represents a 'velocity' that accumulates in directions of persistent gradient descent. The update rules are typically expressed

as:

$$\mathbf{v}_{t+1} = \beta \mathbf{v}_t + \alpha \nabla f(\mathbf{x}_t) \tag{2.27}$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \mathbf{v}_{t+1} \tag{2.28}$$

or sometimes with the learning rate factored differently:

$$\mathbf{m}_{t+1} = \beta \mathbf{m}_t + (1-\beta) \nabla f(\mathbf{x}_t) \tag{2.29}$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \alpha \mathbf{m}_{t+1} \tag{2.30}$$

In these formulations (Equations 2.27-2.28 or 2.29-2.30), $\mathbf{v}_t$ or $\mathbf{m}_t$ explicitly accumulates gradients over time, with $\beta$ controlling the decay rate (how much past gradients influence the current velocity/momentum). $\alpha$ remains the learning rate. This is the approach typically implemented in deep learning frameworks when 'momentum' is specified for optimisers like SGD. It directly addresses the idea of using momentum *on the gradients*.

A popular refinement is Nesterov Accelerated Gradient (NAG) [157], which calculates the gradient at a 'lookahead' position informed by the current momentum, often leading to faster convergence.

These momentum concepts are further extended in adaptive methods like Adam. Adam [70], which has been shown to be particularly effective in high-dimensional memory-dense optimisation problems like the training of neural networks, and is typically used in stochastic settings such as when the loss function is taken with respect to a batch of data. Adam adapts the learning rate for each parameter by estimating the first moment (mean) and the second moment (uncentered variance) of the gradients. The formulation for the Adam algorithm is

as follows:

$$\mathbf{g}_t = \nabla f(\mathbf{x}_t) \tag{2.31}$$

$$\mathbf{m}_t = \beta_1 \cdot \mathbf{m}_{t-1} + (1-\beta_1) \cdot \mathbf{g}_t \tag{2.32}$$

$$\mathbf{v}_t = \beta_2 \cdot \mathbf{v}_{t-1} + (1-\beta_2) \cdot (\mathbf{g}_t \odot \mathbf{g}_t) \tag{2.33}$$

$$\hat{\mathbf{m}}_t = \frac{\mathbf{m}_t}{1-\beta_1^t} \tag{2.34}$$

$$\hat{\mathbf{v}}_t = \frac{\mathbf{v}_t}{1-\beta_2^t} \tag{2.35}$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \alpha \cdot \frac{\hat{\mathbf{m}}_t}{\sqrt{\hat{\mathbf{v}}_t} + \varepsilon} \tag{2.36}$$

where $\alpha$ is the step size (learning rate), $\beta_1$ and $\beta_2$ are exponential decay rates for the moment estimates (typically close to 1, e.g., 0.9 and 0.999 respectively), and $\varepsilon$ is a small constant for numerical stability (e.g., $10^{-8}$). The vector $\mathbf{g}_t$ represents the gradient of the objective function $f$ with respect to the parameters $\mathbf{x}_t$. The vector $\mathbf{m}_t$ is the exponential moving average of the gradient (first moment estimate). The vector $\mathbf{v}_t$ is the exponential moving average of the squared gradient (second raw moment estimate), where $\odot$ denotes element-wise multiplication. $\hat{\mathbf{m}}_t$ and $\hat{\mathbf{v}}_t$ are the bias-corrected moment estimates, counteracting the initialisation bias towards zero, particularly during the initial iterations ($t$ is small). The final parameter update scales the bias-corrected mean gradient $\hat{\mathbf{m}}_t$ inversely by the square root of the bias-corrected second moment estimate $\hat{\mathbf{v}}_t$, effectively providing per-parameter adaptive learning rates.

Figure 2.16 demonstrates the empirical performance differences between small and large step-size gradient descent, momentum based gradient descent, and Adam. 50 iterations of gradient descent are performed using the central-difference approximation of the gradient. Too small a step size may result in slow convergence, and whilst large step sizes may overcome local minima, they tend to oscillate around optimal solutions. Including momentum provides the ability to overcome local minima whilst mitigating against oscillations. In this case,

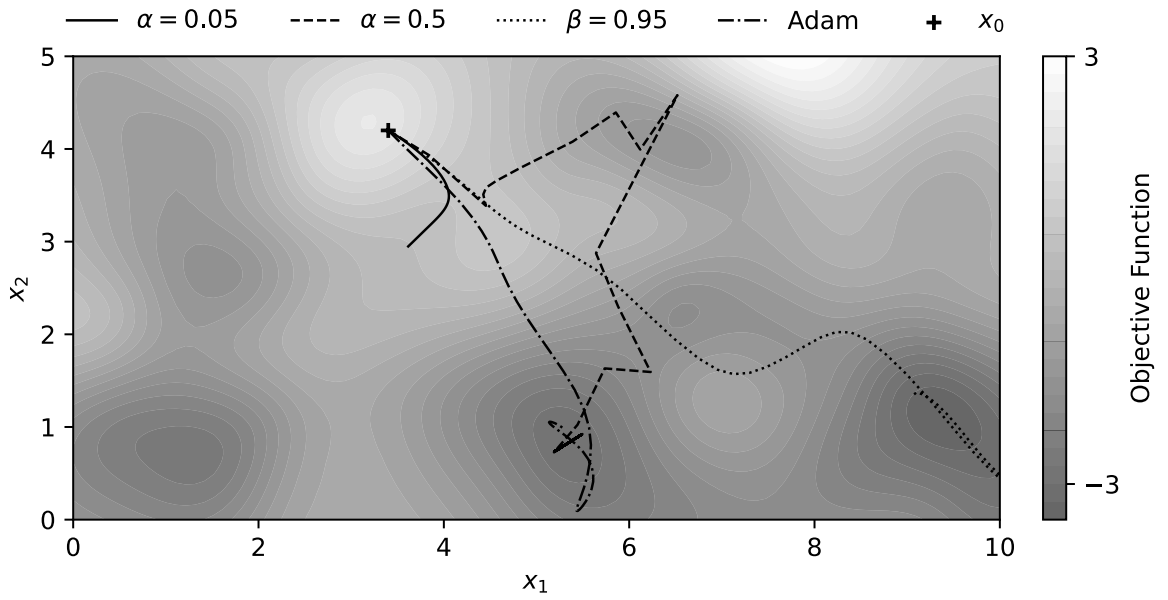



Fig. 2.16 Different first order methods applied for 50 iterations to optimise the test function. Here, $\alpha$ represents step size and $\beta$ represents momentum.

momentum helps to overcome local optima and converge towards the global optima, while the other approaches converge to local optima, or in the case of standard gradient descent do not converge at all.

Other ways of including previous local information include using previous gradients to approximate second-order information, i.e. the Hessian. Methods such as L-BFGS [83] approximate the Hessian (or rather the inverse of the Hessian) using first order information. Depending on the function, and method of calculating gradients, it may be feasible to directly obtain second order information about the function at a given solution.

### Second Order Optimisation

Second order methods take advantage of information about the curvature of a function at a given solution to inform the next iteration. To motivate Newton's method, first suppose that a function at $\mathbf{x}_t + \mathbf{p}$ is approximated using a second order Taylor series, where $\mathbf{p}$ is a small step

or perturbation from the current point $\mathbf{x}_t$:

$$f(\mathbf{x}_t + \mathbf{p}) \approx f(\mathbf{x}_t) + \nabla f_t \mathbf{p}^T + \mathbf{p}\nabla^2 f_t \mathbf{p}^T \tag{2.37}$$

where the gradient and Hessian notation at $\mathbf{x}_t$ has been simplified. In an optimisation scenario, $\mathbf{p}$ should be chosen optimally in order to minimise this second order approximation. Therefore the approximation can be differentiated with respect to $\mathbf{p}$ and equated to zero to locate the minima which has an analytical solution[6], where $\mathbf{p}$ can be solved for as follows

$$\nabla f_t + \nabla^2 f_t \mathbf{p}^T = 0 \tag{2.38}$$

$$\mathbf{p} = -\nabla f_t (\nabla^2 f_t)^{-1} \tag{2.39}$$

where this optimal step from $\mathbf{x}$ is now equivalent to the negative gradient weighted by the inverse of the Hessian. Performing iterative steps in this manner is known as *Newton's method*. Similarly to first order approaches, this step is weighted by a step size which can be defined as previously mentioned. Figure 2.17 demonstrates the differences between first and second order optimisation with the same step-size. The effect of including more local information in the form of the curvature of the function enables second-order methods to reach local optima in fewer iterations. This demonstration across both first-order methods with momentum, and second-order methods, illustrates the importance of applying all the information available within optimisation. This paradigm becomes even more important when neither Hessian nor gradient information is available, leaving decisions to be made with scalar function values.

[6]This can be verified as a minima by taking the second derivative.

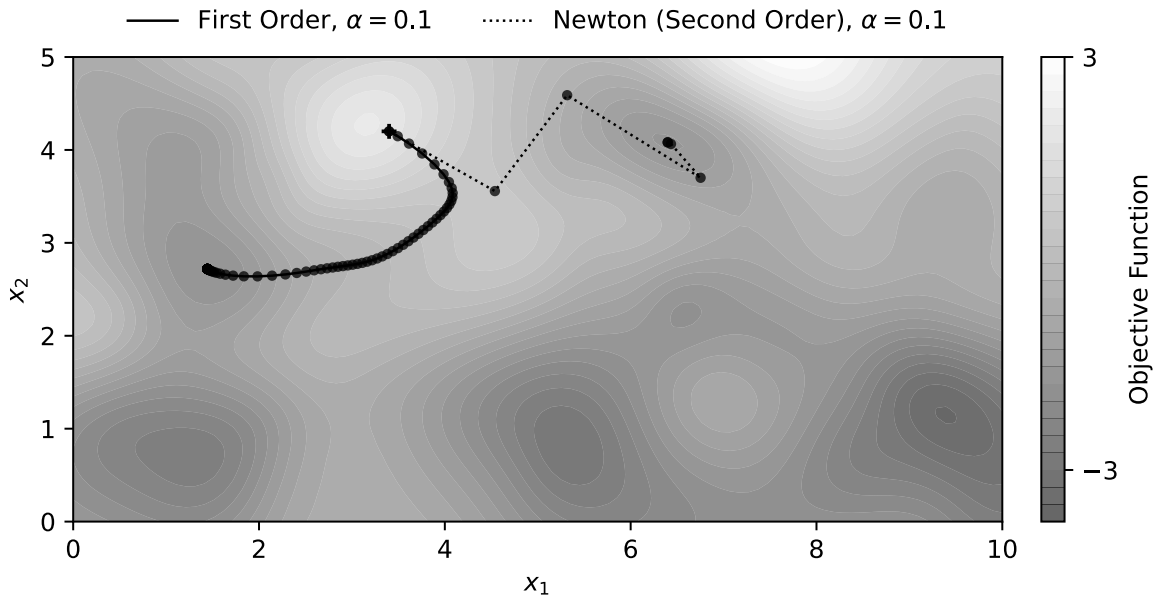



Fig. 2.17 Newton's method and standard gradient descent applied at the same starting point. Individual iterations are marked with crosses.

### 2.2.2 Data-driven (Zero Order) Optimisation

Data-driven optimisation refers to the act of optimising functions using only scalar function outputs. In this setting, data generally refers to scalar function values, though gradient or Hessian observations may be valid 'data' obtained in settings such as CFD. To mitigate this distinction data-driven optimisation is also referred to as zero-order optimisation. The two main paradigms within zero-order methods are those that evaluate the next solution based directly function evaluations (direct methods), and those that infer local or global function information to select the next solution (model-based methods). One could argue that direct methods still contain the notion of a model, albeit a very simple one, though this thesis will distinguish between them by whether they apply a *trained* model, containing an inner-optimisation problem. As model-based approaches often contain an inner-optimisation problem, they are more time consuming than model-free approaches. The difference in performance between the two paradigms is highly dependent on problem structure, function evaluation cost and budget, and dimensionality.

### Model-free Approaches

The most basic and straightforward model-free zero-order optimisation is a random search. This commonly acts as a baseline within data-driven optimisation. $n$ solutions are sampled within the feasible region $\mathcal{X}$, and the optimal solution returned is that with the lowest function value. Basic random search is as follows:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}_i \in \mathbf{X}} f(\mathbf{x}_i) \tag{2.40}$$

$$\mathbf{X} \sim \text{Uniform}(\mathcal{X}), \quad i = 1, \ldots, n \tag{2.41}$$

Figure 2.18 demonstrates the random search method for zero-order optimisation. The two

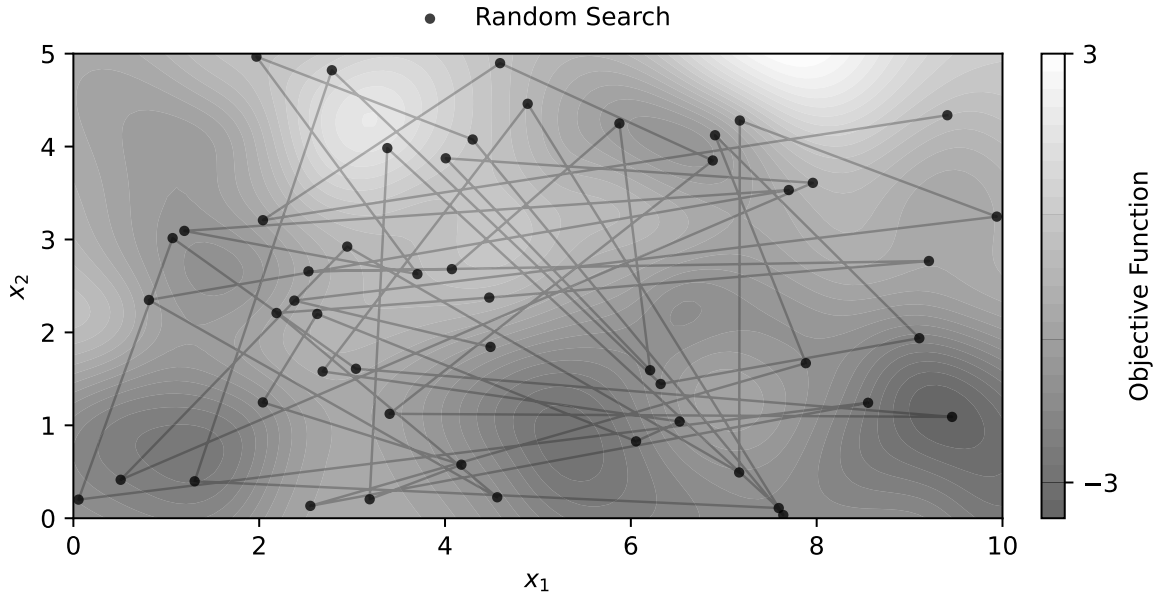
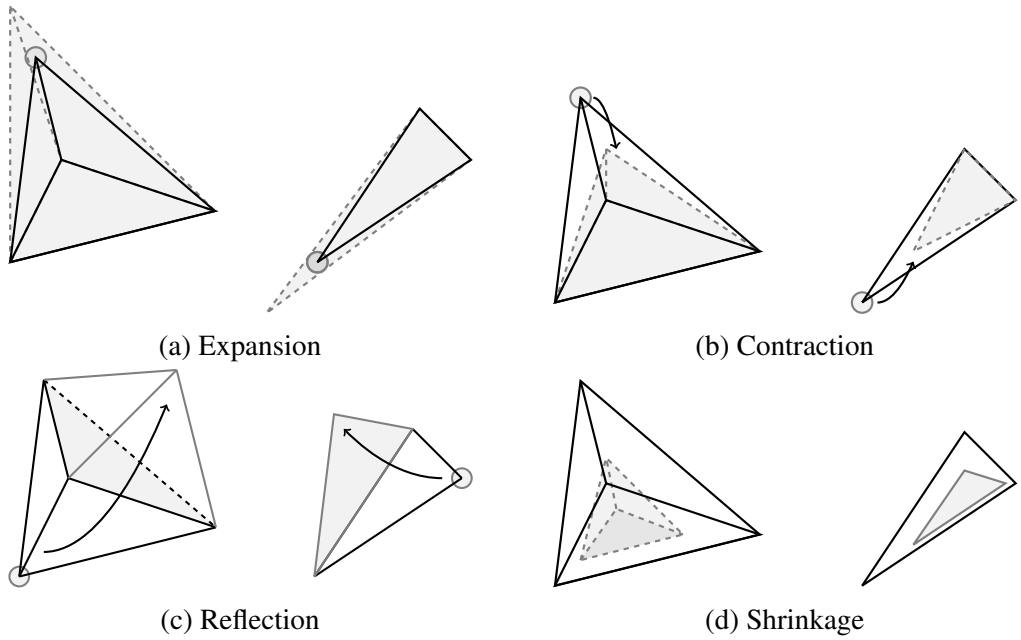



Fig. 2.18 The random search algorithm, connections are drawn between successive solutions.

main limiting factors of random search are that that successive solutions do not take into account previously gained information, i.e. it is effectively non-sequential, and that sampling in increasingly higher dimensions becomes less effective (fills less space) due to the *curse of dimensionality* [11].

Model-free approaches to mitigate these are largely concerned with taking advantage of previously gained information. The Simplex, or Nelder-Mead method [104] initially defines a simplex of $d+1$ solutions for a given $d$ dimensional optimisation problem. Based on the function values at each vertex, the simplex is mutated through operations such as reflection (flipping the simplex by sampling a new vertex opposite the worst point), expansion (extending the reflected point further), contraction (moving the worst point closer to the centroid), or shrinkage (moving all points except the best towards the best point). Figure 2.19 demonstrates these operations for both a two-dimensional and three-dimensional simplex.



(a) Expansion (b) Contraction

(c) Reflection (d) Shrinkage

Fig. 2.19 Nelder-Mead simplex operations

Figure 2.20 demonstrates how successive evaluations converge to an optima despite the lack of available gradient information.

The use of a simplex may be interpreted as using a small discrete model combined with bespoke decision-making rules. The rules may also be interpreted as being made using a discrete gradient-approximation, based on the function values at each vertex. van de Berg et al. [165] provides examples of other methods such the partition-based DiRECT (Dividing

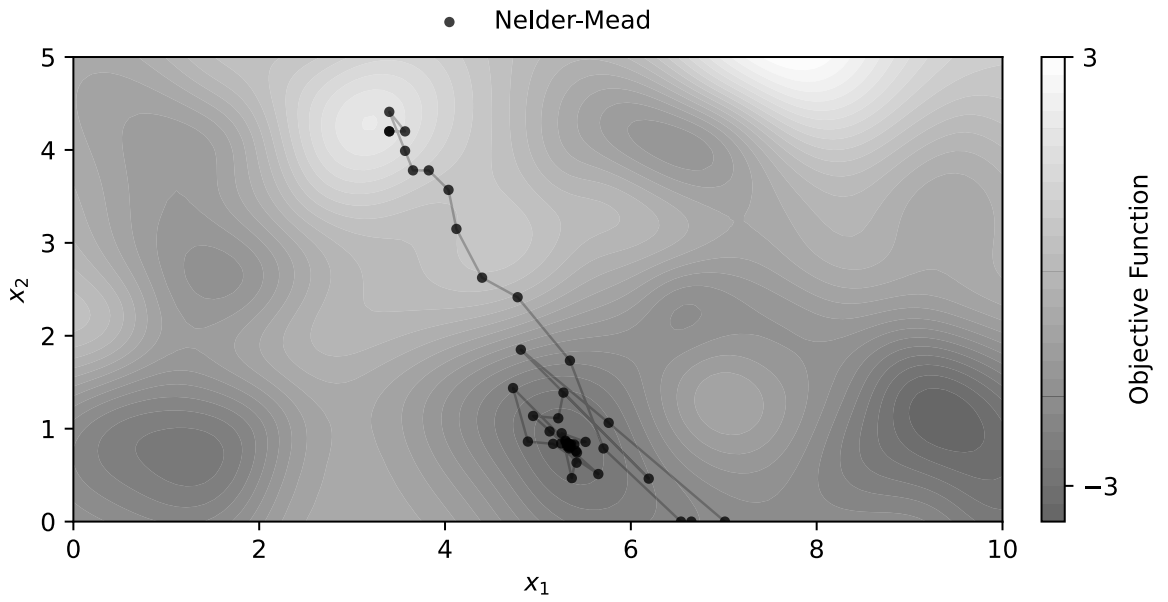



Fig. 2.20 The Nelder-Mead algorithm, connections are drawn between successive evaluations.

Rectangles) approach [62], as well as an comparative evaluation across various process systems engineering scenarios.

A large class of useful model-free zero-order optimisation algorithms are metaheuristics. While these have generally been categorised as nature-inspired optimisation algorithms, the majority of them have been shown to be equivalent [154, 36] and reducible to *local search* approaches such as simulated annealing [71], *population-based* approaches, themselves largely reducible to either particle swarm or directed evolution, or *constructive* approaches such as adaptive neighbourhood search [132]. For the purposes of this thesis, evolutionary methods will be described, though Osaba et al. [111] provides a thorough overview into other methods.

Initially, a population of solutions is defined, which can be done in the same manner as Equation 2.41. Each solution is then evaluated, and selection is performed. The goal of selecting a promising subset is to select both good solutions but maintain an overall distribution (diversity) of solutions throughout the search space. A common approach for selection is tournament selection, where $k$ solutions (typically 2-5) are selected and the best

solution is chosen to continue to the next population. This is repeated until the desired population size. Following this, an operation called crossover is performed. Two solutions are selected at random, a random index chosen in either the vector-valued or binary-valued representation of the solution, and two crossed-over vectors are returned alongside the initial solutions. This results in the population size doubling. Finally, with a random probability, solutions are mutated by randomly assigning new or perturbing variables in a given axis. The process is then repeated and terminated after a given number of iterations. Figure 2.21 demonstrates the characteristic behaviour of an evolutionary algorithm.

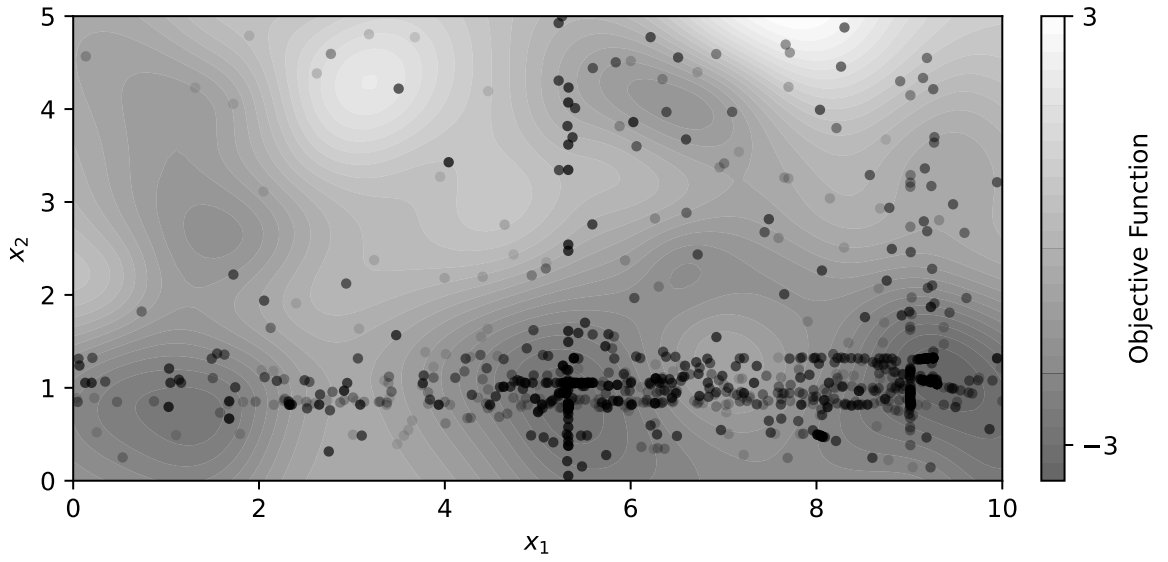



Fig. 2.21 The population of solutions within 100 iterations of an evolutionary algorithm. The transparency of a solution is proportional to the iteration of its population.

Mid-way through the algorithm, the population converges to a a local optima, however the mutation and crossover operations enable the population to 'escape' towards the global optima on the bottom right. Whilst evolutionary algorithms perform well for non-convex and non-continuous functions, they are extremely sample inefficient. Each solution within a population must be evaluated at each iteration. In Figure 2.21, a total of 1000 evaluations were performed,

given the population size of 10. Metaheuristics also contain many hyperparameters that are highly problem dependent, meaning expert-knowledge often places a key role in effective performance

Model-free zero order optimisation algorithms are generally most effective when applied to nonconvex optimisation problems due to their ability to overcome local optima. However, they often use a large number of function evaluations. In situations where the number of function evaluations is limited or there is a cost associated with evaluating the function, it follows that by offsetting resource to the construction of a model and attempting to recover local or global function information may result in more considered choices and a more tractable approach.

### Model-based Approaches

Model-based approaches apply different parametric or non-parametric models to infer local or global information about the underlying function to inform the selection of a solution at each iteration. While Newton's method creates a quadratic approximation to a function using zero, first, and second-order information, model based approaches must produce a model using only zero-order information. The two key aspects within the design of a model-based approach are what type of model to use, and whether it attempts to construct local or global information. Typically, each iteration involves two main stages: first, fitting or updating the chosen model based on the function evaluations observed so far, and second, finding the optimal point according to this model (often within a localised region) to select the next candidate solution for evaluation on the true objective function.

Local approaches define a region of interest around the current solution in which a model is deemed to be accurate, known as a *trust region*. Trust regions are commonly expressed as balls, ellipses or intervals, the size of which are determined based on how accurate the previous prediction was [179]. For a given model $m$, the trust region subproblem at iteration

$k$ can be expressed as:

$$\min_{\mathbf{s}} \quad m_t(\mathbf{x}_t + \mathbf{s}) \tag{2.42}$$

$$\text{subject to} \quad |\mathbf{s}| \leq \Delta_t \tag{2.43}$$

$$\mathbf{x}_t + \mathbf{s} \in \mathcal{X} \tag{2.44}$$

where $\mathbf{s}$ represents a perturbation within a trust region, in this case a ball of radius $\Delta$. To update the size of the trust region, the ratio of actual improvement to predicted improvement is evaluated [107]:

$$\rho_t = \frac{f(\mathbf{x}_t) - f(\mathbf{x}_t + \mathbf{s}_t)}{m_k(\mathbf{x}_t) - m_k(\mathbf{x}_t + \mathbf{s}_t)} \tag{2.45}$$

This ratio $\rho_t$ measures how well the model prediction aligns with the actual function behaviour, and $k$ indexes each model which represents the fact that a unique model is used at each iteration. Based on this ratio, the trust region is updated as follows:

$$\Delta_{t+1} = \begin{cases} \gamma\Delta_t & \text{if } \rho_t < \eta_1 \\ \Delta_t & \text{if } \eta_1 \leq \rho_t < \eta_2 \\ \Delta_t/\gamma & \text{if } \rho_t \geq \eta_2 \end{cases} \tag{2.46}$$

where $\eta_1$ and $\eta_2$ are thresholds for determining the quality of the prediction and $\gamma$ is a factor for shrinking or expanding the trust region.

Successive linear programming (SLP) applies a series of linear approximations (i.e. $m : \theta_0 + \theta_1\mathbf{x}_1 + \cdots + \theta_d\mathbf{x}_d \to y$) based on function observations within a trust-region [113]. A zero-order approximation to the gradient may be used to make this approximation, or it can be constructed by solving a least-squares problem (i.e. using a loss function such as in Figure 2.2). Similarly, Successive Quadratic Programming relies on fitting a quadratic model at

each iteration. This may be under, exactly, or over defined depending on the size of the trust region and amount of data within it [77].

If a trust region is not used, unless the underlying function exactly matches the model used, the model will most likely inaccurately reflect the true global function behaviour. The advantage of smaller models is that they have tractable or easily locatable solutions making iterations fast. Alternatively, a more flexible model such as a Gaussian process may be used such that a global representation of the underlying function can be maintained and used, at the expense of a much more difficult inner-optimisation problem. Rio Chanona et al. [131] combines both these ideas into a Gaussian process trust region framework. The trade off throughout model-based optimisation is therefore between the computational speed and ease of solving smaller models against the information contained in larger, potentially non-convex global models that are more difficult to solve. In situations where function evaluations are expensive, it is often worth maintaining a global function representation and offloading effort into optimising this cheaper, inner optimisation problem.

Figure 2.22 demonstrates the performance of applying a global quadratic model that is successively updated against a local quadratic model. Both models use a zero-order finite central difference approximation to the Hessian. The global quadratic model fails to converge to even a local optima, as the behaviour of the underlying function is not quadratic and the model is not flexible enough to account for this. Trust regions ensure that a step is taken within a region that is approximately quadratic, matching the flexibility of the model used to inform the next solution to evaluate. However, while smaller trust regions enable better approximations to an underlying function, they often inhibit the ability to obtain a global solution. In order to locate globally optimal solutions, a more flexible model must be used, such as a Gaussian process, with the ability to accurately model the entire search space.

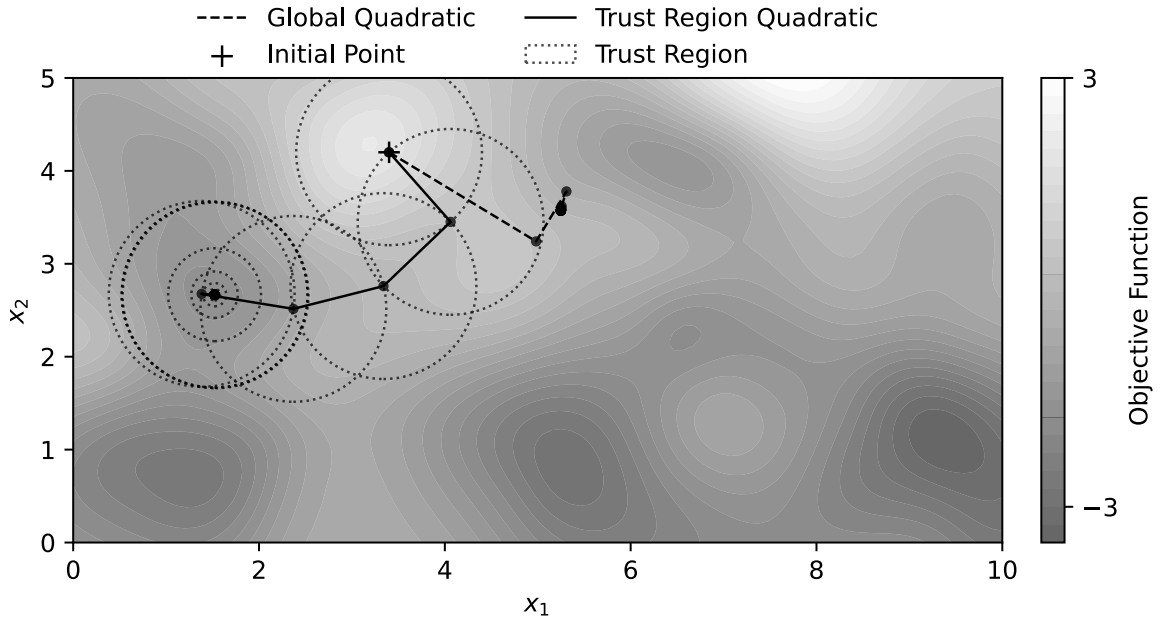



Fig. 2.22 Successive quadratic models used either globally or within a trust region. Trust regions specify smaller local areas that behave closer to the quadratic approximation, resulting in convergence.

### 2.2.3 Bayesian Optimisation

As a fundamentally model-based approach, the general purpose of Bayesian optimisation is to locate *globally* optimal solutions to unknown functions in as few zero-order function evaluations as possible. This situation commonly arises in areas such as laboratory work involving physical experiments or the optimisation of computational fluid dynamics. In both cases the underlying objective function is any combination of inaccessible (i.e. all aspects are unknown and only individual scalar evaluations are available), not useful (for example in the case of a mathematical model with a large number of complex terms, making the calculation of upper and lower confidence bounds required for global optimisation intractable), or expensive, necessitating few function evaluations and by extension the ability to approximate gradients via finite differences. The resulting optimisation problem is often termed *expensive black-box*, but can be either. Due to the problem setting, Bayesian optimisation is more often used for maximisation, a convention that will follow for the remainder of this subsection.

Bayesian optimisation refers to taking the general approach of treating anything not known as uncertain. In the setting of sample efficient model-based optimisation algorithms, as opposed to using a fixed model structure (i.e. linear or quadratic) it is therefore appropriate to use a distribution of potential underlying functions, such as a Gaussian process. Given that this model, and resulting evaluations are probabilistic, the resulting inner-optimisation problem must by extension be solved under uncertainty.

### Expected Improvement

Suppose that a dataset of $n$ observations has been collected $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$. A Gaussian process model is then used to model this dataset, $\mathcal{GP}$. When optimising this under uncertainty to generate the next immediate solution, it follows that the solution with the maximum *expected* function value should be selected, or equivalently the solution with the largest expected improvement from the current best. This function must first be derived, where each location in $\mathcal{X}$ is associated with an expected improvement, generally known as an acquisition or utility function which is denoted $\mathcal{U}$. The expected improvement associated to a solution is as follows:

$$\mathcal{U}(\mathbf{x}; \mathcal{D}) = \mathbb{E}[\max(y^* - \mathcal{GP}_{\mathcal{D}}(\mathbf{x}), 0)] \tag{2.47}$$

where $y^*$ is the current solution with the largest function value (i.e., $f(\mathbf{x}^*)$ where $\mathbf{x}^* = \arg\max_{\mathbf{x}_i \in \mathcal{D}} y_i$). In the case of a Gaussian processes, function evaluations return one dimensional Gaussian distributions and the utility function becomes

$$\mathcal{U}(\mathbf{x}; \mathcal{D}) = \mathbb{E}[\max(y^* - \mathcal{N}(\mu_{\mathcal{D}}(\mathbf{x}), \sigma^2_{\mathcal{D}}(\mathbf{x}))] \tag{2.48}$$

Expressing this using a standard normal distribution through a change of variable results in

$$\mathcal{U}(\mathbf{x}) = \sigma_{\mathcal{D}}(\mathbf{x}) \cdot \mathbb{E}[\max(Z,0)] \quad \text{where} \quad Z = \frac{y^* - \mathcal{N}(\mu_{\mathcal{D}}(\mathbf{x}), \sigma^2_{\mathcal{D}}(\mathbf{x}))}{\sigma_{\mathcal{D}}(\mathbf{x})}. \tag{2.49}$$

The expectation term has an analytical solution as it involves integrating over a standard Gaussian distribution (another advantage of Gaussian processes specifically) which is as follows:

$$\mathcal{U}(\mathbf{x}; \mathcal{D}) = \sigma(\mathbf{x}) \cdot (u \cdot \Phi(u) + \phi(u)) \tag{2.50}$$

where $u = \frac{y^* - \mu_{\mathcal{D}}(\mathbf{x})}{\sigma_{\mathcal{D}}(\mathbf{x})}$, $\Phi$ is the standard cumulative Gaussian function and $\phi$ is the standard Gaussian function. The expected improvement acquisition function *may* be interpreted as balancing exploration and exploitation, though it is important to note that it is not derived with this in mind. Figure 2.23 demonstrates a Gaussian process model of a one-dimensional set of data alongside the respective expected improvement function. Within Bayesian optimisation,

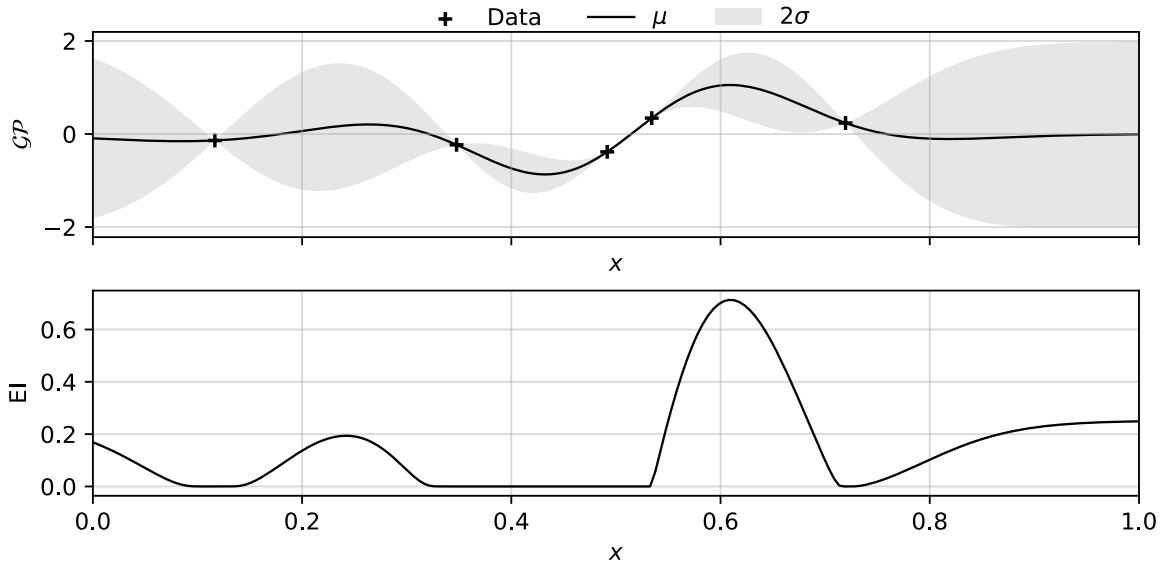



Fig. 2.23 A Gaussian process and respective expected improvement function.

a utility function such as the expected improvement is optimised, the resulting solution of this

inner optimisation problem is then evaluated, added to the dataset and the process repeated until a time or evaluation budget is reached. Empirically, it can be observed that acquisition functions are non convex, and in some cases such as within expected improvement, contain regions that are near flat. Both these properties both contribute towards difficulties when solving the inner optimisation problem, making methods such as meta heuristics often most effective [153] over first-order methods, in spite of the fact that most acquisition functions have analytical derivatives available [42]. Alternatively, the function itself could be modified, as is the case with LogEI proposed by Ament et al. [4], to ensure that gradient-based approaches can overcome flat regions in the search space. The expected improvement utility function presented above contains a number of key assumptions about the problem setting. Namely, observations are themselves noiseless, decisions should be made with respect to the next subsequent improvement (known as a *myopic* policy and valid in the setting where the experimental budget is unknown), and this improvement is with respect to an observed function value as opposed to the maximum of the posterior Gaussian process model. Relaxing or modifying any one of these three assumptions will result in the acquisition function having a different form. Assuming first that the final solution is that associated with the maximum of the final $\mathcal{GP}$ model, as opposed to the maximum evaluated solution, the utility function in Eq 2.47 therefore becomes

$$\mathcal{U}(\mathbf{x};\mathcal{D}) = \mathbb{E}[\max_{\mathbf{x}'\in\mathcal{X}} \mu_{\mathcal{D}'}(\mathbf{x}')|\mathbf{x}_t = \mathbf{x}] - \max_{\mathbf{x}'\in\mathcal{X}} \mu_{\mathcal{D}}(\mathbf{x}') \tag{2.51}$$

where $\mu_{\mathcal{D}}(\mathbf{x})$ is the posterior mean of the Gaussian process at iteration $t$, $\mu_{\mathcal{D}'}(\mathbf{x})$ is the posterior mean after hypothetically sampling at $\mathbf{x}_t$, and $\mathbf{x}'$ represents a inner variable concerning the maximisation of the posterior. Effectively for a given solution $\mathbf{x}$, if this were to be evaluated, what would be the expected difference between the maximum of the old Gaussian process posterior and the updated posterior. As the predicted output is itself a distribution, evaluating this function requires performing an expectation over potential Gaussian processes. This

utility function is called the Knowledge Gradient [40] and empirically favours exploration over expected improvement, due to it's global assessment of improvement as opposed to a local comparison between observed values. However, as the expectation taken with respect to an inner maximisation, the equation must be numerically approximated using methods such as Thompson sampling which approximates solutions to the inner maximisation. Garnett [42] presents methods for numerically estimating a large number of acquisition functions.

### Noisy Expected Improvement

Another assumption that is often useful to relax is that of noiseless observations. Similar to Knowledge Gradient, in the case of a Gaussian process with constant or heteroskedastic noise, it follows that the expected improvement should be with respect to the improvement in the mean of the current and subsequent Gaussian process models. In doing so, we are expressing our interest in optimising the signal and not the noise or values of specific observations. Figure 2.24 shows a Gaussian process with assumed observational noise. The respective

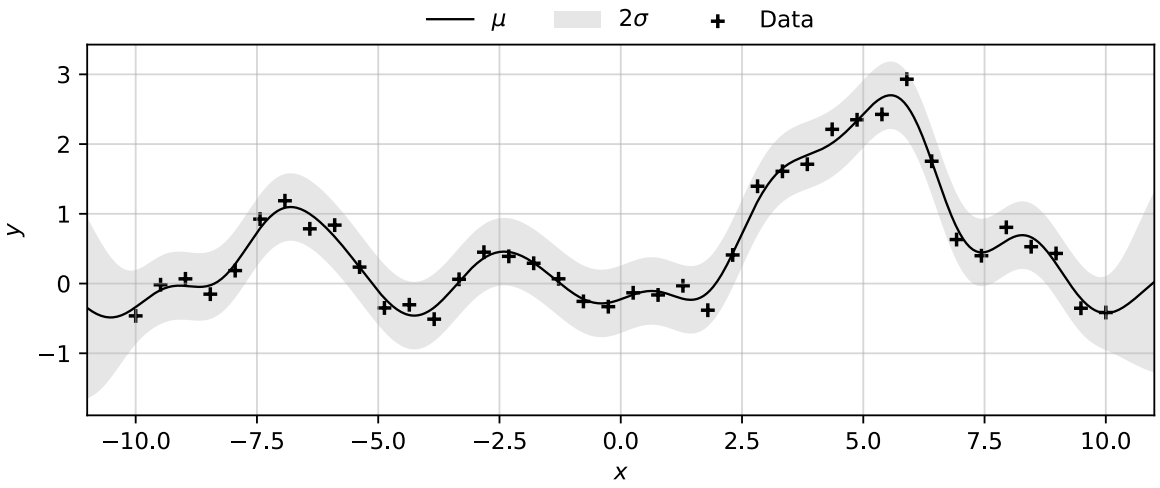



Fig. 2.24 A Gaussian process with assumed noisy observations fit to a one-dimensional data set.

utility function shown in Equation 2.47 therefore becomes

$$\mathcal{U}(\mathbf{x};\mathcal{D}) = \int \left[\max_{\mathbf{x}'\in\mathcal{X}} \mu_{\mathcal{D}'}(\mathbf{x}') - \mu^*\right] \mathcal{N}(y;\mu,s^2)\mathrm{d}y \tag{2.52}$$

where $\mu^* = \max_{\mathbf{x}'\in\mathcal{X}} \mu_{\mathcal{D}}(\mathbf{x}')$, which is a fixed constant value for any value of $\mathbf{x}$ and a given dataset, and as before $\mathcal{D}'$ is the hypothetical updated dataset $\mathcal{D}\cup\{(\mathbf{x},y)\}$. The integral over $y$ must be taken, accounting for all potential observed values which now assumed to stem from normal distribution (i.e. normally distributed noisy measurements). Given a hypothetical observation $y$, the value of the mean of the resulting Gaussian process $\mu_{\mathcal{D}'}$ at given set of potential locations $\mathbf{x}'$ [7] is

$$\mu_{\mathcal{D}'} = \mu_{\mathcal{D}} + \frac{K_{\mathcal{D}}(\mathbf{x}',x)}{s}\frac{y-\mu}{s}, \tag{2.53}$$

where $\mu$ and $s^2$ are the mean and standard deviation of the distribution of potential values $y$ could take. When we express this distribution in terms of the standard normal distribution $z := \mathcal{N}(0,1)$, we have $y = \mu + sz$ and as a result

$$\mu_{\mathcal{D}'} = \mu_{\mathcal{D}}(\mathbf{x}') + \frac{K_{\mathcal{D}}(\mathbf{x}',x)}{s}z. \tag{2.54}$$

Placing this into Eq. 2.52, only the expectation over the standard normal distribution has to be taken, resulting in the following.

$$\mathcal{U}(x;\mathcal{D}) = \int \max_{\mathbf{x}'} \left(\mu_{\mathcal{D}}(\mathbf{x}') + \frac{K_{\mathcal{D}}(\mathbf{x}',x)}{s}z\right)\phi(z)\,\mathrm{d}z - \mu^* \tag{2.55}$$

An important observation here is that the maximisation within the integral is with respect to a discrete set of one dimensional linear functions in $z$. Equivalent to this maximisation,

[7] By discretising this inner maximisation with respect to a limited number of candidates it becomes tractable, though approximated.

the upper envelope of these one-dimensional lines can be taken, resulting in an analytical solution to the integral, by integrating between the vertices of each line segment in the upper envelope. In the case of the Gaussian process shown in Figure 2.24, for a single evaluation of the utility function at $x = 5.0$, the set of discrete one dimensional lines as a function of potential standardised observations $z$ is shown in Figure 2.25, assuming 75 equally spaced values of $\mathbf{x}'$, each corresponding to an individual line. The integral over the line segments

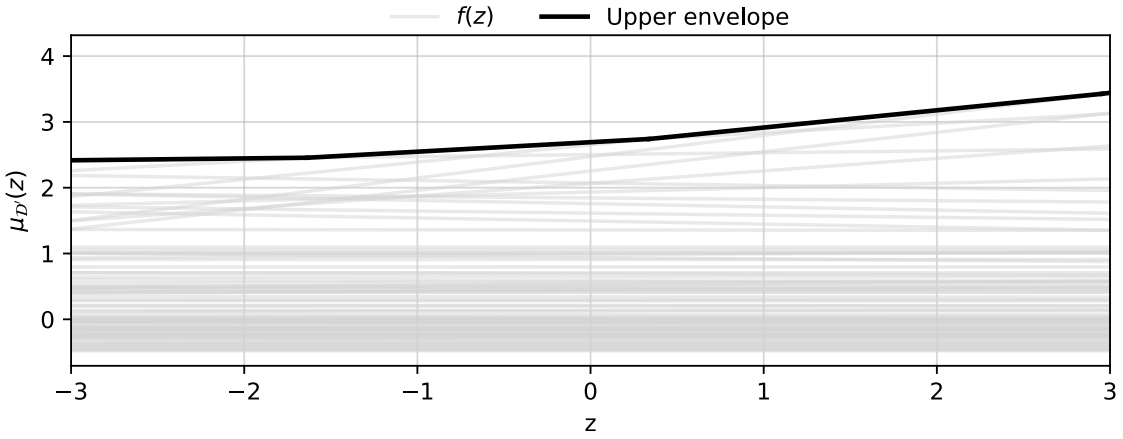



Fig. 2.25 One-dimensional lines representing potential improvements over the current best observation as a function of standardised observations $z$, for a single evaluation of the utility function at $x = 5.0$. The upper envelope of these lines is used to analytically solve the integral in the noisy expected improvement acquisition function.

corresponding to this upper envelope can be analytically evaluated to calculate values of the acquisition function. Figure 2.26 demonstrates this form of a noisy expected improvement acquisition function, where each value corresponds to the expected improvement in the 'signal' of the Gaussian process model.

### Extensions

To further improve the effectiveness of Bayesian optimisation, and particularly decision making under uncertainty, we can take actions that account for decisions beyond the immediate next one. Approaches for *non-myopic* Bayesian optimisation include via successive

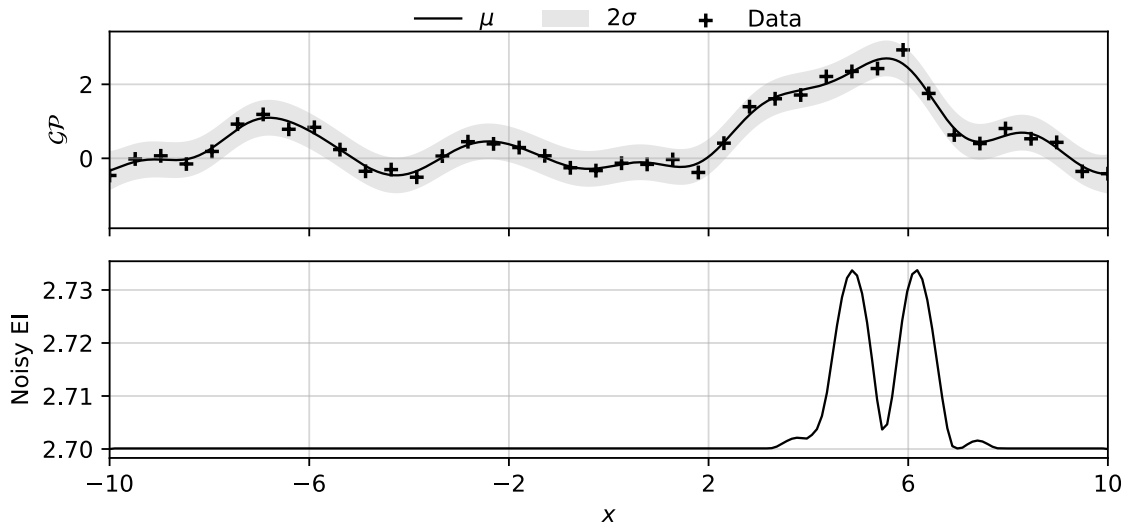



Fig. 2.26 Noisy expected improvement acquisition function for the Gaussian process model shown in Figure 2.24. Each value represents the expected improvement in the 'signal' of the model, accounting for normally distributed noisy measurements.

rollouts [180] or pathwise planning [37]. In some settings such as CFD, simulations are both expensive and gradient observations are accessible. Including gradient observations within Bayesian optimisation is trivial through the creation of a joint zero and first-order Gaussian process. Methods for high-dimensional Bayesian optimisation have also been proposed [34]. Similar to the previous discussion on local and global function information, performing BO in high dimensions generally involved trust-regions that result in more accurate local Gaussian process models, potentially at the expensive of locating a globally optimal solution. Intuitively, in higher dimensions, a trust region omits larger areas than smaller dimensions due to the curse of dimensionality. Alternatively, the dimensionality of the search space itself can be reduced through decomposition, even if the decompositions are performed at random [183].

To conclude, Bayesian optimisation is an intuitive and attractive methodology for many real world optimisation problems. It is sample efficient, global, approach with few hyper-parameters that makes very few assumptions. However, due to the broad nature of the

assumptions that Bayesian optimisation handles, success can often mean identifying when and where assumptions can be relaxed. For example, whether the form of the underlying function is known, or variables are known to be correlated. Or whether function evaluations have different costs in different areas of the search space. In this section, Bayesian optimisation has been demonstrated to serve as a blank canvas, which has the ability to be adapted for domain specific improvements across engineering and scientific discovery.

### 2.2.4 Other Optimisation Paradigms

In the remainder of this section, other optimisation paradigms will be covered, including multi-objective and multi-fidelity optimisation.

#### Multi-objective Optimisation

The majority of realistic problems rarely have a single objective. For example, problems concerning sustainability should balance environmental, economic, and social objectives [134]. Popular empirical approaches to locate solutions that balance multiple objectives include Multi-criteria Decision Analysis (MCDA) [169], or a weighted-sum approach that combines all the objectives into a single value [92]. In these cases, a Pareto front is returned, defining a set of optimal solutions that trade off a number of objectives optimally. For the dual objective-case the weighted sum formulation is as follows:

$$\mathbf{X}_{\text{Pareto}} = \left\{ \mathbf{x}^*(\alpha) : \mathbf{x}^*(\alpha) = \underset{\mathbf{x}\in\mathcal{X}}{\arg\min} \left[\alpha f_1(\mathbf{x}) + (1-\alpha) f_2(\mathbf{x})\right],\ \alpha \in [0,1] \right\} \tag{2.56}$$

where every solution in $\mathbf{X}_{\text{Pareto}}$ is the solution to a specific problem for a given value of $\alpha$. Changing $\alpha$ results in solutions that weight the two competing objectives differently. In practice the set of potential $\alpha$ values is discretised and a finite number of problems are solved.

While MCDA approaches can effectively include qualitative human judgement through the inclusion of preference-based information, they may still result in sub-optimal solutions from a mathematical perspective. Similarly, weighted-sum approaches are known to only be optimal when all objectives are convex. In order to maintain the ability to obtain optimal solutions through an optimisation-based approach, while doing so in non-convex settings, evolutionary methods have been successfully applied for a large number of real world multi-objective optimisation problems. Evolutionary methods such as Non-dominated Sorting Genetic Algorithm (NSGA/-II) [28], use a population of solutions that ensure that the solutions presented within the Pareto front are both optimal and well distributed, ensuring adequate options for a decision maker. As mentioned previously, evolutionary methods also have the ability to overcome local optima and can be applied in non-convex settings typical of problems with competing objectives.

### Multi-fidelity Optimisation

Multi-fidelity optimisation is an approach used when evaluating the objective function at different levels of accuracy or detail is possible, each with associated computational costs. This paradigm is particularly useful in scenarios where high-fidelity evaluations are expensive, but lower-fidelity approximations are available at a reduced cost.

In a multi-fidelity setting, there is typically have access to multiple models or simulations of varying fidelity levels. These can be represented as a set of functions $f_1, f_2, ..., f_M$, where $f_1$ is the lowest fidelity (generally cheapest to evaluate) and $f_M$ is the highest fidelity (most expensive and accurate), though the extension may be made such that the fidelity level is continuous, or there may be multiple independent fidelities. The goal is to apply information from lower-fidelity models to guide the optimisation process, while still converging to the optimal solution of the highest-fidelity model.

The key challenge in multi-fidelity optimisation is balancing the trade-off between exploration of the design space using cheaper, low-fidelity evaluations and exploitation of promising regions using expensive, high-fidelity evaluations. Various strategies have been developed to address this including trust-region methods [90], space mapping techniques involving polynomial or Gaussian process models [44, 60], or ensemble methods [79]. The effectiveness of multi-fidelity optimisation relies on the correlation between low and high-fidelity models [63]. By combining information from models of varying fidelity, multi-fidelity methods enables more efficient use of computational resources and can lead to faster convergence to optimal solutions in complex, computationally expensive problems.

### 2.2.5 Optimisation Summary

Optimisation involves finding minima or maxima of functions, with approaches varying based on available information and problem characteristics. Zero-order methods use only function values, first-order methods incorporate gradients (first derivative), and second-order methods utilise curvature (second derivative) information. Model-free approaches like random search, Nelder-Mead, and evolutionary algorithms directly use function evaluations to inform the next solution to be evaluated, while model-based methods construct approximations of the underlying function to infer higher-order information to guide the search. Bayesian optimisation, particularly relevant for expensive black-box functions, uses probabilistic models like Gaussian processes to balance exploration and exploitation through an acquisition function. Extensions to handle multiple objectives and fidelity levels enable optimisation of real-world problems with competing goals and varying computational costs. The choice of optimisation method ultimately depends on many factors including function evaluation cost, dimensionality, and problem structure.

## 2.3 Chapter Summary

This chapter provided the necessary background on the modelling and optimisation methodologies central to this thesis, viewed through the lens of deductive and inductive reasoning in scientific discovery. It first contrasted physics-based modelling, exemplified by Computational Fluid Dynamics (CFD) derived from first principles, with the inductive paradigm of data-driven models. Parametric approaches like polynomials and neural networks, along with techniques to manage their complexity like regularisation and cross-validation were discussed. Non-parametric methods, including k-nearest neighbours, kernel smoothing, and particularly Gaussian Processes (GPs), were detailed, emphasising GPs' ability to capture uncertainty and model functions effectively from limited data via Bayesian inference.

The discussion then shifted to optimisation, outlining numerical methods utilising gradient (first-order) and Hessian (second-order) information, such as gradient descent variants and Newton's method. Zero-order (derivative-free) optimisation techniques were then explored, differentiating model-free strategies like random search, Nelder-Mead, and evolutionary algorithms from model-based approaches using local (trust-region) or global surrogate models. Bayesian optimisation (BO) was highlighted as a key methodology for efficiently optimising expensive, black-box functions, focusing on its use of GP surrogates and acquisition functions (like Expected Improvement) to balance exploration and exploitation, alongside extensions for noisy data, multi-objective, and multi-fidelity problems. Throughout, the chapter linked these techniques to the practical challenges and goals of engineering design and discovery.

Building on this technical foundation of modelling and optimisation techniques, the subsequent chapter introduces the DARTS framework, applying multi-fidelity Bayesian optimisation specifically to the design and analysis of simulated chemical reactors.

# Chapter 3

# Methods for Data-Driven Design and Analysis of Reactor and Tube Simulations

## Contents


**3.1 Introduction . . . . . 65**

**3.2 Background . . . . . 69**

3.2.1 Multi-fidelity Bayesian optimisation . . . . . 69

**3.3 Method . . . . . 73**

3.3.1 Parameterisation . . . . . 73

3.3.2 Fidelities . . . . . 74

3.3.3 Simulation . . . . . 76

3.3.4 Multi-fidelity Bayesian optimisation . . . . . 78

3.3.5 Stopping criteria . . . . . 79

**3.4 Results . . . . . 81**

3.4.1 Parameterisation . . . . . 83

3.4.2 Model Validation . . . . . 84

3.4.3 Solution Validation . . . . . 91

**3.5 Chapter Conclusions and Future Work . . . . . 93**

# Publication Note

This chapter contains content from the following publications:

- Savage, T., Basha, N., McDonough, J., Matar, O., & del Rio Chanona, E. A. (2023). *Multi-fidelity data-driven design and analysis of reactor and tube simulations* in Computers & Chemical Engineering (Vol. 179). https://doi.org/10.1016/j.compchemeng.2023.108410.
- Savage, T., Basha, N., Matar, O., & del Rio Chanona, E. A. (2022). *Deep Gaussian Process-based Multi-fidelity Bayesian optimisation for Simulated Chemical Reactors* presented at the NeurIPS Workshop on Gaussian Processes, Spatiotemporal modelling, and Decision-making Systems. https://arxiv.org/abs/2210.17213.

The work presented in this chapter explores additive manufacturing, machine learning, and reactor design. By combining multi-fidelity Bayesian optimisation with computational fluid dynamics, we demonstrate how complex 3D reactor geometries can be automatically discovered, leading to experimental performance improvements of 60% over conventional designs and the discovery of new driving fluid flows.

## Modifications from Published Version

The following modifications have been made to the published version:

- Some language has been altered to better fit within the context of this thesis.
- Both publications mentioned have been combined into a single narrative.

# Abstract

Optimising complex reactor geometries is key in order to promote enhanced efficiencies, reducing downstream emissions and costs. In this chapter, I present a framework to solve this nonlinear, computationally expensive, and derivative-free optimisation problem. Gaussian processes are used to learn a multi-fidelity model of reactor simulations correlating multiple independent continuous mesh fidelities. The joint search space of operating conditions and low-dimensional reactor geometries is explored through lower fidelity simulations, evaluated based on a weighted acquisition function, trading off information gain with cost of simulation. Within the framework presented, DARTS, I derive a novel criteria for dictating optimisation termination, ensuring a high fidelity solution is returned before budget is exhausted. I investigate the design of helical-tube reactors under pulsed-flow conditions, which have demonstrated outstanding mixing characteristics. To validate our optimal operating conditions, we experimentally validate the optimal reactor geometry under optimal pulsed-flow conditions, confirming mixing performance. Our approach is applicable to a broad variety of expensive simulation-based optimisation problems, enabling the design of more complex novel parameterised chemical reactors.


## 3.1 Introduction

Chemical processes to transform raw materials into valuable products will always need to occur, and as markets grow practitioners seek to identify novel economically and environ-

mentally friendly alternatives to existing solutions. At the core of these processes, chemical reactors have long been studied, designed, and optimised for the purposes of industrial chemical synthesis. However, as batch processes are being made continuous [136, 114], and speciality products such as biologically-derived chemicals enter industrial viability [161], there is a need for the design of novel case-specific reactors.

Novel reactor configurations have increasingly been considered for chemical synthesis, for example, microfluidic reactors [32, 86], and mesoscale reactors [135]. Microfluidic reactors can enable finer control over local conditions resulting in increased product selectivity, and improved heat transfer resulting in more sustainable processes [32]. 3D printed mesoscale reactors have been proposed as next-generation alternatives to traditionally manufactured designs, lending to their large potential design space. 3D printed reactors for biodiesel production have been shown to provide high yields [135], and helical coil-in-coil reactors, unable to be manufactured otherwise, have been shown to demonstrate 'excellent plug-flow behaviour' [95, 93, 94].

The modelling and design of traditional chemical reactors has largely been considered an art [47], with small improvements in performance resulting in wider impacts on product yield, sustainability, and economic costs. However, with the promise of new reactors comes the need for new analytical techniques to model and optimise in increasingly complex design spaces.

Chemical reactors have been investigated through computational fluid dynamics (CFD) simulations, where systems of partial differential equations (PDEs) with large degrees of freedom are solved iteratively, resulting in large computational costs. In addition to being expensive, gradient information is practically unavailable. In some simulation-based scenarios, optimisation quantities can be obtained directly from a simulation [80]. As such, the adjoint method can be applied to derive gradient quantities which can be used within a gradient-based local optimisation scheme such as the Newton method or BFGS [68, 101, 107].

However, in chemical reactor-based domains, performance is often quantified through a secondary quantity. For example, a tracer is simulated to pass through the reactor and a value[1] is then subsequently derived from this concentration profile through an additional fitting procedure [138]. This results in a scenario in which the gradient of a simulation is practically inaccessible, and derivative-free optimisation must be applied. In addition, derivative-based approaches may converge to local solutions limiting their use for truly global scientific discovery.

Derivative-free optimisation has found significant application in domains where mathematical expressions or gradients are unavailable. Examples include the optimisation of proprietary chemical process software [137, 19], chemical reaction optimisation [35], real-time optimisation [165, 30], and topology optimisation of two-dimensional chemical reactor channels [20]. With the advent of new technologies in reactor design, reactor geometries are becoming more parameterised, and operating conditions are considered jointly alongside geometry, resulting in higher-dimensional, more complex derivative-free optimisation problems.

In many real-world and simulated engineering systems, different quality evaluations of quantities of interest are available. Reactor performance can be quantified by a correlation of dimensionless numbers, a CFD simulation, or a pilot-scale experiment. These all attempt to capture the true underlying performance of a system, with differing accuracies and associated costs. Taking the view that only the industrial-scale reactor in its intended setting will provide a true evaluation of performance: any approximation to this, including pilot and lab-scale experiments, simulations, and basic calculations all become valid lower-fidelity evaluations which may be used simultaneously for design and optimisation. For CFD simulations of a reactor, fidelities are most often associated with the number of finite element cells in a simulation as they dictate the accuracy and computational cost. By motivating the notion that all predicted, unmeasured quantities derive from a lower-fidelity approximation to a desired

[1] For example the number of equivalent tanks-in-series

high-fidelity function, it becomes pertinent to investigate methodologies that apply these approximations to learn about the true system of interest.

The purpose of this chapter is to formulate the joint design of a simulated helical-tube reactor and its operating conditions as a multi-fidelity black-box optimisation problem. The contributions are three-fold. Firstly, a framework for the Design and Analysis of Reactor and Tube Simulations, DARTS, is demonstrated by applying it for the simultaneous optimisation of a helical-tube reactor geometry and operating conditions. The methodology chooses a simulation fidelity at each iteration based on a cost-adjusted estimated information density, with the overall aim being to optimise for the highest fidelity given a time budget. In addition, a novel stopping criteria is derived, ensuring that a high-fidelity solution is returned. Secondly, our framework is applied to the industrially relevant problem of simultaneously optimising the geometry and operating conditions of a simulated pulsed-flow helical-tube reactor. Unlike the majority of approaches, the framework allows for the cost of a simulation to be modelled as a function of both simulation fidelities and decision variables. Two independent fidelities are accounted for, the axial and radial cell count of a simulation, both of which are treated as jointly continuous. Thirdly, assumptions made within the methodology are validated using experimental data. The optimal geometry is 3D printed and evaluated using associated optimal pulsed-flow operating conditions, validating both the performance of the framework as well as our multi-fidelity model. This approach is extensible to a large number of potential design and optimisation problems, ranging from microscale to industrial reactors.

The rest of this chapter is structured as follows. Section 3.2 provides background around approaches for multi-fidelity Bayesian optimisation. Section 4.2 outlines the methodology, starting with the problem setting and including details about simulation properties, reactor parameterisation and fidelities. The specific approach to the multi-fidelity Bayesian optimisation of simulated chemical reactors is then outlined. Section 4.3 presents the results

including experimental validation via 3D printing. Finally, Section 3.5 outlines conclusions and motivations for remaining chapters.

## 3.2 Background

Bayesian optimisation has been motivated and outlined in the Background chapter to this thesis as an attractive method for the optimisation of expensive to evaluate, derivative-free, or intractably complex functions. Equation 3.1 demonstrates the canonical optimisation problem that Bayesian optimisation serves to solve:

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmax}}\ f(\mathbf{x}). \tag{3.1}$$

Recently, domain-specific adaptations to Bayesian optimisation have been developed, leading to improved performance in specific applications [102, 45, 37]. For example, alternative formulations have been proposed to facilitate the setting where function evaluations can be performed in parallel, resulting in larger information gain and potentially faster convergence [45, 17, 105]. Other domain-specific adaptations to Bayesian optimisation include approaches for scenarios where there is an additional cost incurred for evaluating decision variables that are 'far' from previous evaluations [37], various approaches for high-dimensional [33], and multi-objective [26] Bayesian optimisation have also been developed.

### 3.2.1 Multi-fidelity Bayesian optimisation

Multi-fidelity approaches take advantage of one or more, cheaper to evaluate[2], but potentially biased and more uncertain function evaluations. Correctly leveraging this information can lead to the acceleration of design and optimisation. In a multi-fidelity context, the black-box

[2]Often with respect to time, monetary cost, or computational resources.

optimisation problem presented in Eq.3.1 becomes

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmax}}\ f(\mathbf{x}, \mathbf{z}_\bullet), \tag{3.2}$$

where fidelity parameters $\mathbf{z}$ may influence both the accuracy and cost of a function evaluation. Equation 3.2 highlights the emphasis on locating an optimal solution for the highest fidelity function evaluation $\mathbf{z}_\bullet$, given this is the system considered 'most true'. Multi-fidelity Bayesian optimisation algorithms to solve Eq.3.2 have been developed and applied across a number of domains [65, 53, 156, 39, 138, 119, 175, 10]. These approaches all have a common characteristic in that both $\mathbf{x}$ and $\mathbf{z}$ need to be selected at each iteration.

One approach would be to model fidelities discretely, with Deep Gaussian Processes (DGPs) providing a useful model for such a setting [138]. DGPs model the underlying function of interest (in this case the high fidelity function) as a composition of nonlinear functions of lower-fidelity observations. In this formulation, the function at fidelity $\mathbf{z}_i$ is represented as $f(\mathbf{x}, \mathbf{z}_i) = \rho_{i-1}(f(\mathbf{x}, \mathbf{z}_{i-1}), \mathbf{x}) + \phi_{i-1}(\mathbf{x})$, with multi-fidelity DGPs combining these terms into nested GPs. This provides an end-to-end model across all discrete fidelity levels that naturally captures the hierarchical relationship between fidelities, as opposed to forming a one-way relationship from low to high fidelity observations.

After selecting the query point $\mathbf{x}^*$ using standard UCB at the highest fidelity:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{X}}\ \mu_{\hat{f}_t}(\mathbf{x}, \mathbf{z}_\bullet) + \beta^{1/2}\sigma_{\hat{f}_t}(\mathbf{x}, \mathbf{z}_\bullet)$$

The fidelity level is selected based on the trade-off between variance of prediction and simulation cost, inspired by the approach presented by Kandasamy et al. [66] but with fewer hyperparameters:

$$\mathbf{z}^* = \arg\max_{\mathbf{z}}\ \gamma_{\mathbf{z}}\beta^{1/2}\sigma_{\hat{f}_t}(\mathbf{x}^*, \mathbf{z})$$

where $\gamma_{\mathbf{z}} = \tau_{\max}/\tau_{\mathbf{z}}$ weights the uncertainty by the relative computational cost, with $\tau_{\mathbf{z}}$ representing the average simulation time at fidelity $\mathbf{z}$ and $\tau_{\max}$ the maximum simulation time across fidelities.

Figure 3.1 demonstrates a DGP-based approach applied for the optimisation of a basic three dimensional reactor design with 5 discrete fidelities [138].

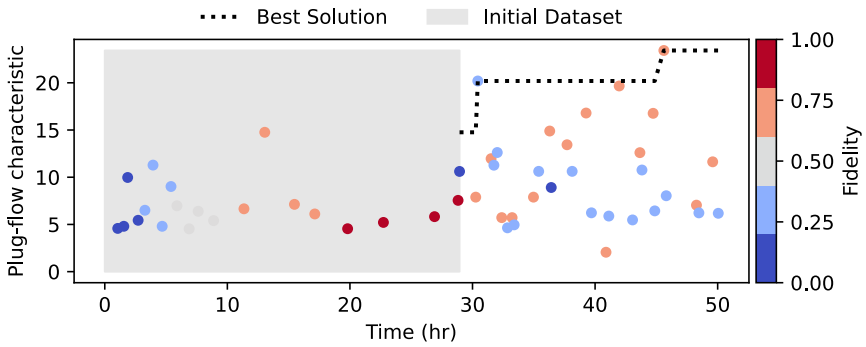



Fig. 3.1 The progression of multi-fidelity Bayesian optimisation using DGPs as a multi-fidelity model. Multiple different fidelity levels are selected for evaluation throughout optimisation.

Applying information from lower-fidelity simulations may accelerate convergence toward optimal solutions. However, as the number of discrete fidelities increases, the resulting multi-fidelity DGP model becomes increasingly complex and difficult to train. This can lead to longer inference times, slower exploration, and potentially less accurate descriptions of the design space. Empirical results suggest that limiting applications to just 2-3 discrete fidelities may be more beneficial in systems with no prior knowledge, despite more fidelities being theoretically available.

This limitation motivates an alternative approach: modelling fidelities as smoothly varying in a continuous space alongside inputs, using a joint space $\mathcal{Z} \times \mathcal{X}$ with a single Gaussian process. This enables continuous approximation of $\mathbb{Z}$ rather than limiting to discrete fidelities, potentially allowing for finer-grained control over the accuracy-cost trade-off. While this approach more naturally handles multiple independent fidelities, it typically assumes simulation cost is independent of $\mathbf{x}$, which may not hold in complex simulation environments.

Kandasamy et al. [65] assume that both inputs and fidelities are smoothly varying in a continuous space, modelling both within a joint space $\mathcal{Z} \times \mathcal{X}$, using a single Gaussian process. As a result the modelling approach serves as a continuous or continuous approximation of $\mathbb{Z}$, not limiting the setting to discrete fidelities. By modelling multi-fidelity data within a joint space, fidelities are able to influence the prediction of objective at other fidelities. A two-step approach is applied to select $\mathbf{x}_{t+1}$ and $\mathbf{z}_{t+1}$. First, decision variables are selected by performing a standard UCB step, conditioned at the highest fidelity. $\mathbf{z}_{t+1}$ is subsequently selected from a candidate set of continuous fidelities. Though the approach scales poorly with number of fidelities, and assumes the cost of a simulation at a given fidelity is independent of $\mathbf{x}$.

He et al. [53] assumes that the cost of a function evaluation is both a function of $\mathbf{x}$ and $\mathbf{z}$, presenting the case study of a finite-element thermal and fluid dynamics model of alloy casting with each simulation takes approximately 10 minutes to complete, with a single fidelity parameter and 2, 3 and 4 design variables. To solve the multi-fidelity optimisation problem, the authors present an augmented acquisition function, weighted by the cost of a simulation. Thodoroff et al. [159] presents a cost-adjusted criterion to select both $\mathbf{x}_t$ and $\mathbf{z}_t$ in a single step, in the context of ice-sheet simulation experimental design. $\mathbf{x}$ and $\mathbf{z}$ are modelled in a joint space by a single Gaussian process $\hat{f}(\mathbf{x}, \mathbf{z})$. Simulation cost is a separately modelled function of both $\mathbf{x}$ and $\mathbf{z}$, with $\mathbf{x}_t$ and $\mathbf{z}_t$ are chosen in a single step. Serani et al. [144] applies multi-fidelity optimisation within the CFD domain to benchmark problems including airfoil and boat hull design. Huang et al. [55] applied a Bayesian optimisation-based technique to a sequentially-trained multi-fidelity model to optimise a metal-forming finite-element simulation. March et al. [89] perform multi-fidelity optimisation of a typical airfoil design problem using a trust-region approach. Mansour et al. [88] optimise the geometry of a coiled tube reactor without pulsed-flow conditions using a genetic algorithm approach. The work assumes a single simulation fidelity, with a result taking over two weeks to produce.

## 3.3 Method

### 3.3.1 Parameterisation

The geometry of a standard helical-tube reactor can be altered by parametrising the coil radius, coil pitch, and inversion. Coil pitch denoted by $\phi$ controls how extended the helical tube is, coil radius denoted by $\rho$ controls how tight the coils are within the helical tube, and the inversion parameter is denoted by $\delta$ controls the change in coil direction. Inversions within helical-tube reactors have been shown to provide effective mixing properties by McDonough et al. [95], Rossi et al. [133], Singh and Nigam [148]; $\delta$ takes a value between 0 and 1, and specifies where along the coil the inversion takes place. The length of the coil is maintained as fixed, resulting in all parametrised coils having the same volume. Within the design of each parametrised reactor, a fixed-length inlet and outlet to the coil is included. The inlet and outlet are horizontal, and a smooth interpolation is used to ensure that the transition from inlet to coil and coil to outlet, is smooth. Figure 3.2 demonstrates the specific parameterisation that is applied within this chapter.

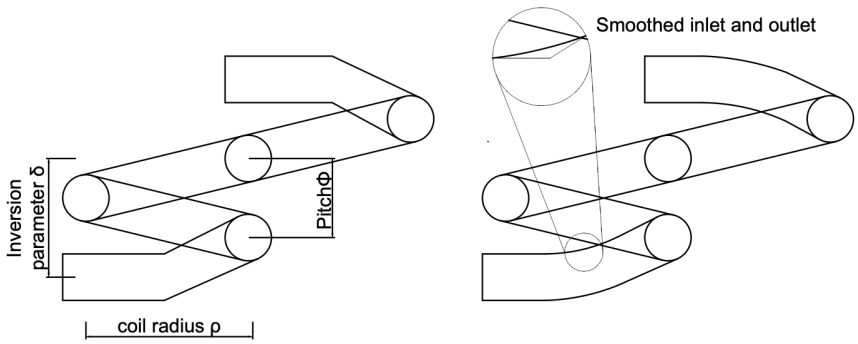



Fig. 3.2 *Left*: A side-view of how coil radius, pitch, and the inversion parameter affects a given coil, with an additional horizontal inlet and outlet. *Right*: When parameterising a coil with horizontal inlet and outlet, a smooth transition is included based on a quadratic interpolation of points.

The reactor inlet flow velocity is defined by fixing the Reynolds number at 50, for which relatively insignificant mixing is expected to take place. A superimposition of oscillatory

velocity is therefore needed to operate under a wide range of plug flow conditions [95]. This oscillatory velocity $v_o$ is achieved through parameters representing oscillation amplitude $a$ and frequency $f$. Oscillatory velocity is defined as

$$v_o = 2\pi f a \sin(2\pi f t).$$

Oscillatory parameters are jointly optimised with the design parameters throughout the optimisation procedure.

Additional information regarding pulsed-flow helical tube reactors can be located within Appendix Section A.1.

### 3.3.2 Fidelities

In addition to geometric design and oscillatory parameters, the output of a simulation is also influenced by one or more fidelities $\mathbf{z}$. A typical fidelity used within CFD simulations is the number of discrete finite elements that are contained within the mesh, known as cell count. Fluid behaviour within pulsed-flow helical-tube reactors is complex due to the transient formation and unravelling vortex structures affecting axial and radial mixing. Therefore, it is important to capture simulation resolution both axially, through the length of the tube, as well as radially throughout the cross-section of the tube for a high-quality simulation. In this chapter, as opposed to applying cell count as a single scalar fidelity parameter, the ability to adapt the resolution of a simulation both axially and radially is maintained, resulting in two independent fidelities. A custom meshing procedure is implemented that allows for axial and radial fidelity to be varied independently. The simulation that serves as the highest fidelity $\mathbf{z}_\bullet$ corresponds to when both the axial and radial fidelities are greatest. Whilst both axial and radial fidelities are discrete due to their meaning within a finite element context, they are

both treated as approximately continuous parameters, and during mesh creation the values of both fidelities are rounded to the nearest integer.

Figure 3.3 demonstrates the final mesh affected by axial fidelity, as well as providing an outline of how meshing is performed. Increased mesh density near the inlet and outlet wall is induced throughout all radial fidelity values to reduce numerical diffusion errors that can compromise the accuracy of the CFD solution. Figure 3.4 demonstrates how the final mesh is affected by radial fidelity, as well as providing an outline of the meshing procedure through the creation of a block-structured mesh. The mesh is refined near the walls to better capture the boundary layer and thereby velocity gradients affecting flow characteristics.

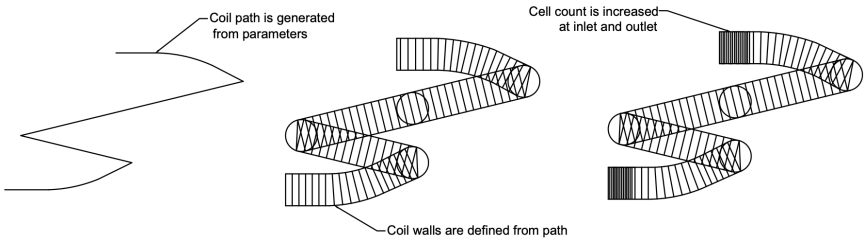



Fig. 3.3 A side-view of the meshing procedure of a coil based on axial fidelity. A central path is first defined based on geometric parameters. Subsequently, circles (tube cross-sections) are defined along this path the number of which depends on the axial fidelity. Finally, the mesh density near the inlet and outlet walls is increased by raising the number of circles to track the tracer concentration with higher accuracy.

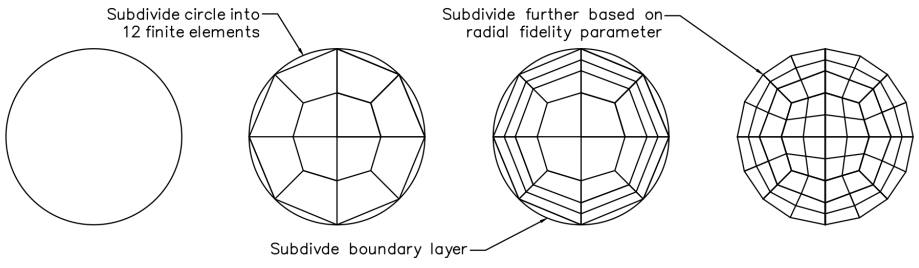



Fig. 3.4 A cross-section of the mesh radially. First, a coarse finite element topology is created. Subsequently, the number of cells near the tube wall are increased in order to better capture the effects of fluid flow near the boundary layer. Finally the mesh is further subdivided based on the radial fidelity value.

### 3.3.3 Simulation

To perform an evaluation of a given reactor mesh and set of operating conditions, a simulation is performed using the open-source code OpenFOAM [58]. A simulated impulse tracer is injected as a scalar field at the inlet of the reactor until $t = 0.15$ s into the water as a medium. The concentration of the tracer is tracked by solving a convection-diffusion equation through scalarTransportFoam. Here, the diffusion coefficient is set as a constant with a small value of $1 \times e^{-10}\ m^2/s$ as diffusion in liquids is slow and is dominated by convection. The pressure-velocity coupled, transient pimpleFOAM solver is used for solving the unsteady momentum equations as time-dependent oscillatory velocities are introduced. The pimpleFoam solver is integrated with the scalarTransportFoam through 'Solver function Objects'. The convection flux on the computational cells was calculated using second-order discretization schemes to ensure the numerical accuracy of the solution. The groovyBC boundary condition is used for imposing oscillatory velocity through swak4Foam library [49]. This oscillatory velocity along with the steady velocity was initialised at the inlet as Hagen-Poiseuille parabolic velocity profile to cut down on the coil length needed for the flow development and the computational cost. Additionally, the solution is terminated by monitoring the tracer concentration at the outlet; this is processed when the tracer concentration drops to values less than $1 \times e^{-7}$ for 10 consecutive iterations. This variable early-stopping criterion based on output accelerates the optimisation procedure by reducing the cost of simulations, unlike other studies where a fixed termination based on certain iterations is enforced [88], but introduces another time dependence on the simulation cost. This also creates an added layer of complexity, as the cost of simulation is additionally dependent on geometry and operating conditions $\mathbf{x}$, given a simulation may terminate early. This OpenFOAM solver is integrated with the optimisation algorithm via the PyFOAM Python library.

The output from a simulation returned from PyFOAM is a set of concentration values and respective times at the outlet of the reactor. This represents the residence time distribution of

the reactor. To convert this distribution to a single optimisation objective, the distribution is transformed to an equivalent number of tanks-in-series, $N$. A larger value of $N$ represents a 'tighter' residence time distribution and a stronger approximation to plug-flow behaviour, desirable to minimise downstream separation costs and process efficiency. First, the time and concentration values are converted to dimensionless quantities, $\theta$ and $E(\theta)$, respectively. Equation 3.3 shows how $E(\theta)$ can be represented as a function of $\theta$ and $N$ [95]:

$$E(\theta) = \frac{N(N\theta)^{N-1}}{(N-1)!} e^{-N\theta}. \tag{3.3}$$

A fitting procedure is performed to obtain a value of $N$ from $E(\theta)$ and $\theta$. This may be done based on least-squares error ($L_2$-norm), however, it is found that due to non-idealities and assumptions made to derive Eq. 3.3, using the difference in maximum predicted value and the maximum value returned from the simulation results in a more robust fit. The estimated tanks in series, $N^*$, is therefore calculated as

$$N^* = \arg\max_N \left| \max\left[E(\theta)\right] - \max\left[\frac{N(N\theta)^{N-1}}{(N-1)!} e^{-N\theta}\right] \right|. \tag{3.4}$$

Figure 3.5 demonstrates $E(\theta)$ against $\theta$ as derived from fitting values of $N$ using Eq. 4.4, for two sets of experimental data.

The case file used can be found within the code for this article. Though the current case considers the flow with two non-reactive, miscible liquids with identical fluid physical properties, this model can be extended to problems with fluids of differing physical properties modelled as multiphase flows or with reactions where chemical-kinetics can be coupled to CFD simulations. In these cases, it is anticipated that the objective function will become more complex, and potentially nonlinear, with convergence times increasing proportionately. Simulations resulting from more complex problems also become increasingly expensive as a variety of fluid effects will need to be captured. This in turn results in multi-fidelity methods

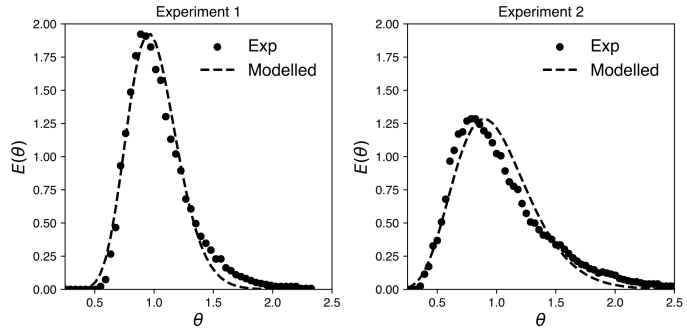



Fig. 3.5 Experimental residence time distribution against the residence time distribution modelled using the ideal tank-in-series relation, given by Eq. 3.3, with a value of equivalent tanks in series derived from Eq. 4.4. Configurations that perform worse, resulting in lower values of N (right), result in a broader residence time distribution, the maximum concentration occurring earlier, and with a lower value.

becoming more important, in order to enable tractable optimisation. As simulations gain complexity there also becomes more potential fidelities available during optimisation such as specific solver options, which include categorical and continuous fidelities, further ensuring the tractable optimisation in more complex cases. Finally, real-world experiments may be used as a distinct fidelity themselves alongside simulations.

### 3.3.4 Multi-fidelity Bayesian optimisation

In addition to dependence on simulation fidelity, the cost of a simulation is also dependent on $\mathbf{x}$ in chemical reactor simulation domains. Variable step-size CFD solvers, along with early stopping criteria both contribute to how $\mathbf{x}$ can influence simulation cost, particularly in cases where operating conditions are optimised over. Therefore, the two-step approach presented by Kandasamy et al. [65] cannot be applied, where $\mathbf{x}$ is selected first, and then $\mathbf{z}$, as the second-step where information and cost is traded off becomes dependent on $\mathbf{x}$. Therefore, the approach that is followed is most similar to He et al. [53] and Thodoroff et al. [159]; by utilising a cost-adjusted acquisition function to trade off information gain and cost in a single step. To do so, the optimisation objectives $\mathbf{y}$ and cost of simulations $\mathbf{c}$ are modeled by two

separate Gaussian processes, $\hat{f}$ and $\lambda$, respectively. This may be solved as a multi-objective optimisation, however to aid tractability and reduce the time to select the next simulation an approximation to this is used by combining the two objectives. The cost-adjusted acquisition function is as follows:

$$\mathbf{x}_{t+1}, \mathbf{z}_{t+1} = \underset{(\mathbf{x},\mathbf{z})\in\mathcal{X}\times\mathcal{Z}}{\operatorname{argmax}} \frac{\mu_{\hat{f}_t}(\mathbf{x}, \mathbf{z}_\bullet) + \beta^{1/2}\sigma_{\hat{f}_t}(\mathbf{x}, \mathbf{z}_\bullet)}{\mu_{\lambda_t}(\mathbf{x}, \mathbf{z})\sqrt{1 - k((\mathbf{x}, \mathbf{z}), (\mathbf{x}, \mathbf{z}_\bullet))^2}}. \tag{3.5}$$

The numerator on the right-hand side represents expected information gained at the highest fidelity since this is the system of interest ($\mathbf{z} = \mathbf{z}_\bullet$). The denominator then weights this expected information by the predicted cost of that simulation, enabling lower fidelity solutions to be selected. The term $\sqrt{1 - k((\mathbf{x}, \mathbf{z}), (\mathbf{x}, \mathbf{z}_\bullet))^2}$ quantifies how much information is lost when evaluating at a lower fidelity [65]. This information loss is based on the correlation between fidelities encoded within the kernel $k$; since correlation is inherently normalised, the overall signal variance ($\sigma_f^2$) is implicitly accounted for and does not appear explicitly. The parameter $\beta^{1/2}$ weights the exploration-exploitation trade-off as is standard in UCB Bayesian optimisation. However, with a cost-adjusted acquisition function, there is no guarantee that a high-fidelity simulation will be selected for evaluation. We address this in the following subsection.

### 3.3.5 Stopping criteria

In traditional Bayesian optimisation, function evaluations are terminated when the computational budget is exhausted, and the evaluated data point with the highest objective is selected as a final solution. In a multi-fidelity framework, this criterion contains subtle nuances. As the underlying function of interest is $f(\mathbf{x}, \mathbf{z}_\bullet)$, the optimal solution returned should be the highest evaluated function-value where $\mathbf{z} = \mathbf{z}_\bullet$. Throughout optimisation $f(\mathbf{x}, \mathbf{z}_\bullet)$ is maximised through lower fidelity simulations, with no guarantee that simulations at the highest

fidelity will be selected to be evaluated. To mitigate this detail in their two-step approach Kandasamy et al. [65] includes criteria to ensure that in some scenarios $\mathbf{z}_\bullet$ is selected in favour of a lower fidelity. It follows that when $f(\mathbf{x}, \mathbf{z}_\bullet)$ has been optimised 'as much as possible' and the most information has been gained, *but before all computational budget has been exhausted*, a function evaluation of the highest fidelity needs to take place, to ensure a valid final solution is returned. Therefore, the notion of a multi-fidelity stopping criteria for Bayesian optimisation is presented. The three key criteria for the return of a final highest-fidelity solution are as follows:

1. This solution need-not be generated by trading off mean and variance (as within a standard acquisition function), and therefore can be performed greedily given that it will be the final evaluation.

2. This solution should be the last solution selected, taking advantage of the maximum amount of information.

3. There must be enough time remaining to evaluate this solution.

Depending on the inclination of the decision-maker, the third criterion may be relaxed and a final solution may be returned without having been evaluated. However, it is assumed that a valid solution must have been evaluated on the true, high-fidelity function for which $\mathbf{z} = \mathbf{z}_\bullet$. At each iteration, this greedy, potentially final, solution is obtained by solving the following equation given by

$$\mathbf{x}_g = \arg\max_{\mathbf{x} \in \mathcal{X}} \mu_{\hat{f}_t}(\mathbf{x}, \mathbf{z}_\bullet). \tag{3.6}$$

where $\mathbf{x}_g$ is a solution that fulfils the first criterion for a final solution. In order to ensure that the second and third criteria are adhered to, the cost of evaluating $\mathbf{x}_g$, and the cost of

evaluating $\mathbf{x}_{t+1}$ are compared with the time or computational budget remaining. As the cost of evaluating a simulation is modelled using a Gaussian process, let us apply both the mean and the standard deviation resulting in a probabilistic upper-bound of each evaluation cost. Equation 3.7 details the maximum time required to evaluate both the next explorative solution as well as the greedy high-fidelity solution

$$c_{\max} = \underbrace{\mu_{\lambda_t}(\mathbf{x}_{t+1}, \mathbf{z}_{t+1}) + p_\lambda \sigma_{\lambda_t}(\mathbf{x}_{t+1}, \mathbf{z}_{t+1})}_{\substack{\text{Predicted maximum cost of} \\ \text{next 'standard' evaluation}}} + \overbrace{\mu_{\lambda_t}(\mathbf{x}_g, \mathbf{z}_\bullet) + p_\lambda \sigma_{\lambda_t}(\mathbf{x}_g, \mathbf{z}_\bullet))}^{\substack{\text{Predicted maximum cost of} \\ \text{high-fidelity 'greedy' evaluation}}} \tag{3.7}$$

where $p_\lambda$ is a parameter weighting the standard deviation, providing the ability to be more conservative and ensure greater probability $\mathbf{x}_g$ can be evaluated. If the value of $c_{\max}$ is greater than the remaining evaluation budget, then $(\mathbf{x}_g, \mathbf{z}_\bullet)$ should be evaluated as a final solution. Otherwise $(\mathbf{x}_{t+1}, \mathbf{z}_{t+1})$ should be evaluated as usual, as it is highly probable that there is enough time remaining for both a standard evaluation as well as a greedy evaluation.
In addition, the objective value returned from the solution of Eq. 3.6, $\mu_{\hat{f}}(\mathbf{x}_g, \mathbf{z}_\bullet)$, provides a proxy for monitoring the progress of multi-fidelity Bayesian optimisation. As function values across different fidelities cannot be compared like-for-like, it enables practitioners to observe the progress of optimisation. Algorithm 1 demonstrates the approach taken for multi-fidelity Bayesian optimisation.

## 3.4 Results

In order to provide a valid comparison between alternative reactor geometries, the length of a given helical coil tube is fixed as 75mm and the tube radius is fixed as 2.5mm. This ensures that all reactors within the design space have the same volume, and therefore equivalent number of tanks-in-series can be compared between solutions.

**Algorithm 1** Multi-fidelity Bayesian optimisation for Simulated Chemical Reactors

**Initialize:** Remaining budget $\Lambda$, Input bounds $\mathcal{X}$, Fidelity bounds $\mathcal{Z}$, objective function $f$, number of initial samples $D$, exploration parameter $\beta^{1/2}$, kernel function $k$.
$\mathcal{D} = \{(\mathbf{x}_i, \mathbf{z}_i, y_i, c_i)\}_{i=1}^{D}$ `// create initial data set through design of experiments`
**while** $\Lambda > 0$ **do**
  $\hat{f}_t \sim \{\mathcal{GP}(\mathbf{0}, k((\mathbf{x},\mathbf{z}),(\mathbf{x}',\mathbf{z}'))\,|\mathcal{D}\}$ `// train objective GP using` $\mathcal{D}$
  $\lambda_t \sim \{\mathcal{GP}(\mathbf{0}, k((\mathbf{x},\mathbf{z}),(\mathbf{x}',\mathbf{z}'))\,|\mathcal{D}\}$ `// train cost GP using` $\mathcal{D}$
  $\mathbf{x}_{t+1}, \mathbf{z}_{t+1} \leftarrow$ Eq. 3.5 `// solve cost-adjusted UCB`
  $\mathbf{x}_g \leftarrow$ Eq. 3.6 `// solve greedy high-fidelity problem`
  $c_{max} \leftarrow$ Eq. 3.7 `// estimate maximum budget for both evaluations`
  **if** $\Lambda > c_{max}$ **then**
    $y_{t+1}, c_{t+1} = f(\mathbf{x}_t, \mathbf{z}_t)$ `// evaluate cost-adjusted UCB solution`
  **else**
    $y_g, c_g = f(\mathbf{x}_g, \mathbf{z}_\bullet)$ `// evaluate greedy high-fidelity solution`
    $\mathbf{x}^* \leftarrow \mathbf{x}_g$
    $y^* \leftarrow y_g$
    **return** $\mathbf{x}^*, y^*$
  **end if**
  $\Lambda \leftarrow \Lambda - c_{t+1}$ `// update remaining budget`
  $\mathcal{D} \leftarrow \mathcal{D} \cup (\mathbf{x}_{t+1}, \mathbf{z}_{t+1}, y_{t+1}, c_{t+1})$ `// update data set`
**end while**

### 3.4.1 Parameterisation

Parameters are assigned reasonable bounds based on physically realisable values as well as bounds dictated by the constraints of the experimental setup. The parameters $\delta$, $\phi$, and $\rho$ are bounded between 0-1, 7.5-15mm, and 3-12.5mm, respectively, while the pulsed-flow amplitude, frequency, and the Reynolds number are bounded between 1-8mm, 2-8 Hz, and 10-50, respectively. These bounds ensure that tube self-intersections cannot occur resulting in an unconstrained problem. Figures 3.6, 3.7, 3.8 demonstrate the effect of $\delta$, $\phi$, and $\rho$ on the helical coil tube geometry, respectively. The mesh for a given set of parameters and fidelities is generated using a custom mesh generation scheme in Python using the `classy_blocks` library, available at https://github.com/OptiMaL-PSE-Lab/pulsed-reactor-optimisation/.

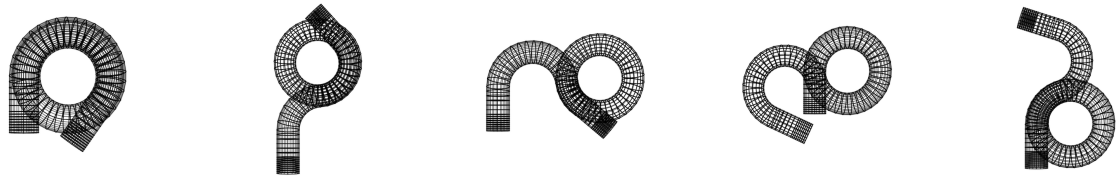


Fig. 3.6 The effect of inversion parameter $\delta$ for a helical coil tube with fixed length, top view. From left to right: $\delta$ is evaluated at 0, 0.15 0.3, 0.6, and 0.75 with coil radius and pitch remaining constant.

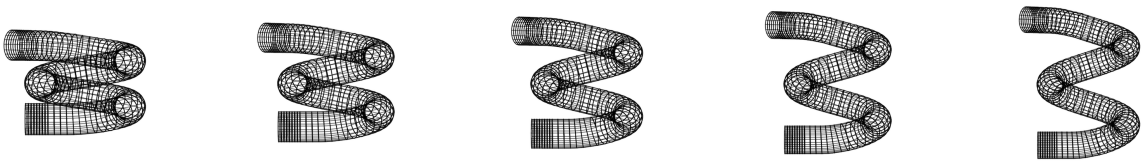


Fig. 3.7 The effect of pitch $\phi$ for a helical coil tube with a fixed length, side view. From left to right: $\phi$ is evaluated from 7.5mm to 15mm with coil radius and inversion location remaining constant.

The lowest axial and radial fidelity values are 1 and 20, and the corresponding highest fidelity values are 5 and 60, respectively. Figure 3.9 demonstrates the effect of both axial and radial fidelities on the final mesh, given a fixed set of design parameters.

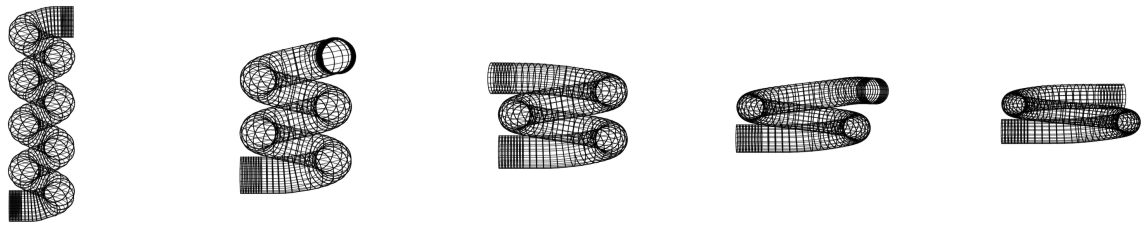
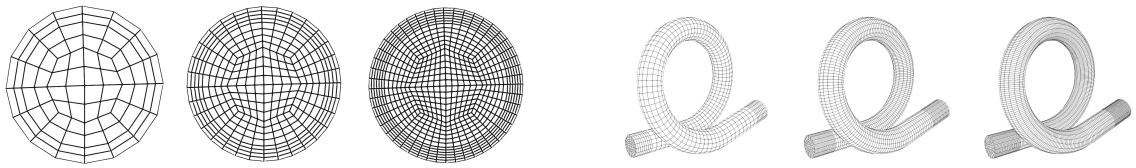


Fig. 3.8 The effect of coil radius $\rho$ for a helical coil tube with a fixed length, side view. From left to right: $\rho$ is evaluated from 3mm to 12.5mm with inversion location and pitch remaining constant.



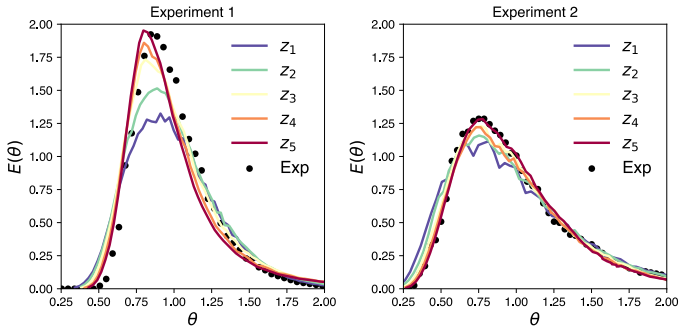
(a) *Radial cross section depicting radial fidelity at values of 1, 3 and 5.*

(b) *View of complete reactor mesh at axial fidelity values of 20, 40, and 60.*

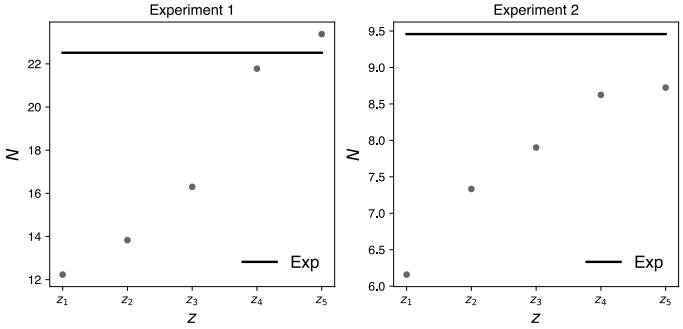
Fig. 3.9 An instance of helical-tube reactor geometry as affected by axial and radial fidelity.

### 3.4.2 Model Validation

In order to gain insight into the effect of a simulation fidelity on the accuracy of $f(\mathbf{x},\mathbf{z})$ for a fixed geometry and operating conditions, simulations with five discrete fidelities were performed for fixed $\mathbf{x}$ and compared with experimental data. The set of fidelities validated are $\{(z_{\text{axial}}=20, z_{\text{radial}}=1), (z_{\text{axial}}=30, z_{\text{radial}}=2), (z_{\text{axial}}=40, z_{\text{radial}}=3), (z_{\text{axial}}=50, z_{\text{radial}}=4), (z_{\text{axial}}=60, z_{\text{radial}}=5)\}$. Figure 3.10 demonstrates the tracer concentration profile of simulations using these five selected fidelity combinations against two sets of experimental data. The experimental data were generated from a 3D-printed reactor with a length of 75.3mm, coil radius of 12mm, coil pitch of 10mm, and no inversion. Pulsed-flow operating conditions were induced with a frequency of 5 Hz, the Reynolds number was equal to 50, and the two experiments differed by applying a flow amplitude of 2mm and 4mm. An experimental value of equivalent tanks-in-series, denoted by $\hat{N}$, was calculated for both experiments from each resulting residence-time distribution.




(a) The outlet concentration profile of a tracer impulse at five combinations of fidelity ranging from lowest ($\mathbf{z}_1$) to highest ($\mathbf{z}_5$) against experimental data. $E(\theta)$ is dimensionless concentration as a function of dimensionless time $\theta$.




(b) The value of $N$ corresponding to the concentration profile from each fidelity, for each of the five selected fidelity combinations. As overall fidelity is increased from $\mathbf{z}_1$ to $\mathbf{z}_5$, the experimentally derived value for $N$, $\hat{N}$ is recovered.

Fig. 3.10 Validation of five discrete mesh fidelities corresponding to different cell counts, across two sets of experimental data under different conditions.[3]

Figure 3.10 also demonstrates how increasing overall fidelity (and therefore cell count) results in a closer approximation to the experimental value of $N$, derived from each concentration profile. For both experiments, the nature with which $N$ approaches $\hat{N}$ when the overall fidelity is increased implies that $f(\cdot,\mathbf{z})$ is smoothly varying. The information that fidelity smoothly influences $f$ validates the assumption that $\mathbf{z} \in \mathcal{Z}$ varies smoothly and continuously enabling a Gaussian process to be applied as a multi-fidelity model.

According to the first steps of Algorithm 1, an initial data set of solutions $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{z}_i, y_i, c_i)\}_{i=1}^{D}$ is first created using a static Latin hypercube design of experiments scheme [97] to maximise the initial information gained, with 25 solutions ($D = 25$). This initial number represents a

hyper-parameter. An alternative approach for initialisation would be to generate promising solutions as defined by an expert. This is explored in later chapters of this thesis, but in this case to ensure the success of modelling the search-space using Gaussian processes, and not to bias the search, a space-filling design was chosen.

All inputs and outputs are normalised to have zero mean and a standard deviation of one at the beginning of each iteration. Before solutions are evaluated, inputs are re-scaled back to their physical quantities. Initially, the hyper-parameter $\beta$ within Eq. 3.5 is set as $\beta = 2.5$, and in Eq. 3.7 $p_\lambda = 2$ resulting in a maximum cost prediction with 95% confidence. The effect of applying different hyper-parameters on the solution is investigated in the Appendix Section A.2. The maximum budget is specified as 64 hours of wall-clock time, not including the budget spent sampling initial solutions. Figure 3.11 shows the optimisation progress throughout the 64 hours. The time to generate, and objective values of initial solutions generated are denoted by using negative iteration and wall-clock time values. Therefore, optimisation itself is denoted as starting at iteration 0, and at a wall-clock time of 0.

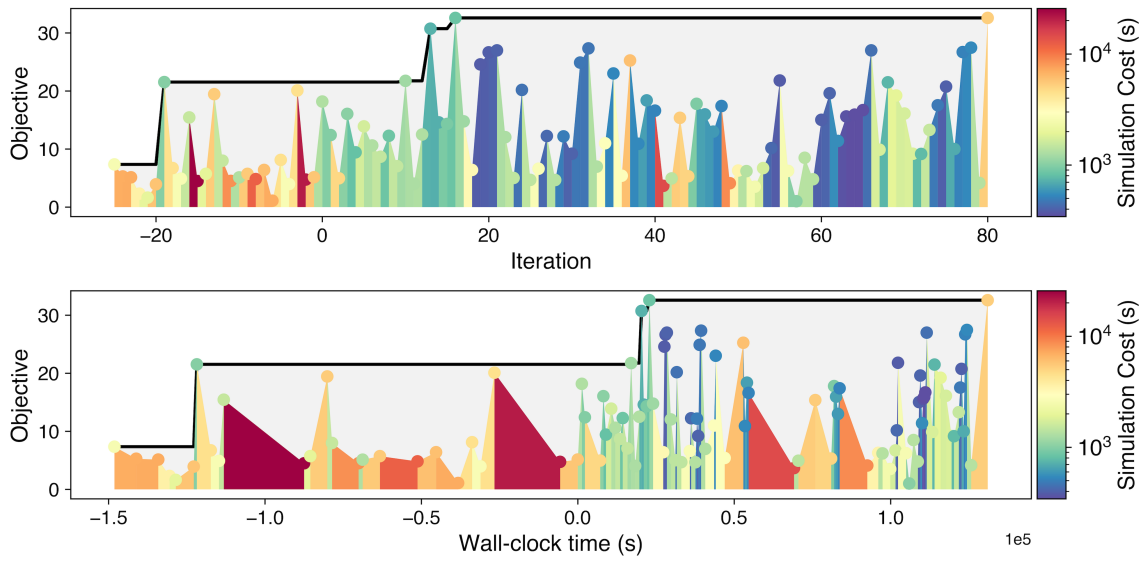



Fig. 3.11 The number of equivalent tanks-in-series evaluated labelled by the respective cost of simulation. The upper half of the figure shows these quantities against iteration and the lower half shows these quantities against wall-clock time, highlighting the importance of lower-cost simulations.

Simulations performed to generate the initial data set of solutions are plotted, and are designated to end at iteration and time 0. The initial data set generated provides an insight into the performance of a 'random' reactor from the design space, with the maximum performance across fidelities at around 20 tanks-in-series. The optimal design represents approximately a 1/3 improvement on this value. When optimisation begins, the search space can be seen to be explored through a number of lower-cost simulations. Figure 3.11 highlights the cost of performing simulations at a high fidelity, in some cases 6-7 hours. Approximately 20 high-fidelity points would be able to be generated in the same time; in the 6-dimensional search space this would be deemed intractable for optimisation for the given time-budget (a very general rule of thumb is $10d$ data-points are required for space-filling a $d$ dimensional space), motivating the multi-fidelity solution approach. As the value of the objective function at different fidelities is assumed to be biased, the objective values cannot be compared like-for-like. However, Fig. 3.11 provides insight into the progression of the algorithm.

Figure 3.12 shows the combinations of axial and radial fidelity evaluated throughout optimisation, alongside respective simulation costs. Generally, simulations performed with lower radial and axial fidelities have lower costs. However, as this relationship is also dependent on reactor geometry and operating conditions, some simulations have higher costs at lower fidelities. The framework tends to evaluate simulations at the lower half of axial fidelity at all five discrete radial fidelity values. This indicates that simulations with different axial fidelities are more correlated than simulations with differing radial fidelities. Less variance across the axial fidelity will result in the acquisition function favouring lower axial fidelity values, as information about the highest fidelity can be gained due to the higher correlation. The final high-fidelity solution can be seen at the component-wise highest-fidelity at the top-right of Fig 3.12. The quantities tracked throughout optimisation in order to ensure this solution is returned are demonstrated in Fig 3.13.

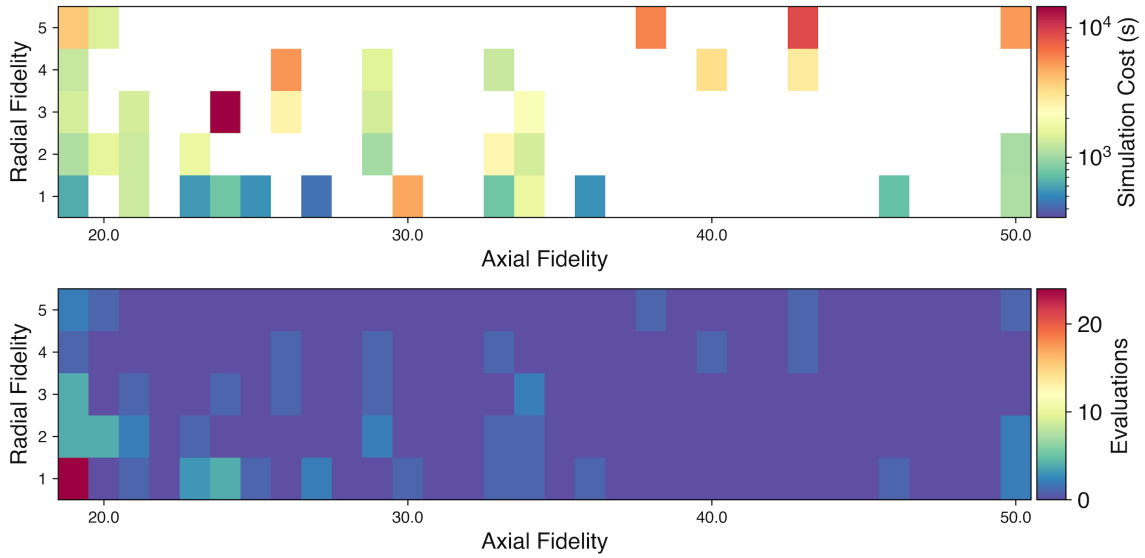



Fig. 3.12 The fidelities selected throughout optimisation within $\mathbb{Z}$. The upper plot demonstrates the average simulation cost at each fidelity. As discrete fidelities are applied which are continuously approximated, the lower plot is presented demonstrating the number of times each discrete fidelity is evaluated. Simulation cost is confirmed to be not only a function of fidelity, but $\mathbf{x}$ as well, as there is no clear distribution of simulation costs across $\mathbb{Z}$.

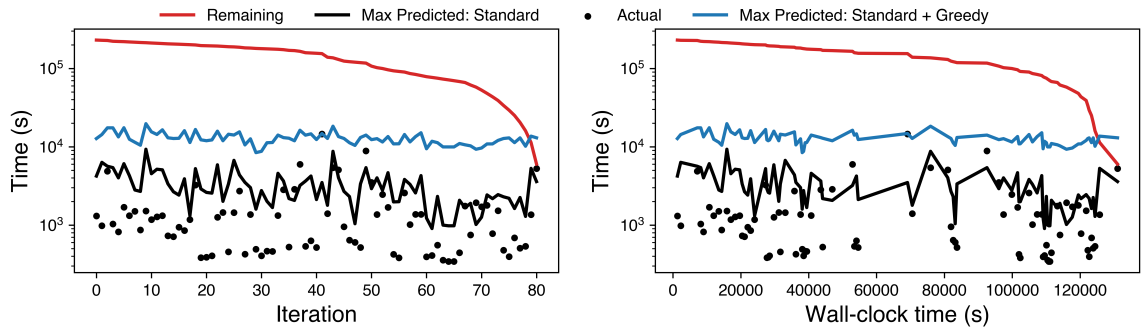



Fig. 3.13 The maximum predicted time to complete a 'standard' iteration compared with the maximum predicted time to complete both a standard iteration as well as a greedy iteration, the actual simulation time, and the remaining time. As the blue line passes above the black line, a greedy iteration must be performed as there is not enough remaining budget to perform both a greedy and standard evaluation

The actual evaluated costs of each simulation generally fall beneath their predicted maximum cost, with the exception of a few simulations. The Gaussian process that models simulation cost can be validated as being accurate as a result. When the time remaining falls beneath the maximum predicted time to perform a standard iteration as well as a greedy final iteration,

then the greedy solution is evaluated and the algorithm terminates. The high-fidelity solution returned at the end of optimisation is demonstrated in Figure 3.14. The optimal coil geometry has a pitch of 1.04cm, radius of 1.25cm, and an inversion that occurs 66% along the coil length. The associated optimal operating conditions are pulsed-flow with an amplitude of 1mm, frequency of 2 Hz and a Reynolds number of 50.

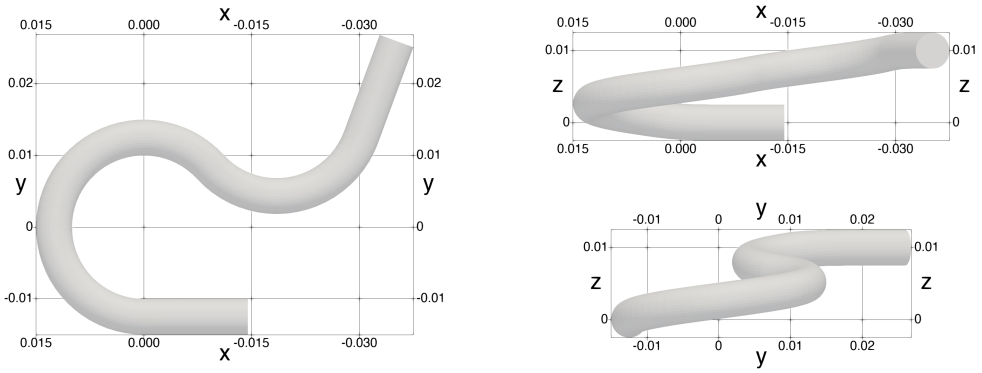



Fig. 3.14 Optimal reactor geometry viewed from different perspectives. The optimal geometry contains an inversion.

The alternative optimal solutions generated via different algorithm hyper-parameters (Section A.2), also contain similar characteristics (broad coil radius, similar operating conditions). Alongside analysis presented by Basha et al. [8], which demonstrates a uni-model space of pulsed-flow operating conditions with a single optimum, it is hypothesised that this specific function space is uni-model and more sensitive to operating conditions than geometry around the optimum.

Figure 3.16 highlights the specific flow behaviour within the optimal reactor throughout a single oscillatory cycle. Streamlines are coloured with tracer concentration to show the movement through the coil. For the positive part of the oscillatory cycle, the tracer moves in the forward direction in a streamlined manner towards the outlet. Specific features start to occur in the negative part of the oscillation cycle where the secondary flow begins to emerge. Cross-sectional flow streamlines transitioning to Dean-type vortices due to the centrifugal forces at the coil turns are shown. These counter-rotating Dean vortices promote radial

mixing of the tracer with the water medium and close to the walls of the computational domain; no tracer dead zones are left behind. Additionally, along with the reverse flow, the swirling motion developed during the negative oscillation cycle redirects the flow in the tangential direction which limits the axial dispersion of the tracer. This combination of promoting radial mixing and inhibiting axial mixing results in high plug flow performance. This flow cycle is periodic in nature for the amplitude of 1 mm and frequency of 2 Hz, and the simulation is continued until the tracer leaves the computational domain.

In the optimal configuration shown in Figs. 3.14 and 3.16, the inversion just before the outlet helps in maintaining the Dean vortex structure until the flow leaves the domain. Whereas, in a standard configuration, the Dean vortices closer to the outlet could be disappearing gradually. Figure 3.15 demonstrates the difference in Dean vortices resulting from a standard coil with the optimal coil containing an inversion. In addition, the inversion helps in the release of

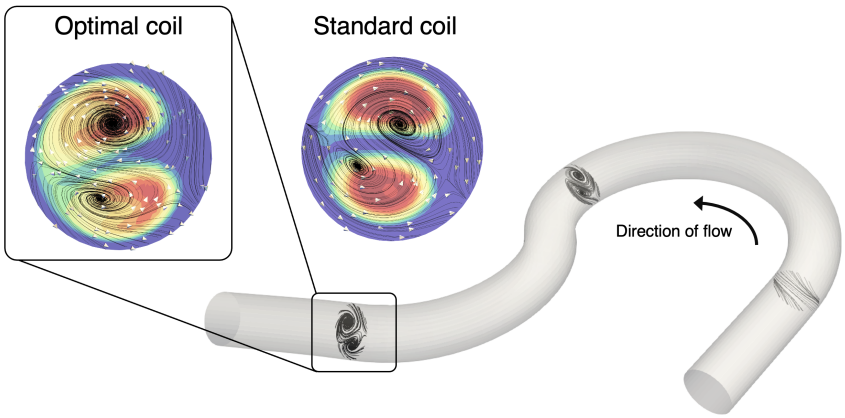



Fig. 3.15 Dean vortices visualised throughout the length of the optimal reactor with the vortices near the outlet of the reactor seen to be stronger than for those associated with a standard, sub-optimal coil configuration.

a highly concentrated tracer that is trapped between the vortices to undergo further radial mixing. Therefore, enhanced radial mixing is achieved with this configuration due to the inversion, leading to an optimal plug flow performance.

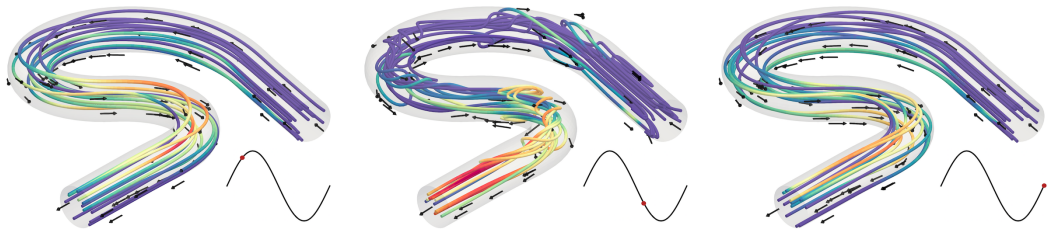
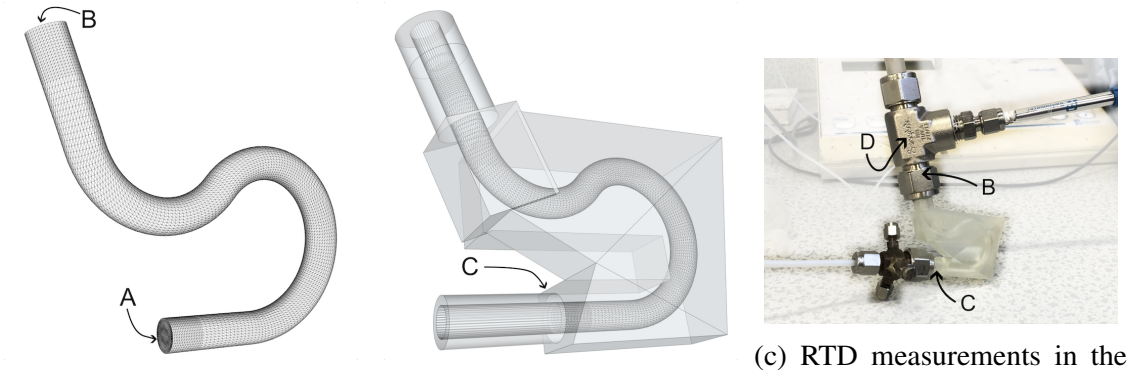


Fig. 3.16 Streamlines indicating tracer concentration within the optimal reactor throughout a single oscillatory pulse, lasting 0.5 seconds. As the oscillatory velocity changes from positive to negative, Dean's vortices form providing mixing throughout the radial direction. The inversion within the coil also contributes to these vortices shifting across the radial direction, providing stronger mixing characteristics. Tracer concentration ranges from 0 (purple) to 1 (red).

### 3.4.3 Solution Validation

The optimal reactor geometry was exported into an STL file format and modified into a 3D printable model by adding a bounding box to give the part volume. The straight sections at the inlet and outlet were given 8 mm and 10 mm OD tube fittings for connection to the experiment apparatus, and the bounding box was then trimmed to reduce the amount of resin needed for the print. The inlet region was also extended by 20 mm to provide additional development length to ensure the experiment matched as closely as possible the parabolic inlet velocity used in the simulations. The optimal geometry was printed using a FormLabs Form3+ using the Clear V4 resin using the default settings. Post-processing involved washing in IPA for 20 min, drying for 24 hours, and post-curing in a FormCure at 60°C for 30 min. Figure 3.17 shows the raw STL file, modified geometry, and printed geometry.

The same RTD method was implemented as reported by McDonough et al. [95]. In brief, RTDs were measured by injecting a 0.1 M KCl aqueous tracer solution into the geometry and measuring the conductivity over time at the outlet. The net flow of deionized water, oscillations, and tracer injection were controlled using three separate OEM syringe pumps (C3000, TriContinent) that were hydraulically linked to the reactor via PTFE tubing routed through a custom Swagelok piece (shown in Figure 3.17c). To minimise the influence of




(a) Raw STL geometry (b) Modified STL geometry (c) RTD measurements in the 3D-printed geometry.

Fig. 3.17 Experimental validation of optimal reactor configuration via additive manufacturing and operation under pulsed-flow operating conditions.

poor mixing inside this Swagelok fitting, the tracer was injected at point C (Figure 3.17) after the additional 20 mm inlet straight section by routing the PTFE tubing through the fitting (see McDonough et al. [95] for further clarity).

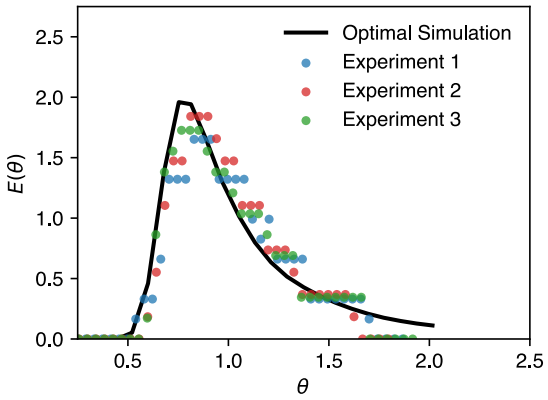



Fig. 3.18 The residence-time distribution predicted via CFD simulation of the optimal, high-fidelity solution returned from the framework, alongside 3 sets of experimental data obtained via the 3D printed reactor.

Figure 3.18 demonstrates the predicted residence-time distribution alongside 3 sets of experimental data obtained from the 3D printed optimal reactor and operating conditions. It can be conjectured that the RTD method is another example of a lower-fidelity approximation to the desired mixing performance. First, the printed geometry is not perfectly smooth due to the well-known stair-stepping effect; this enhanced roughness (wall friction) will differ around

the coil due to variations in the orientation. Second, the conductivity probe was located slightly downstream of the outlet face used in the simulations (point D in Figure 3.17c). For reference, in the simulation the tracer was injected at point A and measured at point B (Figure 3.17a). Nevertheless, the simulated optimal RTD closely matches the 'expected' distribution measured in the experiments across all three replicates, indicating good accuracy of the CFD simulations used to inform the optimisation process. This provides confidence that the output of the optimisation process is meaningful for a reactor geometry operating in a real-world setting.

## 3.5 Chapter Conclusions and Future Work

In this chapter, I have presented a methodology for the identification of optimal pulsed-flow operating conditions and geometry of a helical-tube reactor, framing the problem as a multi-fidelity black-box optimisation problem. I have demonstrated a general framework that takes advantage of different quality simulations to enhance the optimisation of reactor simulations via a multi-fidelity Bayesian optimisation. The framework is validated by applying it to optimise a helical-tube reactor geometry and operating conditions. By motivating the framework with the specific goal of returning a high-fidelity solution, I derive a new criterion for monitoring the progress and dictating the termination of multi-fidelity Bayesian optimisation. I show that this criterion enables a high-quality final solution to be returned before the computational budget is exhausted. The optimal reactor geometry and operating conditions is then 3D printed and the performance is experimentally validated. This approach is extensible to a large number of simulation-based design problems. Work explored in future chapters will serve to apply the framework to investigate alternative reactor parameterisations and expanded classes of reactors, making the leap from optimisation to discovery. Within the solution to more complex CFD problems, a number of solver options may be considered as valid fidelities. Particularly, with respect to investigating which of these fidelities are

important for the purposes of multi-fidelity optimisation of reactor and tube simulations. In addition, different performance indicators may be optimised to investigate the differences between reactor parameterisations optimised for different quantities such as pressure drop.

# Chapter 4

# Machine Learning-assisted Discovery of Flow Reactor Designs

## Contents


**4.1 Introduction . . . . . 99**

**4.2 Methods . . . . . 101**

4.2.1 Parameterisations . . . . . 101

4.2.2 Simulation . . . . . 106

4.2.3 Optimisation . . . . . 107

4.2.4 Experimental Validation . . . . . 108

**4.3 Results . . . . . 109**

4.3.1 Effect of Design Features on Flow Structures . . . . . 110

4.3.2 Convergence Analysis . . . . . 114

4.3.3 Optimal Designs & Comparison with 3D-Printed Reactors . . . . 118

**4.4 Discussion . . . . . 121**

# Publication Note

This chapter has been published as:

Savage, T., Basha, N., McDonough, J., Krassowski, J., Matar, O., & del Rio Chanona, E. A. (2024). *Machine learning-assisted discovery of flow reactor designs.* In Nature Chemical Engineering (Vol. 1, Issue 8, pp. 522–531). Springer Science and Business Media LLC. https://doi.org/10.1038/s44286-024-00099-1

The work presented in this chapter explores additive manufacturing, machine learning, and reactor design. By combining high dimensional parameterisations with previous work on multi-fidelity Bayesian optimisation and computational fluid dynamics, this chapter demonstrates how complex 3D reactor geometries can be discovered, leading to experimental performance improvements of  60% over conventional designs and the discovery of new driving fluid flows.

## Author Contributions

- Tom Savage: Study design and execution, reactor design, parameterisation, meshing, optimization, and manuscript preparation
- Dr Nausheen Basha: Study design and execution, CFD experiments, and manuscript preparation
- Dr Jonathan McDonough: Residence time experiments, reacting flow experiments, and manuscript review
- James Krassowski: Residence time experiments and reacting flow experiments
- Prof Omar Matar: Project supervision, conceptual guidance, and manuscript preparation

- Dr Ehecatl Antonio del Rio Chanona: Project supervision, conceptual guidance, and manuscript preparation

## Modifications from Published Version

The following modifications have been made to the published version:

- Some language has been altered to better fit within the context of this thesis.

# Abstract

Additive manufacturing has enabled the fabrication of advanced reactor geometries, permitting larger, more complex design spaces. Identifying promising configurations within such spaces presents a significant challenge for current approaches. Furthermore, existing parameterisations of reactor geometries are low-dimensional with expensive optimisation limiting more complex solutions. To address this challenge, this chapter establishes a machine learning-assisted approach for the design of new chemical reactors, combining the application of high-dimensional parameterisations, computational fluid dynamics, and multi-fidelity Bayesian optimisation. The development of mixing-enhancing vortical flow structures in coiled reactors is associated with performance, and the approach is used to identify key characteristics of optimal designs. By appealing to the principles of flow dynamics, design features are identified that lead to experimental plug flow performance improvements of $\sim 60\%$ over conventional designs. The results demonstrate that coupling advanced manufacturing techniques with 'augmented-intelligence' approaches can give rise to reactor designs with superior performance.


## 4.1 Introduction

Advances in additive manufacturing have enabled the fabrication of a wide range of complex and potentially counter-intuitive reactor designs. Previously infeasible, or highly impractical designs can now be manufactured and investigated, resulting in substantially larger design

spaces. Coupled with data-driven design tools, such as multi-fidelity Bayesian approaches as described in the previous chapter [140, 82, 53, 38, 158, 76, 98], which have enabled the optimisation of large-scale simulation-based problems, a pathway has emerged towards the identification of optimal reactors from larger design spaces with the potential for improved performance. By exploiting lower-fidelity simulations throughout optimisation, high-quality solutions can be generated in a significantly reduced time; this is particularly the case in scenarios where gradients are unavailable or a more global solution is desired than gradient-based approaches [140, 158].

In this chapter I demonstrate an 'augmented-intelligence' framework that enables the design of reactors that surpass the performance of previously outlined designs by exploiting high dimensional parameterisations. I build upon developments in data-driven optimisation and machine learning that I have outlined in previous chapters as well applying computational fluid dynamics and additive manufacturing, extending the coiled-tube reactor resulting in a shift in focus from optimisation to discovery.

Coiled-tube reactors have received attention across chemical engineering due to their desirable mixing and heat transfer characteristics [94, 95, 2, 61, 171, 126]. Their applications range from flow chemistry [124] and bioprocesses [51], to chemical kinetic experiments [61]. At the mesoscale, coiled-tube reactors have been shown to combine the heat and mass transfer properties of microreactors with the economic benefits of high-throughput larger scale reactors [31, 46]. Previous work, including that presented in the previous chapter has revealed improvement in the plug flow performance of coiled-tube reactors at relatively low flow rates (characterised by Reynolds numbers $Re \leq 50$) by super-imposing pulsed-flow operating conditions, which induce Dean vortices [8, 95, 106] that enhance radial mixing. These vortices arise due to centrifugal forces that develop in a curved pipe which shift the location of maximal velocity towards the outer pipe wall, with a return flow towards the inner curved wall leading to the establishment of counter-rotating cells. In the absence of

pulsed flow, such vortices develop at much higher flow rates and Dean numbers ($Re > 300$, $De > 75$) [27, 41, 81]. It is essential to induce the formation of Dean vortices at low flow rates and Dean numbers, over large proportions of the reactor interior, under steady-state flow, and importantly without the added overhead of pulsed-flow forcing.

To do so, this chapter identifies two parameterisations for the reactor geometry in the radial and axial directions. The simulation-based optimisation problem is solved for all parameterisations using the multi-fidelity Bayesian optimisation formulation applied in the previous chapter, however this time considering steady-state flow conditions at a low flow rate (with $Re = 50$ for which pulsed-flow would have been necessary to drive vortex formation). The composite objective maximised consists of plug flow performance, which is approximated from computational residence-time distributions using a tanks-in-series model, and a non-ideality term that penalises bimodal or unsymmetrical distributions. Optimal solutions are investigated that identify the presence of fully-developed Dean vortices at low Reynolds numbers under steady-state flow conditions. The key driving factors that result in improved performance are identified from which two novel reactors are designed and presented. Final designs are 3D-printed and experimentally validated, confirming their improved performance over a conventional coiled-tube reactor across both tracer and reacting flow experiments.

## 4.2 Methods

### 4.2.1 Parameterisations

In this section, I outline the design of efficient but flexible parameterisations for the discovery of novel coiled-tube reactors, ensuring tractability of the design problem. One where the geometry of the cross-section varies along the length of the reactor, and another where the reactor path itself is allowed to vary. To manage the flexibility/viability trade-off, both

parameterisations themselves are defined by hyper-parameters determining their complexity. To ensure smooth transitions in these parameterisations, interpolating points are used in their formulation, both to define the reactor path, and the cross-section throughout the reactor length. Gaussian processes are used in polar coordinates as a means to generate variable tube cross-sections from interpolating points.

### Polar Gaussian Processes

To define the variable tube cross-sections, it is necessary to generate a smooth and continuous closed curve from a set of interpolating points. A standard Gaussian process is not inherently suited to this task, as its kernel functions typically measure distance in a Euclidean space. For a circular cross-section, this presents a problem: points that are physically adjacent but separated by a full rotation (e.g., at angles 0 and $2\pi$) would be treated as distant, creating a discontinuity where the curve should seamlessly join. To correctly model this periodic domain and ensure a smooth, closed shape, a special class of kernel is required.

Recall from the background chapter that the squared-exponential kernel function assigns decreasing correlation between locations in input space with increasing Euclidian distance, providing an intuitive interpretation for the majority of regression settings. Other kernel functions have been proposed to provide valid covariance matrices, resulting in Gaussian process prior distributions with different properties. The polar squared exponential kernel enables valid covariance matrices to be constructed in polar coordinates and was outlined by [112]. A standard kernel function, dealing with data within data in polar coordinates, will determine that data at $\theta_1 = 0$ and $\theta_2 = \pi/2$ are highly uncorrelated. This is untrue, and in the presence of noiseless-observations these two data points should have perfect correlation ($k(\theta_1, \theta_2) = 1$). Polar kernel functions enable smooth interpolation in polar coordinates by including the ability for proper distances and respective covariances to be calculated between

any two angles [120, 112]. The polar covariance function is written as

$$k(\theta,\theta') = \left| \left(1 + \tau \frac{d(\theta,\theta')}{\pi}\right) \left(1 - \frac{d(\theta,\theta')}{\pi}\right)^{\tau} \right| \quad \tau \geq 4. \tag{4.1}$$

The angular distance metric $d$ is given as

$$d(\theta,\theta') = |(\theta - \theta' + \pi) \mod 2\pi - \pi|,$$

where $\tau$ is a hyper-parameter analogous to length-scale and controls how smooth the prior distribution of functions is. Figure 4.1 demonstrates samples from a Gaussian process prior with polar kernel, as well as a Gaussian process with polar kernel posterior distribution conditioned on data.

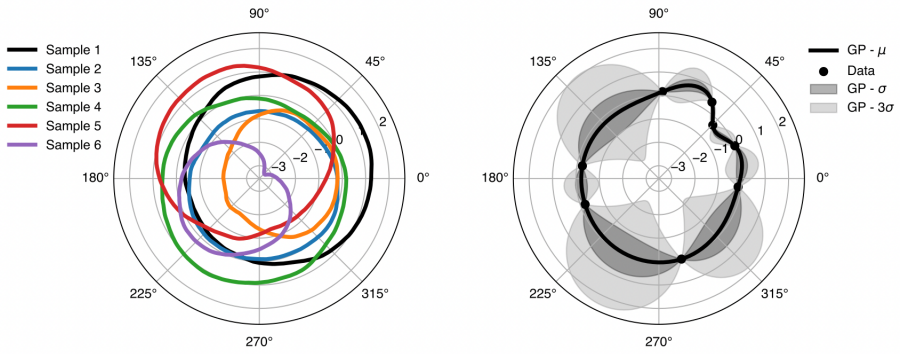



Fig. 4.1 **Left**: Polar Gaussian process parameterisation of coil cross section, demonstrating samples from a Gaussian process prior with a polar kernel. **Right**: The posterior distribution after the polar Prior distribution has been conditioned on data.

### Coil cross-section Parameterisation

Initially, the the number of interpolating points is defined for a given tubular cross-section, denoted $n_c$, and indexed by $i$. These points are then distributed equally along the angle $\theta$ in polar coordinates. The radius coordinate, $r_i \in n_c$ for each inducing point serves as the decision variable, or parameter. Given a set of inducing points corresponding to a specific

cross-section, a polar Gaussian process is used to interpolate between them, resulting in a valid, continuous curve in polar coordinates defining an individual cross-section. Next, the number of interpolating cross-sections denoted $n_l$ and indexed by $j$ is established, equally spaced along the length of the coil. Both $n_c$ and $n_l$ represent hyper-parameters; increasing either one raises the dimensionality of the resulting design problem, whilst resulting in a more flexible parameterisation. Figure 4.2 demonstrates the polar GP interpolation of each specific cross-section along the length of the coil, with $n_c = 6$ and $n_l = 6$. To ensure compatibility with existing fixtures and fittings, two additional cross-sections are defined at the beginning and end of the coil, with constant radius.

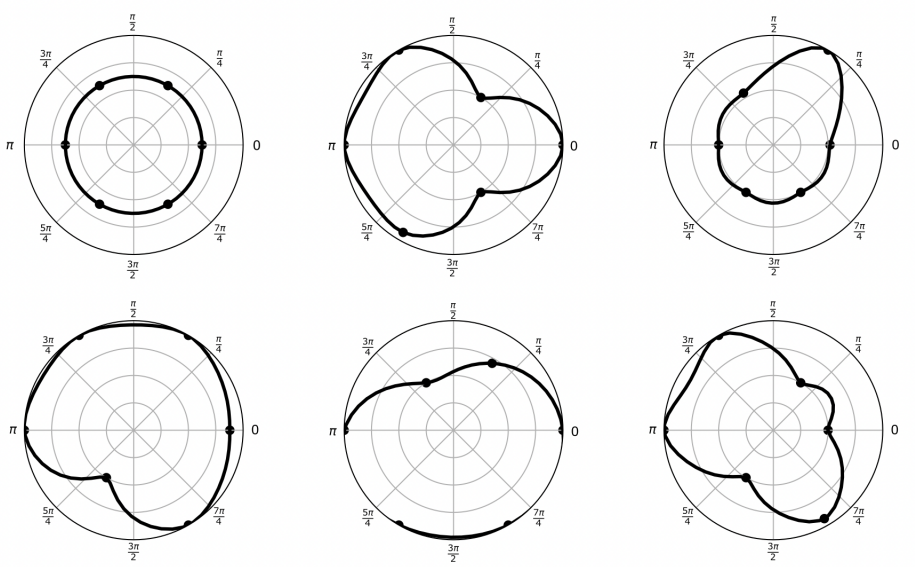


Fig. 4.2 Examples of sets of inducing points at given locations along the length of a coil. A polar Gaussian process is used to interpolate between points, where each data point's radial value is a parameter.

Subsequently, each cross-section is placed along the defined reactor path in 3D space and rotate each cross-section to face perpendicular to the direction of flow. The resulting interpolated cross-sections are defined by a quadratic interpolation between each polar GP posterior, in cylindrical coordinates. Finally, an inlet and outlet is added to the reactor by extending the cross-sections at the beginning and end of the reactor.

Importantly, the meshing procedure enables control over the number of cells throughout the axial direction of the reactor (the direction of fluid flow), as well as the radial direction. Defined by axial and radial fidelity parameters respectively, both values dictate the factor to which blocks are subdivided during mesh creation. By maintaining the ability to adapt the fidelity of the mesh in two independent directions, greater scope is provided for identifying efficient (regarding information gained on a computational expense basis) simulations. The previous chapter has demonstrated how the output of coiled-tube reactor simulations varies approximately smoothly with changes in fidelities [138, 140]. In this chapter it is assumed that they are continuous and round them to the nearest integer when stored or evaluated.

### Coil Path Parameterisation

To achieve a parameterisation of the coil path, a baseline path is defined in cylindrical coordinates, $(\rho_0, \phi_0, z_0)$, note that $z$ is overloaded in this context to refer to the height coordinate. This path is based on a standard coil configuration and serves as a reference for introducing induced deviations. Let $\Delta\rho_j$ and $\Delta z_j$ be the deviations in the radial and height directions, respectively, for each inducing point $j$. In optimising for inducing point deviations, the non-intersection constraint is ensured by design.

Decision variables are therefore the mismatch terms to the baseline path, resulting in a parameterised path given by $(\rho_j, \phi_j, z_j) = (\rho_0 + \Delta\rho_j, \phi_0, z_0 + \Delta z_j)$. After defining the path, circles are placed defining the constant cross-section in 3D space parallel to the direction of flow. Let $n_p$ be the number of inducing points along the coil path, indexed by $j$. This parameterisation is similar to the one presented by [24]. 6 inducing points are used for both the axial direction and radial direction, i.e. $n_p = 6$, however none in the rotational axis $\theta$. Deviations in $\theta$ may be included however it is found that this often results in self-intersections.

Further details regarding the implementation of all parameterisations can be found within Appendix section B.1.

#### Nominal Reactor and Parameter Bounds

A nominal reactor is specified which defines the baseline path of which is used across both parameterisations. Based on results from the proceeding chapter, the standard coiled-tube reactor performs optimally with a low pitch, and large coil radius [140]. The nominal pitch, $p_n$ is specified as 10.4mm and nominal coil radius, $C_{r,n}$ as 12.5mm. It has demonstrated that as the number of coils increase, the number of equivalent tanks-in-series rises approximately linearly [95]. Therefore, two coil rotations are used for the reactor length balancing overall simulation time and reactor size, providing an overall reactor length of $4\pi C_{r,n}$. For cross-sectional interpolating points lower and upper bounds of 2mm and 4mm are applied respectively, indicating minimum and maximum radial values. To maintain feasible coil paths the bounds of height deviations $\Delta z$ are specified as $\pm$1mm, and radial deviations $\Delta\rho$ as $\pm$3.5mm, ensuring no self-intersections.

### 4.2.2 Simulation

The methodology used for the evaluation of reactor designs under specific operating conditions is similar to that in the previous chapter. Utilising OpenFOAM, a simulation is performed where an impulse tracer is injected into the steady-flow water medium, and the resulting concentration is tracked by solving the relevant transport equations. Key aspects of the approach include the application of the transient pimpleFOAM solver for unsteady momentum equations, and the scalarTransportFoam to handle convection-diffusion effects with diffusion coefficient constant at a low value. The integral of the concentration over the outlet is recorded at each timestep and returned from the simulation. The evaluation

process additionally involves monitoring the tracer concentration at the outlet to terminate the solution at a specific threshold, ensuring minimal solution times.

### 4.2.3 Optimisation

The optimisation of highly-parameterised reactor simulations is high dimensional and involves computationally expensive function evaluations. This problem is solved using multi-fidelity Bayesian optimisation as presented in the previous chapter.

A maximum time budget is defined approximately depending on the number of parameters in each parameterisation optimised, ranging between 72 and 168 hours in total. The framework involves two hyper-parameters to be decided, $\beta$ defining the exploration level, and $p_c$ which dictates how conservative the stopping criteria is. These are selected as $\beta = 1.5$ and $p_c = 2$ based on values that were deemed to be robust from previous work.

#### Optimisation Objective

The residence time distribution (RTD) from a simulation is approximated using a tanks-in-series model, with a larger number of theoretical tanks representing stronger plug flow performance. In the previous chapter RTDs were observed to have been unimodal and relatively symmetrical. Due to the extensive design space applied within this chapter, it was found that simulations can return non-ideal distributions that contain particularly long tails, or are bimodal. As more complex fluid flows are induced by the reactor designs, the resulting distributions become less ideal, and the tanks-in-series model provides a poor approximation. Therefore, a composite objective function is applied that serves to not only maximise plug flow performance, but also encourages symmetric and unimodal RTDs, accounting for designs to return non-ideal concentration distributions.

Equation 4.4 demonstrates the quantity, denoted $f$, that is minimised during optimisation.

$$\hat{E}_i(N) = \frac{N(N\theta_i)^{N-1}}{(N-1)!} e^{-N\theta_i} \quad i = 1, \ldots, d \tag{4.2}$$

$$N^* = \arg\min_N \frac{1}{d} \sum_{i=1}^{d} \left(E_i - \hat{E}_i(N)\right)^2 \tag{4.3}$$

$$f = \underbrace{\frac{\alpha}{d} \sum_{i=1}^{d} \left(E_i - \hat{E}_i(N^*)\right)^2}_{\substack{\text{Weighted} \\ \text{MSE}}} - \underbrace{N^*}_{\substack{\text{Equivalent} \\ \text{tanks-in-series}}} \tag{4.4}$$

where $d$ is the number of data points contained within an RTD returned from a simulation, $\hat{E}$ and $E$ are the predicted and returned dimensionless concentration values respectively, $\theta$ represents dimensionless time, and $N$ represents tank-in-series. Based on initial testing, a value of 100 is assigned to $\alpha$, ensuring that the tank-in-series, and the non-ideality error are weighted approximately equally (approximately between 50-100 each).

All code was evaluated using 64 CPUs, a single RTX6000 GPU to aid with training and evaluating Gaussian processes, and 64Gb of RAM. All code used within this chapter can be found within the associated repository [1]. Code containing only the respective parameterisations and function evaluations for benchmarking purposes can be found within the associated repository[2].

### 4.2.4 Experimental Validation

The selected designs were exported into the Standard Tessellation Language (.STL) file format from their mesh representation and modified resulting into 3D printable models. This was achieved by incorporating a bounding cylinder to encompass each reactor. The linear sections at the inlet and outlet were specifically configured with 8 mm and 10 mm outside diameter (OD) tube fittings, a requirement for interfacing with the experimental

[1] https://github.com/OptiMaL-PSE-Lab/pulsed-reactor-optimisation
[2] https://github.com/trsav/reactor_benchmark

apparatus. Finally, the bounding cylinder was reduced with the internal volume between the coil removed to minimise the resin required for printing. The designs were printed using a FormLabs Form3+ 3D printer, with a Clear V4 resin, following the manufacturer's default settings. A post-processing phase was performed, comprising of a washing stage in Isopropyl Alcohol (IPA) for 20 minutes, a drying period extending 24 hours, and a concluding post-cure treatment in a FormCure chamber, maintained at a temperature of 60$^{o}$C for a 30-minute interval.

The RTD method was the same as reported by McDonough et al. [95] and applied in the previous chapter. A 0.1 M KCl aqueous tracer solution was injected at the inlet of each reactor, and the conductivity at the outlet was measured throughout time. The net flow of deionized water and tracer injection were controlled using three separate OEM syringe pumps (C3000, TriContinent) that were hydraulically linked to the reactor via PTFE tubing routed through a custom Swagelok piece.

The concentrations used for the Villermaux–Dushman reaction were 0.0075M $H_2SO_4$, 0.008M KI, 0.0015M $KIO_3$, 0.023M NaOH, and 0.0225M $H_3BO_4$ (all in mol/L). The reactants were introduced into the reactor using separate syringe pumps, and the absorbance at 353 nm caused by tri-iodate ions was measured at the reactor outlet using a UV-Vis spectrometer in a 10 mm pathlength cuvette. Three repeats were performed for each reactor design to ensure reproducibility. Figure 4.3 provides an overview of the methodology of this chapter.

## 4.3 Results

Each parameterisation was optimised under steady-state flow conditions, with a Reynolds number of 50. To ensure tractability of the problem, the joint parameterisation employs a sequential strategy whereby the parameters of the optimal coil path design were fixed before the introduction of parameters to manipulate the cross-section. Figure 4.4 demonstrates the

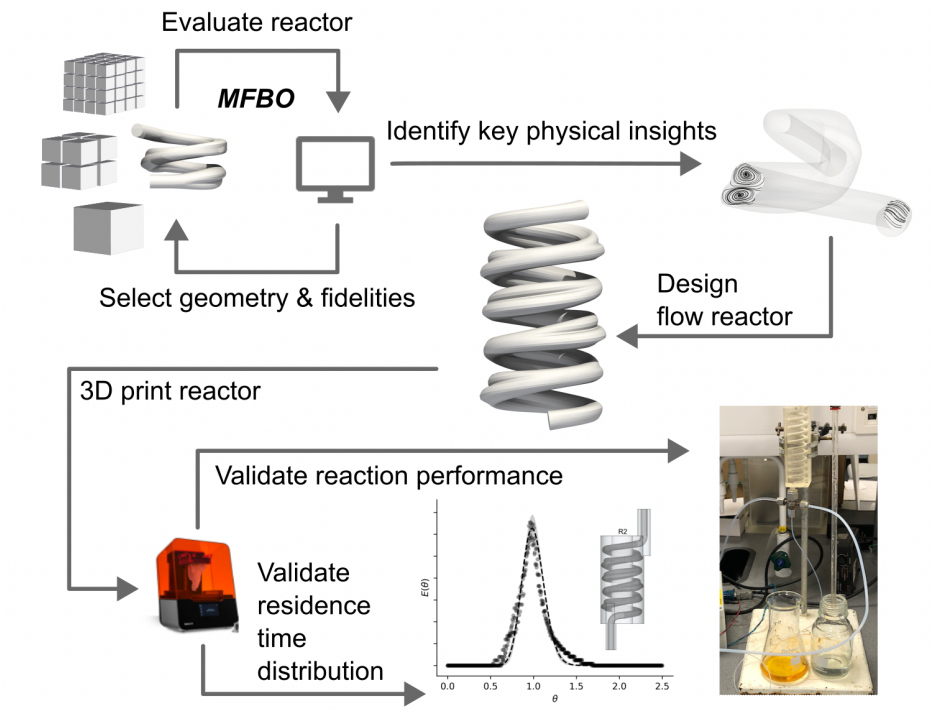



Fig. 4.3 A flow-diagram of the main aspects of the methodology of this chapter.

most optimal geometries for each parameterisation. We denote the control reactor as design 'i', and designs resulting from optimal cross-section, coil path, and combined parameters as 'ii', 'iii', and 'iv', respectively, corresponding to the subplots within Fig. 4.4.

### 4.3.1 Effect of Design Features on Flow Structures

Let us first consider design 'ii', where the shape of the tube cross-section throughout the length of the reactor is allowed to vary. The first key feature of design 'ii' is that the cross-section undergoes periodic expansions and contractions approximately every half-turn. Secondly, the design comprises a pinch constricting the flow when the cross-sectional area is greatest, during the expansion phase. Next, let us consider the optimal solution resulting from the parameterisation of the coil path corresponding to design 'iii' where the path deviates from a nominal configuration via the interpolation of cylindrical coordinates. The coil radius of curvature of design 'iii' begins relatively large, and subsequently reduces along the length

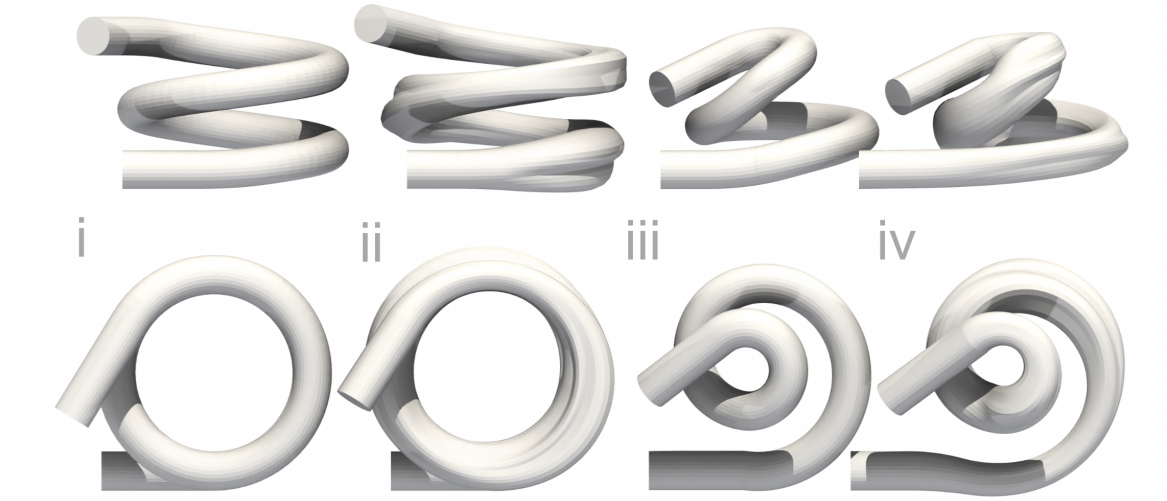



Fig. 4.4 The conventional coiled-tube reactor (i) alongside optimal coiled-tube reactors generated by parameterising the cross-section (ii), the coil path (iii), and a joint parameterisation (iv).

of the reactor, resulting in a tighter design than that in 'ii'. The pitch of the coil in design 'iii' begins small before rising approximately halfway along the length of the reactor then decreasing near the reactor outlet to the extent that the coil path points downwards. Within design 'iv', the path is fixed as in 'iii' and the cross-section is allowed to vary; inspection of design 'iv' reveals that it possesses features that are present in both designs 'ii' and 'iii'. To further investigate why these solutions are deemed optimal, different flow characteristics are demonstrated. Figure 4.5 depicts changes in fluid velocity within design 'iv' compared to a conventional coil (design 'i').

The expansion and contraction features in the cross-section alter the velocity distribution along the coil length in design 'iv', resulting in higher and lower velocities during contraction and expansion, respectively. Conversely, the conventional coil exhibits a relatively uniform velocity distribution along the coil length. The velocity changes observed in design 'iv', due to acceleration and deceleration, induce stronger pressure gradients, leading to the formation of Dean vortices that significantly enhance radial mixing. Figure 4.6 demonstrates the

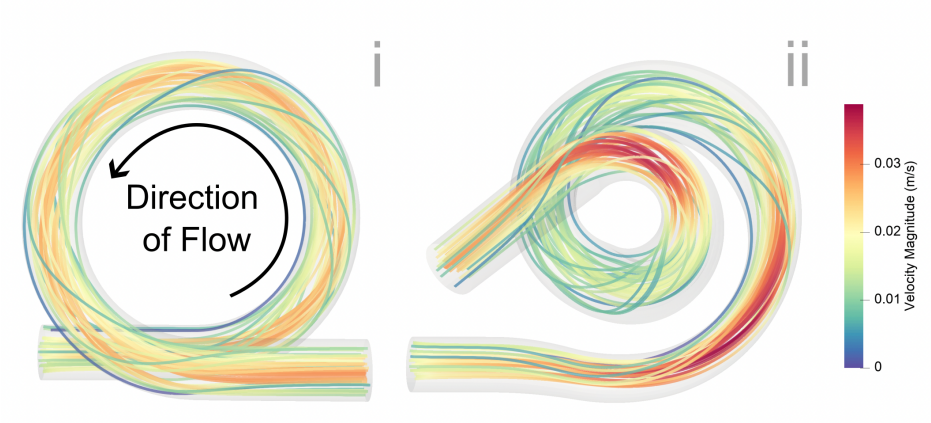



Fig. 4.5 Velocity streamlines coloured with velocity magnitude within a standard coil (i) and within optimal joint parameterisation (ii).

formation of Dean vortices in both a standard coil and design iv, as shown by sub-plots i and ii in Figure 4.6, respectively.

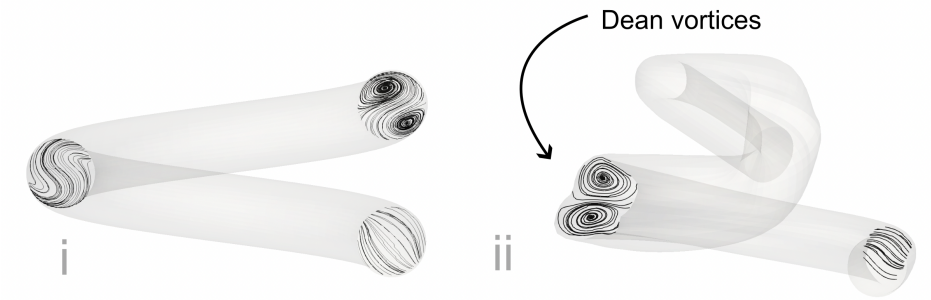



Fig. 4.6 Secondary flow streamlines at various cross-sections of the coil.

While Dean vortices are also formed in a standard coil, they are only partially established close to the tube outlet. The earlier formation of fully-developed Dean vortices in design 'iv' demonstrates the impact of reactor geometry resulting from the application of our framework, suggesting the potential for enhanced plug flow performance within more compact reactors, at lower *Re*.

Promoting plug flow depends on the relationship between axial and radial fluid mixing. To control axial dispersion, the consideration of streamwise velocity distribution becomes imperative. Therefore, velocities across the cross-sectional area of the coil are presented.

Figure 4.7 represents the velocity distribution along the length of design ‘i’ and design ‘iv’, illustrated by sub-plots i and ii, respectively.

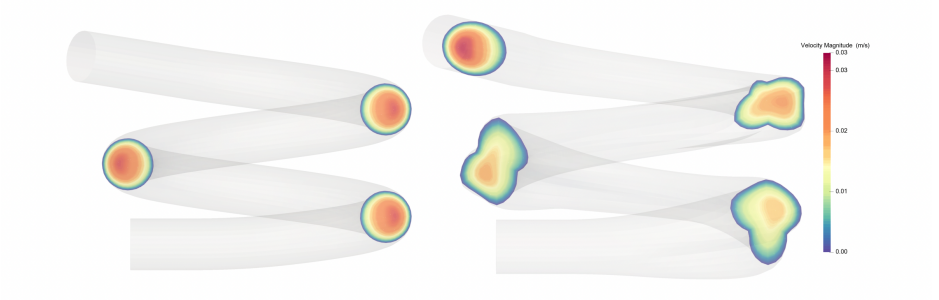



Fig. 4.7 Cross-sectional plane across the coil demonstrating streamwise velocities.

In design ‘i,’ it can be observed that the peak velocity is closer to the outer walls, with lower velocities near the inner wall throughout the coil length. This configuration induces faster tracer movement along the outer walls and slower movement along the inner walls, resulting in an extended time for the tracer to exit the coil, and increased axial dispersion. In contrast, the inclusion of the characteristic pinch feature in design ‘iv’ plays a key role in redistributing velocity across the coil cross-section by altering the radial position of peak velocities along the coil length. Design ‘iv’ demonstrates peak velocity at various radial positions along the coil length: at the top, bottom, and towards the outer wall, as depicted in Figure 4.7. This leads to the acceleration and deceleration of the tracer movement along the coil length, promoting a comparatively uniform axial movement of the tracer over time. Consequently, this results in a narrow profile for the residence time distribution of the tracer.

To further illustrate the enhanced characteristics of the optimised designs over conventional coil tube reactors, Figure 4.8 depicts streamlines representing fluid flow within the initial length of design ‘ii’.

Blue streamlines originate from the coil centre, while black streamlines start near the coil wall. As the cross-section of the reactor expands throughout the initial curve, the slow-moving fluid radially furthest from the centre of the coil moves initially outwards. Subsequently, as the cross-sectional area is greatest, the fluid within the central region moves towards the

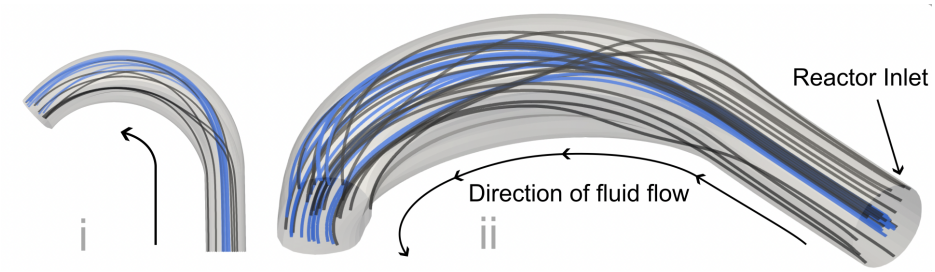



Fig. 4.8 The presence of induced Dean vortices within the optimal joint parameterisation coil (ii) compared with a standard coil (i).

outer walls of the coiled-tube. The fluid closest to the outer walls is then acted upon by the change in cross-section of the tube in the form of a pinch forcing fluid towards the centre of the tube via a swirling motion. The cross-section then contracts again, enabling a repetition of the mechanism for radial mixing under steady-state flow conditions.

### 4.3.2 Convergence Analysis

In this section, the convergence of the designs is analysed from an optimisation perspective in order to highlight the potential optimality of the solutions provided.

Given the high-dimensional design space, t-Distributed Stochastic Neighbour Embedding (t-SNE) [166] is applied to identify emerging trends throughout optimisation, investigate design convergence, and highlight the behaviour of the GP hyperparameters. This is a probabilistic dimensionality reduction technique, enabling high-dimensional data to be plotted whilst preserving local structure. The convergence of the cross-section design problem is analysed in this chapter, with the trends associated with the other parameterisations relegated to Appendix section B.3. The cross-section optimisation problem has better convergence properties than the other two design problems, motivating the process of identifying optimal driving characteristics as opposed to specific designs themselves. Figure 4.9 demonstrates the composite objective function against iteration (i), and wall-clock time (ii), demonstrating how lower-cost simulations are applied to provide less-expensive exploration.

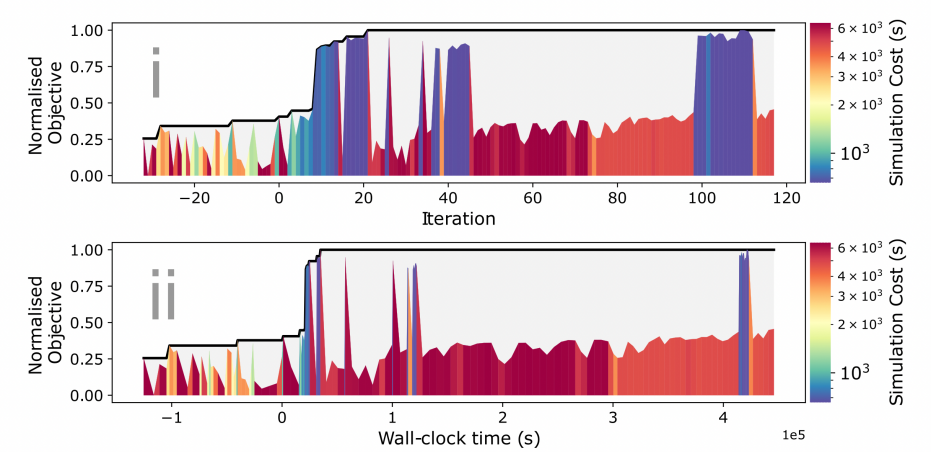



Fig. 4.9 The optimisation objective against iteration (i) and wall-clock time (ii). The objective has been normalised. The initial data set generated via design-of-experiments is denoted as negative iterations and wall-clock time. Hence, optimisation begins at the $0^{\text{th}}$ iteration and at $t = 0$.

The initial design of experiments used within optimisation is represented as occurring before the first iteration (so-called 'negative' iterations), and before $t = 0$. This sample contains reactor simulations with a variety of computational costs, as simulations across the spectrum of fidelities are performed. After optimisation begins, low-cost, low-fidelity simulations are selected during an initial phase of exploration. Simulations are selected that are progressively less expensive until approximately the $20^{\text{th}}$ iteration, as the updated GPs within the framework gain a better representation of simulation cost, and are able to select more efficient simulations. The framework then undergoes a period largely applying either the highest fidelity simulations, or the lowest fidelity simulations.

To investigate the properties of the GP that model the objective throughout optimisation, let us observe the kernel function hyper-parameters at a number of iterations. Figure 4.10 (top) demonstrates the lengthscale of each dimension within the kernel function within the objective GP as optimisation progresses.

Initially, the majority of lengthscales are relatively large indicating a uniform function space, capturing broad trends, as expected in a low-data regime in high dimensions. As optimisation progresses, and the GP is trained with more data, the lengthscales broadly decrease indicating

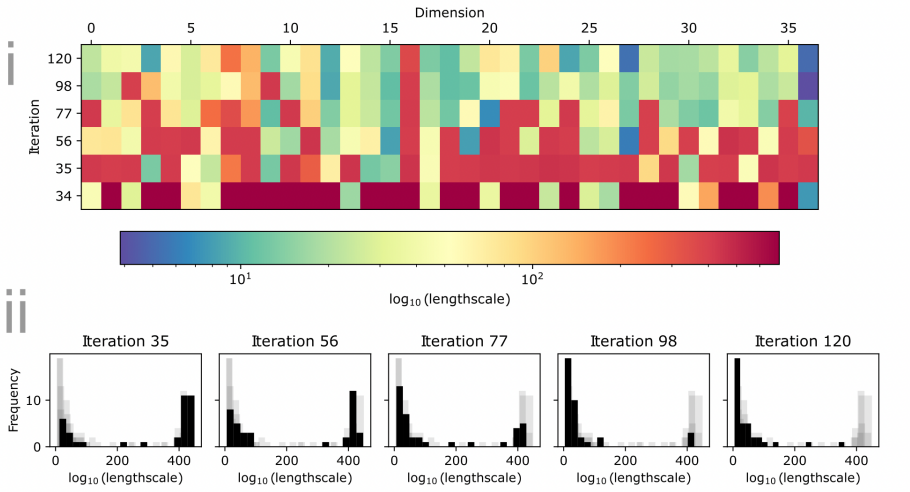



Fig. 4.10 Gaussian process dimension lengthscales throughout Bayesian optimisation iterations (top) alongside histograms demonstrating the distribution of GP lengthscales changing throughout optimisation (bottom).

an improved representation of the design space. Figure 4.10 (bottom) demonstrates the distribution of lengthscale hyper-parameters as optimisation progresses. The distribution of hyper-parameters tends towards lower values, and the GP becomes more nonlinear as data become available and correlations are more accurately captured.

As different simulation fidelities bias the objective function and the parameterisations are high-dimensional, t-SNE is applied to the convergence data of the cross-section parameterisation, observing data in the space of design parameters but not fidelities, colouring data points based on number of properties. Figure 4.11 shows the two-dimensional t-SNE embeddings of data points $\in \mathcal{X}$, labelled by axial and radial fidelities, objective value, and iteration.

Feature **i** shows the initial data set generated before optimisation begins. From here, the optimisation can be seen to progress along the path denoted by **ii**, demonstrating systematic changes in parameters indicative of convergent behaviour. Along this path, a number of different fidelities are evaluated (feature **iii**), demonstrating how lower-fidelity simulations are applied.

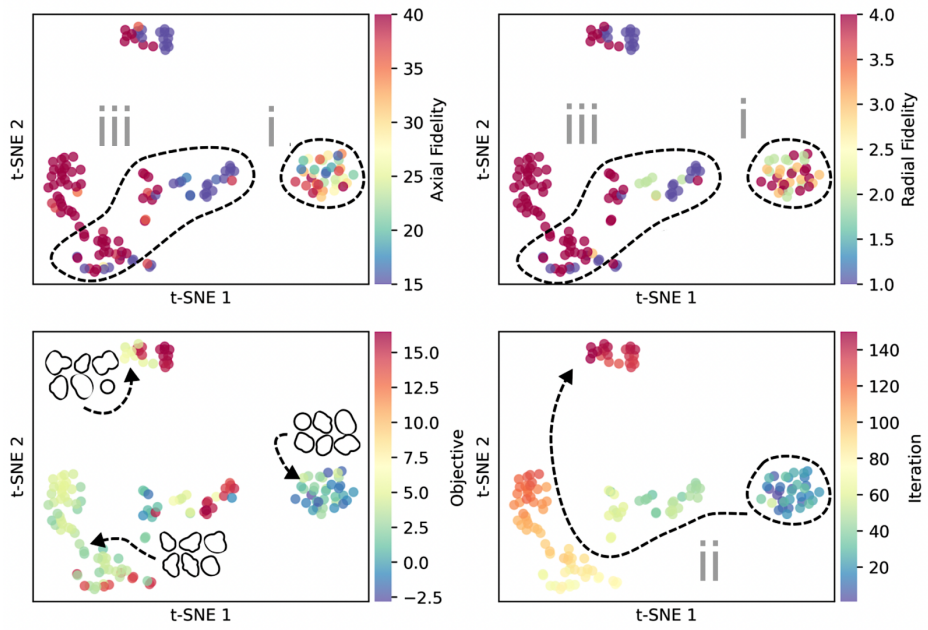



Fig. 4.11 t-SNE analysis of the data generated throughout optimisation in design parameter space ($\mathcal{X}$), reducing the dimensionality of design parameters to two dimensions, and labelled with different respective quantities.

A sample of the inducing cross-sections is presented from various clusters of data within the analysis against optimisation objective. The cross-sections in the original data set (feature **i**), as expected contain no distinct characteristics. As the optimisation progresses, the cross-sections defined by inducing points become more distinct, with certain cross-sections gaining symmetrical forms. Finally, at the end of optimisation, the cross-sections are most distinct, consisting of alternating small and large cross-sectional areas, with pinches throughout.

Lastly, let us define parameter variability as the normalised inverse of each GP lengthscale, each corresponding to a specific parameter. This property is calculated at the end of optimisation and present the value for each inducing parameter (the collection of which defines the form of the design space), indicating the variability in objective each specific parameter is responsible for. As can be seen from Figure 4.12, which demonstrates parameter variability throughout the reactor, the variability resulting from parameters in the tube cross-section is

greater where pinches occur, than those associated with the bottom and top of the tubular cross-section.

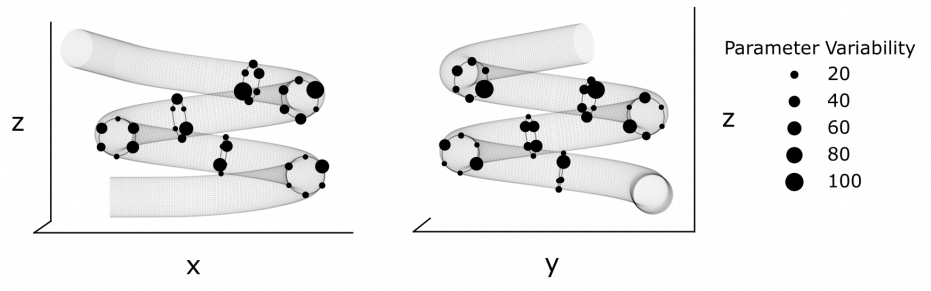



Fig. 4.12 Parameter variability, defined as a function of the GP lengthscale corresponding to each parameter, plotted for each inducing parameter on a nominal coil.

These trends indicate that the tube cross-section, and specifically the pinch, has a significant effect on plug flow performance, confirming our observations.

### 4.3.3 Optimal Designs & Comparison with 3D-Printed Reactors

The design concepts identified through the use of data-driven optimisation and machine learning are as follows:

- Expansion and subsequent contraction of reactor cross-section every half turn;
- Distinct pinch in cross-section throughout the reactor;
- Changes in the direction of flow including constriction of coil radius.

These concepts are now applied to two reactors corresponding to longer coils than those studied above; for one, changes are included to the tube cross-section, and for the other, variations to both cross-section and coil path.

Figure 4.13 demonstrates the final coiled-tube reactor designs, containing features identified as being responsible for the optimality of both the cross-section and coil path parameterisations.

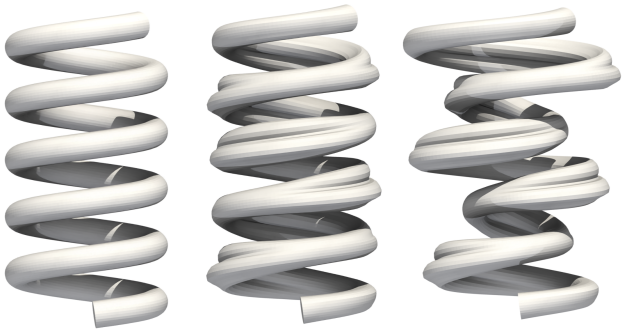


Fig. 4.13 **Comparisons with nominal coiled-tube designs.** The nominal coiled-tube (**left**, R1) alongside extrapolated steady-state flow coil designs containing aspects from the optimal cross-section (**centre**, R2) and both cross-section and coil path (**right**, R3).

The designs shown in Figure 4.13 were 3D-printed, and residence-time distribution experiments (see Methods) were performed for each configuration using steady-state flow at a $Re = 50$. **Figure 4.14** demonstrates the aggregated experimental data alongside tank-in-series models for each reactor configuration.

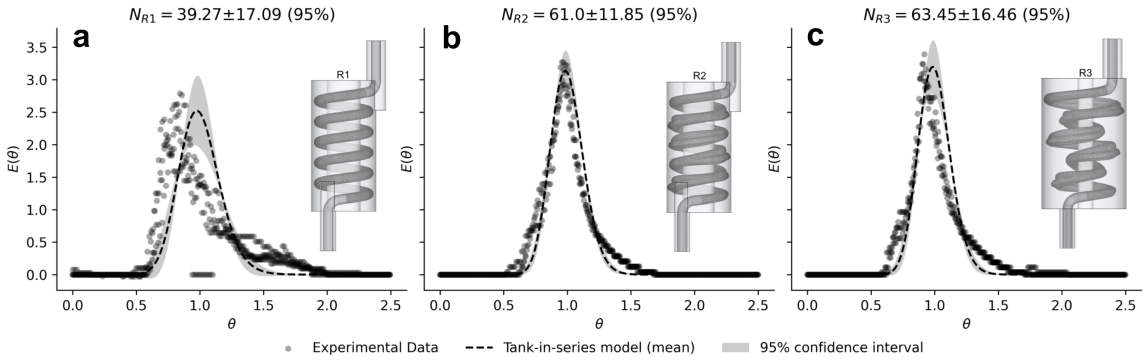



Fig. 4.14 Data generated across three experiments for each reactor configuration, designed to maximise equivalent tanks-in-series and minimise non-symmetric RTDs. A tanks-in-series model is used to estimate the performance of each reactor across the experiments performed. Dimensionless concentration at the outlet of the reactor ($E(\theta)$) is plotted against dimensionless time ($\theta$). **a**, the control reactor (R1). **b**, the reactor with variable cross-section (R2). **c**, the reactor with variable cross-section and path (R3).

The standard coil (R1) produces a skewed distribution with a mean equivalent tanks-in-series value across all experiments of of 39.27. The two distributions resulting from proposed reac-

tors with variable cross-section (R2) and cross-section with coil path (R3) are symmetrical. The equivalent tanks-in-series for these two reactors are higher than the standard coil at 61 (R2) and 63.45 (R3), representing a 55% and 62% increased value of mean equivalent tanks-in-series respectively. The RTD resulting from R1 is to be expected, as typically high *Re* is required for Dean vortices to form. The improvements in R2 and R3 demonstrate that it is indeed possible to avoid the need for oscillatory conditions to induce Dean vortices and promote mixing at low *Re*, through the inclusion of the identified design concepts. Other factors such as multiphase flows and chemistry may be explored to further exploit the discovery of designs through machine learning. An uncertainty analysis of the experimental data can be found in the Appendix section B.4.

To further validate the effectiveness of the optimised reactor designs, experiments were performed using the Villermaux–Dushman reaction system [121], a widely used test reaction for characterising mixing performance in reactors. The Villermaux–Dushman reaction is particularly suitable for this purpose as it is a fast, mixing-sensitive reaction that can provide insights into the mixing efficiency of the reactors [96].

The Villermaux-Dushman reaction system is as follows:

$$H_2BO_3^- + H^+ \rightleftharpoons H_3BO_3 \tag{4.5}$$

$$IO_3{}^- + 5I^- + 6H^+ \rightleftharpoons 3I_2 + 3H_2O \tag{4.6}$$

$$I_2 + I^- \rightleftharpoons I_3{}^- \tag{4.7}$$

Under poor mixing conditions reaction 4.6 is prohibited by the lack of acid, consumed within the faster first reaction. Good mixing conditions results in the formation of $I_2$ which in turn reacts to produce $I_3{}^-$ tri-iodide ions, measured at 353nm [96].

**Table 4.1** presents the absorbance values at 353 nm, corresponding to the presence of tri-iodate, measured at the outlet of each reactor. A higher absorbance value indicates a

Table 4.1 Absorbance values at 353 nm at the outlet of each reactor following a Villermaux–Dushman reaction.

| | **Absorbance (353 nm)** | | | |
|---|---|---|---|---|
| Reactor | Repeat 1 | Repeat 2 | Repeat 3 | Average |
| R1 | 0.267 | 0.276 | 0.275 | 0.273 |
| R2 | 0.292 | 0.288 | 0.300 | 0.293 |
| R3 | 0.326 | 0.324 | 0.325 | **0.325** |

higher conversion of the limiting reagent. The optimized reactors, R2 and R3, exhibit higher average absorbance values (0.293 and 0.325, respectively) compared to the nominal design, R1 (0.273), demonstrating improved reactive flow performance at Re=50. These results confirm that for mass transfer-limited reactions, such as the Villermaux–Dushman reaction, optimizing the residence time distribution serves as an effective proxy for enhancing overall reactor performance.

It is noted that the power consumption of the experimental syringe pumps was not measured. However, the pressure drop between the standard and optimised coils can be analysed computationally via CFD simulations. For a single coil turn, the optimised geometry has a 29.3 % higher pressure drop compared to the standard coil. This difference would further increase with the coil length. Therefore, the proposed designs almost certainly have a larger pressure drop than the standard coil. Future work may propose to treat this, or a similar case as a multi-objective black-box optimisation problem, trading off pressure-drop and plug flow performance.

## 4.4 Discussion

Throughout optimisation, it cannot be guaranteed that the global optimum of the parameterisations has been identified. However, the global methodology applied provides more varied solutions than a gradient-based local approach such as the adjoint method, and key features are identified that drive the behaviour of fluid within these reactors, including the presence of

induced Dean vortices. Subsequently, the reactors that are experimentally validated can be interpreted as 'augmented intelligence'-assisted designs, with features based on the optimal solutions.

This workflow is proposed for the design of highly-parameterised reactors as the interpretability of solutions is maintained. To support the design of algorithms for the optimisation of high-dimensional expensive black-box problems, all parameterisations are released as benchmark problems[3]. The parameterisations optimised contain feasible reactor geometries by design, resulting in an unconstrained optimisation problem. Highlighting the importance of parameterisation specification, the emergent behaviour in designs 'iii' and 'iv' contains a coil path that unintendedly pitches down near the reactor outlet. It is worth noting that features of the optimal coil path design bear some resemblance to the wavering coiled flow inverter design proposed by Singh et al. [149], though this design primarily relies on inversions within the coil. Future work may perform comparisons over a wide range of alternate and existing coil designs.

[3]Found at https://github.com/trsav/reactor_benchmark

# Chapter 5

# Human-algorithm Collaborative Bayesian Optimisation for Engineering Systems

## Contents


**5.1 Introduction . . . . . 128**

5.1.1 Contributions . . . . . 131

**5.2 Background . . . . . 132**

**5.3 Methodology . . . . . 132**

5.3.1 Expert-augmented Design of Experiments . . . . . 133

5.3.2 Defining Alternate Solutions . . . . . 134

5.3.3 Algorithm Characteristics . . . . . 136

5.3.4 Selecting from a Set of Solutions . . . . . 139

**5.4 Case Studies . . . . . 139**

5.4.1 Benchmark Functions . . . . . 140

5.4.2 Pulsed-flow Coiled Tube Reactor optimisation . . . . . 142

5.4.3 Black-box Bioprocess optimisation . . . . . 143

5.4.4 Human-in-the-loop Experimental Protocol . . . . . 146

**5.5 Results & Discussion . . . . . 147**

5.5.1 Effect of Noise . . . . . 147

5.5.2 Effect of Dimensionality . . . . . 149

5.5.3 Effect of Number of Alternative Solutions . . . . . 151

5.5.4 Effect of Specific Function Properties . . . . . 152

5.5.5 Coiled Tube Reactor Design . . . . . 154

5.5.6 Fed-batch Bioprocess Control . . . . . 155

**5.6 Conclusions . . . . . 157**

# Publication Note

This chapter has been published as:

Savage, T., & del Rio Chanona, E. A. (2024). *Human-algorithm collaborative Bayesian optimization for engineering systems*. In Computers & Chemical Engineering (Vol. 189, p. 108810). Elsevier BV. https://doi.org/10.1016/j.compchemeng.2024.108810

The work presented in this chapter explores how human expertise can be incorporated into Bayesian optimisation workflows to support faster optimisation and discovery across science and engineering.

## Author Contributions

- Tom Savage: Conceptualisation, methodology, software implementation, data analysis, visualisation, and writing of the original manuscript
- Dr Ehecatl Antonio del Rio Chanona: Project supervision, conceptual guidance, and manuscript review

## Modifications from Published Version

The following modifications have been made to the published version:

- Spelling has been changed from American to British English.
- Some language has been altered to better fit within the context of this thesis.

# Abstract

In this thesis and beyond, Bayesian optimisation has proven effective for optimising expensive-to-evaluate functions across chemical engineering. However, valuable physical insights from domain experts are often overlooked in favour fully-automated decision making, potentially limiting convergence to promising solutions. This chapter introduces a collaborative Bayesian optimisation approach that re-integrates human input into the data-driven decision-making process. By combining high-throughput Bayesian optimisation with discrete decision theory, this chapter serves to enable experts to influence the selection of experiments via a discrete choice. A multi-objective approach is used to generate a set of high-utility and distinct solutions, from which the expert selects the desired solution for evaluation at each iteration. This methodology maintains the advantages of Bayesian optimisation while providing a means to incorporate expert knowledge and improve accountability. The approach is demonstrated across various case studies, including bioprocess optimisation and reactor geometry design, demonstrating that even with an uninformed practitioner, the algorithm recovers the regret of standard Bayesian optimisation. By including continuous expert opinion, the proposed method enables faster convergence and improved accountability for Bayesian optimisation in engineering systems.

## 5.1 Introduction

Bayesian optimisation has been successfully applied in complex domains where sampling is expensive and derivatives are often not available, such as those that involve experiments, simulation or propriety software. Previous chapters have exploited this for the optimisation and discovery of novel coiled tube reactor geometries. By removing the human from decision-making processes in favour of maximising statistical quantities such as expected improvement, unknown functions can be optimised efficiently [116, 181, 164]. However, these engineering systems are often engaged with by domain experts such as engineers or chemists, and as such the behaviour of the underlying function cannot be considered completely unknown a-priori [170, 7, 23]. Therefore, there exists significant scope to apply Bayesian optimisation in optimising expensive derivative-free problems, whilst enabling domain experts to inform the decision-making process, putting the human back into the loop. In previous work, Kanarik et al. [64] demonstrated the potential for experts to enhance the convergence of Bayesian optimisation when applied to a complex semi-conductor chip process, particularly during the initial exploration phase. However, expert behaviour, often consisting of changing one or two variables at a time, has been hypothesised to inhibit progress later on, where Bayesian optimisation alone was deemed more efficient. Kanarik et al. [64] highlights the difficulties in fusing expert and computational knowledge with respect to cultural challenges such as trust and transparency, and acknowledges the important role humans will play alongside computational tools. Recent attempts have been made to integrate expert knowledge algorithmically within Bayesian optimisation [85, 57, 25]. Hvarfner et al. [57] and Ramachandran et al. [128] weight the acquisition function based on a user-specified prior in an effort to guide convergence towards expert-specified regions. In the case of a poor prior the approach has been found to be over-sensitive [85]. Liu [85] modified the approach by diminishing the weight of the expert-specified prior in the optimisation, favouring it throughout the exploration-phase. By guiding the optimisation

towards the general 'area' of human intuition, methodologies that weight the acquisition function ensure the expert has no guaranteed say in which solutions are evaluated throughout the optimisation. Moreover, these approaches are considered static with expert-information only derived before the optimisation. Whilst updating the prior throughout optimisation is an option to include human-feedback, this may be time-consuming for a practitioner and previous issues remain. Additionally, it is always possible to include a (simplified) model as the mean function to a Gaussian process, acting as a way to insert domain knowledge within the Bayesian optimisation framework. Gupta et al. [50] presents an alternative approach, where an expert has continuous input throughout the optimisation process. The expert, approximated by a linear Gaussian process, also selects a solution at a given stage, which is evaluated alongside the solution from a standard acquisition function. By approximating human behaviour, the authors present sub-linear regret bounds, faster than either the human-model or stand-alone Bayesian optimisation. Kumar et al. [75] demonstrates an approach where alternative solutions are proposed by both the expert and the acquisition function, and the information is combined to inform an evaluated experiment, in which the expert has the final say. Our work contributes broadly to the field of human-AI collaboration, in the same body of literature. Reverberi et al. [130], Cisse et al. [25] provide evidence that humans have the potential to internally condition information throughout complex medicine decision-making processes, to enhance AI systems, whilst Wang et al. [168] provides an overview into the field of Human-AI collaboration within data-science.

In this chapter, these issues of how to efficiently include human-input, provide accountability, and maintain ease-of-use are mitigated by presenting a methodology for human-in-the-loop collaborative Bayesian optimisation that enables continuous input, whilst remaining practical for an expert to perform. Importantly, the approach presented enables an expert to maintain influence over decision-making with minimal effort through a discrete selection step that may be performed remotely, quickly, and agnostic to platform (i.e. can be communicated

efficiently online). Therefore removing a number of barriers of entry to applying human-in-the-loop methods in the real world such as the continuous time and effort an expert expected to contribute during the optimisation process.

High-throughput (batch) Bayesian optimisation [45] is used alongside a discrete decision mechanism to enable domain experts to influence the selection of optimal experiments. This methodology exploits the hypothesis that humans are more efficient at making discrete choices rather than continuous ones, whilst also enabling experts to influence important early decisions through an expert-informed design-of-experiments step that enables evaluation of specifically desired solutions. At each iteration a high-throughput (batch) Bayesian optimisation problem is solved using a multi-objective formulation, optimising the trade off between the sum of the alternative utility function values (also referred to as acquisition function values) and the determinant of their covariance matrix, equivalent to their total variability. By taking the solution at the knee point of the Pareto front, a set of alternative solutions is returned that have both high utility values and are reasonably distinct. Subsequently, these alternatives are provided to a domain expert to select their desired evaluation in the knowledge that any one choice presents information gain about the optimal solution, whilst ensuring the choices are distinct to the human expert. Alternative solution information such as utility function value, predicted output distribution, and visualisations are provided to the expert as prior knowledge. The decision-maker then effectively performs discrete Bayesian reasoning, internally conditioning their provided information with their own expertise and knowledge of the solutions provided (acting equivalently to a prior)[108]. In addition, this approach enables improved accountability in the decision making process and final solution, as the decision-maker has the final say in what is evaluated at each iteration. This approach works with any utility function and NSGA-II [28] is applied for multi-objective optimisation, efficiently handling the non-convex utility-space. Figure 5.1 demonstrates the main aspects

of the methodology presented in this chapter for proposing alternative solutions to a human expert using a high-throughput formulation solved using multi-objective optimisation.

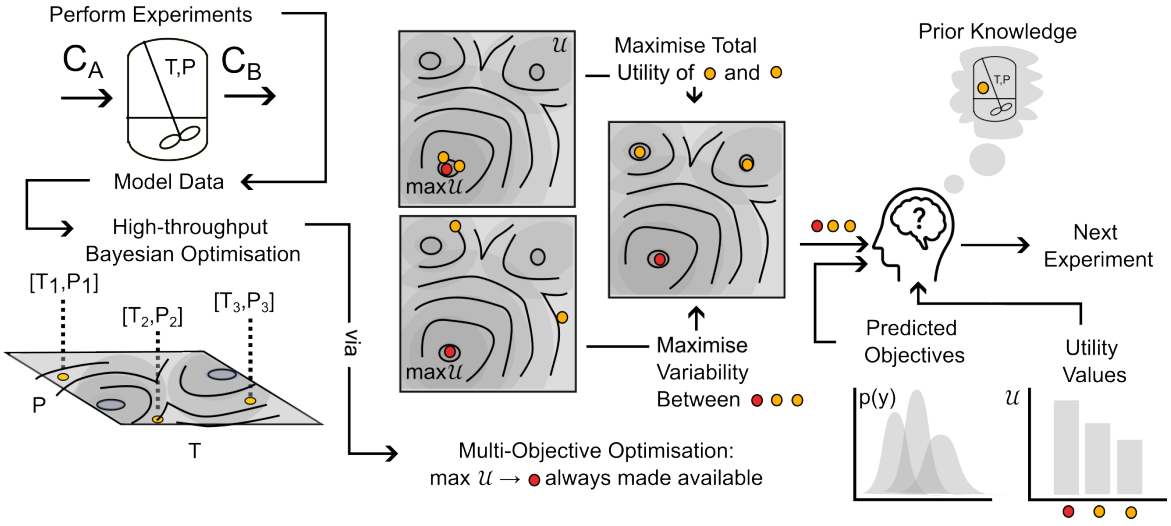



Fig. 5.1 Methodology overview for proposing alternative solutions to a human expert using a high-throughput formulation solved with multi-objective optimisation. The trade-off between the sum of utility function values and the determinant of their covariance matrix is optimised to obtain a set of high-utility and distinct solutions, from which the expert selects the most promising one for evaluation.

The methodology is benchmarked on a number of numerical and applied case studies, and across different human behavioural policies including complete expertise, no knowledge, and an adversarial or misaligned practitioner. In doing so insights are provided into when expert-assisted Bayesian optimisation approaches may be beneficial with respect to problem dimensionality, noise level, and function properties.

### 5.1.1 Contributions

This chapter presents a novel methodology for human-in-the-loop collaborative Bayesian optimisation that enables continuous expert input while remaining practical and easy to use. The key contributions are as follows:

1. A collaborative approach to Bayesian optimisation exploiting both human domain knowledge, and efficiency in making discrete choices.

2. A novel batch/high throughput Bayesian optimisation formulation using multi-objective optimisation to balance utility and diversity across alternative solutions.

3. A novel expert-informed design-of-experiments step to include specifically desired solutions within a space-filling initial set of solutions.

4. Benchmarking insights for collaborative Bayesian optimisation across various benchmark and real world scenarios, demonstrating improved performance across a range of expert abilities.

## 5.2 Background

I will first provide a background into the various components that contribute towards the methodology. Gaussian processes have been previously covered and a thorough description can be found within the background chapter. Similarly, the reader is referred to the sections on Bayesian optimisation and various noiseless and noisy acquisition functions in the background chapter, which are used throughout this chapter.

## 5.3 Methodology

In this section, I outline the specific approach for human-in-the-loop Bayesian optimisation. The methodology consists of three steps. Firstly, an expert-informed initial design of experiments methodology is outlined. Secondly, a high-throughput (batch) approach for Bayesian optimisation using a multi-objective formulation is described, and used in order to provide a set of alternative solutions. Finally, the human decision making procedure involved is described in detail.

Let us first begin with a description of an optional step, wherein an expert provides a number of specific solutions that they wish to be evaluated. By performing this step before optimisa-

tion begins, and not during optimisation, the expert is given the agency to consider continuous solutions without prohibitive time constraints that may be present during optimisation, as well as ensure that specific solutions are evaluated. By including this step, the benefits discussed in Kanarik et al. [64] and Reverberi et al. [130] are incorporated. This is done by applying an augmented initial design of experiments, enabling the number of specific solutions to be as large or small as the expert deems appropriate relative to the total number of space-filling points.

### 5.3.1 Expert-augmented Design of Experiments

Let us first define an initial number of $m$ potential $d$-dimensional solutions as specified by an expert as $\mathbf{X}_{\text{expert}} \in \mathbb{R}^{m \times d}$. Subsequently the $t-m$ remaining solutions are presented as $\mathbf{X}_{\text{design}} \in \mathbb{R}^{(t-m) \times d}$, where $t$ is the total number of initial solutions required. If $m \geq t$ then there is no need to proceed with the following step and the final design of experiments becomes $\mathbf{X}_{\text{expert}}$. In order to distribute the remaining initial solutions throughout the design space the following optimisation problem is solved:

$$\min_{\mathbf{X}_{\text{design}}} -|K_{\mathbf{X}_{\text{aug}}}| \tag{5.1}$$

where $\mathbf{X}_{\text{aug}}$ represents the concatenation of $\mathbf{X}_{\text{design}}$ and $\mathbf{X}_{\text{expert}}$, and $K_{\mathbf{X}_{\text{aug}}} = [k(\mathbf{X}_{\text{aug},i}, \mathbf{X}_{\text{aug},j})]_{i,j=1}^{p}$ is the covariance matrix between all solutions within the augmented set using a given kernel function (i.e. distance based). Equation B.14 reflects the principle of maximising the determinant of the covariance matrix, equivalent to a *volume of information* spanned by all solutions. By optimising over the remaining solutions, it is ensured these are evenly distributed throughout the design space, taking into account the fixed locations within $\mathbf{X}_{\text{expert}}$.

Figure 5.2 demonstrates this approach for expert-informed generation of an initial set of solutions for use as a dataset in Bayesian optimisation.

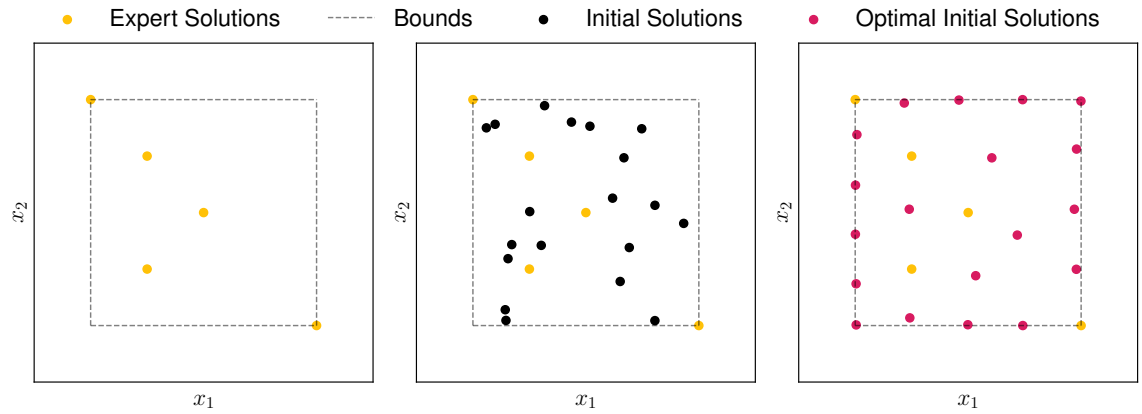



Fig. 5.2 The expert-augmented design of experiments procedure outlined using a two dimensional example. **Left:** the expert defined set of solutions. **Center:** the initial random set of solutions defined as $\mathbf{X}_{\text{design}}$. **Right:** the optimal set of solutions comprising of both $\mathbf{X}_{\text{design}}$ alongside $\mathbf{X}_{\text{expert}}$, resulting in the final design of experiments.

Let us now turn our attention to the Bayesian optimisation loop itself, beginning with a focus on how to define alternative sets of solutions to provide to an expert at each iteration.

### 5.3.2 Defining Alternate Solutions

To begin the approach, at each iteration a given utility function $\mathcal{U}$ is first maximised for a given dataset $\mathcal{D}_t := \{(\mathbf{x}_i, y_i)\}_{i=1}^t$:

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathcal{X}\subseteq\mathbb{R}^d}{\arg\max}\ \mathcal{U}(\mathbf{x}), \tag{5.2}$$

resulting in the optimal next evaluation, $\mathbf{x}^*$, from a standard utility sense (for example expected-improvement). This step remains the same as in standard Bayesian optimisation, where this solution would be evaluated, added to the dataset, and the procedure repeated. What provides the key different in this approach is that a number of alternatives is desired, from which the expert may choose their desired solution. By considering utility values

and differences between these solutions, the goal is to ensure that all have large expected improvement values, whilst remaining distinct within the design space. Let $p$ be the number of alternate solutions provided to the expert and construct the decision variable matrix $\mathbf{X} \in \mathbb{R}^{(p-1)\times d}$ by concatenating $p-1$ alternate solutions $\mathbf{X} := [\mathbf{x}_1, \ldots, \mathbf{x}_{p-1}]$. Let us then define the high-throughput (batch) utility function $\hat{\mathcal{U}}$ which is specified as the sum of the individual utilities of alternate solutions within $\mathbf{X}$

$$\hat{\mathcal{U}}(\mathbf{X}) = \sum_{i=0}^{p-1} \mathcal{U}(\mathbf{x}_i). \tag{5.3}$$

Similarly, $\hat{\mathcal{S}}$ is introduced as a measure for capturing the variability among both the optimal and alternative solutions. Specifically, let $\hat{\mathcal{S}}$ be the determinant of the covariance matrix $K_{\mathbf{X}_{\text{aug}}}$ for the augmented matrix of solution vectors $\mathbf{X}_{\text{aug}} = \mathbf{X} \cup \mathbf{x}^*$, containing the current set of alternate solutions:

$$\hat{\mathcal{S}}(\mathbf{X}, \mathbf{x}^*) = |K_{\mathbf{X}_{\text{aug}}}| \tag{5.4}$$

$$K_{\mathbf{X}_{\text{aug}}} = [k(\mathbf{x}_{\text{aug},i}, \mathbf{x}_{\text{aug},j})]_{i,j=1}^{p}, \tag{5.5}$$

where $\hat{\mathcal{S}}$ quantifies the 'volume of information' spanned by the alternative solutions $\mathbf{X}$ as well as the optimal solution $\mathbf{x}^*$. Maximising $\hat{\mathcal{U}}$ will result in all alternative solutions proposed being the same as $\mathbf{x}^*$, that is $[\mathbf{x}_1^*, \ldots, \mathbf{x}_{p-1}^*]$. Contrary to this, maximising $\hat{\mathcal{S}}$ will result in a set of solutions that are maximally-spaced both with respect to other alternatives, but also to $\mathbf{x}^*$. This can be empirically visualised in Figure 5.1.

At iteration $t$, the following multi-objective optimisation problem is solved:

$$[\mathbf{X}_{\hat{\mathcal{U}}}^*, \ldots, \mathbf{X}_{\hat{\mathcal{S}}}^*] = \max_{\mathbf{X}} \left( \hat{\mathcal{U}}(\mathbf{X}; \mathcal{D}_t), \hat{\mathcal{S}}(\mathbf{X}, \mathbf{x}^*) \right), \tag{5.6}$$

resulting in a set of solutions along the Pareto front of both objectives $\hat{\mathcal{U}}$ and $\hat{\mathcal{S}}$. From this set of optimal solutions, $\mathbf{X}_k^*$ is defined as the solution at knee-point of the Pareto front. The $p-1$ individual solutions contained within $\mathbf{X}_k^*$ optimally trade off the sum of their utility values with their variability. By presenting the knee-solution of the multi-objective problem, it is ensured that when provided to an expert alongside $\mathbf{x}^*$, any individual solution will have high expected information gain, and the solutions themselves will be distinct enough to ensure the expert is not made to make an effective gradient calculation (i.e. solutions are very close and practically indistinguishable). The practitioner is then made to choose a solution to evaluate from this set of alternatives. To do so, the practitioner is provided with information such as the utility value of each solution, expected output distributions (obtained from the Gaussian process), and information regarding previous solutions that they may wish to draw upon. In doing so, the practitioner effectively performs an internal discrete Bayesian reasoning, conditioning previous prior information and expert opinion with the mathematical quantities provided to make an informed decision.

Algorithm 2 illustrates the methodology. The approach allows for an expert to include a number of continuous solutions within the initial dataset $\mathcal{D}$, which can be performed as outlined in Section 5.3.1. Additionally, the algorithm may be used alongside other approaches which allow for an expert to specify a distinct solution to be evaluated at every iteration.

### 5.3.3 Algorithm Characteristics

Figure 5.3 demonstrates the intended behaviour of our approach. A one-dimensional case study is presented for demonstration, optimising a function obtained through sampling a Gaussian process prior specified by a Matérn 5/2 kernel with lengthscale $l = 0.5$. In this case study 4 alternatives are provided to an expert, from which a solution at random for illustrative purposes.

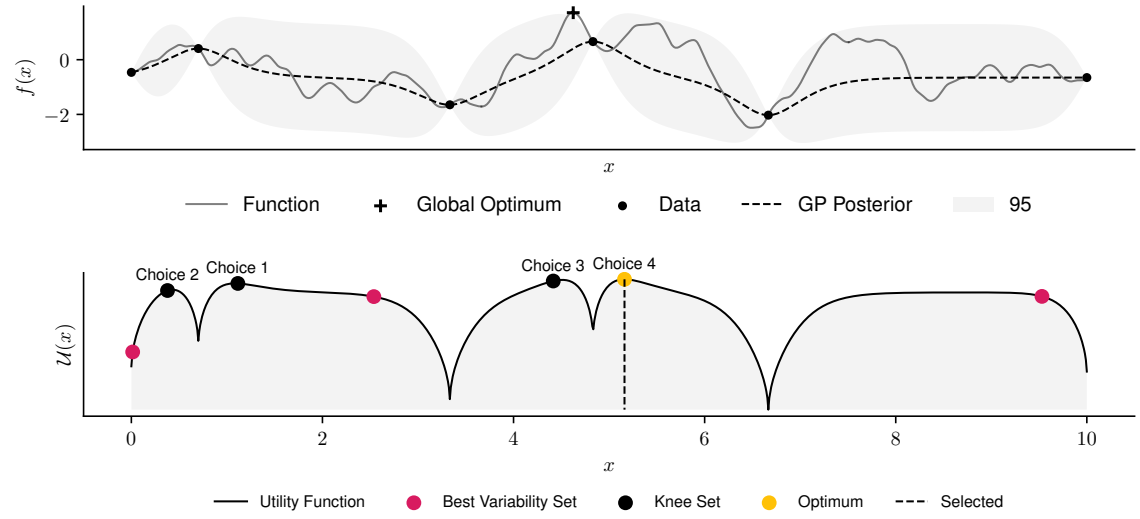
**Algorithm 2** Expert-Guided Bayesian optimisation

**Initialise:** Objective function $f$, Initial data $\mathcal{D}$, Domain $\mathcal{X}$
Alternative choices $p$, Utility function $\hat{\mathcal{U}}$, Variability function $\hat{\mathcal{S}}$, Termination Criteria
**if** `size`$(\mathcal{D})$ is less than desired **then**
  $\mathcal{D} \leftarrow$ Expert-guided DoE $(\mathcal{D})$ `// Section 5.3.1`
**end if**
**while** Termination Criteria is False **do**
  $\mathbf{x}^* \leftarrow \arg\max_{x \in \mathcal{X}} \mathcal{U}(x)$ `// Standard Bayesian optimisation Step`
  `// Multi-Objective optimisation`
  $[\mathbf{X}_1^*, \ldots, \mathbf{X}_m^*] \leftarrow \max_{\mathbf{X}}(\hat{\mathcal{U}}(\mathbf{X}; \mathcal{D}), \hat{\mathcal{S}}(\mathbf{X}, \mathbf{x}^*))$
  $\mathbf{X}_k^* \leftarrow \text{knee}([\mathbf{X}_1^*, \ldots, \mathbf{X}_m^*])$ `// Select Knee-solution`
  `// Expert selects from alternatives and standard optimal`
  $\mathbf{x}_{\text{eval}} \leftarrow \underset{\mathbf{x} \in \mathbf{X}_k^* \cup \mathbf{x}^*}{\text{expert}}\, \mathbf{x}$
  $\mathcal{D} \leftarrow \{(\mathbf{x}_{\text{eval}}, f(\mathbf{x}_{\text{eval}}))\} \cup \mathcal{D}$ `// Update dataset`
**end while**

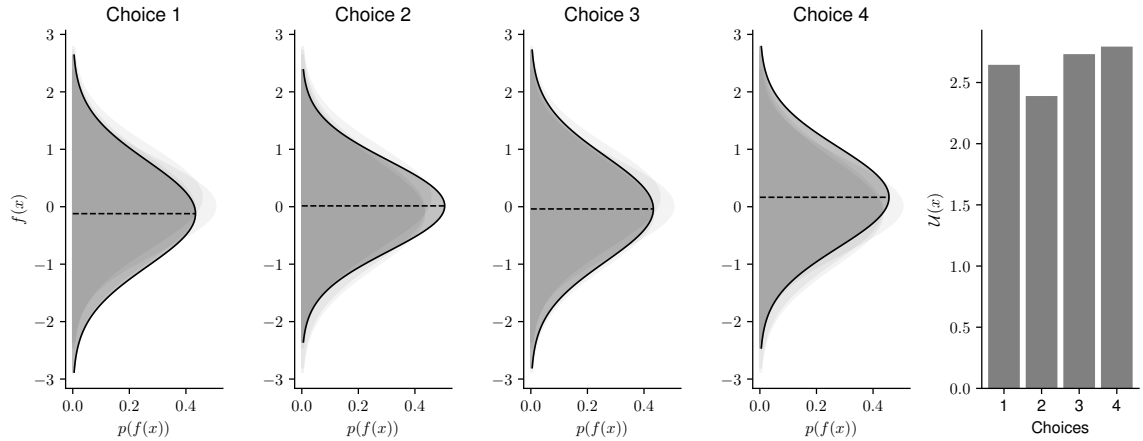
Figure 5.3a demonstrates the three alternate solutions that maximise the solution variability in red, originating from an extreme solution on the Pareto front, while the black solutions denote those contained within the knee-solution of the Pareto front. The yellow optimal solution is included alongside these three alternatives to an expert. In this case choice four is selected randomly from the alternatives.

Figure 5.3b demonstrates how in this case choice one and choice three have relatively similar utility values and predicted output distribution to choice four (the optimal of the acquisition function). The expert is then allowed to distinguish between these similar solutions in a way the computer cannot through their prior domain knowledge. By conditioning their prior information on the given values, the expert is effectively performing internal Bayesian reasoning.

To illustrate the importance of including the optimum of the utility function within the set of alternatives, consider a scenario where the utility function possesses a single optimal solution, representing a singular clearly desirable experiment. In such a case, the multi-objective approach which focuses on maximising both aggregate utility and variability could lead to




(a) The objective function and utility function after five function evaluations.




(b) The information provided to the expert regarding the three alternative solutions alongside input values and if available solution visualisations.

Fig. 5.3 A standard iteration of our algorithm on a one-dimensional case-study.

suboptimal outcomes, as the optimal set of alternatives may attempt to straddle the single optimal experiment. In this case, the solutions provided to the expert have ignored the clear solution in pursuit of increased variability. By including the solution that maximises the utility function within the set of alternatives and calculating the variance with respect to this, this behaviour is mitigated, breaking any symmetry that may occur within the multi-objective

formulation. In doing so the expert is allowed to simply fall back to standard Bayesian optimisation by selecting the solution with the largest utility value.

### 5.3.4 Selecting from a Set of Solutions

Having presented an approach to provide a set of promising and distinct alternative solutions to the expert, let us draw our attention to how to present these to a practitioner. As previously mentioned, the expert effectively performs discrete Bayesian reasoning, conditioning the information provided with their prior beliefs of the underlying function. While the utility function summarises the predictive output distribution of a given solution, the output distribution is still included alongside utility values. This enables the practitioner to infer their own internal solution utility. For example, the expert practitioner may have more interest in regions of high uncertainty, which, while potentially being sub-optimal from an expected improvement perspective, will provide them with the confidence to implement the resulting optimal solution.

Solution information provided to the practitioner may be provided in a string-based format. However, for problems with a large number of variables this may become prohibitive. Therefore, it is proposed that for applied high dimensional problems an image-based solution representation may be used. For example in black-box bioprocess control, a visualisation of the control profile as we illustrate in our case study.

## 5.4 Case Studies

In this section the performance of the approach is investigated through a number of mathematical functions and real world chemical engineering case studies. Standard Bayesian optimisation is used as a baseline for our comparisons. To characterise how our method will behave depending on the level of expertise of the human making decisions a number

of different human behaviours are hypothesised. The 'Expert' practitioner represents an ideal, where the best solution (that is the one with the highest true objective value) is always selected. Equivalently, to test the performance of our approach under the influence of a practitioner with misaligned knowledge, an 'Adversarial' practitioner is presented who consistently selects the solution with the lowest true objective value. In addition, a probabilistic practitioner is presented, who selects the solution with the best true objective value with some probability. In practice, the human will condition the information provided (e.g. objective function value, visuals of the possible solutions) with their prior beliefs (e.g. expert knowledge). In the outlined approach this includes information regarding the expected distribution of the objective of each solution, as well as the utility value of each solution. Therefore, finally a 'Trusting' practitioner is presented who selects the solution with the largest utility, equivalent to standard Bayesian optimisation as this solution is obtained through standard single objective optimisation. The human behaviours applied are summarised within Table 5.1.

Table 5.1 Human behaviours Applied for Benchmarking

| **behaviour Type** | **Description** |
|---|---|
| Expert | Selects the solution with the best true function value. |
| Adversarial | Selects the solution with the worst true function value. |
| Trusting | Selects the solution with the maximum utility value. |
| $p(\text{Best})$ | Selects the solution with the best true function value with probability $p(\text{Best})$, otherwise selects a random solution. |

### 5.4.1 Benchmark Functions

This approach is benchmarked on four well-known optimisation benchmark functions. For each standard function the performance of different expert behaviours is evaluated for the two

dimensional, five dimensional, and ten dimensional form. The equations for these benchmark functions can be located within Appendix C.1.

In addition to benchmarking on structured functions, random functions are generated by sampling from a Gaussian process prior, enabling the ability to benchmark our approach in expectation across all functions of a given lengthscale. Each function is equivalent to a sample path from a Gaussian process. Random two, three and five dimensional functions are generated from priors with a lengthscale of 0.05 within the bounds 0 and 1. The same random seed is used across all benchmarks to ensure a fair comparison between independent variables. Further information about how Gaussian process sample paths can be used to assess the convergence of Bayesian optimisation algorithms can be found in [43]. A visualisation of lower-dimension sampled functions can be located in Appendix C.2.

For each test function, Bayesian optimisation is performed across a number of noise levels, where the standard deviation of the noise represents 0%, 5% and 10% of the range of each function (difference between the min $f$ and max $f$).

The approach for human-in-the-loop Bayesian optimisation is assessed across both UCB and EI utility functions. In the noisy case, when $\sigma > 0$ (Gaussian distributed observation noise) an approximation is applied with the expectation taken over either the UCB or EI acquisition function across eight noiseless Gaussian processes, each trained with a sampled set of outputs from a Gaussian process trained with noise. In the noiseless case, the standard form of EI or UCB is used to assess solution utility. To locate the optimum of the acquisition function, gradient-based optimisation is performed using a multi-start of 256 initial solutions evaluated in parallel, using the L-BFGS-B solver [182] and a low tolerance of 1E-10. Analytical gradients are obtained using the JAX library [15]. A number of different approaches are available to locate a standard solution that optimises the acquisition function. In this chapter a multi-start gradient-based approach is used due to the ability to obtain cheap analytical gradients. However, the impact of using an evolutionary approach is investigated within

Appendix C.4. The multi-objective problem outlined in Eq. 5.6 is solved using the NSGA-II algorithm [28] with a population size of 50. Given these are test functions initial solutions are generated using a static Latin-hypercube design-of-experiments with eight solutions, selected to provide a stable initial Gaussian process model across all problems.

For each standard optimisation test function and respective noise level, Bayesian optimisation is performed for 48 iterations a total of 32 times with different initial conditions, and take the average regret across all runs. When optimising the functions from a Gaussian process prior of a given length scale and dimension, 50 unique functions are sampled and keep the initial conditions the same across all runs.

### 5.4.2 Pulsed-flow Coiled Tube Reactor optimisation

In addition to benchmark problems, the method is applied to two applied chemical engineering black-box optimisation tasks.

The first task concerns the joint optimisation of operating conditions and geometry of a pulsed-flow helical tube reactor as outlined in more detail in Chapter 3. Table 5.2 outlines the specific parameters involved.

Table 5.2 The parameters and respective bounds for the optimisation of pulsed-flow reactor geometry and operating conditions.

| | **a (m)** | **f (Hz)** | **Re** | **p (m)** | **r (m)** |
|---|---|---|---|---|---|
| Lower Bound | 0.001 | 2 | 10 | 0.0075 | 0.005 |
| Upper Bound | 0.008 | 8 | 50 | 0.015 | 0.015 |

The variable $a$ represents the amplitude of pulsed-flow at the inlet of the reactor, $f$ represents the pulsed-flow frequency, $Re$ represents the Reynolds number, which in turn defines the velocity at the inlet of the reactor. Finally, $p$ and $r$ represent the pitch and radius of the coil respectively.

In order to enable rapid benchmarking, 75 reactor configurations were distributed throughout the design space using a Latin hypercube design of experiments, and each evaluated in OpenFOAM. A Gaussian process was then trained using this data, and the resulting mean used as the ground-truth function.

For this case study, previous visualisations of each result in the form of the resulting flow-field would typically be available for the expert to inspect. However, as a dataset of solutions is used which is fit to a Gaussian process providing a surrogate the true function, this information is not available. We can however visualise solutions before evaluation in the form of plots of both geometry and operating conditions, which is made available to assist the expert in making discrete choices. An example of this solution visualisation is shown in Fig 5.4.

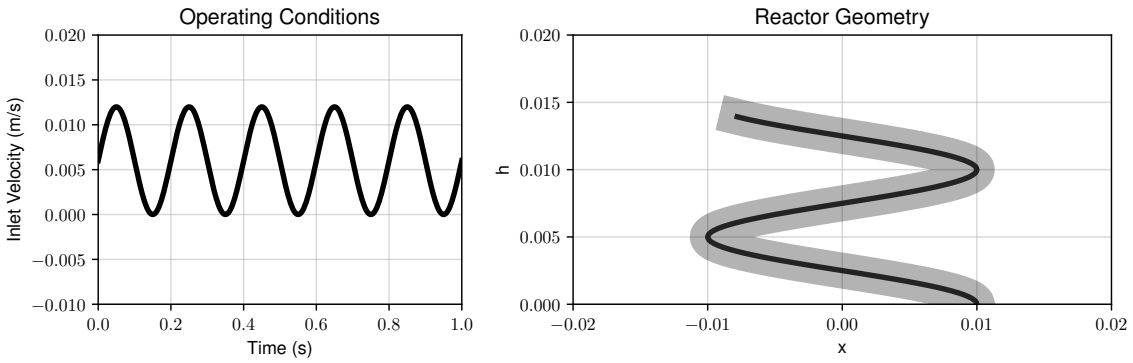



Fig. 5.4 A reactor configuration visualised in order to aid the expert in making discrete choices. In doing so, the expert had a reduced need to interpret higher-dimensional decision vectors.

### 5.4.3 Black-box Bioprocess optimisation

This task concerns the fed-batch optimisation of phycocyanin synthesised by cyanobacterium Arthrospira platensis via photo-production. To evaluate the system, non-smooth bioprocess dynamics are integrated for a total of 200 simulated hours. Details of the dynamics can be located in the work by Petsagkourakis et al. [118]. For the purposes of this case study it is

assumed that the dynamics of the reaction are not able to reliably be learned due to the time involved to collect data and the small amount of data, necessitating a black-box approach. Both nitrogen feed rate and light intensity serve as the optimisation variables, and can be changed eight equally spaced times throughout the time-horizon at values between 0-400 $mgL^{-1}$ and 0-40 Lux respectively. The objective is to maximise the total end concentration of phycocyanin product.

This black-box bioprocess case-study represents an expensive high-dimensional experimental system with a non linear objective, where domain knowledge may be used alongside other contextual information to provide potential improvements.

To aid the expert in making a decision, a visualisation of the proposed control scheme is provided corresponding to a given solution. As this is a simulated case study, it is postulated that online measurements are available after each control scheme has been evaluated. This additional information, which is reasonably available in the engineering domain, may enable the expert to reason about the choices based on previous evaluations more easily. Figure 5.5 demonstrates this result visualisation made available for all evaluated control schemes.

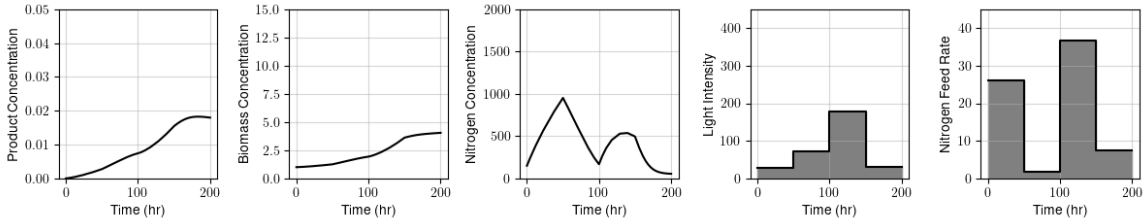



Fig. 5.5 A reactor configuration visualised in order to aid the expert in making discrete choices. In doing so, the expert had a reduced need to interpret higher-dimensional decision vectors.

Figure 5.6 demonstrates an example of the information made available about the four alternative solutions, to the expert.

The optimisation variables and respective bounds are shown in Table 5.3.

Where $I$ represents the constant value of light intensity between two fixed times denoted within the table, and $F_n$ represents the respective inflow rate of nitrogen to the reactor. We

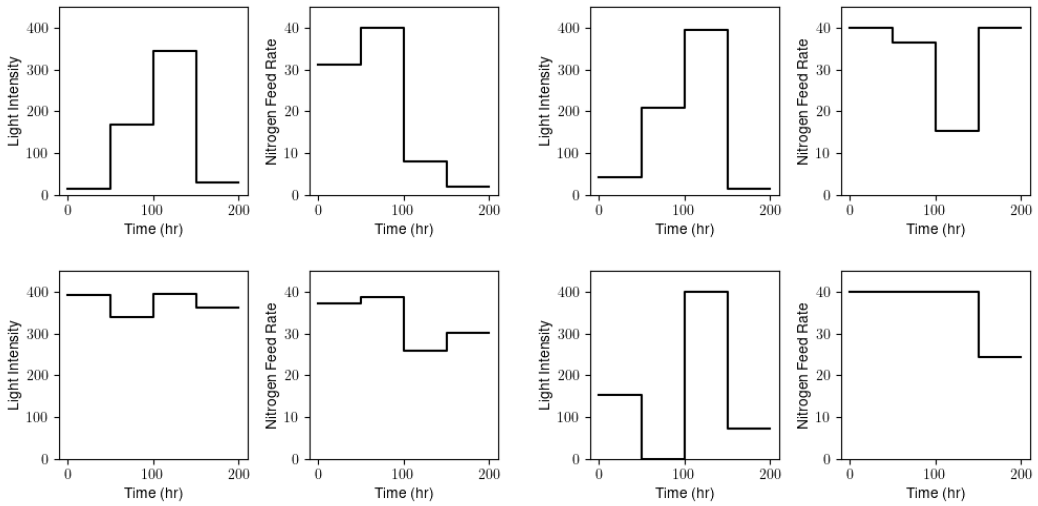



Fig. 5.6 Four alternative control solutions, as made available to the expert in order to facilitate easier reasoning particularly in higher dimensions.

Table 5.3 The parameters and respective bounds for the optimisation of a fed batch bioprocess.

| **Bound** | **I (0-50)** | **I (50-100)** | **I (100-150)** | **I (150-200)** | **$F_n$ (0-50)** | **$F_n$ (50-100)** | **$F_n$ (100-150)** | **$F_n$ (150-200)** |
|---|---|---|---|---|---|---|---|---|
| L | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| U | 400 | 400 | 400 | 400 | 40 | 40 | 40 | 40 |

note that knowledge of the underlying dynamics are not made available to the humans performing the selection of the solutions, and only basic contextual information is provided that would reasonably be available to an expert.

Figure 5.7 demonstrates an instance of the textual information made available to the expert at each iteration, including information regarding the previous iterations and the choices available.

All code can be located within the repository https://github.com/OptiMaL-PSE-Lab/humbo/.

Below are the previous 5 iterations and objectives.

| Objective | I(0-50) | I(50-100) | I(100-150) | I(150-200) | Fn(0-50) | Fn(50-100) | Fn(100-150) | Fn(150-200) |
|---|---|---|---|---|---|---|---|---|
| 0.0215613 | 131.191 | 189.19 | 89.648 | 223.75 | 36.567 | 14.937 | 31.371 | 9.567 |
| 0.0254919 | 73.899 | 203.248 | 376.875 | 100.243 | 18.181 | 9.736 | 28.862 | 25.63 |
| 0.0307188 | 198.445 | 89.908 | 350.6 | 309.593 | 31.282 | 14.422 | 32.851 | 37.276 |
| 0.0151289 | 181.478 | 287.325 | 236.092 | 132.862 | 4.454 | 2.18 | 8.052 | 35.804 |
| 0.0101658 | 99.114 | 398.827 | 242.546 | 223.298 | 33.358 | 1.155 | 8.997 | 9.395 |

Here are the current solution choices:

| Choice | Utility | Sol Mean | Sol STD | I(0-50) | I(50-100) | I(100-150) | I(150-200) | Fn(0-50) | Fn(50-100) | Fn(100-150) | Fn(150-200) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 4.859 | 0.0228799 | 0.027104 | 199.783 | 88.116 | 350.298 | 316.622 | 31.409 | 14.514 | 32.209 | 36.341 |
| 2 | 4.859 | 0.0208041 | 0.0272468 | 142.077 | 266 | 333.015 | 48.546 | 26.261 | 30.578 | 24.305 | 28.053 |
| 3 | 4.859 | 0.0208041 | 0.0272468 | 371.439 | 287.84 | 214.561 | 376.934 | 7.355 | 17.148 | 38.747 | 26.816 |
| 4 | 4.859 | 0.0208041 | 0.0272468 | 59.608 | 235.294 | 396.863 | 123.922 | 27.451 | 8.157 | 39.216 | 14.275 |

Remember to check solution visualisations in bo/bioprocess_profile_human/tom//choices/ before making a choice.
Enter choice:

Fig. 5.7 An instance of the textual information made available to the expert at each iteration.

### 5.4.4 Human-in-the-loop Experimental Protocol

To evaluate the performance of this expert-guided Bayesian optimisation approach with real human input, experiments were conducted with human participants on the fed-batch bioprocess optimisation and reactor design case studies.

All participants were chemical engineering postgraduate students with knowledge of optimisation, design, and process control. Each participant was provided with an introduction to the problem, including the context, optimisation variables, objective function, and any relevant background information that an expert in the field would typically have access to. Participants were then given the opportunity to select up to eight initial solutions to initialise the experimental design. They were encouraged to apply their domain knowledge in making these selections.

Participants then engaged in the iterative Bayesian optimisation process, consisting of: Reviewing information from previous iterations, including visualisations of evaluated control strategies and their corresponding objective values; Analysing the current set of alternatives, which included predicted performance metrics (e.g., mean objective value) and visualisations

of the proposed control strategies; Selecting the most promising solution from the alternatives based on their expertise and the provided information.

The selected solution was evaluated, and the results were added to the dataset for the next iteration. The optimisation process continued for 32 iterations. Objective values and convergence metrics were recorded throughout the optimisation process for each participant. In addition, qualitative data such as comments and questions were recorded throughout.

For comparison, standard Bayesian optimisation (without human input) was performed 32 times for each case study to generate a baseline for expected convergence. The experiments were designed to mimic a realistic expert-guided optimisation scenario while providing controlled conditions for comparing the performance of human-in-the-loop optimisation against standard Bayesian optimisation.

## 5.5 Results & Discussion

Firstly, the effect of different problem features on convergence is presented, enabling conclusions regarding the performance of the collaborative approach. By comparing the convergence of standard Bayesian optimisation with stochastic expert policies conclusions can be made as to the potential of the expert-guided approach to improve convergence across a range of features. Specifically, this enables conclusions regarding when our approach is favourable over standard Bayesian optimisation with respect to expert ability, problem class, and number of alternative solutions.

### 5.5.1 Effect of Noise

To investigate the effect of objective noise, Figure 5.8 demonstrates the results of the approach across general 2D and 3D objective functions sampled from a Gaussian process prior, with

differing levels of noise. Only the mean performance is displayed as opposed to including the standard deviation of regret to maintain the interpretability of the results.

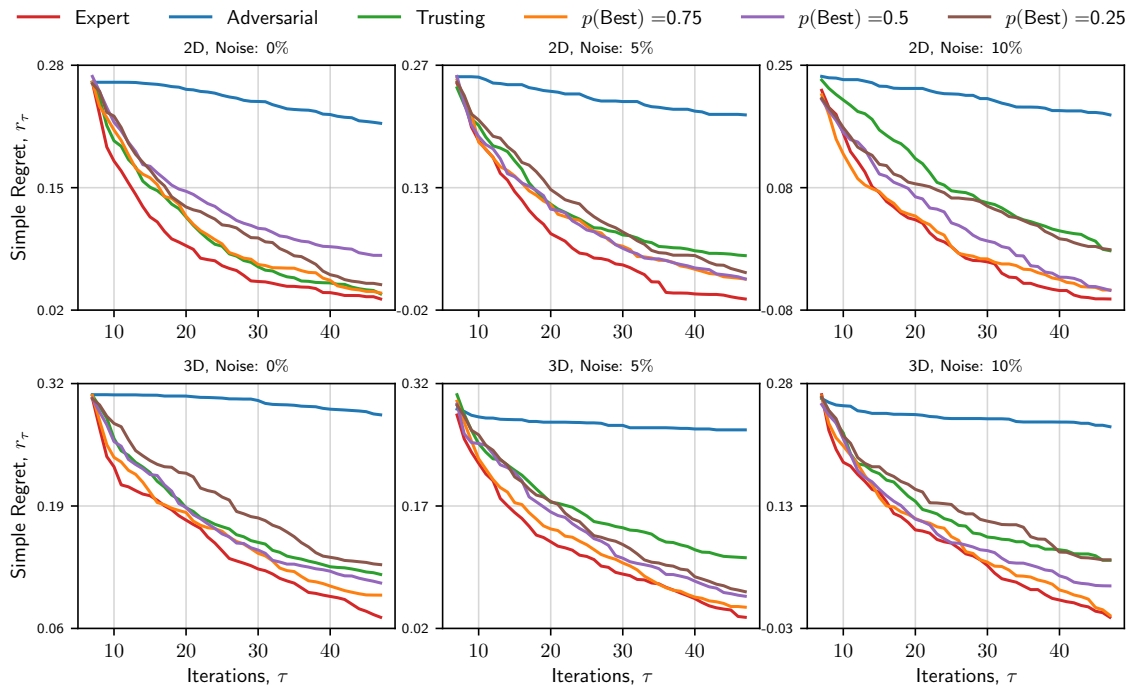



Fig. 5.8 The effect of the level of observation noise on the difference in convergence for various expert behaviours across general 2D and 3D functions of a given length scale.

The adversarial or misaligned selection policy presents an upper bound on the rate of convergence, and the expert policy generally provides a lower bound on rate of convergence for non-myopic (one-step ahead) Bayesian optimisation. Generally, increasing the ability of the expert, by increasing the probability of selecting the best choice results convergence between these upper and lower bounds. Standard Bayesian optimisation which is denoted the 'trusting' policy (selects the solution with the largest utility value) also falls between these bounds.

Observing the convergence results for 2D functions, with no observational noise convergence is only faster than standard Bayesian optimisation when the expert selects the best solution with 75% accuracy. At a 5% observational noise level, the required accuracy to recover standard convergence drops to 25% with four alternate choices, reflecting the noise level at

which an expert selecting randomly recovers the standard level of convergence. This trend continues when increasing to 10% observational noise, where an expert who selects randomly becomes more favourable than standard Bayesian optimisation.

In higher-dimensional noisy problems, the quality of the expert required to recover the convergence of standard Bayesian optimisation reduces across all noise levels. Performance across all noise levels across general 3D functions is similar, indicating that benefits are maintained in the presence of noise.

Based on these results it can be concluded that the approach becomes more competitive for problems with greater observation noise, as the quality of expert required to improve on standard convergence becomes lower.

In addition, it is hypothesised that this methodology is favourable in noisy Bayesian optimisation settings as it enables an expert to separate signal to noise internally, and to focus on prior 'non-noisy' trends, reducing the overall number of iterations.

### 5.5.2 Effect of Dimensionality

To demonstrate the effect of problem dimensionality, Figure 5.9 presents the convergence results with respect to two, five, and ten dimensional functions with 2.5% noise using the expected improvement acquisition function. The adversarial policy has been omitted to improve interpretability of the plot, as the value of regret is significantly and consistently higher than other behaviours.

Relative differences in convergence between quality of experts across all functions and increasing in dimensionality is similar. The convergence of a random expert ($p = 0.25$) matches or exceeds standard Bayesian optimisation (trusting), indicating that in some problem settings the increased exploration that results from the random selection of a choice generated specifically through an evolutionary approach, is beneficial. As dimension increases the gap between the performance of an expert practitioner and standard Bayesian optimisation

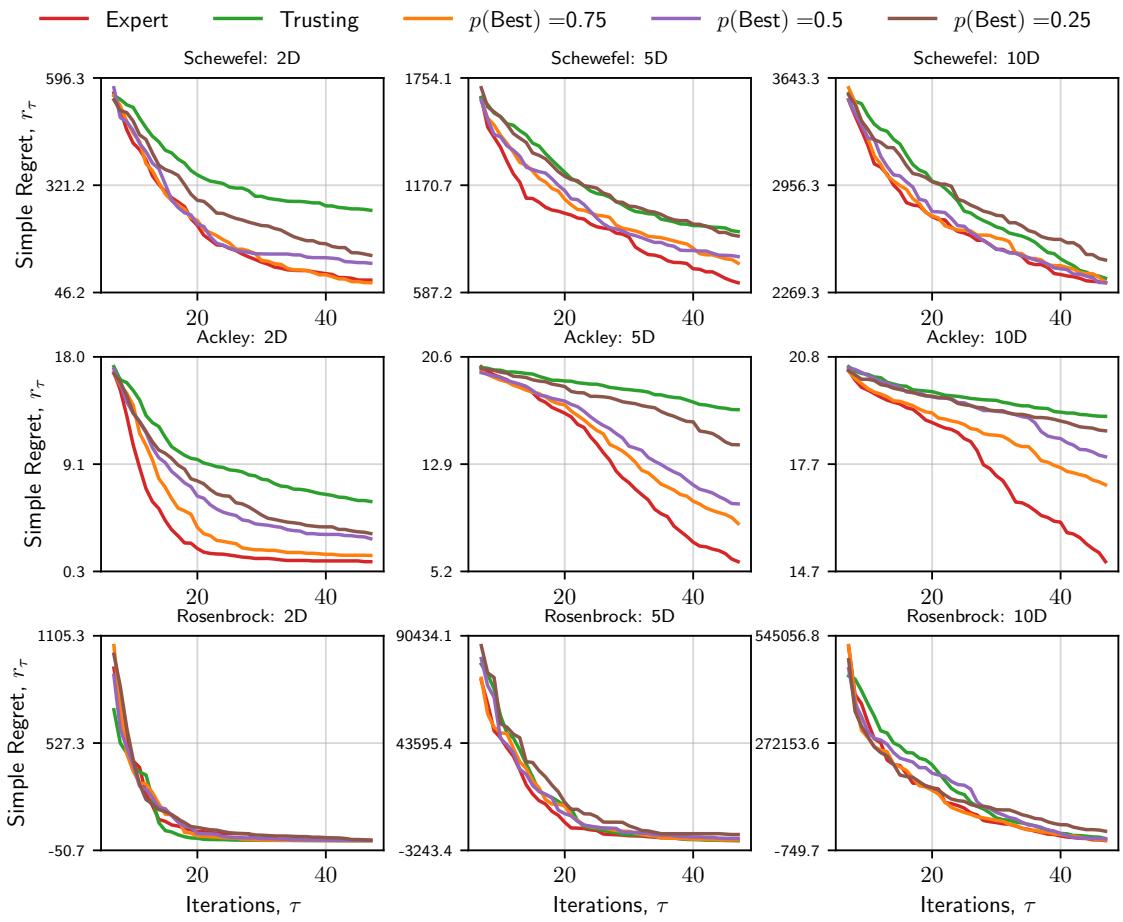



Fig. 5.9 The effect of the dimensionality of the problem on the difference in convergence for various expert behaviours.

diminishes. However, any expert that performs better than randomly still improves on the convergence of standard Bayesian optimisation.

This would indicate that the benefits of human-in-the-loop approaches diminish in higher dimensional cases, disregarding the ability to select specific solutions at the beginning of the process, though this is omitted for benchmarking purposes.

These results demonstrate that benefits of our approach are maintained in reasonably high dimensional and practical problem settings.

### 5.5.3 Effect of Number of Alternative Solutions

The number of alternative solutions made available the hypothesised experts is now investigated with respect to performance. It is noted that making more choices available has the potential to increase performance, as more solutions are made available to the expert, any one of which may be spotted as high performing. However, a human will be able to reason less effectively about the relative differences between a larger amount of solutions and may perform worse, indicating a trade-off in practice. When benchmarking computationally there is no consideration as to the increased difficult ability to reason about larger numbers of solutions due to the assumption of an underlying knowledge of the true function. Figure 5.10 demonstrates the convergence results when increasing the alternate solutions from three to six for the two dimensional Ackley function.

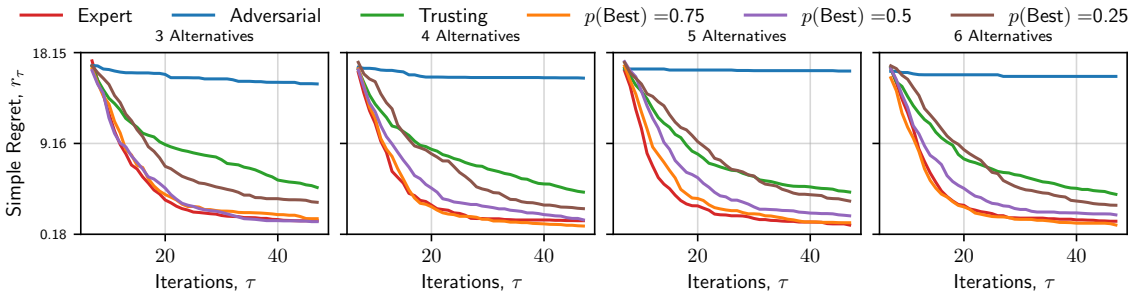



Fig. 5.10 The effect of the number of alternative solutions made available on convergence for various expert behaviours.

Convergence generally becomes faster as a larger number of alternative solutions are made available at each iteration. Adversarial practitioners become slightly worse with more potential solutions, as there is more scope to make a poor selection. While overall choice increases alongside the potential for improvement, it becomes more important to ensure that the expert is not misaligned.

To summarise, a larger number of alternatives provides increased scope for convergence improvements. However, at the same time it becomes more important to be confident in the quality of the expert due to increased risk of misguided selections.

### 5.5.4 Effect of Specific Function Properties

The Rastrigin function [129] is highly multi-modal, with a wide bowl-shape punctuated by individual peaks. Figure 5.11 shows convergence on the 2D Rastrigin function with no observation noise.

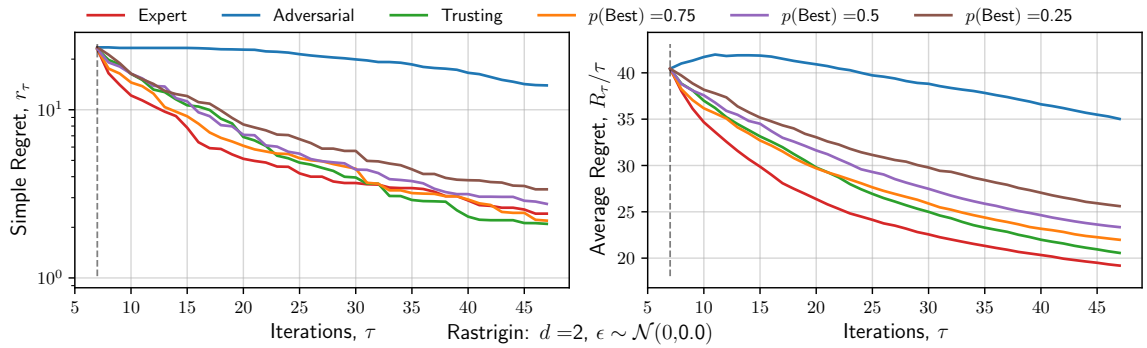



Fig. 5.11 Convergence of collaborative Bayesian optimisation of the 2D Rastrigin function with no observation noise.

The convergence distribution for all policies are tight, with the exception of the hypothetical adversarial practitioner. Initially, expert policies have the same performance than standard Bayesian optimisation, highlighting the ability for experts to influence Bayesian optimisation in early stages. However, this benefit is negated in the later stages of optimisation.

Whilst the Ackley function has similar multi-modal characteristics to the Rastrigin function, the underlying form is less bowl shaped, resulting in the the value of the global optima being significantly different to the value of other local optima. Figure 5.12 shows convergence on the 2D Rastrigin function with no observation noise.

The distribution of convergence characteristics between policies is widened, and therefore the ability for the expert to provide an improvement is increased. In fact, selecting randomly from the four alternate solutions improves on standard Bayesian optimisation, with policies that are better than random providing further improvements. It is hypothesised that in this situation the underlying GP may have a higher length scale than is representative of the underlying function, enabling the expert to spot high performing solutions that are

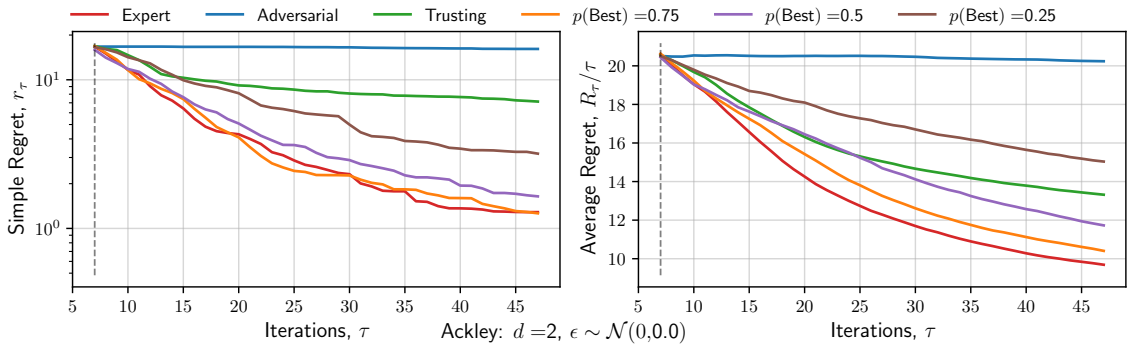



Fig. 5.12 Convergence of collaborative Bayesian optimisation of the 2D Ackley function with no observation noise.

not predicted. Differences between these two functions, and the increased expert-guided exploration highlights the ability of our approach to identify solutions in situations where the value of the global optima is potentially significantly different to the value of other local optima.

Finally, a more complex five dimensional multi-modal case with 5% observation noise is tested. Figure 5.13 shows convergence on the 5D Schewefel function with 5% observation noise.

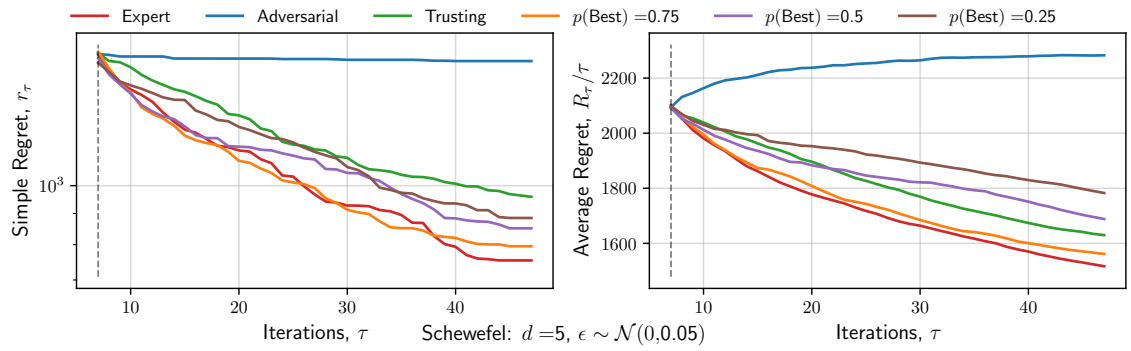



Fig. 5.13 Convergence of collaborative Bayesian optimisation of the 5D Schewefel function with 5% observation noise.

The random expert with a 25% chance of selecting the best solution out performs standard Bayesian optimisation, indicating that in high-dimensional multi-modal problems the additional exploration achieved through the selection of alternate solutions is favourable.

### 5.5.5 Coiled Tube Reactor Design

In addition to using stochastic expert policies to benchmark our approach based on expert skill and problem type, we also apply the approach to a real-world chemical engineering case study involving human participants. These subjects are all chemical engineering postgraduate students, with knowledge of optimisation, design, and broadly control.

Six participants were used to generate the human results, and standard Bayesian optimisation was performed 32 times to generate the expected convergence metrics. Figure 5.14 presents the convergence results for standard Bayesian optimisation and our approach when simultaneously optimising the geometry and operating conditions of a pulsed-flow coiled tube reactor.

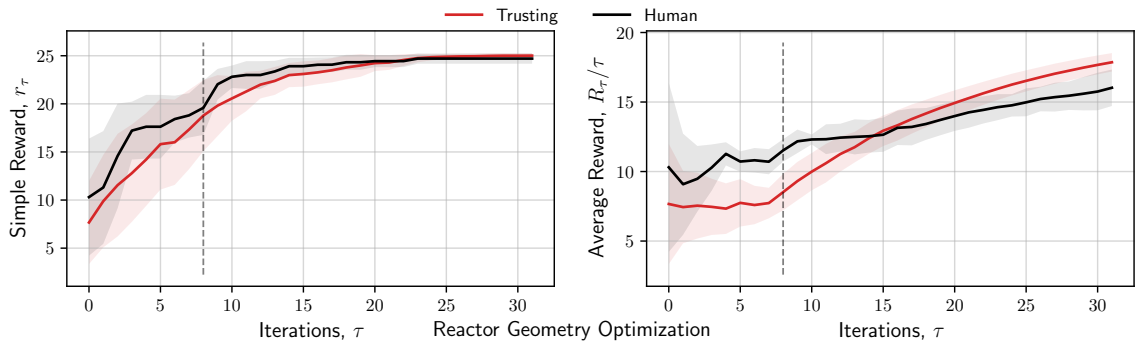



Fig. 5.14 Simple and average reward, corresponding to the number of equivalent tanks-in-series for a given reactor geometry and operating conditions, presented alongside the reward obtained from standard Bayesian optimisation. Filled sections represent one standard deviation and solid lines represent the mean reward.

By applying collaborative Bayesian optimisation reactor configurations were able to be located with a greater equivalent tanks-in-series (i.e. better performing), maintaining an

improvement over standard Bayesian optimisation throughout the design process. As the iterations progress the advantage diminishes, consistent with observations in literature. While average reward is more commonly applied in reinforcement learning settings, as opposed to Bayesian optimisation where only the best solution is of interest, in this situation it provides clear evidence for this diminishing advantage.

This case study demonstrates the potential benefit of including human expertise and intuition to enhance the optimisation of real-world systems such as the simultaneous optimisation of reactor geometry and operating conditions.

### 5.5.6 Fed-batch Bioprocess Control

For this case study, each participant is given the problem context described above, and the ability to select any number of initial solutions below 8. They can view previous solutions, a visualisation of current choices at each iteration, and choice information such as predicted mean objective.

Figure 5.15 demonstrates the average and standard deviation of reward, corresponding to the final product concentration of the participants against standard Bayesian optimisation.

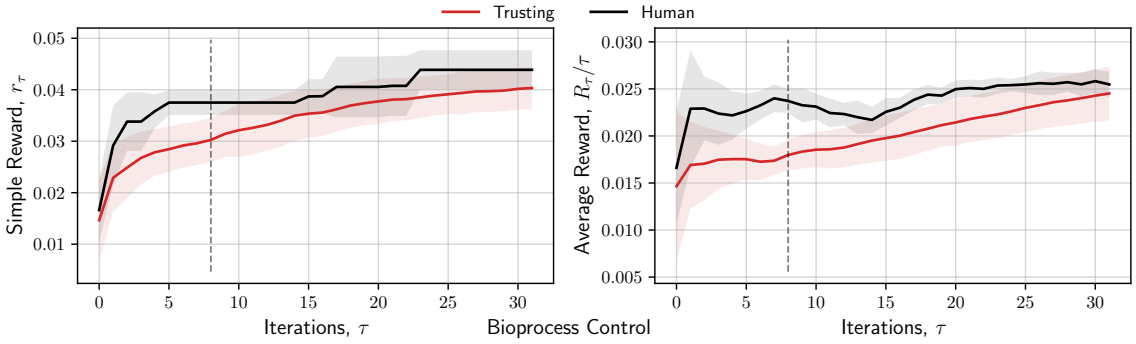



Fig. 5.15 Simple and average reward presented alongside the reward obtained from standard Bayesian optimisation. Filled sections represent one standard deviation and solid lines represent the mean reward.

On average, the inclusion of human-input throughout Bayesian optimisation improves the convergence when compared to standard Bayesian optimisation. It is noted that significant improvements are made during the initial sampling stage, where the participants had the ability to select initial solutions.

Initial solutions selected largely followed heuristics such as constant light or nitrogen feed rate, or ramps up or down. These heuristics prove more effective in this case study than a random control profile providing improved initial convergence. Overall, it can be demonstrated how the combination of the initial data generation, and continuous input contribute to improved performance.

### Qualitative Feedback

When performing these experiments, an amount of qualitative feedback was generated from the participants as they were performing the task. In this subsection a number of these thoughts are presented for discussion in groups.

> ***Participant 1****: I wish I could say that I don't like* ***any*** *of the solutions.*
>
> ***Participant 3****: Can I say that I don't want to see that solution again?*

Expressing a negative sentiment for a solution was a common theme among participants. In the event that no solutions are desirable, the framework does enable the expert to simply choose their own (though in benchmarking this was not allowed). Alternatively, the solution with the largest utility value can be selected. It was decided not to incorporate the ability to express a negative choice as it may result in biasing the search space (through a weighted function or otherwise). Future work may consider the inclusion of preference-based models as well as choice, however when taking this 'negative' viewpoint care must be taken that globally optimal solutions are not erroneously disregarded. In comparison the approach presented in this chapter can be considered broadly 'optimistic' as no human bias is able to be introduced to the model itself, only at the stochastic decision making stage.

***Participant 2****: I know this won't be a good solution, but I'll select it to stop it getting recommended.*

***Participant 3****: I will choose this one because I want to see more of that type of profile*

In these discussions, participants demonstrate non-myopic reasoning (that is, the preference or ability to make decisions that optimise beyond one-step ahead). This reasoning can be included within standard Bayesian optimisation, however our acquisition function remains myopic avoiding the computational expense of a non-myopic acquisition function. For a full treatment, a non-myopic acquisition function may be used. Participant two raises an undesirable feature of our approach which, where undesirable solutions that a human chooses not to select are not evaluated, and therefore consistently proposed. Non-myopic reasoning (selecting an undesirable solution) serves to break the deadlock in this case, but future work may be concerned with ensuring that solutions are selected that are different from previous iterations.

## 5.6 Conclusions

In this chapter an approach for including expert-opinion continuously throughout Bayesian optimisation is presented, empirically minimising expert-time and effort. It relies on the proposal of a number of promising solutions from which the expert selects, these solutions are defined through the combination of high-throughput Bayesian optimisation and multi-objective optimisation. For a large class of problems this enables improvement on standard Bayesian optimisation, with benefits maintained across high dimensional, noisy, and real world case studies. It is important to highlight that for certain problems, incorporating expert knowledge only matches the convergence of standard Bayesian optimisation. Moreover, identifying the specific classes of problems where human input provides significant improvements

remains challenging, making it difficult to predict the potential benefits of human-in-the-loop optimisation a priori.

The improved performance of the random expert over standard Bayesian optimisation in some cases can also be attributed to the effectiveness of the evolutionary optimisation method on specific classes of problems. For multi-modal problems such as the Ackley function, standard gradient-based methods may struggle to locate the best performing solutions. In these cases, the methodology provides improvement simply by applying an evolutionary approach, overcoming the many local optima. A detailed comparison between using an evolutionary against a gradient-based approach to locate the 'true' optimal solution is provided within Appendix C.3.

The approach is validated using human participants on a real-world bioprocess optimisation case study, as well as for the simultaneous design of coiled tube reactor operating conditions and geometry, achieving improved convergence over standard Bayesian optimisation.

This chapter presents a step towards the inclusion of domain experts within Bayesian optimisation, linking discrete decision making with existing state-of-the art computational methodologies to support the optimisation and discovery of engineering systems.

# Chapter 6

# Accelerating Convergence with LLM-in-the-Loop Bayesian Optimisation

## Contents


**6.1 Introduction . . . . . 164**

6.1.1 Contributions . . . . . 166

**6.2 Background . . . . . 167**

6.2.1 Human-in-the-Loop Optimisation . . . . . 167

6.2.2 Large Language Models for Scientific Tasks . . . . . 167

**6.3 Methodology . . . . . 168**

6.3.1 Generating Candidate Solutions . . . . . 169

6.3.2 Parsing Candidate Solutions for LLMs . . . . . 170

6.3.3 LLM Decision-Making Procedure . . . . . 171

6.3.4 Output Format and Selection Decision . . . . . 176

**6.4 Results . . . . . 176**

6.4.1 Performance on Classical Test Functions . . . . . 176

6.4.2 Comparison of Prompting Strategies . . . . . 179

6.4.3 Performance on Random Functions . . . . . 181

6.4.4 Chemical Optimisation Case Study . . . . . 184

**6.5 Conclusions . . . . . 187**

# Publication Note

This chapter has been published online as a preprint as:

Savage, T., & del Rio Chanona, E. A. (2024). *Accelerating Convergence with LLM-in-the-Loop Bayesian Optimisation*.

The work presented in this chapter explores how large-language models can be incorporated into Bayesian optimisation workflows to improve the optimisation convergence while maintaining full automation.

## Author Contributions

- Tom Savage: Conceptualisation, methodology, software implementation, data analysis, visualisation, and writing of the original manuscript
- Dr Ehecatl Antonio del Rio Chanona: Project supervision, conceptual guidance, and manuscript review

## Modifications from Preprint Version

The following modifications have been made to the published version:

- Spelling has been changed from American to British English.
- Some language has been altered to better fit within the context of this thesis.

# Abstract

In this chapter I present in-silico improvements to the convergence of Bayesian optimisation using large language models as computational decision-makers. Methods for human-in-the-loop Bayesian optimisation have become increasingly popular due to the ability to incorporate valuable prior expert knowledge. However, the benefits of fully automated decision-making may limit their application for large-scale self-driving lab and scientific discovery tasks. In this work large language models are used to make the selection between candidate solutions. In doing so, the method recovers full automation of Bayesian optimisation whilst demonstrating convergence improvements gained from expert intervention. The approach is benchmarked on a large number of random and structured test functions across a number of dimensionalities as well as a challenging, high-dimensional chemical optimisation problem derived from experimental data, demonstrating improvements to closed-loop convergence. The approach is also presented as a novel test set leakage-proof benchmark for large-language models, prompts, and inference strategies, where improved performance associated with specific models and prompting strategies such as mixture-of-experts. It is hypothesised that LLMs are able to spot trends in data, and observe empirical improvements in all but random functions with small length scales. This approach illustrates how LLMs can be combined with Bayesian optimisation in a sustainable and robust manner, mitigating against hallucinations and mistakes, resulting in improved scientific discovery pipelines and more effective self-driving laboratories.

## 6.1 Introduction

As demonstrated previously in this thesis (Chapters 3,4, and 5), domains such as drug-discovery, self-driving labs where decisions are made autonomously, and modern reactor design and discovery require solving expensive black box optimisation problems. Improving methods to solve these problems may lead to faster, more efficient scientific discovery, with applications ranging from new flow reactor designs to novel battery technologies [141, 38]. The solutions to these problems have traditionally been guided by human experts including chemists, biologists, and engineers, who design experiments based on domain knowledge and intuition.

In opposition to this, Bayesian optimisation (BO) has emerged as a powerful automated approach for such problems. By combining probabilistic modelling with decision making under uncertainty, BO identifies promising experiments while minimising expensive function evaluations. Its derivative-free nature and efficiency compared to alternatives like Nelder-Mead, metaheuristics, or DIRECT methods make it particularly attractive when objective functions are expensive to evaluate [104, 62]. The ability to automate this procedure has made BO a pivotal tool within self-driving laboratories to support scientific discoveries.

Work presented in the previous chapter on human-in-the-loop BO, as well as in the wider literature, has demonstrated significant improvements in optimisation performance through expert knowledge integration [82, 139]. These approaches typically involve experts making periodic interventions in the optimisation process, selecting promising regions of the search space, or adjusting algorithm parameters based on observed trends [75, 57]. However, this creates a fundamental trade-off. While human input may accelerate convergence, it comes at the cost of full and continuous automation that makes BO valuable in applications like self-driving labs [18]. This chapter therefore aims to answer the following question: can the benefits of expert knowledge be included within the loop while maintaining fully automated optimisation and discovery?

The emergence of large language models (LLMs) as powerful reasoning tools presents a potential solution. Recent work has shown LLMs' capabilities in scientific reasoning, mathematical problem-solving, and pattern recognition [84]. However, their application to scientific discovery has been limited by challenges in evaluation, verification, and the risk of hallucination. Optimisation provides a unique opportunity for LLMs to contribute to science and engineering, where decisions can be evaluated quantitatively through an objective function, creating a robust framework for assessing LLM reasoning capabilities and grounding decisions based on predefined metrics.

In this chapter, a novel approach is presented that addresses these challenges by using LLMs as computational experts within the BO framework. The methodology presented maintains full automation while demonstrating convergence improvements similar to those gained from expert intervention. By using LLMs to make selections between discrete candidate solutions, two key benefits are achieved: preserving the automated nature of BO while incorporating pattern recognition capabilities that mimic expert intuition.

The approach is validated across both structured and random test functions of varying dimensionality, demonstrating significant improvements in optimisation performance for structured problems. Notably, it is observed that LLMs effectively identify trends in the optimisation trajectory, leading to enhanced convergence in all cases except random functions with small length scales. However, the success on the chemical case study underscores that many real-world scientific problems, despite their complexity, possess sufficient underlying structure for LLMs to identify and exploit, leading to performance gains. This aligns with the intuition that LLMs, like human experts, excel at recognising meaningful patterns but struggle with purely random data. This work presented in this chapter sits at the intersection of two important trends in scientific AI: making LLMs more reliable for scientific applications and capturing expert decision-making processes in closed-loop optimisation methodologies.

### 6.1.1 Contributions

The contributions are threefold: First, a framework is introduced for integrating LLMs with Bayesian optimisation that preserves full automation while capturing expert-like insights. Second, empirical and quantitative evidence is provided across both synthetic benchmarks and a high-dimensional chemical optimisation case study derived from experimental data, that LLMs can effectively serve as computational experts in optimisation tasks, particularly for structured problems and realistic applications. Third, this approach is presented as a novel, data-leakage-proof benchmark for evaluating LLM capabilities in scientific decision-making tasks, offering a quantitative framework for assessing LLM reasoning in optimisation contexts.

By demonstrating how LLMs can be combined with BO in a robust manner that mitigates potential hallucinations and mistakes, this final chapter demonstrates a pathway towards more effective automated optimisation systems that benefit from AI-driven expertise while maintaining full automation. Furthermore, our benchmark framework offers a new approach for evaluating and improving LLM performance in scientific decision-making tasks, with direct implications for their broader application in scientific discovery.

The remaining content of this chapter is organised as follows. Section 6.2 provides background on existing human-in-the-loop approaches for Bayesian optimisation as well as the use of LLMs in scientific tasks. Section 6.3 presents the LLM-enhanced BO methodology and prompting framework. Section 6.4 analyses results across various test functions, comparing performance against standard BO and expert benchmarks, culminating in the chemical optimisation case study. Finally, section 6.5 concludes with implications for human and non-human in the loop BO, self-driving laboratories and future directions.

## 6.2 Background

This section provides a background on human-in-the-loop approaches to Bayesian optimisation. A background on the main prerequisites (Gaussian processes and Bayesian optimisation) can be found in Chapter 2.

### 6.2.1 Human-in-the-Loop Optimisation

Human-in-the-loop optimisation approaches take advantage of expert knowledge to guide the optimisation process in a number of ways. These include search space design and constraint specification [117], selection between candidate solutions [139], early stopping and restart decisions [1], and hyperparameter tuning and kernel selection [177].

Recently, and in the previous chapter, these interventions have demonstrated success over fully automated approaches in theory and in practice [177, 139]. For example, chemists' intuition about reaction conditions has helped identify promising regions of search spaces, while process engineers' knowledge of operating constraints has guided optimisation toward practically feasible solutions [146]. However, these interventions introduce human-dependent bottlenecks that can limit throughput in automated systems, particularly in settings like self-driving laboratories where continuous operation is desired.

The success of human experts often comes from their ability to either recognise and reason about patterns from past experiments [21], apply domain-specific heuristics [178], or make informed choices between competing alternatives [69].

### 6.2.2 Large Language Models for Scientific Tasks

Recent developments in LLMs have demonstrated their ability to perform complex reasoning tasks. LLMs learn to process and generate text through pre-training on vast corpora of data,

developing capabilities that extend beyond simple pattern matching to encompass elements of planning and reasoning [87, 84, 173].

Recent work has shown that LLMs can perform tasks such as mathematical proof verification, scientific paper analysis, and experimental design [147]. Particularly notable successes include discovering novel materials through analysis of scientific literature, validating mathematical proofs, and proposing experimental protocols [152, 59]. Their ability to process sequential information 'in-context' makes them particularly well-suited for optimisation tasks where historical trajectory information can inform future choices.

Previous attempts to apply LLMs to scientific tasks have often struggled with their tendency to hallucinate or make unverifiable claims [162]. Common approaches to address these limitations include restricting output space through careful prompt engineering [67], implementing verification steps for model outputs [84], or combining LLMs with traditional scientific computing tools [22].

The framework presented within this chapter builds on these approaches while addressing their limitations. By constraining the LLM's role to selection between well-defined alternatives, natural boundaries are created for the model's decision space. The quantitative nature of optimisation provides immediate feedback on decision quality, allowing for objective evaluation of the LLM's reasoning capabilities. This creates a robust framework that exploits an LLM's pattern recognition and reasoning capabilities while mitigating the risk of poor decisions stemming from its limitations.

## 6.3 Methodology

This section outlines the approach for LLM-enhanced Bayesian optimisation. It begins by recapping the framework for generating candidate solutions using standard Bayesian optimisation approaches, equivalent to the methodology in the previous chapter. Subsequently, a structured mechanism is presented for preparing information regarding candidate solutions

for LLM-based decision-making. Finally, the LLM decision-making procedure is detailed, including various agent architectures, prompting strategies tested, and the experimental setup.

### 6.3.1 Generating Candidate Solutions

Similar to the expert-guided approach presented previously (Chapter 5), at each iteration $t$, a given utility function $\mathcal{U}$ is first maximised for a given dataset $\mathcal{D}_t := (x_i, y_i)_{i=1}^t$:

$$x^* = \underset{x \in \mathcal{X} \subseteq \mathbb{R}^d}{\arg\max}\, \mathcal{U}(x; \mathcal{D}_t), \tag{6.1}$$

resulting in the optimal next evaluation, $x^*$, from a standard utility sense (for example, expected improvement, EI). A set of $p-1$ alternative solutions, denoted by the matrix $X \in \mathbb{R}^{(p-1)\times d}$ where $X := [x_1, \ldots, x_{p-1}]^T$, is then generated by solving the multi-objective optimisation problem introduced in the previous chapter.

This involves maximising two competing objectives, the aggregate utility ($\hat{\mathcal{U}}$) which denotes the sum of the individual utilities of the alternative solutions within $X$:

$$\hat{\mathcal{U}}(X; \mathcal{D}t) = \sum_{i=1}^{p-1} \mathcal{U}(x_i; \mathcal{D}_t), \tag{6.2}$$

and the solution diversity ($\hat{\mathcal{S}}$), a measure capturing the variability among both the optimal solution $x^*$ and the alternative solutions $X$. This is quantified by the determinant of the covariance matrix $K_{X_{\text{aug}}}$ for the augmented set of solution vectors $X_{\text{aug}} = X \cup x^*$:

$$\hat{\mathcal{S}}(X, x^*) = |K_{X_{\text{aug}}}| \tag{6.3}$$

where $K_{X_{\text{aug}}} = [k(x_{\text{aug},i}, x_{\text{aug},j})]_{i,j=1}^{p}$, and $k(\cdot,\cdot)$ is the kernel function of the Gaussian process model. The multi-objective optimisation problem is formulated as:

$$[\mathbf{X}^*_{\hat{\mathcal{U}}}, \ldots, \mathbf{X}^*_{\hat{\mathcal{S}}}] = \max_{\mathbf{X}} \left( \hat{\mathcal{U}}(\mathbf{X}; \mathcal{D}_t), \hat{\mathcal{S}}(\mathbf{X}, \mathbf{x}^*) \right), \tag{6.4}$$

yielding a set of non-dominated solutions representing the Pareto front. The Python implementation uses the NSGA-II algorithm for this step. From the Pareto front, the knee-point solution $X^k$ is selected. This is achieved by normalising the objective values on the front and finding the solution closest to the ideal point (utopia point) in the normalised objective space, effectively balancing high aggregate utility and high diversity. The final set of $p$ candidate solutions presented to the LLM consists of the $p-1$ solutions from $X^*_k$ and the original optimal solution $x^*$.

### 6.3.2 Parsing Candidate Solutions for LLMs

A key challenge in applying LLMs to scientific tasks is ensuring that the information provided is structured in a way that enables effective reasoning while mitigating against hallucination. Unlike the previous chapter, it cannot be relied on that the 'expert' (in this case an LLM as opposed to a human) will reliably process the information. The approach presented addresses this through careful normalisation and structuring of the data provided in the prompt.

#### Data Normalisation and Formatting

All input variables (from candidate solutions) and previous objective values (from the history $\mathcal{D}_t$) are normalised to an integer scale of 0-100 using min-max normalisation across the entire set of available data (history + candidates). Following normalisation, each variable and objective value is rounded to the nearest integer. This normalisation serves multiple purposes. It creates a consistent scale for variables, making it easier for the LLM to reason

about relative values, irrespective of the original physical scales. It also prevents the LLM from making assumptions based on the absolute magnitude of variables, focusing instead on relative relationships and patterns. Finally, it improves tokenisation efficiency by representing numerical values as single tokens where possible, as illustrated in Figure 6.1. The historical data and candidate solutions are presented to the LLM in a structured format, typically JSON, clearly delineating past observations (inputs and corresponding normalised objectives) and the current set of alternative solutions.




Fig. 6.1 The solution in JSON format `{"x1":1.4056,"x2":2.3033,"x3":5.690,"x4":23.349}` tokenised using OpenAI's tokeniser `tiktoken`, pre-normalisation and rounding (above) and post-normalisation and rounding (below). Fewer tokens are used post-processing resulting in a single token for each variable value.

### 6.3.3 LLM Decision-Making Procedure

This chapter explores three distinct prompting strategies for LLM-based decision-making, representing different trade-offs between reasoning complexity, computational cost, and performance. Decisions are made using various LLMs including Claude 3 models (Sonnet 3.5) [5], DeepSeek models (Deepseek, Deepseek-Reasoner) [29], GPT-4 models (4o-mini) [110] in order to test comparative performance, highlighting the methodologies ability to serve as a test-set leakage-free benchmark for scientific discovery. Different approaches to decision-making are described as follows:

#### Basic One-Shot Selection

This simplified approach condenses the decision-making into a single LLM call for efficiency. The LLM receives the normalised historical data and the normalised candidate solutions. It

is prompted to directly select the most promising solution based on this information. The system prompt focuses on the essential task and uses XML tags for improved focus. The opening context is provided here:

```
<task>
You are a decision maker who must make a choice about which solution to
    select based on the provided information. Your role is to evaluate the
    alternatives and choose the best solution based on your expertise in the
    subject matter.
You will be provided previous solutions and their objective values
Your task is to choose a solution that you believe will perform the best
    based on the information provided.
You will be given a few alternative solutions in JSON format, and you must
    use this data to determine the key distinctions among the solutions.
</task>
```

This method prioritises speed and reduced computational cost, suitable when latency is a primary concern, potentially sacrificing depth of reasoning. An example of the full prompt can be found within the Appendix section D.1.

### Multi-Agent System

This approach employs a system of three specialised agents collaborating on the decision, separating aspects of the decision-making process in an attempt to improve reasoning abilities. **Historian Agent**: Analyses the optimisation history ($\mathcal{D}_t$) to identify patterns and trends associated with high and low objective values. The context and task of the historian agent is as follows, and the remaining can be found in Appendix section D.1:

```
<personality>
You are an analytical data historian who objectively examines solutions and
    identifies trends in the data that suggest both poor and good performance.
```

```
    Your role within the team is to provide balanced insights into the
    characteristics of solutions based on their objective values.
</personality>

<task>
Your task is to describe the properties of solutions with characteristically
    low and high objective values, identifying both bad and good aspects of
    the solutions.
You will be given previous solutions in JSON format, and you must use this
    data to determine which variables, or combinations of variables, and
    respective values are associated with low and high objective values.
</task>
```

### Differentiator Agent

This agent examines the current set of candidate solutions to highlight key distinctions between them. The context and task of the differentiator agent is as follows, and the remaining can be found in Appendix section D.1:

```
<personality>
You are a concise differentiator who identifies the key distinctions between
    solutions. Your role is to provide brief, clear descriptions of the main
    differences among the provided solutions.
Comment only on values of the variables and their relative differences.
</personality>

<task>
Your task is to succinctly describe the main differences between the provided
    solutions, focusing on the most relevant variables.
```

```
You will be given a few alternative solutions in JSON format, and you must
    use this data to determine the key distinctions among the solutions.
</task>
```

### Decision-Maker Agent

This agent integrates the analyses from the Historian and Differentiator, along with the candidate solution data, to make the final selection. The context and task of the decision maker agent is as follows, and the remaining can be found in Appendix section D.1:

```
    <task>
You are a decision maker who must make a choice about which solution to
    select based on the provided information. Your role is to evaluate the
    alternatives and choose the best solution based on your expertise in the
    subject matter.
You will be provided information about previous trends, which variables are
    most important, and the differences between solutions.
Your task is to choose a solution that you believe will perform the best
    based on the information provided.
You will be given a few alternative solutions in JSON format, and you must
    use this data to determine the key distinctions among the solutions.
</task>
```

This structured collaboration allows for more specialised and potentially deeper reasoning compared to the one-shot approach.

### Chain-of-Thought Reasoning

This strategy encourages the LLM to perform extensive self-questioning and iterative analysis, mimicking a detailed human thought process. This approach prompts the LLM towards

a more deliberative reasoning style, aiming to induce connected reasoning steps from the LLM at inference time and encouraging a thorough exploration of the decision space before selecting a solution. This behaviour can be optimised for during training time using methods such as Group Relative Policy Optimisation (GRPO) [145]. The chain of thought approach uses a system prompt emphasising thorough exploration and can be considered an extension to the one-shot approach, as well as orthogonal to the multi-agent approach. A sample is provided here but the full prompt can be found in Appendix section D.1. The following prompt is used in addition to the one-shot prompt:

```
You engage in extremely thorough, self-questioning reasoning. Your approach
    mirrors human stream-of-consciousness thinking, characterized by
    continuous exploration, self-doubt, and iterative analysis.

<format>
<contemplator>
[Your extensive internal monologue goes here]
- Begin with small, foundational observations
- Question
each step thoroughly
- Show natural thought progression
- Express doubts and
uncertainties
- Revise and backtrack if you need to
- Continue until natural resolution
</contemplator>

<response>
FINAL ANSWER HERE
```

```
</response>
</format>
```

This method aims for the most robust reasoning but incurs the highest computational cost due to the extended generation length.

### 6.3.4 Output Format and Selection Decision

Regardless of the decision-making approach, the LLM's final output is structured as a JSON object containing both the solution choice and a brief justification for the selection. For example:

```
{"choice": 3,
  "reason": "Based on previous iterations, solution 3 is optimal as it
    balances high values of x2 and x3, which are strongly correlated with
    good performance."}
```

The solution corresponding to the chosen index is then evaluated using the true objective function $f$, and the new data point (input and objective value) is added to the dataset $\mathcal{D}_{t+1}$ for the subsequent iteration.

## 6.4 Results

This section describes the results of benchmarking across multiple test functions and experimental conditions.

### 6.4.1 Performance on Classical Test Functions

Initial analysis focuses on three classical optimisation functions chosen for their distinct characteristics. Each function is optimised for both 2 and 5 dimensions, and considering 2, 3,

and 4 alternate candidate solutions to select from. Each configuration is then repeated 36 times with different initial datasets, sampled randomly.

Performance is evaluated through a two-stage normalisation process designed to enable fair comparison across different functions and approaches. First, raw objective values are normalised to a [0,100] scale using min-max normalisation across all methods:

$$y_{\text{normalised}} = 100 \cdot \frac{y - y_{\min}}{y_{\max} - y_{\min}} \tag{6.5}$$

This creates a consistent scale where 100 represents the best observed performance and 0 the worst across all approaches. Second, relative performance is calculated as a percentage of the improvement range between expert and adversarial baselines:

$$\text{Performance } (\%) = 100 \cdot \frac{y_{\text{method}} - y_{\text{adversarial}}}{y_{\text{expert}} - y_{\text{adversarial}}} \tag{6.6}$$

This metric quantifies how much of the potential improvement range each method achieves, with 100% indicating expert-level performance and 0% indicating adversarial-level performance. Table 6.1 highlights the results for the multimodal Ackley function where Sonnet 3.5 is used as an LLM alongside a multi-agent prompting strategy.

LLMBO shows significant improvements over standard BO in most scenarios, particularly in higher dimensions. For d=2, LLMBO achieves 69.58% of expert performance with two alternatives (a=2), substantially outperforming standard BO's 6.45%. In five dimensions (d=5), LLMBO consistently outperforms BO across all numbers of alternatives, with performance improvements ranging from 31.85% to 47.49% of expert performance. The only exception is at d=2, a=3, where standard BO slightly outperforms LLMBO (39.63% vs 33.18%). No-

Table 6.1 Benchmark results for the Ackley function using LLM-BO with Sonnet 3.5 and a multi-agent approach

| **Ackley Function** | | | | | | |
|---|---|---|---|---|---|---|
| | $d=2$ | | | $d=5$ | | |
| | $a=2$ | $a=3$ | $a=4$ | $a=2$ | $a=3$ | $a=4$ |
| Expert | 65.96 | 21.21 | 40.73 | 88.27 | 90.87 | 95.74 |
| Adversarial | 0.00 | 1.52 | 0.49 | 19.65 | 7.46 | 7.71 |
| BO | 4.25 | 9.32 | 3.24 | 31.08 | 31.24 | 27.08 |
| LLMBO | 45.89 | 8.05 | 5.40 | 50.36 | 47.07 | 35.75 |
| BO (% Performance) | 6.45 | **39.63** | 6.84 | 16.66 | 28.51 | 22.01 |
| LLMBO (% Performance) | **69.58** | 33.18 | **12.2** | **44.76** | **47.49** | **31.85** |

tably, both approaches significantly outperform the adversarial baseline, demonstrating the effectiveness of the optimisation strategies.

Table 6.2 highlights the results for the Rastrigin function under the same conditions.

Table 6.2 Benchmark results for the Rastrigin function using LLM-BO with Sonnet 3.5 and a multi-agent approach

| **Rastrigin Function** | | | | | | |
|---|---|---|---|---|---|---|
| | $d=2$ | | | $d=5$ | | |
| | $a=2$ | $a=3$ | $a=4$ | $a=2$ | $a=3$ | $a=4$ |
| Expert | 71.83 | 52.07 | 55.29 | 97.58 | 96.51 | 97.0 |
| Adversarial | 33.83 | 7.83 | 0.00 | 89.57 | 77.76 | 71.79 |
| BO | 50.94 | 52.44 | 40.5 | 93.47 | 95.03 | 95.49 |
| LLMBO | 64.07 | 50.75 | 51.34 | 95.98 | 93.61 | 93.75 |
| BO (% Performance) | 45.03 | **100.85** | 73.25 | 48.73 | **92.11** | **94.01** |
| LLMBO (% Performance) | **79.58** | 97.01 | **92.86** | **79.96** | 84.52 | 87.12 |

In two dimensions (d=2), LLMBO achieves 79.58% of expert performance with two alternatives (a=2), outperforming standard BO's 45.03%. However, with three alternatives, standard BO slightly exceeds expert performance (100.85%), while LLMBO maintains strong performance at 97.01%. In higher dimensions (d=5), both approaches perform well, with BO showing slight advantages for more alternatives (a=3,4). While LLMBO leads with two alternatives (79.96% vs 48.73%), standard BO achieves superior performance with three and

four alternatives (92.11% and 94.01% respectively). Both methods consistently outperform the adversarial baseline, particularly in higher dimensions.

Table 6.3 highlights the results for the Schwefel function under the same conditions.

Table 6.3 Benchmark results for the Schwefel function using LLM-BO with Sonnet 3.5 and a multi-agent approach

| **Schwefel Function** | | | | | | |
|---|---|---|---|---|---|---|
| | $d = 2$ | | | $d = 5$ | | |
| | $a = 2$ | $a = 3$ | $a = 4$ | $a = 2$ | $a = 3$ | $a = 4$ |
| Expert | 35.27 | 46.87 | 49.7 | 85.77 | 85.43 | 90.80 |
| Adversarial | 10.43 | 3.74 | 0.00 | 75.52 | 70.98 | 68.15 |
| BO | 28.03 | 28.04 | 25.34 | 80.37 | 83.12 | 81.89 |
| LLMBO | 31.98 | 33.96 | 19.49 | 80.29 | 83.53 | 81.92 |
| BO (% Performance) | 70.84 | 56.34 | **50.98** | **47.32** | 83.99 | 60.62 |
| LLMBO (% Performance) | **86.73** | **70.06** | 39.22 | 46.56 | **86.84** | **60.76** |

In two dimensions (d=2), LLMBO shows stronger performance with two and three alternatives, achieving 86.73% and 70.06% of expert performance respectively, compared to BO's 70.84% and 56.34%. However, with four alternatives, BO maintains better performance (50.98% vs 39.22%). In five dimensions (d=5), both approaches perform similarly with two alternatives ( 47%), while LLMBO shows slight advantages with three and four alternatives, reaching 86.84% and 60.76% of expert performance respectively.

### 6.4.2 Comparison of Prompting Strategies

In this subsection, the same experimental setup is used to evaluate the impact of different prompting strategies on LLMBO performance, comparing both the multi-agent and one-shot prompts. This analysis is performed just for the two alternative case (a=2) across all three test functions in both 2 and 5 dimensions.

Figure 6.2 illustrates the convergence trajectories of different optimisation approaches over 30 iterations. Each plot shows the normalised performance against the iteration number for a specific function and dimensionality combination. The solid lines represent mean

performance across 36 random initializations, while the shaded regions indicate the 95% confidence intervals around these means.

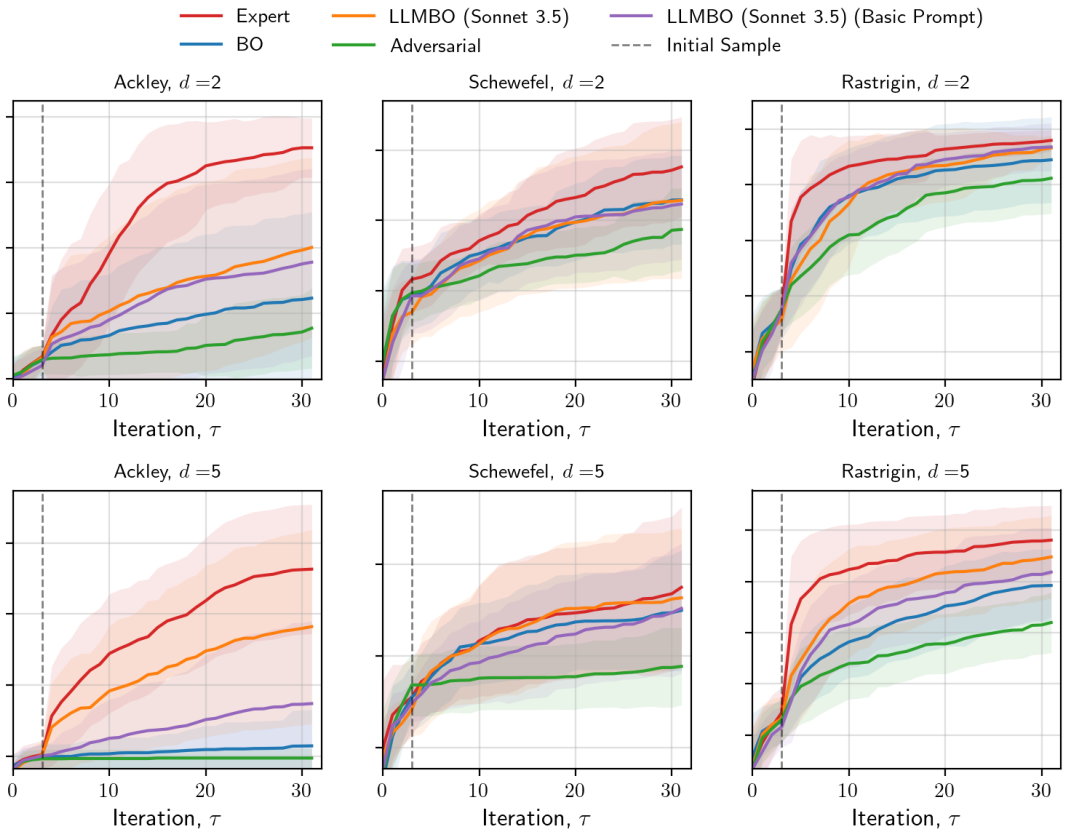
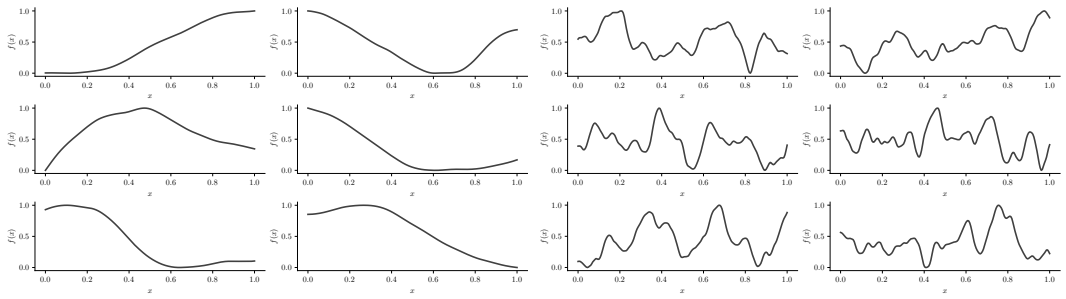



Fig. 6.2 Convergence comparison of optimisation strategies across test functions when presented with two alternative solutions, highlighting the difference in performance between the basic one-shot prompt and the multi-agent strategy.

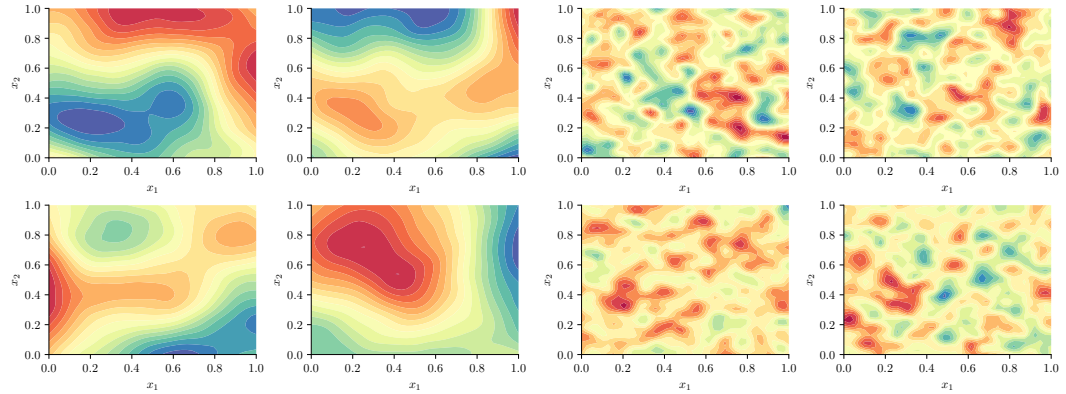
The convergence plots show multi-agent prompting either outperforming basic one-shot prompting, with the performance gap most pronounced for the 5D Ackley and Rastrigin functions. Both LLMBO variants surpass standard BO in all test cases, though they remain below expert selection performance as expected. These results give evidence that prompt engineering does impact LLM performance in optimisation and discovery contexts, with the multi-agent approach enabling more effective reasoning about complex optimisation landscapes resulting in further improvement over standard Bayesian optimisation.

### 6.4.3 Performance on Random Functions

In this subsection experiments are performed by optimising random functions that are sampled from Gaussian process priors. By varying the lengthscale of the kernel function used to generate these random functions from 0.05 to 0.5 (note the function is bounded between 0 and 1), performance can be assessed on the spectrum of 'linear-like' to completely random functions. Similarly to above these are performed using two alternative solutions. Figure 6.3 demonstrates sampled functions for one and two dimensions at both the lowest and highest lengthscales used.



(a) One-dimensional random functions sampled with lengthscales of 0.5 (left) and 0.05 (right).



(b) Two-dimensional random functions sampled with lengthscales of 0.5 (left) and 0.05 (right).

Fig. 6.3 Example random functions sampled from Gaussian process priors at different lengthscales. As the lengthscale decreases from 0.5 to 0.05, the functions transition from smoother to more complex behavior.

Figures 6.4 and 6.5 present the convergence results for random functions generated with lengthscales of 0.5 and 0.05, respectively. These experiments were conducted using two

alternative solutions (a=2) across dimensions d=2,3,4,5. The Deepseek LLM was used with both the one-shot and multi-agent (MoA) prompting strategies.

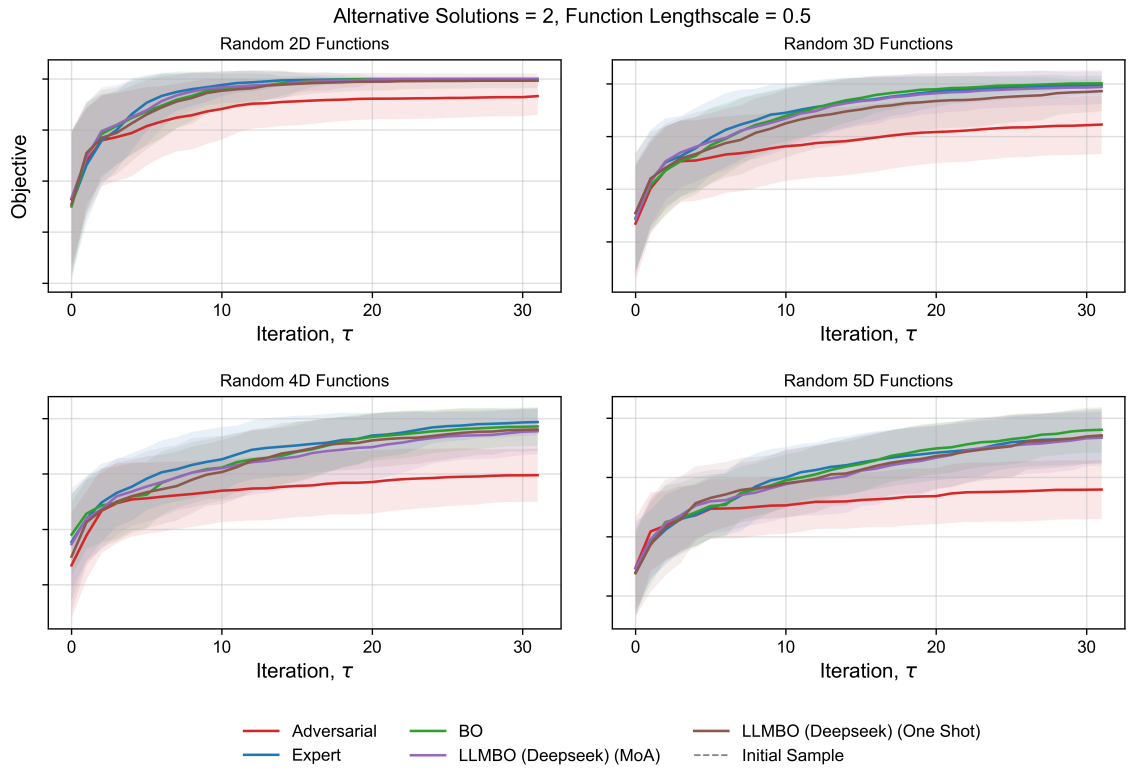



Fig. 6.4 Convergence comparison on random functions with lengthscale 0.5, using two alternative solutions across various dimensions. Performance of LLMBO (Deepseek One-Shot and MoA) is compared against Expert, standard BO, and Adversarial baselines.

For the smoother random functions with a lengthscale of 0.5 (Figure 6.4), the LLMBO methods (One-Shot and MoA) closely mirror the performance of standard BO across all tested dimensions. While expert selection remains superior, there is no clear advantage or disadvantage to using the LLM in this regime compared to standard BO. The underlying structure, although present, does not seem distinct enough for the LLM to consistently exploit beyond the capabilities of the standard BO utility function.

In contrast, when optimising the more complex random functions with a lengthscale of 0.05 (Figure 6.5), the performance of LLMBO slightly degrades relative to standard BO, particularly noticeable in 3D and 5D. The increased randomness and lack of discernible

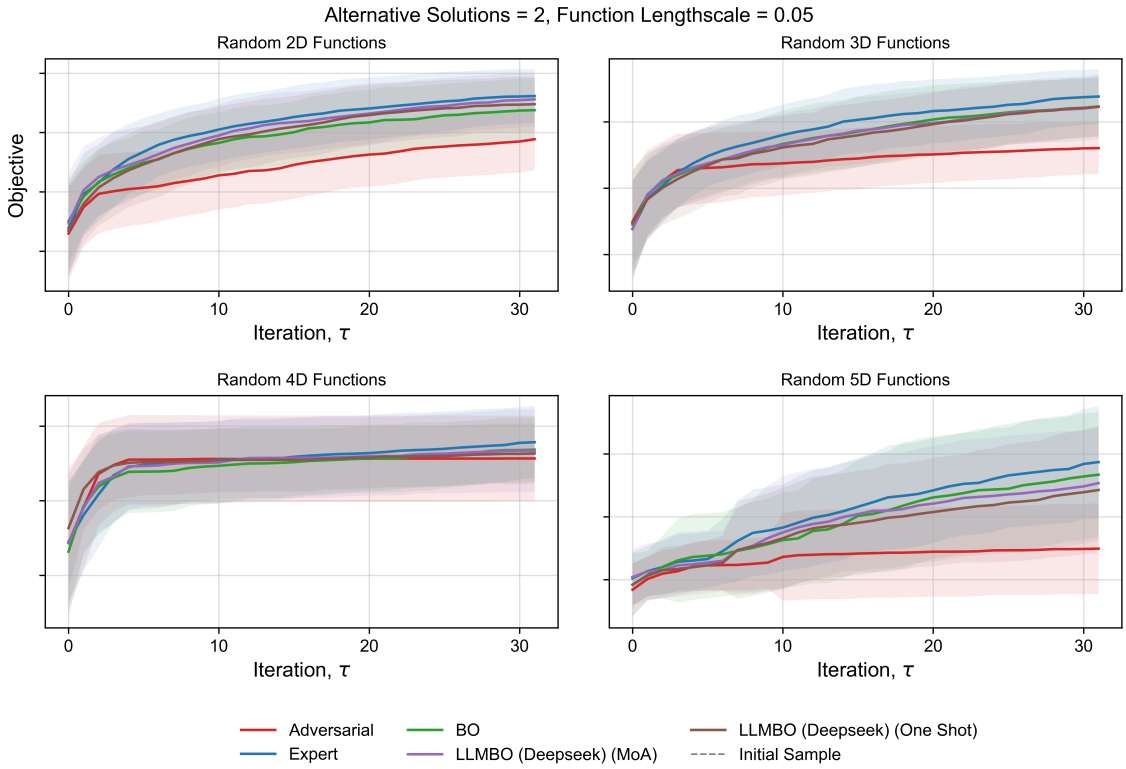



Fig. 6.5 Convergence comparison on random functions with lengthscale 0.05, using two alternative solutions across various dimensions. Performance of LLMBO (Deepseek One-Shot and MoA) is compared against Expert, standard BO, and Adversarial baselines.

global trends make it difficult for the LLM to identify reliable patterns from the history. Consequently, the LLM's choices do not improve upon, and sometimes slightly hinder, the convergence compared to standard BO. The gap between expert performance and both BO and LLMBO widens significantly under these conditions, especially in higher dimensions, illustrating the challenge posed by these complex, noisy function landscapes.

These results indicate that the effectiveness of LLMBO is linked to the presence of in-context learnable structure within the objective function. The LLM provides benefits over standard BO for structured functions (as seen in previous sections) where it can emulate expert intuition by identifying trends. However, this advantage diminishes for smoother random functions and can disappear or reverse for highly complex, low-lengthscale random functions where patterns are obscured by noise and local variability.

In light of this, Figure 6.6 demonstrates the performance of different LLMs and prompting strategies across functions of lengthscale equal to 0.5.

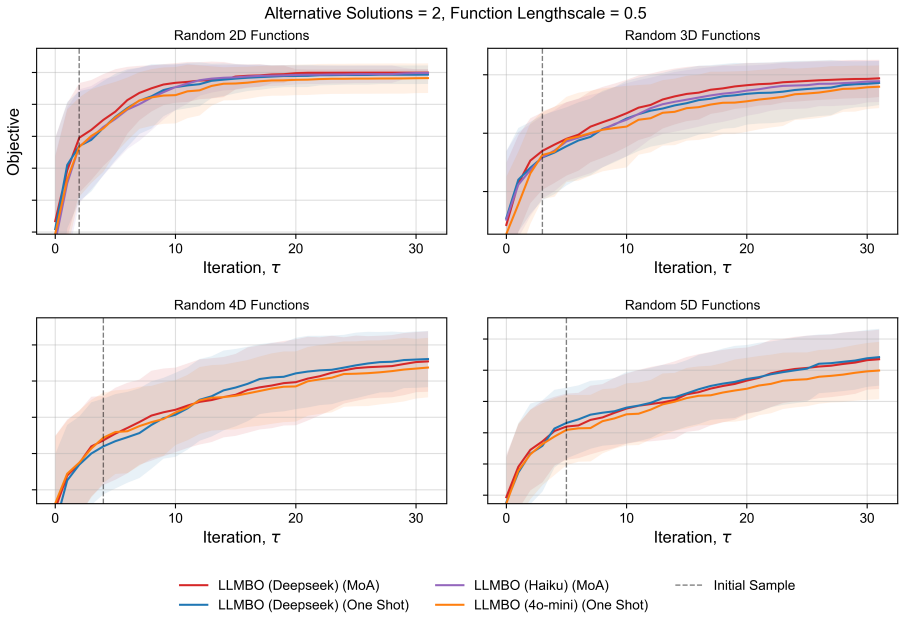



Fig. 6.6 Convergence comparison of different LLM models and prompting strategies on random functions with lengthscale 0.5, using two alternative solutions across dimensions $d = 2, 3, 4, 5$. Performance of LLMBO (Deepseek V3 with one-shot and multi-agent prompts and Anthropic Haiku with one-shot prompt) is compared.

While performance differences are subtle, both the Deepseek multi-agent approach and one-shot approach outperform the smaller 4o-mini model showing slightly better convergence, with the Multi-agent approach improving on the One-shot approach when using Deepseek V3 [29]. These results illustrate the benchmark's utility in differentiating LLM reasoning capabilities for optimisation tasks.

### 6.4.4 Chemical Optimisation Case Study

Following the extensive benchmarking on synthetic and random functions, the LLMBO methodology is now evaluated on a complex, high-dimensional optimisation problem derived directly from real-world experimental data. This step serves to validate the approach beyond

theoretical constructs and assess its potential applicability within genuine scientific discovery workflows, particularly those involving automated experimentation platforms or self-driving laboratories.

The case study selected is based on the 10-variable photocatalysis optimisation task originally formulated and solved by Burger et al. [18] using a pioneering mobile robotic chemist. The original experiment sought to enhance the activity of the P10 polymer photocatalyst using the bioderived L-cysteine as a scavenger, simultaneously investigating five distinct chemical hypotheses through automated experimentation. This important work demonstrated the impact of integrating robotics and AI to explore vast chemical parameter spaces, seeking novel materials for hydrogen production and demonstrating the acceleration of scientific discovery through automation [18]. The high dimensionality (ten interacting chemical components), coupled with the direct relevance to materials science and catalysis, makes this system an ideal benchmark for LLMBO.

The original study employed a Bayesian optimisation strategy to guide the robot through 688 autonomous experiments [18]. For the present work, the data generated from these experiments was subsequently used to train a Gaussian Process Regression (GPR) model, creating a high-fidelity simulation of the chemical system's response surface (Hydrogen Evolution Rate, HER). This GPR-based simulation, previously utilised to evaluate the Hypothesis Bayesian Optimization (HypBO) algorithm [25], provides a realistic, experimentally-grounded, yet computationally tractable objective function.

This case study therefore tests the key hypothesis of this chapter that LLMs can act as computational experts, identifying patterns within the optimisation trajectory of a real-world system to enhance convergence beyond that of standard BO. Evaluating LLMBO in this context allows for an assessment of its practical utility for accelerating discovery pipelines in computationally and experimentally intensive domains such as catalysis development.

The LLMBO methodology was applied using the Deepseek model with both a chain-of-thought reasoning-based prompt and the basic one-shot prompting strategy, enabling a comparison of how effective it is to induce reasoning at inference time for this task. Performance is compared against standard BO.

Figure 6.7 presents the convergence results for this case study, comparing the performance of standard BO against LLMBO using both strategies.

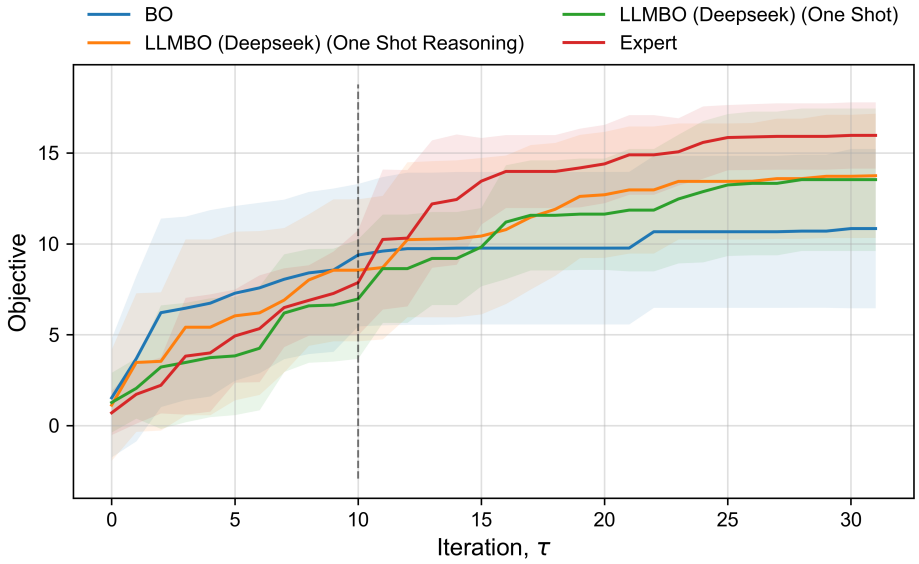



Fig. 6.7 Convergence comparison for the simulated 10-variable photocatalysis optimisation task. Performance of standard Bayesian Optimisation (BO) is compared against LLMBO using the Deepseek model with both a basic One-Shot prompt and a Chain-of-Thought (CoT) reasoning prompt. Results show the mean normalised performance over multiple runs, with shaded areas representing confidence intervals.

As depicted in the figure, both LLMBO variants (One Shot and One Shot Reasoning) eventually outperform standard BO as the optimisation progresses after the initial static design of experiments with 10 evaluations.

While the 'One Shot Reasoning' strategy appears to yield slightly faster improvements during the middle iterations (roughly iterations 10 to 20), both LLMBO approaches converge to a similar final performance level, which is clearly superior to that achieved by standard BO

alone. The difference in final convergence between the two LLMBO prompting strategies is marginal based on the mean trajectories shown.

Although neither LLMBO approach reaches the performance level of the theoretical 'expert' baseline (representing perfect selection at each step) which is to be expected, they both demonstrate a distinct advantage over standard BO in this high-dimensional chemical space. This suggests that the LLM, regardless of the specific one-shot prompting variant used here, was able to identify and leverage patterns in the historical data more effectively than the standard BO acquisition function alone. These results support the core hypothesis that LLMs can function as effective computational experts, enhancing automated optimisation by reasoning over historical data in realistic search spaces. This successful application implies that integrating LLMs holds considerable promise for accelerating discovery in real-world automated science platforms, such as self-driving laboratories, even if the benefits between specific prompting strategies like basic one-shot and reasoning-enhanced one-shot might be subtle for certain problems.

## 6.5 Conclusions

In this chapter, LLMBO was introduced; a novel framework integrating large language models into the Bayesian optimisation loop to serve as computational decision-makers. By applying the methodology developed for human-in-the-loop BO (Chapter 5) to generate diverse, high-utility candidate solutions via multi-objective optimisation, and replacing the human expert with an LLM, the benefits of full automation can be maintained while potentially capturing some benefits of expert intuition. The results presented demonstrate that this approach can significantly enhance convergence compared to standard Bayesian optimisation, not only on structured synthetic benchmarks but also, crucially, on a challenging, high-dimensional chemical optimisation problem derived from real-world experimental data.

The approach was benchmarked extensively across classical optimisation functions, random functions sampled from Gaussian process priors with varying complexity (lengthscales), and an applied high-dimensional chemical discovery problem.

Key findings indicate that LLMBO consistently improved performance over standard BO on classical structured test functions, particularly in higher dimensions, suggesting LLMs can mimic expert behaviour by exploiting data trends. This performance advantage was successfully replicated in the 10-dimensional chemical optimisation case study, confirming that LLMs can apply pattern-recognition and reasoning capabilities to navigate high-dimensional realistic optimisation landscapes grounded in experimental data more effectively than standard BO. The effectiveness of LLMBO was influenced by the prompting strategy, with a multi-agent approach generally outperforming a basic one-shot prompt. However, performance was strongly linked to problem structure: while LLMBO matched standard BO on smoother random functions, it slightly underperformed on highly complex random functions and the specific reactor design case study, indicating its benefits are not universal and depend on the presence of learnable patterns within the optimisation landscape. Finally, the methodology proved effective as a robust, leakage-proof benchmark for quantitatively evaluating LLM reasoning in scientific decision-making contexts.

This work demonstrates that LLMs can be integrated into automated scientific discovery workflows like Bayesian optimisation in a meaningful and robust way. By constraining the LLM's role to selecting from a predefined set of high-quality candidate solutions, the risks associated with hallucination or poor suggestions are mitigated. The LLM acts as an enhancer, processing historical context to potentially accelerate convergence, particularly when underlying patterns exist, though this benefit is not guaranteed for all problems.

This chapter presents a validated approach for developing more intelligent, fully automated optimisation systems, particularly relevant for applications like self-driving laboratories where continuous, unattended operation is key. It represents a step towards harnessing the

pattern-recognition and in-context reasoning abilities of LLMs for scientific discovery while grounding their contributions in the framework of Bayesian optimisation. Future work could explore more sophisticated prompting strategies, integrating uncertainty information more explicitly into the LLM prompt, applying the framework to a wider range of real-world experimental optimisation tasks, and further investigating the problem characteristics that determine whether LLMBO provides an advantage over standard BO.

# Chapter 7

# Conclusion

This thesis has investigated the application of data-driven optimisation techniques to improve the design and analysis of chemical engineering systems, with a specific focus on reactor technology. The work proceeded by framing the challenge of advancing chemical process design through combining established engineering principles, modelled deductively, with inductive, data-oriented machine learning approaches. It was proposed that integrating these perspectives using modern optimisation algorithms provides a pathway towards improved process efficiency and the discovery of novel chemical engineering solutions. The research presented detailed the development and evaluation of optimisation frameworks tailored to specific chemical engineering problems. This final chapter first summarises the main contributions from the preceding chapters, followed by concluding remarks and potential directions for future work.

# Summary of Contributions

1. **Automated Reactor Design using Multi Fidelity Optimisation:** Chapter 3 introduced a structured framework, DARTS, for chemical reactor design employing multi-fidelity Bayesian optimisation integrated with computational fluid dynamics (CFD). This involved developing a parameterisation for helical tube reactors and utilising continuous mesh fidelities to explore the design space efficiently. A cost-adjusted acquisition function and a unique termination rule were formulated, facilitating the selection of lower-cost simulations while ensuring a final, validated high-fidelity reactor design was achieved within computational limits. Applied to pulsed-flow reactors, the framework automatically identified geometries, including one with a flow inversion, that improved mixing performance, a finding confirmed through experimental testing of 3D printed prototypes. This demonstrated the capacity of automated optimisation to uncover non-intuitive designs leading to enhanced fluid dynamics, specifically the generation of beneficial Dean vortices.

2. **Machine Learning Assisted Discovery of Flow Reactor Geometries:** Chapter 4 advanced the reactor design approach by incorporating high-dimensional parameterisations for both the reactor's cross-section and its path, applying polar Gaussian processes for geometric flexibility. Optimising for enhanced plug flow performance under steady-state conditions (Re = 50) revealed designs with features like periodic expansions, contractions, and pinches. These geometric modifications were shown through CFD and subsequent experiments to induce Dean vortices and improve mixing characteristics even at low Reynolds numbers, conditions where conventional designs often require external forcing like pulsed flow. The resulting 3D printed reactors showed approximately 60% improvement in plug flow performance metrics over standard coils, illustrating a successful application of machine learning to discover fundamentally better reactor configurations for chemical processes.

3. **Human-Algorithm Collaborative Bayesian Optimisation for Chemical Engineering Systems:** Chapter 5 addressed the integration of chemical engineering expertise into the optimisation workflow. A collaborative Bayesian optimisation method was developed, allowing domain experts (like chemical engineers or chemists) to guide the process by selecting from a diverse set of high-utility candidate solutions. These candidates were generated using a multi-objective batch Bayesian optimisation formulation. An expert-augmented initial sampling strategy was also introduced. Application to chemical engineering case studies, including reactor design and the optimisation of a fed-batch bioprocess for phycocyanin production, showed that incorporating expert choices accelerated convergence compared to standard Bayesian optimisation, especially in complex, noisy scenarios relevant to process development. This approach provides a practical way to leverage valuable process knowledge while maintaining algorithmic rigor.

4. **In Silico Chemical Process Expertise using Large Language Models:** To retain full automation, crucial for applications like self-driving labs in chemical discovery, Chapter 6 substituted the human expert with a Large Language Model (LLM) in the collaborative framework. The LLM was tasked with selecting the next experiment from the candidate solutions, based on structured historical data and solution features. This LLM-enhanced Bayesian optimisation method demonstrated convergence improvements over standard Bayesian optimisation, particularly for problems with underlying structure, mimicking the benefits seen with human experts. The LLM appeared capable of identifying trends in the optimisation data, effectively acting as an in silico expert for guiding process optimisation, though its advantage diminished for highly random function landscapes. This offers a route to incorporating sophisticated pattern recognition into automated chemical experimentation platforms.

## Concluding Remarks and Future Directions

The research presented in this thesis demonstrates the potential benefits of combining data-driven optimisation, machine learning, chemical engineering principles, and expert input (human or artificial). The work moved beyond standard optimisation applications towards discovery, yielding novel reactor designs with improved mixing performance linked to specific induced fluid dynamic effects. The automation of design tasks using frameworks like DARTS, alongside the ability to find non-intuitive geometries, highlights the utility of these computational methods.

The effectiveness of multi-fidelity optimisation for managing computationally expensive CFD simulations in reactor design was established, offering a practical method for exploring complex design spaces within computational constraints. Furthermore, the investigations into collaborative optimisation frameworks showed that integrating expert knowledge, either directly from humans or via LLMs, can improve the efficiency of identifying optimal process

conditions and designs, particularly in complex or noisy situations. The finding that LLMs can identify patterns in data suggests a potential role as **in silico** experts in automated discovery platforms.

Future research could build on these results. Extending the high-dimensional reactor parameterisations to address multi-objective problems, such as balancing mixing performance against pressure drop, is a relevant direction. Applying these frameworks to different reactor types or reaction systems, possibly including reaction kinetics alongside transport phenomena, would further assess their general applicability. Enhancing methods for incorporating expert knowledge, perhaps using preference elicitation beyond discrete choices, could improve human-algorithm interaction. For LLM integration, tailoring prompts and fine-tuning strategies for chemical process optimisation, potentially using chemical knowledge graphs or simulation tools, could enhance their reliability as automated guides.

Continued development of such integrated, data-driven methods can accelerate innovation in chemical reactor engineering, process intensification, and process systems engineering more broadly, pioneering a new generation of tools for the discovery of engineering systems.

The following is a discussion around what research into the topics covered in this thesis may look like within 5-10 years.

I believe research into unifying multi-modal sources of data for autonomous decision-making will gain impact, particularly as new sensors and measurements will be supported by new models. The methodologies in this thesis largely rely on optimising a single, scalar objective function, such as the number of equivalent tanks-in-series, which is a useful but simplified representation of reactor performance. Future algorithms will have access to much richer data streams from a range of sensors, including high-resolution imaging that captures flow patterns, spectroscopic data indicating real-time composition, and pressure sensor arrays mapping fluctuations within the reactor. The key research challenge will be to move beyond optimising a single metric and instead learn from and make decisions based on this high-dimensional,

heterogeneous information. This will necessitate a shift from standard surrogate models like Gaussian Processes to more sophisticated approaches capable of handling structured, vector-valued, or even functional outputs.

This integration of richer data sources naturally leads to closing the optimisation loop with physical hardware, for example through the direct inclusion of 3D printing. While this thesis used additive manufacturing for post-optimisation validation of designs, the next step will involve incorporating the design-print-test cycle directly into the optimisation loop. Such a hardware-in-the-loop approach would enable fully autonomous discovery, where the algorithm not only suggests a new geometry but also oversees its fabrication and subsequent experimental evaluation. This presents challenges related to the time cost of manufacturing, experimental noise, and physical constraints. However, it also opens up possibilities for reactors that can adapt their own geometry in a continuous manufacturing setup in response to changing feed compositions or catalyst deactivation, moving towards truly self-optimising systems.

The necessity for such hardware-in-the-loop optimisation becomes even more apparent when adapting reactor design to multiphase or reactive flows, where state-of-the-art simulations are too computationally expensive or fail to accurately capture interfacial physics. For these challenging domains, the purely simulation-based optimisation approach detailed in this thesis reaches its practical limits. Future progress will depend on frameworks where physical experiments are an integral, high-fidelity data source within the optimisation loop itself. This highlights the importance of the multi-fidelity concepts explored in Chapter 3. Future research may present a system where a cheap, inaccurate CFD model might serve as the lowest fidelity, a bench-scale experiment an intermediate one, and a pilot-plant run the highest, all integrated to guide the design process for systems that are to complex to be modelled through simulation alone.

Finally, there is significant scope to expand the very definition of multi-fidelity optimisation. This thesis primarily defined fidelity in the context of simulation mesh resolution. Taking a wider view of the problem may treat every available information source—from foundational first-principles models and empirical correlations to multi-scale simulations and physical experiments of varying sizes—as a different fidelity observation of the "true" target system, which is ultimately the final, industrial-scale reactor. The central research ambition, therefore, is to create a unified optimisation framework that can ingest and learn from this heterogeneous collection of sources. Developing methods to accurately model the non-linear and non-obvious correlations between these fidelities will enable the design of previously infeasible systems and fully realise the synthesis of deductive and inductive reasoning that formed the core theme of this thesis.

# References


[1] Masaki Adachi, Brady Planden, David A. Howey, Michael A. Osborne, Sebastian Orbell, Natalia Ares, Krikamol Muandet, and Siu Lun Chau. Looping in the human collaborative and explainable bayesian optimization. 2023. doi: 10.48550/ARXIV. 2310.17273. URL https://arxiv.org/abs/2310.17273.

[2] Shobha Agrawal and K.D.P. Nigam. Modelling of a coiled tubular chemical reactor. *Chem. Eng. J.*, 84(3):437–444, December 2001. doi: 10.1016/s1385-8947(00) 00370-3.

[3] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, December 1974. ISSN 0018-9286. doi: 10.1109/tac.1974.1100705. URL http://dx.doi.org/10.1109/TAC.1974.1100705.

[4] Sebastian Ament, Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Unexpected improvements to expected improvement for bayesian optimization, 2023. URL https://arxiv.org/abs/2310.20708.

[5] Anthropic. Introducing the next generation of Claude. https://www.anthropic.com/news/claude-3-family. [Accessed 17-04-2025].

[6] James P. Bailey, Gauthier Gidel, and Georgios Piliouras. Finite regret and cycles with fixed step-size via alternating gradient descent-ascent, 2019. URL https://arxiv.org/abs/1907.04392.

[7] Annabel R. Basford, Steven K. Bennett, Muye Xiao, Lukas Turcani, Jasmine Allen, Kim. E. Jelfs, and Rebecca L. Greenaway. Streamlining the automated discovery of porous organic cages. *Chemical Science*, 2024. ISSN 2041-6539. doi: 10.1039/d3sc06133g. URL http://dx.doi.org/10.1039/D3SC06133G.

[8] Nausheen Basha, Thomas Savage, Jonathan McDonough, Ehecatl Antonio Del-Rio Chanona, and Omar K. Matar. Discovery of mixing characteristics for enhancing coiled reactor performance through a Bayesian optimisation-CFD approach. *Chem. Eng. J.*, page 145217, 2023. ISSN 1385-8947. doi: 10.1016/j.cej.2023.145217.

[9] G K Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, Cambridge, England, February 2000.

[10] Rohit Batra, Ghanshyam Pilania, Blas P. Uberuaga, and Rampi Ramprasad. Multifidelity information fusion with machine learning: A case study of dopant formation energies in hafnia. *ACS Appl. Mater. Interfaces*, 11(28):24906–24918, April 2019. doi: 10.1021/acsami.9b02174. URL https://doi.org/10.1021/acsami.9b02174.

[11] Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

[12] R Byron Bird, Warren E Stewart, and Edwin N Lightfoot. *Transport Phenomena*. John Wiley & Sons, Chichester, England, November 2006.

[13] David Bostock. *The Theory that Perception is Knowledge*, page 40–83. Oxford University Press, February 1991. doi: 10.1093/acprof:oso/9780198239307.003.0002. URL http://dx.doi.org/10.1093/acprof:oso/9780198239307.003.0002.

[14] George E. P. Box. Science and statistics. *Journal of the American Statistical Association*, 71(356):791–799, December 1976. ISSN 1537-274X. doi: 10.1080/01621459.1976.10480949. URL http://dx.doi.org/10.1080/01621459.1976.10480949.

[15] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

[16] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021. URL https://arxiv.org/abs/2104.13478.

[17] B.Shahriari, K. Swersky, Z. Wang, R.P. Adams, and N. de Freitas. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, 104 (1):148–175, January 2016. doi: 10.1109/jproc.2015.2494218.

[18] Benjamin Burger, Phillip M. Maffettone, Vladimir V. Gusev, Catherine M. Aitchison, Yang Bai, Xiaoyan Wang, Xiaobo Li, Ben M. Alston, Buyi Li, Rob Clowes, Nicola Rankin, Brandon Harris, Reiner Sebastian Sprick, and Andrew I. Cooper. A mobile robotic chemist. *Nature*, 583(7815):237–241, July 2020. ISSN 1476-4687. doi: 10.1038/s41586-020-2442-2. URL http://dx.doi.org/10.1038/s41586-020-2442-2.

[19] J. Caballero and I.E. Grossmann. An algorithm for the use of surrogate models in modular flowsheet optimization. *AIChE journal*, 54(10):2633–2650, 2008.

[20] H. Cai, K. Guo, H. Liu, C. Liu, and A. Feng. Derivative-free level-set-based multi-objective topology optimization of flow channel designs using lattice Boltzmann method. *Chemical Engineering Science*, 231:116323, February 2021. doi: 10.1016/j.ces.2020.116323.

[21] Tanmay Chakraborty, Christian Wirth, and Christin Seifert. Explainable bayesian optimization, 2024. URL https://arxiv.org/abs/2401.13334.

[22] Sijia Chen, Yibo Wang, Yi-Feng Wu, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Lijun Zhang. Advancing tool-augmented large language models: Integrating insights from errors in inference trees, 2024. URL https://arxiv.org/abs/2406.07115.

[23] Kwanghyun Cho, Ketong Shao, and Ali Mesbah. Run-indexed time-varying Bayesian optimization with positional encoding for auto-tuning of controllers: Application to a plasma-assisted deposition process with run-to-run drifts. *Computers & Chemical*

*Engineering*, 185:108653, June 2024. ISSN 0098-1354. doi: 10.1016/j.compchemeng.2024.108653.

[24] Guglielmo Cimolai, Iman Dayyani, and Qing Qin. Multi-objective shape optimization of large strain 3D helical structures for mechanical metamaterials. *Mater. Design*, 215: 110444, March 2022. doi: 10.1016/j.matdes.2022.110444.

[25] Abdoulatif Cisse, Xenophon Evangelopoulos, Sam Carruthers, Vladimir V. Gusev, and Andrew I. Cooper. HypBO: Accelerating Black-Box Scientific Experiments Using Experts' Hypotheses, 2023.

[26] S. Daulton, D. Eriksson, M. Balandat, and E. Bakshy. Multi-objective Bayesian optimization over high-dimensional search spaces. In James Cussens and Kun Zhang, editors, *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pages 507–517. PMLR, 01–05 Aug 2022. URL https://proceedings.mlr.press/v180/daulton22a.html.

[27] W.R. Dean. Note on the motion of fluid in a curved pipe. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 4(20):208–223, July 1927. doi: 10.1080/14786440708564324.

[28] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6 (2):182–197, April 2002. ISSN 1089-778X. doi: 10.1109/4235.996017. URL http://dx.doi.org/10.1109/4235.996017.

[29] Deepseek. Introducing DeepSeek-V3. https://api-docs.deepseek.com/news/news1226. [Accessed 17-04-2025].

[30] E. A. del Rio Chanona, P. Petsagkourakis, E. Bradford, J. E. Alves Graciano, and B. Chachuat. Real-time optimization meets bayesian optimization and derivative-free optimization: A tale of modifier adaptation. *Comp. Chem. Eng.*, 147:107249, April 2021. doi: 10.1016/j.compchemeng.2021.107249. URL https://doi.org/10.1016/j.compchemeng.2021.107249.

[31] Zhengya Dong, Stefan D.A. Zondag, Matthias Schmid, Zhenghui Wen, and Timothy Noël. A meso-scale ultrasonic milli-reactor enables gas–liquid-solid photocatalytic reactions in flow. *Chem. Eng. J*, 428:130968, January 2022. doi: 10.1016/j.cej.2021.130968.

[32] K.S. Elvira, X. Casadevall i Solvas, R.C.R. Wootton, and A.J. deMello. The past, present and potential for microfluidic reactor technology in chemical synthesis. *Nature Chemistry*, 5(11):905–915, October 2013. doi: 10.1038/nchem.1753.

[33] D. Eriksson, M. Pearce, J. Gardner, R.D. Turner, and M. Poloczek. Scalable Global Optimization via Local Bayesian Optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/6c990b7aca7bc7058f5e98ea909e924b-Paper.pdf.

[34] David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local Bayesian optimization. In *Advances in Neural Information Processing Systems*, pages 5496–5507, 2019. URL http://papers.nips.cc/paper/8788-scalable-global-optimization-via-local-bayesian-optimization.pdf.

[35] K.C. Felton, J.G. Rittig, and A.A. Lapkin. Summit: Benchmarking Machine Learning Methods for Reaction Optimisation. *Chemistry–Methods*, 1(2):116–122, February 2021. doi: 10.1002/cmtd.202000051.

[36] Iztok Fister Jr, Uroš Mlakar, Janez Brest, and Iztok Fister. A new population-based nature-inspired algorithm every month: is the current era coming to the end. In *Proceedings of the 3rd student computer science research conference*, pages 33–37. University of Primorska Press, 2016.

[37] Jose Pablo Folch, Shiqiang Zhang, Robert M Lee, Behrang Shafei, David Walz, Calvin Tsay, Mark van der Wilk, and Ruth Misener. Snake: Bayesian optimization with pathwise exploration, 2022. URL https://arxiv.org/abs/2202.00060.

[38] Jose Pablo Folch, Robert M. Lee, Behrang Shafei, David Walz, Calvin Tsay, Mark van der Wilk, and Ruth Misener. Combining multi-fidelity modelling and asynchronous batch Bayesian Optimization. *Comp. Chem. Eng.*, 172:108194, 2023. ISSN 0098-1354. doi: https://doi.org/10.1016/j.compchemeng.2023.108194.

[39] J.P. Folch, R.M. Lee, B. Shafei, D. Walz, C. Tsay, M. van der Wilk, and R. Misener. Combining Multi-Fidelity Modelling and Asynchronous Batch Bayesian Optimization, 2022.

[40] Peter Frazier, Warren Powell, and Savas Dayanik. The knowledge-gradient policy for correlated normal beliefs. *INFORMS Journal on Computing*, 21(4):599–613, November 2009. ISSN 1526-5528. doi: 10.1287/ijoc.1080.0314. URL http://dx.doi.org/10.1287/ijoc.1080.0314.

[41] Hua Gao, Jian Zhou, Mohammad Moein Naderi, Zhangli Peng, and Ian Papautsky. Evolution of focused streams for viscoelastic flow in spiral microchannels. *Microsyst. Nanoeng.*, 9(1), June 2023. doi: 10.1038/s41378-023-00520-4.

[42] R. Garnett. *Bayesian Optimization*. Cambridge University Press, 2023.

[43] Roman Garnett. *Bayesian Optimization*. Cambridge University Press, January 2023. doi: 10.1017/9781108348973.

[44] M. Giselle Fernández-Godino. Review of multi-fidelity models. *Advances in Computational Science and Engineering*, 1(4):351–400, 2023. ISSN 2837-1739. doi: 10.3934/acse.2023015. URL http://dx.doi.org/10.3934/acse.2023015.

[45] Leonardo D. González and Victor M. Zavala. New paradigms for exploiting parallel experiments in Bayesian optimization. *Computers & Chemical Engineering*, 170:108110, February 2023. ISSN 0098-1354. doi: 10.1016/j.compchemeng.2022.108110.

[46] Carlos A Grande, Richard Blom, Vesna Middelkoop, Dorota Matras, Antonis Vamvakeros, Simon DM Jacques, Andrew M Beale, Marco Di Michiel, Kari Anne Andreassen, and Aud M Bouzga. Multiscale investigation of adsorption properties of novel 3d printed utsa-16 structures. *Chemical Engineering Journal*, 402:126166, 2020.

[47] Don W Green and Robert H Perry. *Perry's chemical engineers' handbook, eighth edition*. McGraw-Hill Professional, New York, NY, 8 edition, November 2007.

[48] Benjamin Grimmer. Provably faster gradient descent via long steps, 2023. URL https://arxiv.org/abs/2307.06324.

[49] BF Gschaider. The incomplete swak4foam reference. *Tech. Rep.*, 131:202, 2013. URL https://openfoamwiki.net/images/d/db/Swak4FoamReference.pdf.

[50] Sunil Gupta, Alistair Shilton, Arun Kumar A, Shannon Ryan, Majid Abdolshah, Hung Le, Santu Rana, Julian Berk, Mahad Rashid, and Svetha Venkatesh. BO-Muse: A human expert and AI teaming framework for accelerated experimental design, 2023.

[51] J. Hagedorn and F. Kargi. coiled-tube membrane bioreactor for cultivation of hybridoma cells producing monoclonal antibodies. *Enzyme Microb. Tech.*, 12(11): 824–829, November 1990. doi: 10.1016/0141-0229(90)90017-k.

[52] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer New York, 2009. ISBN 9780387848587. doi: 10.1007/978-0-387-84858-7. URL http://dx.doi.org/10.1007/978-0-387-84858-7.

[53] X. He, R. Tuo, and C.F.J. Wu. Optimization of Multi-Fidelity Computer Experiments via the EQIE Criterion. *Technometrics*, 59(1):58–68, January 2017. doi: 10.1080/00401706.2016.1142902.

[54] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, January 1989. ISSN 0893-6080. doi: 10.1016/0893-6080(89)90020-8. URL http://dx.doi.org/10.1016/0893-6080(89)90020-8.

[55] D. Huang, T. T. Allen, W. I. Notz, and R. A. Miller. Sequential kriging optimization using multiple-fidelity evaluations. *Structural and Multidisciplinary Optimization*, 32 (5):369–382, May 2006. URL https://doi.org/10.1007/s00158-005-0587-0.

[56] Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, March 1964. ISSN 0003-4851. doi: 10.1214/aoms/1177703732. URL http://dx.doi.org/10.1214/aoms/1177703732.

[57] Carl Hvarfner, Danny Stoll, Artur Souza, Marius Lindauer, Frank Hutter, and Luigi Nardi. $\pi$BO: Augmenting Acquisition Functions with User Beliefs for Bayesian Optimization, 2022.

[58] Hrvoje Jasak, Aleksandar Jemcov, Zeljko Tukovic, et al. Openfoam: A c++ library for complex physics simulations. In *International workshop on coupled methods in numerical dynamics*, volume 1000, pages 1–20, 2007. URL http://csabai.web.elte.hu/http/simulationLab/jasakEtAlOpenFoam.pdf.

[59] Shuyi Jia, Chao Zhang, and Victor Fung. Llmatdesign: Autonomous materials discovery with large language models, 2024. URL https://arxiv.org/abs/2406.13163.

[60] Ping Jiang, Tingli Xie, Qi Zhou, Xinyu Shao, Jiexiang Hu, and Longchao Cao. A space mapping method based on gaussian process model for variable fidelity metamodeling. *Simulation Modelling Practice and Theory*, 81:64–84, February 2018. ISSN 1569-190X. doi: 10.1016/j.simpat.2017.11.010. URL http://dx.doi.org/10.1016/j.simpat.2017.11.010.

[61] Michael Jokiel, Nicolas Maximilian Kaiser, Péter Kováts, Michael Mansour, Katharina Zähringer, Krishna Deo Prasad Nigam, and Kai Sundmacher. Helically coiled segmented flow tubular reactor for the hydroformylation of long-chain olefins in a thermomorphic multiphase system. *Chem. Eng. J.*, 377:120060, December 2019. doi: 10.1016/j.cej.2018.09.221.

[62] Donald R Jones, Cary D Perttunen, and Bruce E Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of optimization Theory and Applications*, 79: 157–181, 1993.

[63] Edmund Judge, Mohammed Azzouzi, Austin M. Mroz, Antonio del Rio Chanona, and Kim E. Jelfs. Applying multi-fidelity bayesian optimization in chemistry: Open challenges and major considerations, 2024. URL https://arxiv.org/abs/2409.07190.

[64] Keren J. Kanarik, Wojciech T. Osowiecki, Yu Lu, Dipongkar Talukder, Niklas Roschewsky, Sae Na Park, Mattan Kamon, David M. Fried, and Richard A. Gottscho. Human-machine collaboration for improving semiconductor process development. *Nature*, 616(7958):707–711, March 2023. doi: 10.1038/s41586-023-05773-7.

[65] K. Kandasamy, G. Dasarathy, J. Schneider, and B. Poczos. Multi-fidelity Bayesian Optimisation with Continuous Approximations, 2017.

[66] Kirthevasan Kandasamy, Gautam Dasarathy, Junier B. Oliva, Jeff Schneider, and Barnabas Poczos. Multi-fidelity gaussian process bandit optimisation, 2016. URL https://arxiv.org/abs/1603.06288.

[67] Sanyam Kapoor, Nate Gruver, Manley Roberts, Katherine Collins, Arka Pal, Umang Bhatt, Adrian Weller, Samuel Dooley, Micah Goldblum, and Andrew Gordon Wilson. Large language models must be taught to know what they don't know, 2024. URL https://arxiv.org/abs/2406.08391.

[68] G.K.W. Kenway, C.A. Mader, P. He, and J.R.R.A. Martins. Effective adjoint approaches for computational fluid dynamics. *Progress in Aerospace Sciences*, 110: 100542, October 2019. doi: 10.1016/j.paerosci.2019.05.002.

[69] Myunghee Kim, Ye Ding, Philippe Malcolm, Jozefien Speeckaert, Christoper J. Siviy, Conor J. Walsh, and Scott Kuindersma. Human-in-the-loop bayesian optimization of wearable device parameters. *PLOS ONE*, 12(9):e0184054, September 2017. ISSN 1932-6203. doi: 10.1371/journal.pone.0184054. URL http://dx.doi.org/10.1371/journal.pone.0184054.

[70] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. URL https://arxiv.org/abs/1412.6980.

[71] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983. ISSN 1095-9203. doi: 10.1126/science.220.4598.671. URL http://dx.doi.org/10.1126/science.220.4598.671.

[72] Thomas S Kuhn. *The structure of scientific revolutions*, volume 962. University of Chicago press Chicago, 1997.

[73] Thomas S Kuhn. *The essential tension: Selected studies in scientific tradition and change*. University of Chicago press, 2011.

[74] Thomas S Kuhn et al. Second thoughts on paradigms. *The structure of scientific theories*, 2:459–482, 1974.

[75] Arun Kumar, Santu Rana, Alistair Shilton, and Svetha Venkatesh. Human-ai collaborative bayesian optimisation. *Advances in Neural Information Processing Systems*, 35:16233–16245, 2022.

[76] Rémi Lam, Douglas L. Allaire, and Karen E. Willcox. Multifidelity Optimization using Statistical Surrogate Modeling for Non-Hierarchical Information Sources. In *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, January 2015. doi: 10.2514/6.2015-0143.

[77] Jeffrey Larson, Matt Menickelly, and Stefan M. Wild. Derivative-free optimization methods. *Acta Numerica*, 28:287–404, 2019. URL https://doi.org/10.1017/S0962492919000060.

[78] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

[79] Zengcong Li, Shu Zhang, Hongqing Li, Kuo Tian, Zhizhong Cheng, Yan Chen, and Bo Wang. On-line transfer learning for multi-fidelity data fusion with ensemble of deep neural networks. *Advanced Engineering Informatics*, 53:101689, August 2022. ISSN 1474-0346. doi: 10.1016/j.aei.2022.101689. URL http://dx.doi.org/10.1016/j.aei.2022.101689.

[80] R.H. Liebeck and A.I. Ormsbee. Optimization of airfoils for maximum lift. *Journal of Aircraft*, 7(5):409–416, September 1970. doi: 10.2514/3.44192.

[81] Phillip M. Ligrani. A study of dean vortex development and structure in a curved rectangular channel with aspect ratio of 40 at dean numbers up to 430. *ntrs.nasa.gov*, 07 1994. URL https://ntrs.nasa.gov/citations/19950005258.

[82] Marius Lindauer, Katharina Eggensperger, Matthias Feurer, André Biedenkapp, Joshua Marben, Philipp Müller, and Frank Hutter. BOAH: A Tool Suite for Multi-Fidelity Bayesian Optimization; Analysis of Hyperparameters, 2019. URL https://arxiv.org/abs/1908.06756.

[83] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

[84] Hanmeng Liu, Zhizhang Fu, Mengru Ding, Ruoxi Ning, Chaoli Zhang, Xiaozhang Liu, and Yue Zhang. Logical reasoning in large language models: A survey, 2025. URL https://arxiv.org/abs/2502.09100.

[85] Peng Liu. Human-in-the-loop Bayesian Optimization with No-Regret Guarantees. October 2022. Available at SSRN: https://ssrn.com/abstract=4262945.

[86] M.W. Losey, R.J. Jackman, S.L. Firebaugh, M.A. Schmidt, and K.F. Jensen. Design and fabrication of microfluidic devices for multiphase mixing and reaction. *Journal of Microelectromechanical Systems*, 11(6):709–717, December 2002. doi: 10.1109/jmems.2002.803416.

[87] Ziming Luo, Zonglin Yang, Zexin Xu, Wei Yang, and Xinya Du. Llm4sr: A survey on large language models for scientific research, 2025. URL https://arxiv.org/abs/2501.04306.

[88] Michael Mansour, Katharina Zähringer, Krishna D.P. Nigam, Dominique Thévenin, and Gábor Janiga. Multi-objective optimization of liquid-liquid mixing in helical pipes using Genetic Algorithms coupled with Computational Fluid Dynamics. *Chemical Engineering Journal*, 391:123570, July 2020. URL https://doi.org/10.1016/j.cej.2019.123570.

[89] A. March, K. Willcox, and Q. Wang. Gradient-based multifidelity optimisation for aircraft design using Bayesian model calibration. *The Aeronautical Journal (1968)*, 115(1174):729–738, 2011. URL https://doi.org/10.1017/S0001924000006473.

[90] Andrew March and Karen Willcox. Provably convergent multifidelity optimization algorithm not requiring high-fidelity derivatives. *AIAA Journal*, 50(5):1079–1089, May 2012. ISSN 1533-385X. doi: 10.2514/1.j051125. URL http://dx.doi.org/10.2514/1.J051125.

[91] Deniele L. R. Marini. The planetary machine by johannes kepler, 2022. URL https://arxiv.org/abs/2208.04815.

[92] R. Timothy Marler and Jasbir S. Arora. The weighted sum method for multi-objective optimization: new insights. *Structural and Multidisciplinary Optimization*, 41(6):853–862, December 2009. ISSN 1615-1488. doi: 10.1007/s00158-009-0460-7. URL http://dx.doi.org/10.1007/s00158-009-0460-7.

[93] J.R. McDonough, S.M.R. Ahmed, A.N. Phan, and A.P. Harvey. The development of helical vortex pairs in oscillatory flows – A numerical and experimental study. *Chemical Engineering and Processing - Process Intensification*, 143:107588, September 2019. URL https://doi.org/10.1016/j.cep.2019.107588.

[94] J.R. McDonough, J. Armett, R. Law, and A.P. Harvey. Coil-in-Coil Reactor: Augmenting Plug Flow Performance by Combining Different Geometric Features Using 3D Printing. *Industrial & Engineering Chemistry Research*, 58(47):21363–21371, November 2019. doi: 10.1021/acs.iecr.9b04239.

[95] J.R. McDonough, S. Murta, R. Law, and A.P. Harvey. Oscillatory fluid motion unlocks plug flow operation in helical tube reactors at lower Reynolds numbers (Re $\leq$ 10). *Chem. Eng. J.*, 358:643–657, February 2019. doi: 10.1016/j.cej.2018.10.054.

[96] J.R. McDonough, M.F. Oates, R. Law, and A.P. Harvey. Micromixing in oscillatory baffled flows. *Chem. Eng. J*, 361:508–518, April 2019. ISSN 1385-8947. doi: 10.1016/j.cej.2018.12.088.

[97] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239, May 1979. doi: 10.2307/1268522.

[98] Mark McLeod, Michael A. Osborne, and Stephen J. Roberts. Practical Bayesian Optimization for Variable Cost Objectives. *arXiv preprint*, 2017. doi: 10.48550/ARXIV.1703.04335.

[99] Donald Michie and Rory Johnston. *The creative computer*. Penguin Books, Harlow, England, April 1985.

[100] David Mindell. *Between Human and Machine*. Johns Hopkins University Press, 2003. ISBN 9780801868955. doi: 10.56021/9780801868955. URL http://dx.doi.org/10.56021/9780801868955.

[101] J.-D. Müller and P. Cusdin. On the performance of discrete adjoint CFD codes using automatic differentiation. *International Journal for Numerical Methods in Fluids*, 47 (8-9):939–945, 2005. doi: 10.1002/fld.885.

[102] Jonggeol Na, Ji Hyun Bak, and Nikolaos V. Sahinidis. Efficient bayesian inference using adversarial machine learning and low-complexity surrogate models. *Comput. Chem. Eng.*, 151:107322, August 2021. doi: 10.1016/j.compchemeng.2021.107322. URL https://doi.org/10.1016/j.compchemeng.2021.107322.

[103] E. A. Nadaraya. On estimating regression. *Theory of Probability amp; Its Applications*, 9(1):141–142, January 1964. ISSN 1095-7219. doi: 10.1137/1109020. URL http://dx.doi.org/10.1137/1109020.

[104] John A. Nelder and Roger Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[105] T.N. Nguyen, T.T.P Nhat, K. Takimoto, A. Thakur, S. Nishimura, J. Ohyama, I. Miyazato, L. Takahashi, J. Fujima, K. Takahashi, and T. Taniike. High-Throughput Experimentation and Catalyst Informatics for Oxidative Coupling of Methane. *ACS Catalysis*, 10(2):921–932, December 2019. doi: 10.1021/acscatal.9b04293.

[106] Nivedita Nivedita, Phillip Ligrani, and Ian Papautsky. Dean Flow Dynamics in Low-Aspect Ratio Spiral Microchannels. *Sci. Rep.*, 7(1), March 2017. doi: 10.1038/srep44072.

[107] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2006. ISBN 9780387400655.

[108] Mike Oaksford and Nick Chater. Précis ofbayesian rationality: The probabilistic approach to human reasoning. *Behavioral and Brain Sciences*, 32(1):69–84, February 2009. ISSN 1469-1825. doi: 10.1017/s0140525x09000284.

[109] J. Tinsley Oden. Adaptive multiscale predictive modelling. *Acta Numerica*, 27: 353–450, 2018. doi: 10.1017/S096249291800003X.

[110] OpenAI. GPT-4o mini. fast, affordable small model for focused tasks. https://platform.openai.com/docs/models/gpt-4o-mini. [Accessed 17-04-2025].

[111] Eneko Osaba, Esther Villar-Rodriguez, Javier Del Ser, Antonio J. Nebro, Daniel Molina, Antonio LaTorre, Ponnuthurai N. Suganthan, Carlos A. Coello Coello, and Francisco Herrera. A tutorial on the design, experimentation and application of metaheuristic algorithms to real-world optimization problems. *Swarm and Evolutionary Computation*, 64:100888, July 2021. ISSN 2210-6502. doi: 10.1016/j.swevo.2021.100888. URL http://dx.doi.org/10.1016/j.swevo.2021.100888.

[112] Espéran Padonou and O Roustant. Polar Gaussian Processes for Predicting on Circular Domains. working paper or preprint, February 2015. URL https://hal.science/hal-01119942.

[113] F. Palacios-Gomez, L. Lasdon, and M. Engquist. Nonlinear optimization by successive linear programming. *Management Science*, 28(10):1106–1120, October 1982. ISSN 1526-5501. doi: 10.1287/mnsc.28.10.1106. URL http://dx.doi.org/10.1287/mnsc.28.10.1106.

[114] M.M. Papathanasiou, B. Burnak, J. Katz, N. Shah, and E.N. Pistikopoulos. Assisting continuous biomanufacturing through advanced control in downstream purification. *Computers &amp Chemical Engineering*, 125:232–248, June 2019. doi: 10.1016/j.compchemeng.2019.03.013.

[115] Josh Patterson and Adam Gibson. *Deep Learning*. O'Reilly Media, Sebastopol, CA, September 2017.

[116] Joel A. Paulson and Calvin Tsay. Bayesian optimization as a flexible and efficient design framework for sustainable process systems, 2024.

[117] Valerio Perrone, Huibin Shen, Matthias Seeger, Cedric Archambeau, and Rodolphe Jenatton. Learning search spaces for bayesian optimization: Another view of hyperparameter transfer learning, 2019. URL https://arxiv.org/abs/1909.12552.

[118] P. Petsagkourakis, I.O. Sandoval, E. Bradford, D. Zhang, and E.A. del Rio-Chanona. Reinforcement learning for batch bioprocess optimization. *Computers & Chemical Engineering*, 133:106649, February 2020. ISSN 0098-1354. doi: 10.1016/j.compchemeng.2019.106649.

[119] Panagiotis Petsagkourakis, Benoit Chachuat, and Ehecatl Antonio del Rio-Chanona. Safe real-time optimization using multi-fidelity gaussian processes. In *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, December 2021. doi: 10.1109/cdc45484.2021.9683599. URL https://doi.org/10.1109/cdc45484.2021.9683599.

[120] Thomas Pinder and Daniel Dodd. GPJax: A Gaussian Process Framework in JAX. *J. Open Source Softw.*, 7(75):4455, 2022. doi: 10.21105/joss.04455.

[121] J. Pinot, J.-M. Commenge, J.-F. Portha, and L. Falk. New protocol of the Villermaux–Dushman reaction system to characterize micromixing effect in viscous media. *Chem. Eng. Sci.*, 118:94–101, October 2014. ISSN 0009-2509. doi: 10.1016/j.ces.2014.07.010.

[122] Max Planck. *Scientific autobiography: And other papers*. Open Road Media, 1949.

[123] B.T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, January 1964. ISSN 0041-5553. doi: 10.1016/0041-5553(64)90137-5. URL http://dx.doi.org/10.1016/0041-5553(64)90137-5.

[124] Riccardo Porta, Maurizio Benaglia, and Alessandra Puglisi. Flow Chemistry: Recent Developments in the Synthesis of Pharmaceutical Products. *Org. Process Res. Dev.*, 20(1):2–25, December 2015. doi: 10.1021/acs.oprd.5b00325.

[125] Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets, 2022. URL https://arxiv.org/abs/2201.02177.

[126] Arjun Kumar Pukkella, Naga Ravikumar Varma Nadimpalli, Venkataramana Runkana, and Sivakumar Subramanian. A novel spiral infinity reactor for continuous hydrothermal synthesis of nanoparticles. *Sci. Rep.*, 12(1), May 2022. doi: 10.1038/s41598-022-11141-8.

[127] Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6 (65):1939–1959, 2005. URL http://jmlr.org/papers/v6/quinonero-candela05a.html.

[128] Anil Ramachandran, Sunil Gupta, Santu Rana, Cheng Li, and Svetha Venkatesh. Incorporating expert prior in Bayesian optimisation via space warping. *Knowl. Based Syst.*, 195:105663, May 2020. doi: 10.1016/j.knosys.2020.105663.

[129] Leonard Andreevič Rastrigin. Systems of extremal control. *Nauka*, 1974.

[130] Carlo Reverberi, Tommaso Rigon, Aldo Solari, Cesare Hassan, Paolo Cherubini, and Andrea Cherubini. Experimental evidence of effective human–AI collaboration in medical decision-making. *Scientific reports*, 12(1):14952, 2022.

[131] E.A. del Rio Chanona, J.E. Alves Graciano, E. Bradford, and B. Chachuat. Modifier-adaptation schemes employing gaussian processes and trust regions for real-time optimization. *IFAC-PapersOnLine*, 52(1):52–57, 2019. ISSN 2405-8963. doi: 10.1016/j.ifacol.2019.06.036. URL http://dx.doi.org/10.1016/j.ifacol.2019.06.036.

[132] Stefan Ropke and David Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40 (4):455–472, November 2006. ISSN 1526-5447. doi: 10.1287/trsc.1050.0135. URL http://dx.doi.org/10.1287/trsc.1050.0135.

[133] D. Rossi, L. Gargiulo, G. Valitov, A. Gavriilidis, and L. Mazzei. Experimental characterization of axial dispersion in coiled flow inverters. *Chemical Engineering Research and Design*, 120:159–170, April 2017. doi: 10.1016/j.cherd.2017.02.011.

[134] Jeffrey D. Sachs, Guido Schmidt-Traub, Mariana Mazzucato, Dirk Messner, Nebojsa Nakicenovic, and Johan Rockström. Six transformations to achieve the sustainable development goals. *Nature Sustainability*, 2(9):805–814, August 2019. ISSN 2398-9629. doi: 10.1038/s41893-019-0352-9. URL http://dx.doi.org/10.1038/s41893-019-0352-9.

[135] H.S. Santana, A.C. Rodrigues, M.G.M. Lopes, F.N. Russo, J.L. Silva, and O.P. Taranto. 3D printed millireactors for process intensification. *Chinese Journal of Chemical Engineering*, 28(1):180–190, January 2020. doi: 10.1016/j.cjche.2018.12.013.

[136] M. Sarkis, A. Bernardi, N. Shah, and M.M. Papathanasiou. Emerging challenges and opportunities in pharmaceutical manufacturing and distribution. *Processes*, 9(3):457, March 2021. doi: 10.3390/pr9030457.

[137] T. Savage, F. Almeida-Trasvina, E.A. del Rio Chanona, R. Smith, and D. Zhang. An integrated dimensionality reduction and surrogate optimization approach for plant-wide chemical process operation. *AIChE Journal*, 67(11), July 2021. doi: 10.1002/aic.17358.

[138] T. Savage, N Basha, O. Matar, and E.A. del Rio Chanona. Deep Gaussian Process-based Multi-fidelity Bayesian Optimization for Simulated Chemical Reactors, 2022.

[139] Tom Savage and Ehecatl Antonio del Rio Chanona. Human-algorithm collaborative bayesian optimization for engineering systems. *Computers amp; Chemical Engineering*, 189:108810, October 2024. ISSN 0098-1354. doi: 10.1016/j.compchemeng.2024.108810. URL http://dx.doi.org/10.1016/j.compchemeng.2024.108810.

[140] Tom Savage, Nausheen Basha, Jonathan McDonough, Omar K Matar, and Ehecatl Antonio del Rio Chanona. Multi-Fidelity Data-Driven Design and Analysis of Reactor and Tube Simulations. *arXiv preprint*, 2023. doi: 10.48550/arXiv.2305.00710. Accepted: Comp. Chem. Eng.

[141] Tom Savage, Nausheen Basha, Jonathan McDonough, James Krassowski, Omar Matar, and Ehecatl Antonio del Rio Chanona. Machine learning-assisted discovery of flow reactor designs. *Nature Chemical Engineering*, 1(8):522–531, August 2024. ISSN 2948-1198. doi: 10.1038/s44286-024-00099-1. URL http://dx.doi.org/10.1038/s44286-024-00099-1.

[142] Rylan Schaeffer, Mikail Khona, Zachary Robertson, Akhilan Boopathy, Kateryna Pistunova, Jason W. Rocks, Ila Rani Fiete, and Oluwasanmi Koyejo. Double descent demystified: Identifying, interpreting amp; ablating the sources of a deep learning puzzle, 2023. URL https://arxiv.org/abs/2303.14151.

[143] Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2), March 1978. ISSN 0090-5364. doi: 10.1214/aos/1176344136. URL http://dx.doi.org/10.1214/aos/1176344136.

[144] A. Serani, R. Pellegrini, J. Wackers, C.-E. Jeanson, P. Queutey, M. Visonneau, and M. Diez. Adaptive multi-fidelity sampling for CFD-based optimisation via radial basis function metamodels. *International Journal of Computational Fluid Dynamics*, 33 (6-7):237–255, August 2019. URL https://doi.org/10.1080/10618562.2019.1683164.

[145] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

[146] Benjamin J. Shields, Jason Stevens, Jun Li, Marvin Parasram, Farhan Damani, Jesus I. Martinez Alvarado, Jacob M. Janey, Ryan P. Adams, and Abigail G. Doyle. Bayesian reaction optimization as a tool for chemical synthesis. *Nature*, 590(7844):89–96, February 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03213-y. URL http://dx.doi.org/10.1038/s41586-021-03213-y.

[147] Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers, 2024. URL https://arxiv.org/abs/2409.04109.

[148] J. Singh and K.D.P. Nigam. Pilot plant study for effective heat transfer area of coiled flow inverter. *Chemical Engineering and Processing: Process Intensification*, 102: 219–228, April 2016. doi: 10.1016/j.cep.2016.02.001.

[149] Jogender Singh, Alejandro Montesinos-Castellanos, and Krishna Deo Prasad Nigam. Thermal and hydrodynamic performance of a novel passive mixer 'wavering coiled flow inverter'. *Chem. Eng. Process.*, 141:107536, July 2019. ISSN 0255-2701. doi: 10.1016/j.cep.2019.107536.

[150] Patrick Slade, Christopher Atkeson, J. Maxwell Donelan, Han Houdijk, Kimberly A. Ingraham, Myunghee Kim, Kyoungchul Kong, Katherine L. Poggensee, Robert Riener, Martin Steinert, Juanjuan Zhang, and Steven H. Collins. On human-in-the-loop optimization of human–robot interaction. *Nature*, 633(8031):779–788, September 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07697-2. URL http://dx.doi.org/10.1038/s41586-024-07697-2.

[151] Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2005. URL https://proceedings.neurips.cc/paper_files/paper/2005/file/4491777b1aa8b5b32c2e8666dbe1a495-Paper.pdf.

[152] Peiyang Song, Kaiyu Yang, and Anima Anandkumar. Lean copilot: Large language models as copilots for theorem proving in lean, 2024. URL https://arxiv.org/abs/2404.12534.

[153] Xingyou Song, Qiuyi Zhang, Chansoo Lee, Emily Fertig, Tzu-Kuo Huang, Lior Belenki, Greg Kochanski, Setareh Ariafar, Srinivas Vasudevan, Sagi Perel, and Daniel Golovin. The vizier gaussian process bandit algorithm, 2024. URL https://arxiv.org/abs/2408.11527.

[154] Kenneth Sörensen. Metaheuristics—the metaphor exposed. *International Transactions in Operational Research*, 22(1):3–18, February 2013. ISSN 1475-3995. doi: 10.1111/itor.12001. URL http://dx.doi.org/10.1111/itor.12001.

[155] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

[156] R. Stroh, J. Bect, S. Demeyer, N. Fischer, D. Marquis, and E. Vazquez. Sequential Design of Multi-Fidelity Computer Experiments: Maximizing the Rate of Stepwise Uncertainty Reduction. *Technometrics*, 64(2):199–209, July 2021. doi: 10.1080/00401706.2021.1935324.

[157] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1139–1147, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/sutskever13.html.

[158] Shion Takeno, Hitoshi Fukuoka, Yuhki Tsukada, Toshiyuki Koyama, Motoki Shiga, Ichiro Takeuchi, and Masayuki Karasuyama. Multi-fidelity Bayesian Optimization with Max-value Entropy Search and its parallelization. *arXiv preprint*, 2019. doi: 10.48550/ARXIV.1901.08275.

[159] P. Thodoroff, M. Kaiser, R. Williams, R. Arthern, S. Hosking, N. Lawrence, J. Byrne, and I. Kazlauskaite. Multi-fidelity experimental design for ice-sheet simulation. *NeurIPS Workshop on Gaussian Processes, Spatiotemporal Modeling, and Decision-making Systems*, 2022. URL https://gp-seminar-series.github.io/neurips-2022/assets/camera_ready/4.pdf.

[160] Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In David van Dyk and Max Welling, editors, *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 567–574, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR. URL https://proceedings.mlr.press/v5/titsias09a.html.

[161] Appurva Tiwari, V.M. Rajesh, and Sanjeev Yadav. Biodiesel production in micro-reactors: A review. *Energy for Sustainable Development*, 43:143–161, April 2018. doi: 10.1016/j.esd.2018.01.002.

[162] S. M Towhidul Islam Tonmoy, S M Mehedi Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. A comprehensive survey of hallucination mitigation techniques in large language models, 2024. URL https://arxiv.org/abs/2401.01313.

[163] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

[164] Niki Triantafyllou, Nilay Shah, Maria M. Papathanasiou, and Cleo Kontoravdi. *Combined Bayesian optimization and global sensitivity analysis for the optimization of simulation-based pharmaceutical processes*, page 381–386. Elsevier, 2023. ISBN 9780443152740. doi: 10.1016/b978-0-443-15274-0.50061-5.

[165] Damien van de Berg, Thomas Savage, Panagiotis Petsagkourakis, Dongda Zhang, Nilay Shah, and Ehecatl Antonio del Rio-Chanona. Data-driven optimization for process systems engineering applications. *Chemical Engineering Science*, 248:117135, February 2022. ISSN 0009-2509. doi: 10.1016/j.ces.2021.117135. URL http://dx.doi.org/10.1016/j.ces.2021.117135.

[166] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *J. Mach. Learn. Res.*, 9(11), 2008.

[167] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. URL https://arxiv.org/abs/1706.03762.

[168] Dakuo Wang, Justin D. Weisz, Michael Muller, Parikshit Ram, Werner Geyer, Casey Dugan, Yla Tausczik, Horst Samulowitz, and Alexander Gray. Human-AI Collaboration in Data Science: Exploring Data Scientists' Perceptions of Automated AI. *Proc. ACM Hum.-Comput. Interact.*, 3(CSCW), nov 2019. doi: 10.1145/3359313.

[169] Jiang-Jiang Wang, You-Yin Jing, Chun-Fa Zhang, and Jun-Hong Zhao. Review on multi-criteria decision analysis aid in sustainable energy decision-making. *Renewable and Sustainable Energy Reviews*, 13(9):2263–2278, December 2009. ISSN 1364-0321. doi: 10.1016/j.rser.2009.06.021. URL http://dx.doi.org/10.1016/j.rser.2009.06.021.

[170] Ke Wang and Alexander W Dowling. Bayesian optimization for chemical products and functional materials. *Current Opinion in Chemical Engineering*, 36:100728, June 2022. ISSN 2211-3398. doi: 10.1016/j.coche.2021.100728.

[171] Li Wang, Peiyong Ni, and Guannan Xi. The effect of off-center placement of twisted tape on flow and heat transfer characteristics in a circular tube. *Sci. Rep.*, 11(1), March 2021. doi: 10.1038/s41598-021-86285-0.

[172] Geoffrey S. Watson. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, 26(4):359–372, 1964. ISSN 0581572X. URL http://www.jstor.org/stable/25049340.

[173] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2022. URL https://arxiv.org/abs/2201.11903.

[174] Veit Wild, Motonobu Kanagawa, and Dino Sejdinovic. Connections and equivalences between the nyström method and sparse variational gaussian processes, 2021. URL https://arxiv.org/abs/2106.01121.

[175] J.M. Winter, R. Abaidi, J.W.J. Kaiser, S. Adami, and N.A. Adams. Multi-fidelity Bayesian optimization to solve the inverse Stefan problem. *Comput. Methods Appl. M.*, 410:115946, May 2023. doi: 10.1016/j.cma.2023.115946.

[176] Philip Wolfe. Convergence conditions for ascent methods. *SIAM Review*, 11(2): 226–235, April 1969. ISSN 1095-7200. doi: 10.1137/1011036.

[177] Wenjie Xu, Masaki Adachi, Colin N. Jones, and Michael A. Osborne. Principled bayesian optimisation in collaboration with human experts, 2024. URL https://arxiv.org/abs/2410.10452.

[178] Wenjie Xu, Yuning Jiang, Emilio T. Maddalena, and Colin N. Jones. Lower bounds on the noiseless worst-case complexity of efficient global optimization. *Journal of Optimization Theory and Applications*, 201(2):583–608, March 2024. ISSN 1573-2878. doi: 10.1007/s10957-024-02399-1. URL http://dx.doi.org/10.1007/s10957-024-02399-1.

[179] Ya-xiang Yuan. Recent advances in trust region algorithms. *Mathematical Programming*, 151:249–281, 2015.

[180] Xubo Yue and Raed AL Kontar. Why non-myopic bayesian optimization is promising and how far should we look-ahead? a study via rollout. In *International Conference on Artificial Intelligence and Statistics*, pages 2808–2818. PMLR, 2020.

[181] Jiyizhe Zhang, Naoto Sugisawa, Kobi C. Felton, Shinichiro Fuse, and Alexei A. Lapkin. Multi-objective Bayesian optimisation using q-noisy expected hypervolume improvement (qNEHVI) for the Schotten–Baumann reaction. *Reaction Chemistry & Engineering*, 9(3):706–712, 2024. ISSN 2058-9883. doi: 10.1039/d3re00502j.

[182] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on mathematical software (TOMS)*, 23(4):550–560, 1997.

[183] Juliusz Ziomek and Haitham Bou-Ammar. Are random decompositions all we need in high dimensional bayesian optimisation?, 2023. URL https://arxiv.org/abs/2301.12844.

# Appendix A

# Methods for Data-Driven Design and Analysis of Reactor and Tube Simulations

This is the appendix for the chapter: Methods for Data-Driven Design and Analysis of Reactor and Tube Simulations

## A.1 Pulsed-flow Helical-tube Reactors

A promising class of reactor is the helical-tube reactor under pulsed-flow conditions. Pulsed-flow conditions have been shown to induce Dean's vortices within the helical geometry. These vortices promote radial mixing within the tube and at the same time inhibit axial mixing, resulting in desirable plug-flow mixing characteristics [95, 94]. The pulsed-flow oscillation intensity has been suggested to cause vortex shedding or swirling flows which in turn impact plug flow performance. [95] confirms this by demonstrating plug-flow behaviour switching 'on' and 'off' with changes in oscillation intensity. Based on this, optimal operating conditions have been shown to occur at ratios of Dean to Reynolds numbers of between 2 and 8 and for Strohls numbers between 1 and 2 for the specific helical-tube reactor applied within

the study. However, the dependence on reactor geometry, as well as geometry optimisation has not been performed.

## A.2 Effect of hyperparameters

The optimisation procedure was repeated multiple times to investigate the effect of hyper-parameters. These include $\beta$ controlling exploration tendencies, $\gamma$ controlling how cost is weighted, and $p_\lambda$ controlling how conservative the stopping criteria becomes. We use the same initial dataset generated to provide a more fair comparison between runs. Here we reduce the exploration parameter relative to the cost-adjusting parameter, with both being reduced relative to expected objective function, with $\gamma = 1/2$, $\beta = 1/2$. Figure A.1 demonstrates the objective and cost of evaluated simulations throughout time.

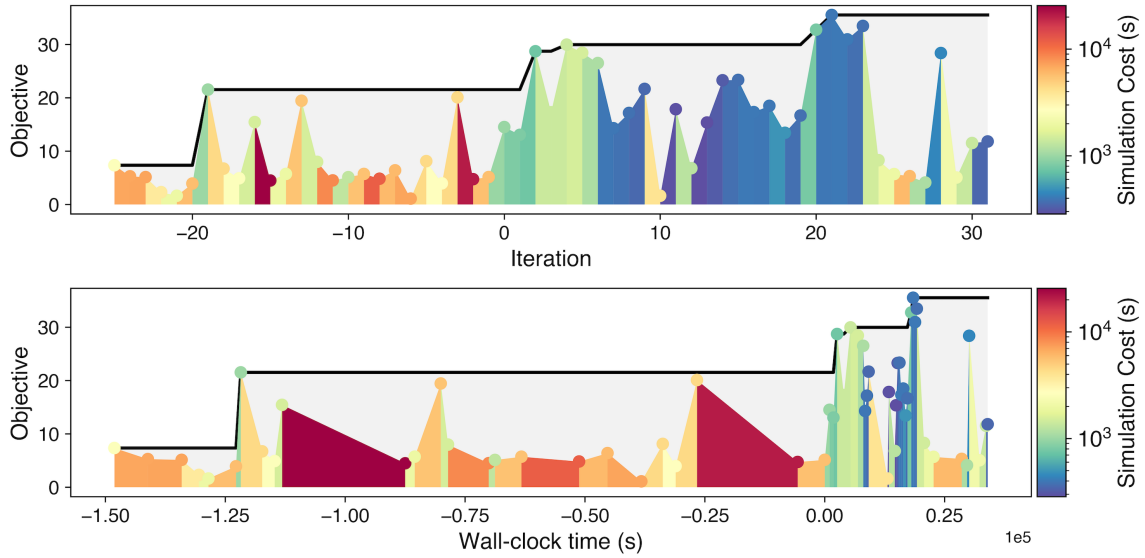



Fig. A.1 The number of equivalent tanks-in-series evaluated coloured by the respective cost of simulation. optimisation is performed with hyper-parameters $\gamma = 1/2$ and $\beta = 1/2$. The upper half of the figure shows these quantities against iteration and the lower half shows these quantities against wall-clock time, highlighting the importance of lower-cost simulations.

Figure A.2 shows the fidelities selected throughout optimisation, alongside average simulation cost, and number of evaluations.

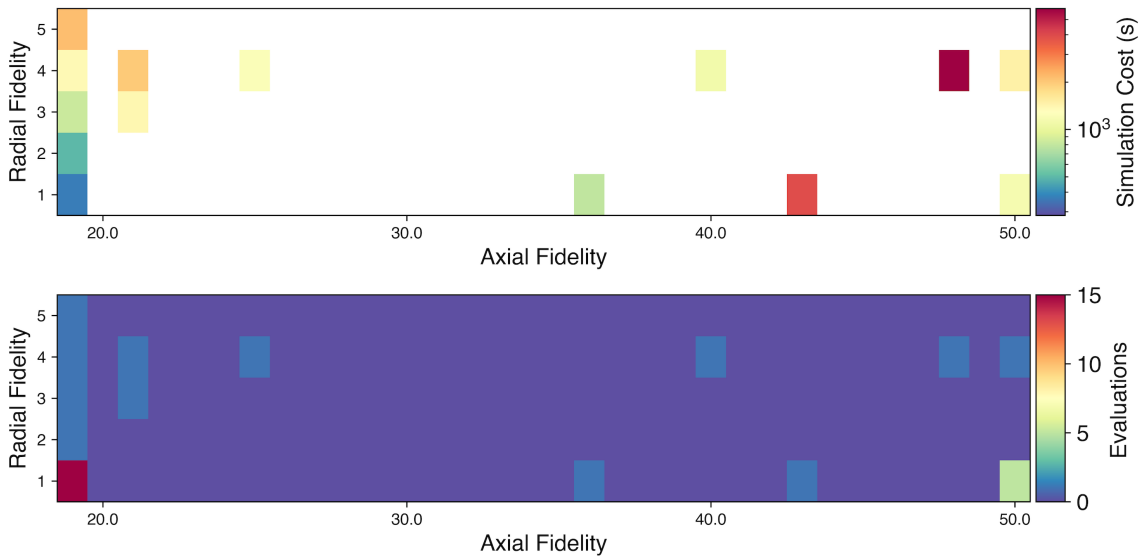



Fig. A.2 The fidelities selected throughout optimisation within $\mathbb{Z}$. The upper plot demonstrates the average simulation cost at each fidelity. optimisation is performed with hyper-parameters $\gamma = 1/2$ and $\beta = 1/2$.

Figure A.3 demonstrates the geometry of the optimal reactor resulting from running the framework with this set of hyper-parameters.

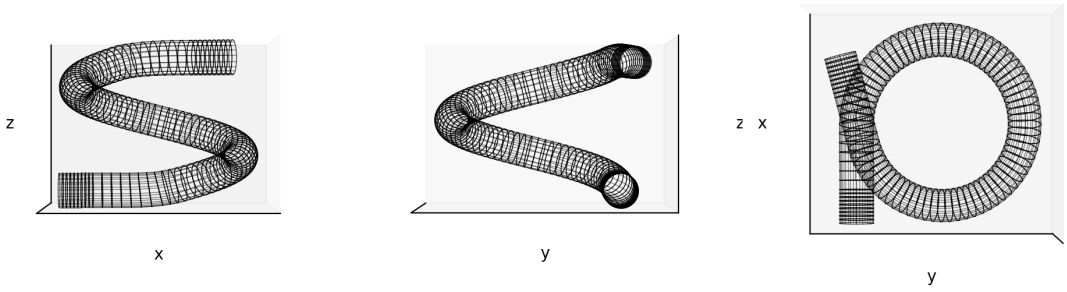



Fig. A.3 The optimal solution resulting from optimisation with hyper-parameters $\gamma = 1/2$ and $\beta = 1/2$, the solution visually is similar to the solution presented in the main part of this article. The number of equivalent tanks-in-series of this configuration and operating conditions is 34.5.

Subsequently, we increase the exploration parameter relative to the cost-adjusting parameter and expected objective function, with $\gamma = 1/2$, $\beta = 1$. Figure A.4 demonstrates the objective and cost of evaluated simulations throughout time.

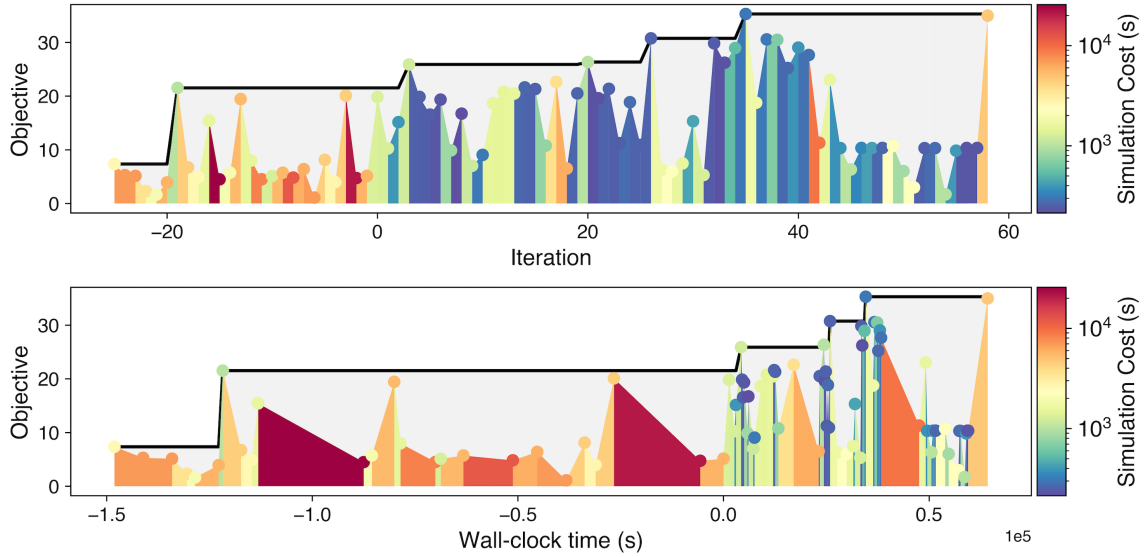



Fig. A.4 The number of equivalent tanks-in-series evaluated coloured by the respective cost of simulation. optimisation is performed with hyper-parameters $\gamma = 1/2$ and $\beta = 1$. The upper half of the figure shows these quantities against iteration and the lower half shows these quantities against wall-clock time, highlighting the importance of lower-cost simulations.

Figure A.5 shows the fidelities selected throughout optimisation, alongside average simulation cost, and number of evaluations.

Figure A.6 demonstrates the geometry of the optimal reactor resulting from running the framework with this set of hyper-parameters.

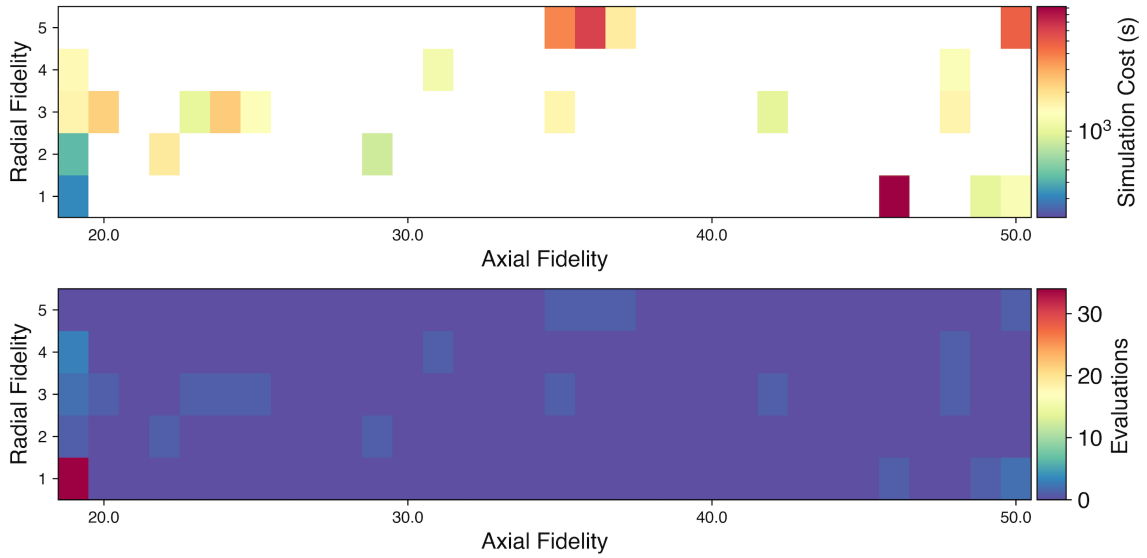



Fig. A.5 The fidelities selected throughout optimisation within $\mathbb{Z}$. The upper plot demonstrates the average simulation cost at each fidelity. optimisation is performed with hyper-parameters $\gamma = 1/2$ and $\beta = 1$.

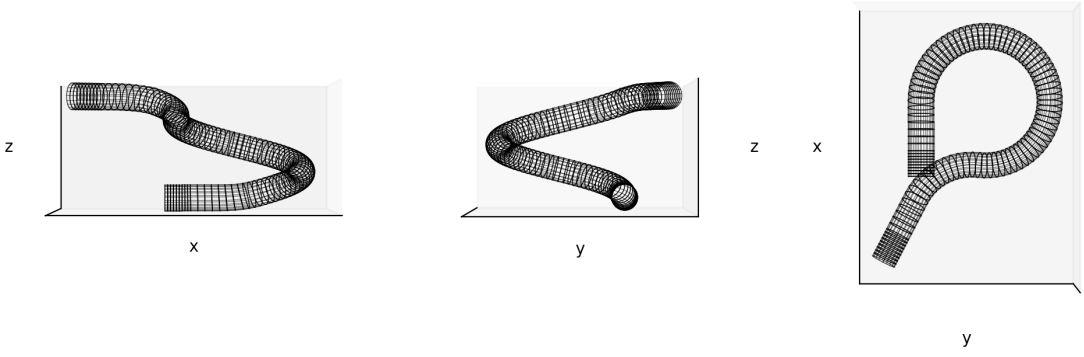



Fig. A.6 The optimal solution resulting from optimisation with hyper-parameters $\gamma = 1/2$ and $\beta = 1$, the solution differs in that it does not contain an inversion, however the coil radius and operating conditions are consistent with the other optimal solutions. The number of equivalent tanks-in-series of this configuration and operating conditions is 26.8.

# Appendix B

# Machine Learning-assisted Discovery of Flow Reactor Designs

This is the appendix for the chapter: Machine Learning-assisted Discovery of Flow Reactor Designs.

## B.1 Parameterisation & Meshing

In this section we provide further detail into the complete parameterisation and meshing procedure that occurs. In order to evaluate a given reactor design, the transformation from scalar values defining inducing points to geometry needs to occur, and in turn the transformation from geometry to a valid mesh and finally simulation. We further outline simulation details in a subsequent section. Code for the described can be found within Github repositories https://github.com/OptiMaL-PSE-Lab/pulsed-reactor-optimisation (containing optimisation procedure and meshing) and https://github.com/trsav/reactor_benchmark (containing only the meshing and reactor simulation packaged as a benchmark function, to be evaluated within a Docker container via a REST API).

### B.1.1 Cross Section

Within this chapter, 6 sets of 6 inducing points are applied for the parameterisation of cross section, although this methodology theoretically and practically allows for any combination of cross section flexibility (varying both the number of inducing cross-sections and points within a specific cross section). For simplicity we will focus on outlining the case of 6 sets of cross sections, each with 6 inducing points. The bounds of which have been previously outlined. In this section, visual aids are used to elucidate the methodology. These 36 parameters are grouped into 6 groups of 6 cross-sections, where each value defines the radius at a point within a specific cross section. A set of points containing constant values (defining a constant radius cross-section) is appended to both the start and the end of this list. These additional cross-sections enable for the reactor to interface with existing equipment and tubes, also with constant radius. Figure B.1 presents the parameters at this stage of the process.

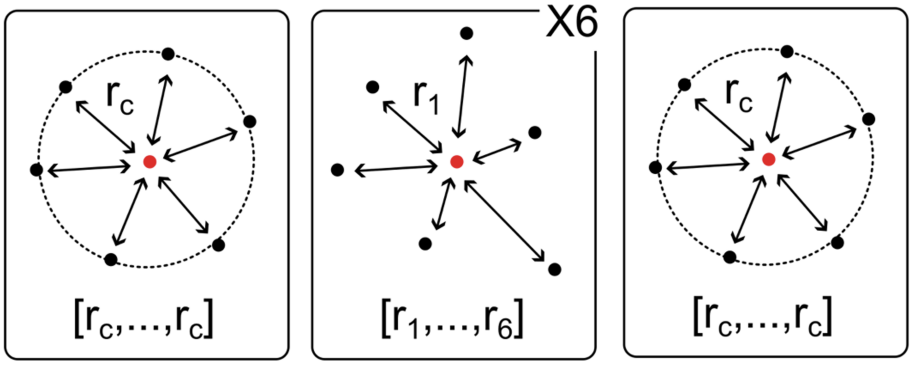



Fig. B.1 The radial variables that define the cross section of the reactor throughout the length, alongside appended cross-sections of constant radius at the start and end of the reactor.

Following this, the set of parameters defining each distinct cross-section are used within a polar Gaussian process (GP) in order to provide a smooth and fully connected interpolation. For a given set, an angle is determined between 0 and $2\pi$, equally spacing the points within polar coordinates. This angle represents the input of the GP and the radius represents the output. A GP is subsequently trained, and inference is performed across 64 equally spaced angular values, returning 64 radial values (the mean of the GP) that combined define each

cross-section. The variance of the GP is ignored. A larger number of values may be used. Figure B.2 demonstrates the GP interpolation of each cross-section.

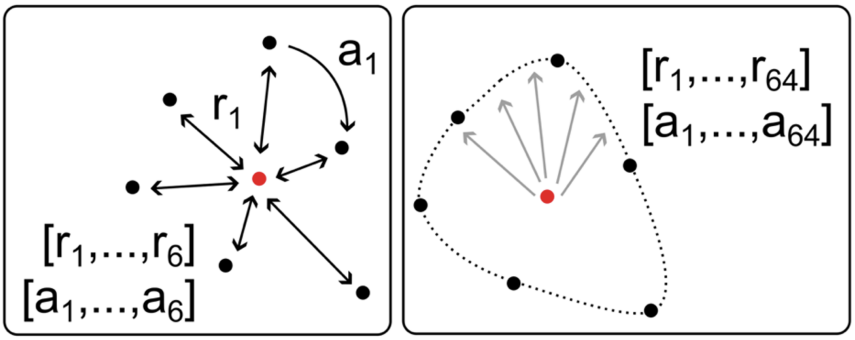



Fig. B.2 Polar GP interpolation of the 6 inducing points for each distinct cross-section. 64 points are returned which represent the highest-fidelity representation of coil cross-section.

Figure B.3 demonstrates a real example of radial inducing variables equally spaced in angular coordinates and associated Polar GP posterior mean predictions.

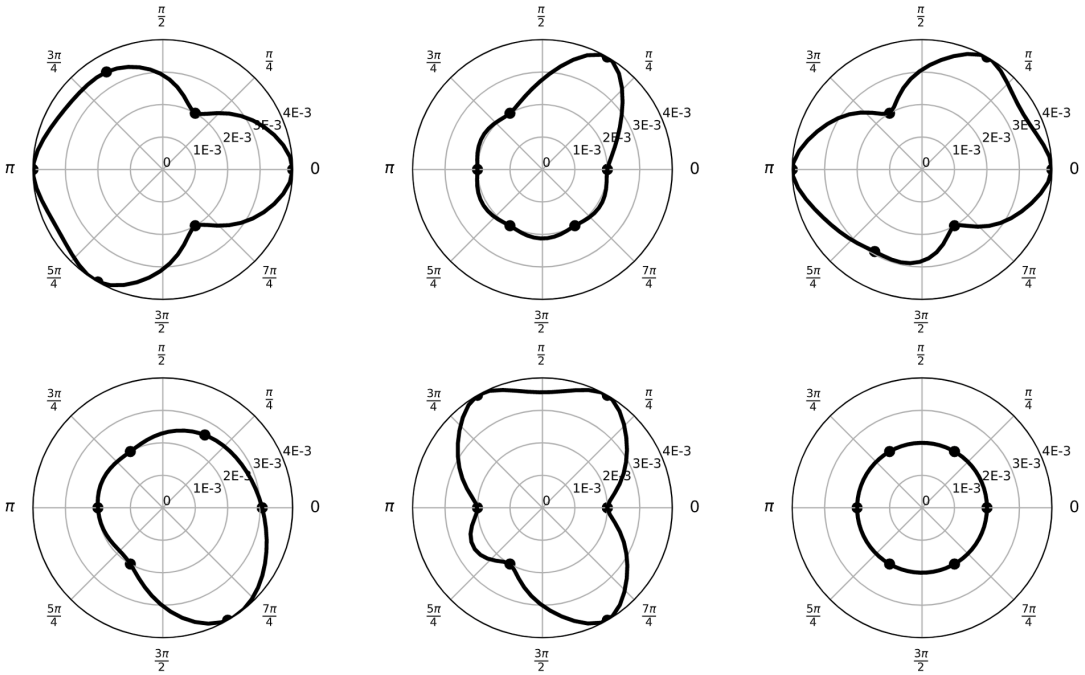



Fig. B.3 An example of cross-section inducing points and Polar GP interpolations.

The nominal reactor path is then specified as described within the main text. The resulting sets of 64 angular and radial values (herein referred to each 'cross-section') are spaced along the length of this path. Subsequently they are rotated within 3D space such that they are normal to the direction of the coil path (the direction of fluid flow). The coil path used contains an inlet length and outlet length to account for developed flow. These lengths are specified such that they smoothly transition with the reactor coil. A conversion from polar coordinates to Cartesian coordinates is performed, resulting in 8 sets of 64 points in 3D Cartesian space. Figure B.4 demonstrates the resulting data.

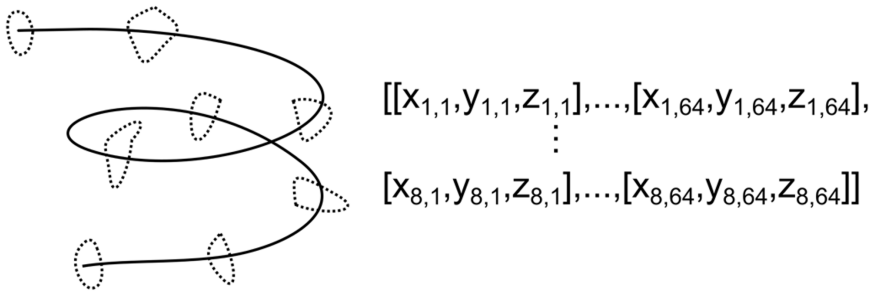



Fig. B.4 The resulting data after GP interpolation of each cross-section, and appropriate spacing in 3D cartesian coordinates.

Figure **??** demonstrates an example from the main text of cross-sections distributed along the length of the reactor.

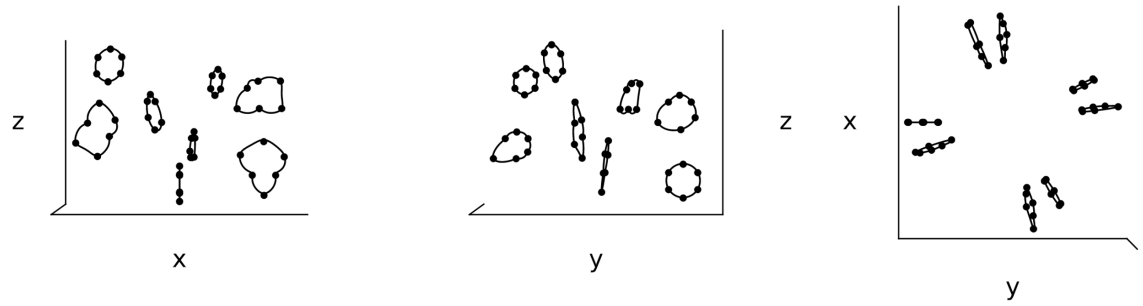



Fig. B.5 Different perspectives of interpolated cross-sections distributed along coil path.

To generate the data that defines the walls of the reactor we interpolate again, this time along the length. That is, perform 64 individual interpolations, each of 8 points, defining a path

from inlet to outlet along the wall. To perform this within Cartesian coordinates would result in non-coil-like reactors. Therefore, for each set of 8 points we make the conversion, this time to cylindrical coordinates ensuring the preservation of increasing angular coordinates to result in valid coils. Here, we generate a number of new points proportional to the axial fidelity value (whilst we could have generated Polar GP points proportional to the radial fidelity value, we opt to instead generate the full set for visualisation purposes, later limiting the values available to the mesh). In this case we will present a hypothetical 100 axial separations. These final points are what define the walls of a given reactor. Figure B.6 demonstrates the resulting data, where each interpolated 'line' from inlet to outlet is plotted, the collection of which defines the reactor wall.

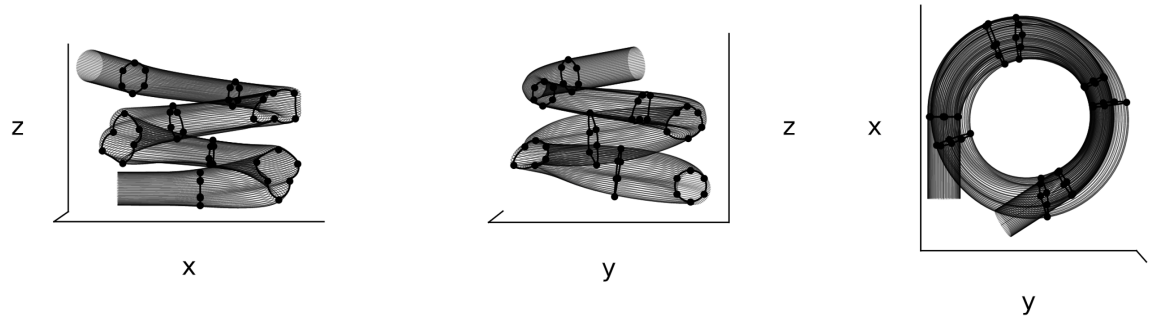



Fig. B.6 Final data made available as a result of cross-section parameterisation.

### B.1.2 Coil Path

For the parameterisation of coil path, we exploit the simplified representation of a coil within cylindrical coordinates. The procedure is much the same as the cross-section parameterisation. As opposed to GP interpolation to determine cross-section values, a set of fixed radius values are applied. The coil path that is used to distribute individual cross-sections comes from interpolation of points within cylindrical coordinates, where individual parameters represent deviations from a nominal path. However, the final information provided to the meshing procedure is the same.

## B.2 Simulation

The simulation is performed with the open-source solver OpenFOAM v1906. It is a transient simulation as the tracer as scalar field is injected into water medium for 0.15 s and the simulation is run until the tracer leaves the fluid domain. The time step size in the simulations was governed by the Courant number, with a maximum Courant number limit set at 5. To address the time-dependent nature of oscillatory velocities, the pressure-velocity coupled, incompressible, transient 'pimpleFOAM' solver was used for solving the unsteady momentum equations. Although unsteadiness was observed in the flow, it still remained within the laminar range, and as such, no turbulence models were incorporated into this study.

The 'scalarTransportFoam' solver capable of solving advection-diffusion equation is used to track the tracer concentration in the water medium. This solver was integrated with the pimpleFoam solver, facilitating the simultaneous solution of continuity, momentum, and advection-diffusion equations at each time step. The coupling of these solvers was achieved through 'Solver function objects'. Note that this scenario represents an advection-dominated flow, thereby we set an extremely low diffusion coefficient of D = $1x10^{-10}$ m2/s, with the Peclet number Pe »1. (Pecelet number is a dimensionless number that defines the ratio of the rate of advection of the tracer quantity to the diffusion of the same quantity). We normalise the inlet with the tracer concentration as [0,1], and oscillatory frequency, amplitude and time-dependent velocities were imposed on the inlet using the 'groovyBC' boundary condition from the swak4Foam library. Furthermore, the inlet velocity was initialised with a Hagen-Poiseuille parabolic velocity profile to reduce the length of the coil, thereby conserving computational resources required for flow development.

The solution scheme was chosen in a way that they provide a reasonable balance between accuracy and solution time. For instance, the time discretisation scheme for the time derivative term is set to 'Euler', a first order scheme chosen for efficient calculations. The convection flux on the computational cells was calculated using second-order discretisation schemes

to ensure the numerical accuracy of the solution. Convergence of the numerical solution at each time step was ascertained when the normalised residuals for the solved equations fell below 10-5. Additional iterations were performed within each time step through using parameters within the settings such as 'nCorrectors' for a more accurate representation of transient flow phenomena and 'nNonOrthogonalCorrectors' to account for non-orthogonal mesh cells which could have been generated due to the complex shape of the reactor and can introduce numerical error. Nonorthogonal cells are those whose faces do not form right angles with each other.

We monitor for velocity and tracer concentration field at the outlet through 'Field function objects'. These monitored data are instrumental in deriving the residence time distribution (RTD) and serve as a basis for terminating the simulation case. The termination criteria relies on the tracer concentration dropping below a stringent tolerance level of 10-7 for 10 consecutive time steps. This variable stopping criterion based on tracer concentration at the outlet accelerates the optimisation process.

## B.3 Optimisation Progress

### B.3.1 Coil Path Parameterisation

In this subsection we present the convergence results for the optimisation of coil path. 6 inducing points are applied in cylindrical for both the radial and height coordinate, but not the angular coordinate. Figure B.7 presents the optimisation progression of the coil path design problem. The objective has been normalised. The initial data set generated via design-of-experiments is denoted as negative iterations and wall-clock time. Hence, optimisation begins at the 0th iteration and at t=0. Objective values at different fidelities are biased and cannot be compared like-for-like.

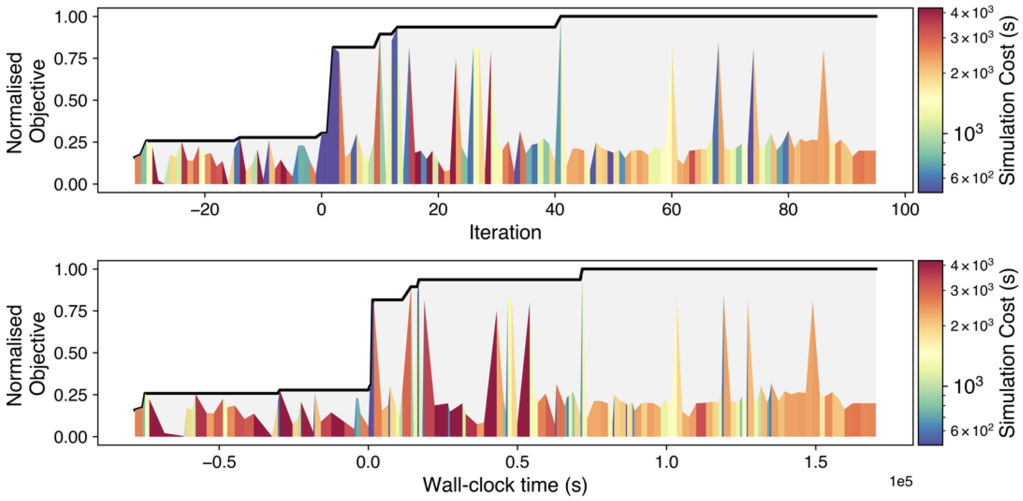



Fig. B.7 The optimisation objective against iteration and wall-clock time for the optimisation of coil path.

Cheaper simulations can be seen to be applied early on throughout optimisation, as is the case with the cross-section problem. Figure B.8 demonstrates t-SNE analysis of the coil path problem. Some structure can be observed regarding progression. However, characteristics are less distinct than the cross-section problem.

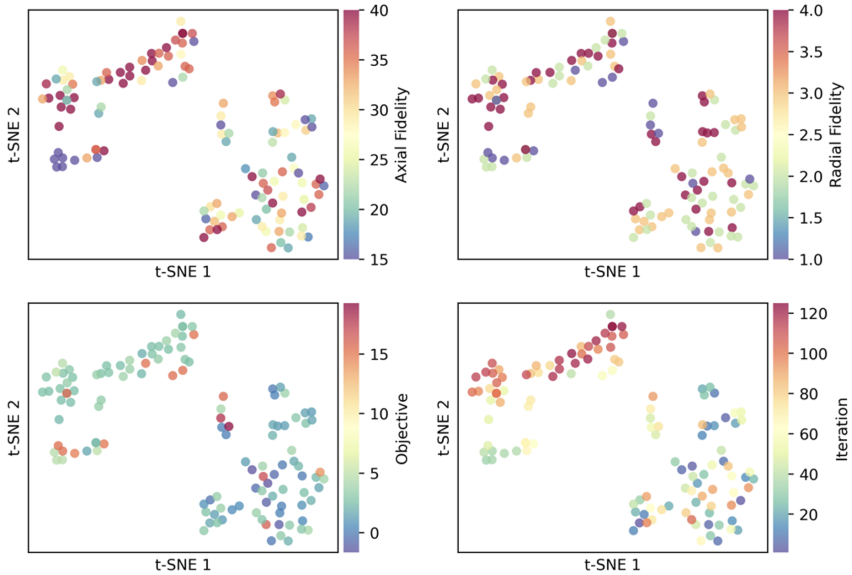



Fig. B.8 t-SNE analysis of the data generated throughout optimisation in design parameter space, reducing the dimensionality of design parameters to two dimensions, and labelled with different respective quantities.

### B.3.2 Joint Parameterisation

In this subsection we present the convergence results for joint parameterisation optimisation. The coil path was fixed as the solution to the previous problem and the cross section was allowed to vary. Figure B.9 demonstrates the relative objective values evaluated throughout this optimisation of the coil path parameterisation to maximise the composite objective function.

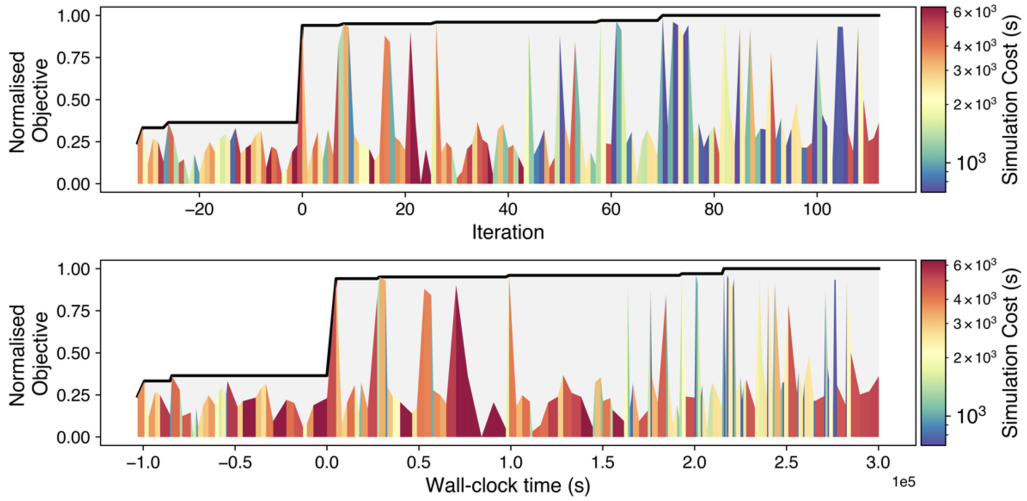



Fig. B.9 The optimisation objective against iteration and wall-clock time for the optimisation of coil path, given the fixed coil path geometry.

Systematic structures are more difficult to observe, as a broad range of fidelities, and their resulting simulation costs are evaluated. Through the solutions evaluated throughout optimisation do contain values larger than those within the initial design of experiments.

Figure B.10 presents a t-SNE analysis of the design parameters of the optimisation of cross-section given a fixed optimal coil path.

Few structures can be observed with in the t-SNE analysis of the joint parameterisation convergence data, indicating the systematic change of parameters was less likely to occur making the optimal features more difficult to identify. A small cluster of distinct solutions near the end can be observed. For this reason, we focus on individual characteristics identified in the previous two individual parameterisations.

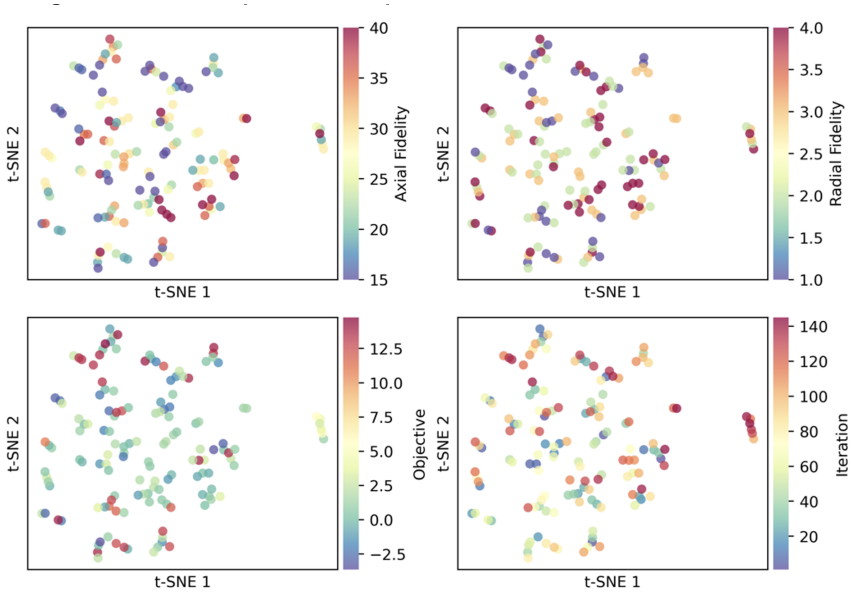



Fig. B.10 t-SNE analysis of the data generated throughout optimisation of coil cross-section given a fixed optimal coil path in design parameter space, reducing the dimensionality of design parameters to two dimensions, and labelled with different respective quantities.

## B.4 Experimental Uncertainty Quantification

We present the experimental data for each experiment, for each reactor tested including the control (reactor 1) and two bespoke designs (reactors 2 and 3). The measured quantities for each reactor can be found within the associated .CSV files. Below, we present the measured quantities after converting to dimensionless residence time distribution using the equation presented within the main chapter. For each experiment we calculate a value for the equivalent tanks-in-series by approximating the experiment using the tanks-in-series model. Figure B.11 demonstrates the experimental data for the control reactor, where values of concentration and time have been converted to dimensionless equivalents.

Figure B.12 demonstrates the experimental data for reactor 2, where values of concentration and time have been converted to dimensionless equivalents.

Figure B.13 demonstrates the experimental data for reactor 3, where values of concentration and time have been converted to dimensionless equivalents.

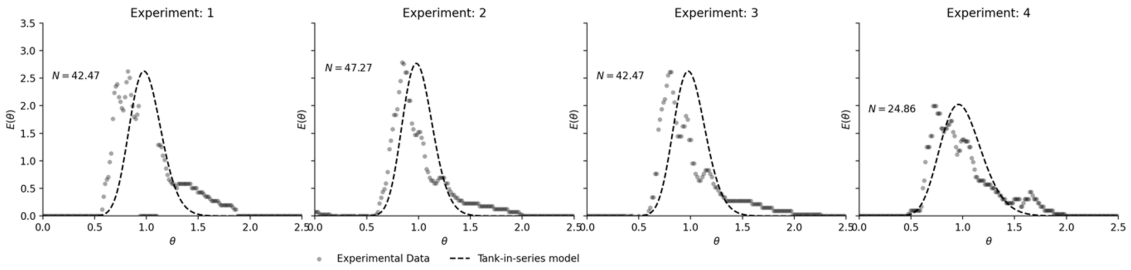



Fig. B.11 Experimental data and associated 'optimal' tank-in-series model for reactor 1 (control).

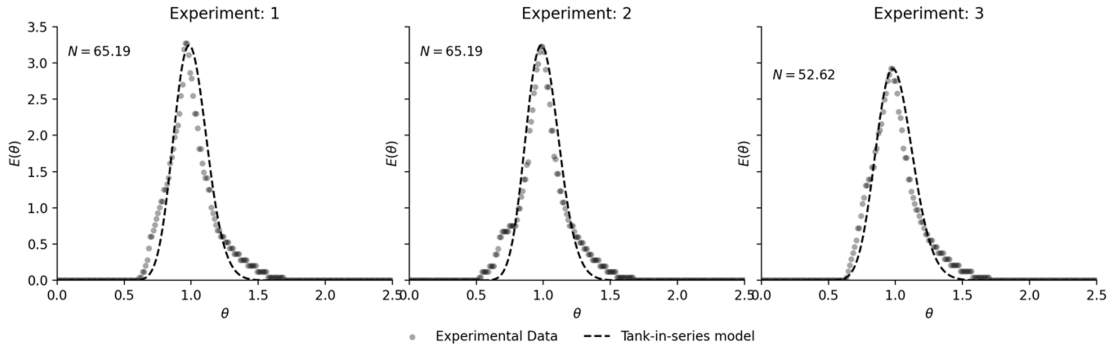



Fig. B.12 Experimental data and associated 'optimal' tank-in-series model for reactor 2 (varying cross-section).

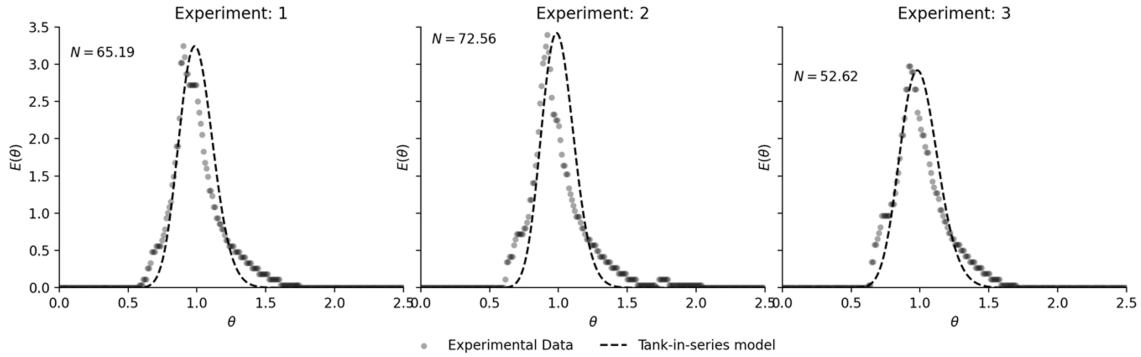



Fig. B.13 Experimental data and associated 'optimal' tank-in-series model for reactor 3 (varying cross-section and path).

For each experiment, for each reactor, the value of equivalent tanks-in-series is calculated. As a result, we are left with values of N for each reactor across all experiments, which we assume are Gaussian distributed. Figure B.14 demonstrates samples from each of these approximate distributions.

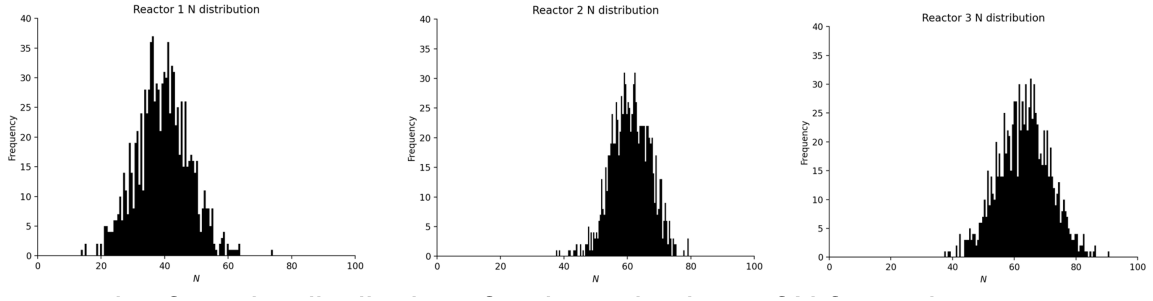



Fig. B.14 Samples from the estimated distribution of N values for each reactor.

To obtain realistic uncertainty estimates of tanks-in-series model predictions, we apply samples from the distribution of estimated values of N for each reactor to produce a number of different predictions. These result in a set of RTDs, which we subsequently process into a mean RTD and standard deviation. Subsequently, these values are plotted against the data across all experiments as shown in Figure B.15.

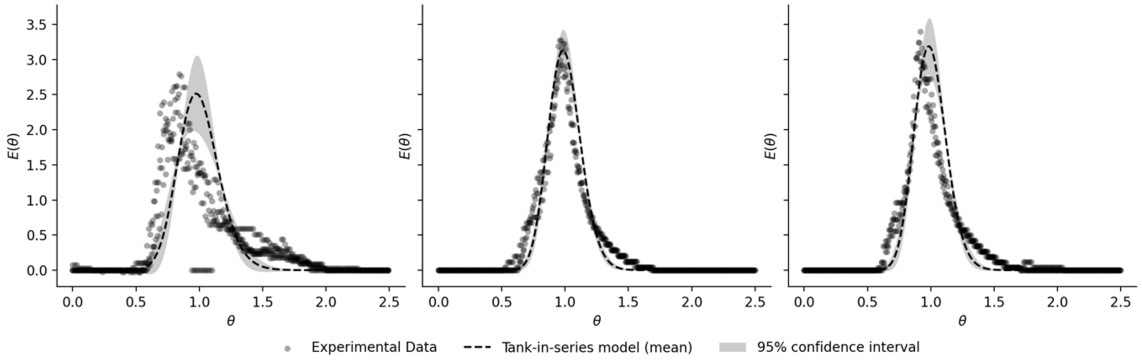



Fig. B.15 All experimental data across all three reactors, alongside uncertainty estimates of 'optimal' tanks-in-series models.

# Appendix C

# Human-algorithm Collaborative Bayesian Optimisation for Engineering Systems

This is the appendix for the chapter: Human-algorithm Collaborative Bayesian Optimisation for Engineering Systems.

## C.1 Optimisation Benchmark Functions

### C.1.1 Rastrigin

$$f(\mathbf{x}) = 10d + \sum_{i=1}^{d} \left[x_i^2 - 10\cos(2\pi x_i)\right] \tag{C.1}$$

### C.1.2 Rosenbrock

$$f(\mathbf{x}) = \sum_{i=1}^{d-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right] \tag{C.2}$$

### C.1.3 Ackley

$$f(\mathbf{x}) = -a\exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d}x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right) + a + \exp(1) \tag{C.3}$$

where $a = 20$, $b = 0.2$, $c = 2\pi$.

### C.1.4 Schewefel

$$f(\mathbf{x}) = 418.9829d - \sum_{i=1}^{d} x_i \sin(\sqrt{x_i}) \tag{C.4}$$

## C.2 Sampled Benchmark Functions

Here we visualise sampled functions from a Gaussian process prior in order to provide further intuition into our results.

Figure C.1 demonstrates 1D functions sampled from a Gaussian process prior with Matérn kernel.

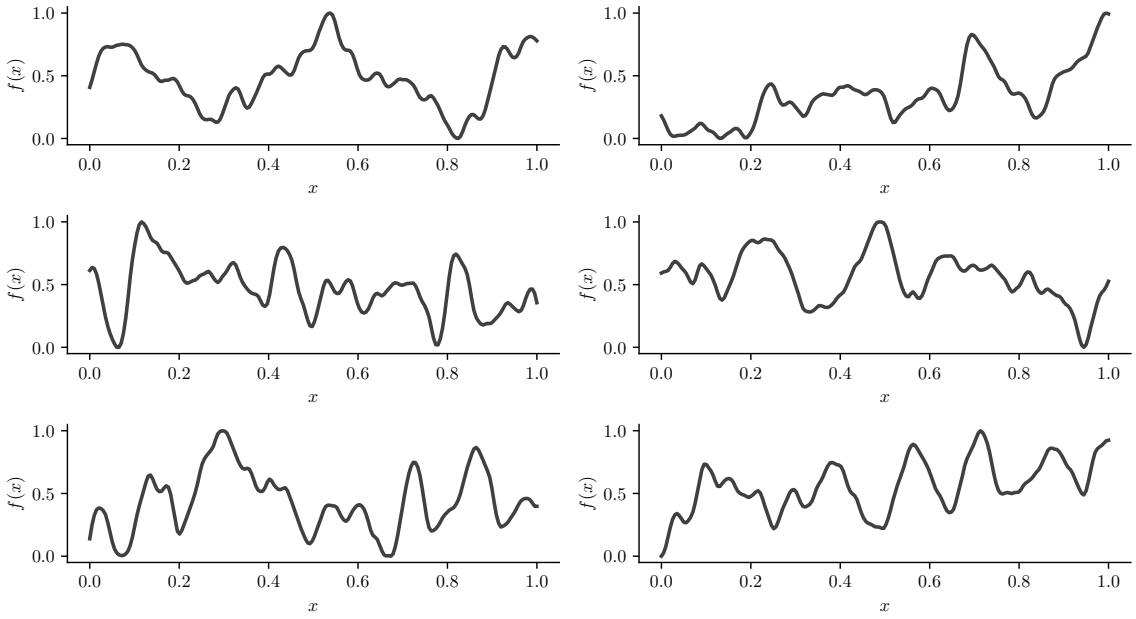



Fig. C.1 Functions sampled from a Gaussian process prior used for 1D benchmarking $f \sim \mathcal{GP}(\mu \equiv 0, K_M(d, \nu = 0.05))$ where $K_M$ is the Matérn 5/2 kernel function.

Figure C.2 demonstrates 2D functions sampled from a Gaussian process prior with Matérn kernel.

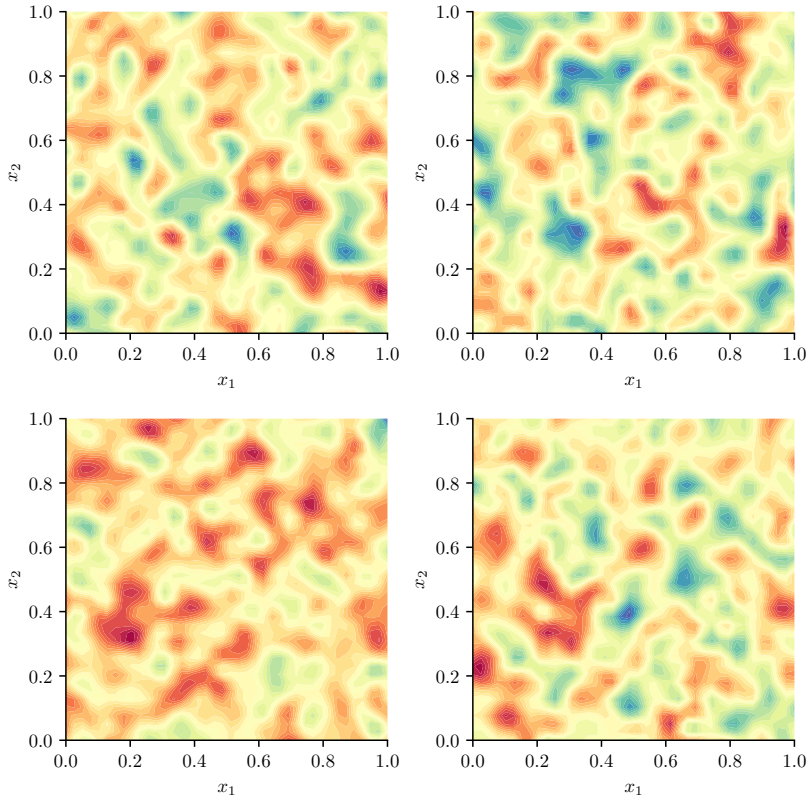


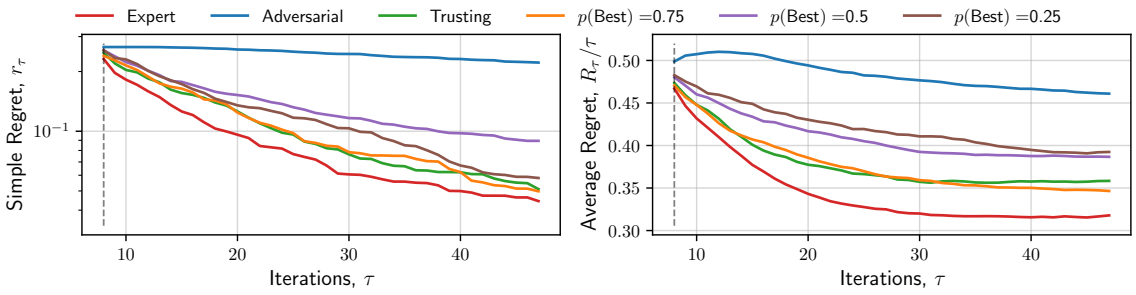

Fig. C.2 Functions sampled from a Gaussian process prior used for 2D benchmarking $f \sim \mathcal{GP}(\mu \equiv 0, K_M(d, \nu = 0.04))$ where $K_M$ is the Matérn 5/2 kernel function.

In order to generate these functions, a Gaussian process prior was sampled at $300d$ locations throughout the unit bounds. This sampled data was then used to train a noiseless Gaussian process which serves as a continuous and persistent proxy to the underlying sampled function. However, any approach that interpolates between points may be used. All code for the function generation can be located within the associated Github repository.

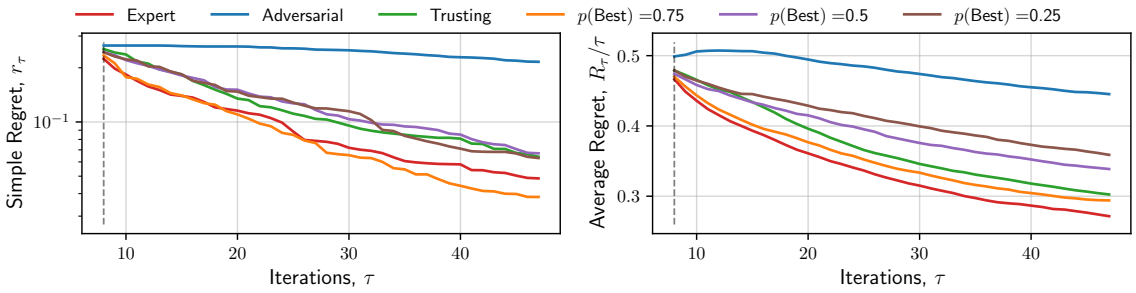
## C.3 Effect of acquisition function

In addition to other results we present the effect of the specific acquisition function here. Whilst the acquisition function itself contains a number of assumptions, differences can result in improved performance [4].

Figure C.3 demonstrates the performance of expected improvement and upper confidence bound for noiseless two dimensional functions sampled from a Gaussian process prior as previously described.




(a) Expected Improvement.

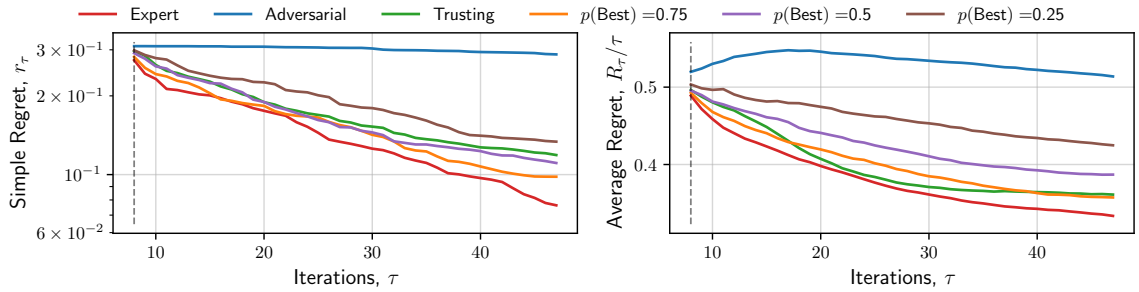



(b) Upper Confidence Bound.

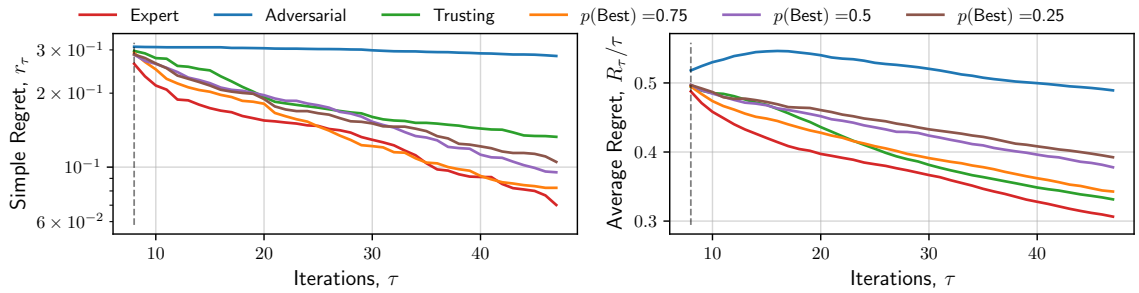
Fig. C.3 The effect of the acquisition function on the difference in convergence for sampled 2D functions.

When observing the simple regret, the convergence for $p(\text{Best}) = 0.75$ appears to outperform the expert decision, which was previously defined as the lower bound on the rate of convergence. However, with a higher number of repeated experiments, we expect the mean expert convergence to fall below that of $p(\text{Best}) = 0.75$. This hypothesis is supported by the average regret quantity, where the expert provides a more consistent lower bound. The discrepancy between simple and average regret in this case highlights the importance of considering multiple performance metrics when assessing the convergence of Bayesian optimisation algorithms.

Figure C.4 demonstrates similar results across sampled 3D functions.




(a) Expected Improvement.




(b) Upper Confidence Bound.

Fig. C.4 The effect of the acquisition function on the difference in convergence for sampled 3D functions.

Overall performance and trends across the two acquisition functions is largely similar. Final regret values for the upper confidence bound function are slightly lower than expected improvement, potentially due to the more pronounced local optima than expected improvement. However, the trend is too small to draw concrete conclusions.

## C.4 Effect of using an evolutionary approach to optimise acquisition function

In some cases, randomly selecting from the alternative solutions ($p(\text{Best}) = 0.25$) provides improved performance over standard Bayesian optimisation. This is counter-intuitive. We hypothesise that this is due to the improved ability for an evolutionary approach (used to locate alternative solutions) to sometimes find better optima of some acquisition functions

than a multi-start gradient based approach (applied to locate the 'true' optima used for standard Bayesian optimisation).

To investigate this, we perform the same benchmarks as outlined in Section 5.5.2. However, we use an evolutionary approach to optimise the acquisition function at each iteration for the 'true' optima. Figure C.5 demonstrates the convergence results across these case studies. The number of repeats and other hyperparameters remain same as described within the main article.

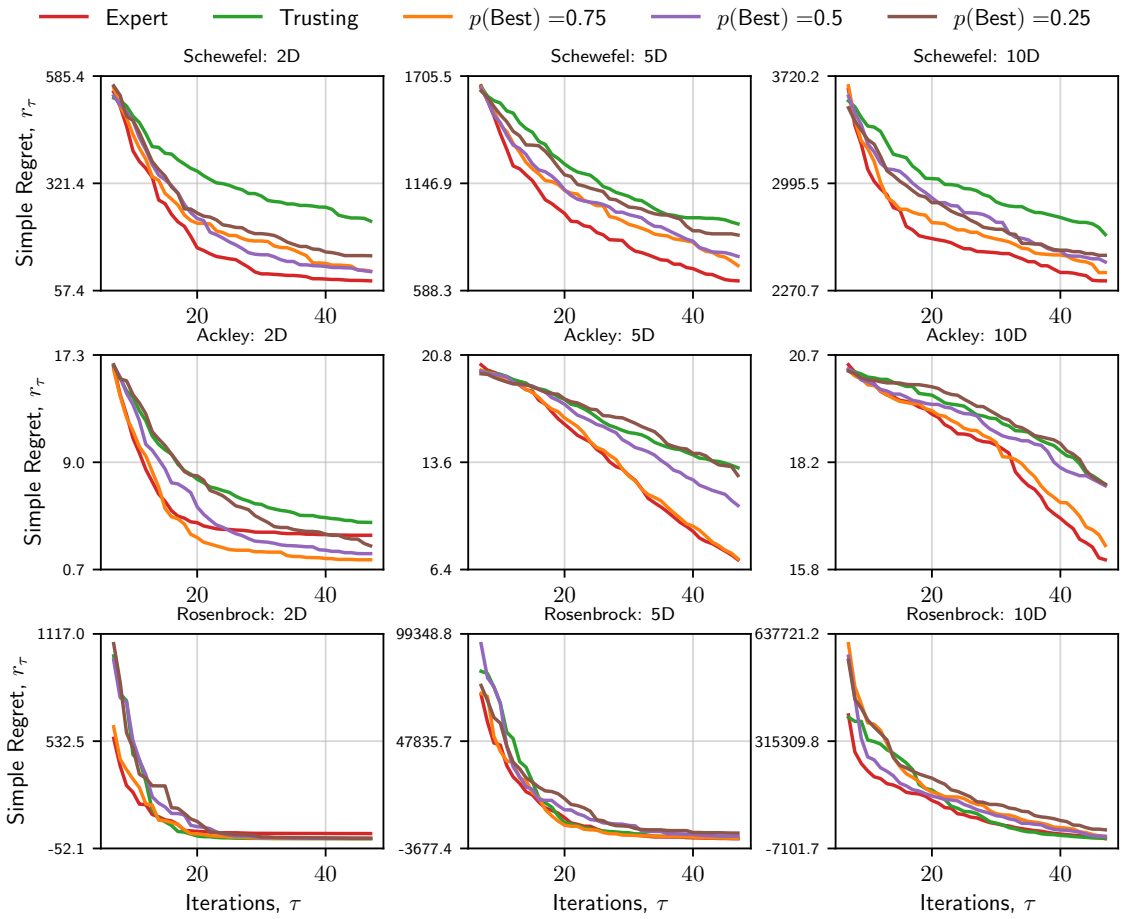



Fig. C.5 The effect of the dimensionality of the problem on the difference in convergence for various expert behaviours when using an evolutionary approach to locate the single optima of the acquisition function.

In some cases where the random expert ($p(\text{Best}) = 0.25$) performs significantly better when using a multi-start gradient-based approach (for example in the 2D and 5D and 10D Ackley function as seen in Figure 5.9), using an evolutionary approach to optimise the acquisition function results in less relative difference between the convergence of standard Bayesian optimisation and random selection between alternatives. In these cases, an evolutionary approach is more effective to optimise the acquisition function, and improvements in convergence when considering expert abilities is still maintained.

# Appendix D

# In-silico Improvements to Bayesian Optimisation using Large Language Models

This is the appendix for the chapter: In-silico Improvements to Bayesian Optimisation using Large Language Models.

## D.1 LLM Prompts for Decision Making

### D.1.1 Basic One-Shot Prompt

```
<task>
You are a decision maker who must make a choice about which solution to
    select based on the provided information. Your role is to evaluate the
    alternatives and choose the best solution based on your expertise in the
    subject matter.
You will be provided previous solutions and their objective values
```

```
Your task is to choose a solution that you believe will perform the best
    based on the information provided.
You will be given a few alternative solutions in JSON format, and you must
    use this data to determine the key distinctions among the solutions.
</task>

<additional-information>
You must return your solution choice as a JSON object with the key 'choice'
    and the value as the index of the best solution.
You must also provide a reason for your choice in the JSON object with the
    key 'reason'.
The reason must be one or two unformatted sentences only.
</additional-information>



Here is an example of what to expect:

<example>
<previous-iterations>
[{"x1":10,"x2":20,"x3":50,"objective":11},{"x1":20,"x2":48,"x3":59,"objective
    ":28},{"x1":31,"x2":42,"x3":80,"objective":39}]
</previous-iterations>

<data>
Solution 1: {"x1":54,"x2":26,"x3":34}
Solution 2: {"x1":21,"x2":23,"x3":25}
Solution 3: {"x1":23,"x2":49,"x3":53}
</data>
```

```
<response>
{"choice": 3,"reason": "Based on previous iterations and the main differences
    between the solutions, solution 3 is the best choice because it has the
    highest x2 and x3 values, which are associated with better objective
    values."}
</response>
</example>
```

### D.1.2 Historian Prompt

```
    <personality>
You are an analytical data historian who objectively examines solutions and
    identifies trends in the data that suggest both poor and good performance.
    Your role within the team is to provide balanced insights into the
    characteristics of solutions based on their objective values.
</personality>

<task>
Your task is to describe the properties of solutions with characteristically
    low and high objective values, identifying both bad and good aspects of
    the solutions.
You will be given previous solutions in JSON format, and you must use this
    data to determine which variables, or combinations of variables, and
    respective values are associated with low and high objective values.
</task>

<additional-information>
```

```
All variables and objectives range between 0 and 100.
Be clear and concise in your description, providing a balanced analysis of
    the solutions' characteristics.
Only discuss:
1. properties of bad solutions
2. properties of good solutions
3. general comments
</additional-information>

Here are some examples of what to expect:

<example>
<data>
[{"x1":10,"x2":20,"x3":50,"objective":11},{"x1":20,"x2":48,"x3":59,"objective
    ":28},{"x1":31,"x2":42,"x3":80,"objective":39}]
</data>

<trends>
Based on the provided data, the following trends can be observed:

Bad solutions have:
- Low values for x1: Solutions with lower x1 values are associated with lower
     objective values.
- Low values for x2: Solutions with lower x2 values correspond to lower
    objective values.
- Low values for x3: Similarly, solutions with lower x3 values are linked to
    lower objective values.
```

```
Good solutions have:
- Higher values for x1: As x1 values increase, the objective values tend to
    improve.
- Higher values for x2: Solutions with higher x2 values are associated with
    better objective values.
- Higher values for x3: Higher x3 values correspond to improved objective
    values.

In summary, solutions with low values for x1, x2, and x3 variables tend to
    have worse objective values, while solutions with higher values for these
     variables are likely to perform better.
</trends>
</example>

<example>
<data>
[{"x1":54,"x2":26,"x3":34,"objective":16},{"x1":21,"x2":23,"x3":25,"objective
    ":59},{"x1":23,"x2":49,"x3":53,"objective":41}]
</data>

<trends>
Based on the provided data, the following trends can be observed:

Bad solutions have:
- High values for x1: Solutions with higher x1 values are associated with
    lower objective values.
- Low values for x3: Similarly, solutions with lower x3 values are linked to
    lower objective values.
```

```
Good solutions have:
- Lower values for x1: Solutions with lower x1 values tend to have higher
    objective values.
- When all variables are low at the same time: Solutions with low values for
    x1, x2, and x3 tend to have better objective values.

x2 doesn't seem to have a clear relationship with the objective values.
In summary, solutions with high values for x1 and low values for x2 and x3
    variables tend to have low objective values, while solutions with lower
    x1 values and higher x2 and x3 values are likely to perform better.
</trends>
</example>
```

### D.1.3 Differentiator Prompt

```
    <personality>
You are a concise differentiator who identifies the key distinctions between
    solutions. Your role is to provide brief, clear descriptions of the main
    differences among the provided solutions.
Comment only on values of the variables and their relative differences.
</personality>

<task>
Your task is to succinctly describe the main differences between the provided
     solutions, focusing on the most relevant variables.
You will be given a few alternative solutions in JSON format, and you must
    use this data to determine the key distinctions among the solutions.
```

```
</task>

<additional-information>
All variables range between 0 and 100.
Be brief and to the point in your description, providing a focused analysis
    of the main differences between the solutions.
</additional-information>

Here are some examples of what to expect:

<example>
<data>
Solution 1: {"x1":10,"x2":20,"x3":50}
Solution 2: {"x1":20,"x2":48,"x3":59}
Solution 3: {"x1":31,"x2":42,"x3":80}
</data>

<response>
The main differences between the solutions are:
- Solution 1 has the lowest value for x1
- Solution 2 has the highest value for x2 and moderate values for x1 and x3
- Solution 3 has the highest value for x1 and x3
</response>
</example>

<example>
<data>
Solution 1: {"x1":54,"x2":26,"x3":34}
```

```
Solution 2: {"x1":21,"x2":23,"x3":25}
Solution 3: {"x1":23,"x2":49,"x3":53}
</data>

<response>
The main differences between the solutions are:
- Solution 1 has a significantly higher x1 value compared to the other
    solutions.
- Solution 2 has the lowest values for all variables.
- Solution 3 has the highest x2 and x3 values.
</response>
</example>
```

### D.1.4 Decision-maker Prompt

```
    <task>
You are a decision maker who must make a choice about which solution to
    select based on the provided information. Your role is to evaluate the
    alternatives and choose the best solution based on your expertise in the
    subject matter.
You will be provided information about previous trends, which variables are
    most important, and the differences between solutions.
Your task is to choose a solution that you believe will perform the best
    based on the information provided.
You will be given a few alternative solutions in JSON format, and you must
    use this data to determine the key distinctions among the solutions.
</task>
```

```
<additional-information>
You must return your solution choice as a JSON object with the key 'choice'
    and the value as the index of the best solution.
You must also provide a reason for your choice in the JSON object with the
    key 'reason'.
The reason must be one or two unformatted sentences only.
</additional-information>



Here is an example of what to expect:


<example>
<data>
Solution 1: {"x1":54,"x2":26,"x3":34}
Solution 2: {"x1":21,"x2":23,"x3":25}
Solution 3: {"x1":23,"x2":49,"x3":53}
</data>


<solution-differences>
The main differences between the solutions are:
- Solution 1 has a significantly higher x1 value compared to the other
    solutions.
- Solution 2 has the lowest values for all variables.
- Solution 3 has the highest x2 and x3 values.
</solution-differences>


<previous-iterations>
In previous iterations, bad solutions have:
```

```
- Low values for x1: Solutions with lower x1 values are associated with lower
    objective values.
- Low values for x2: Solutions with lower x2 values correspond to lower
   objective values.
- Low values for x3: Similarly, solutions with lower x3 values are linked to
   lower objective values.

In previous iterations, good solutions have:
- Higher values for x1: As x1 values increase, the objective values tend to
   improve.
- Higher values for x2: Solutions with higher x2 values are associated with
   better objective values.
- Higher values for x3: Higher x3 values correspond to improved objective
   values.
In summary, solutions with low values for x1, x2, and x3 variables tend to
   have worse objective values, while solutions with higher values for these
    variables are likely to perform better.
</previous-iterations>

<response>
{"choice": 3,"reason": "Based on previous iterations and the main differences
    between the solutions, solution 3 is the best choice because it has the
   highest x2 and x3 values, which are associated with better objective
   values."}
</response>
</example>
```

**Chain-of-thought Prompt**

```
    You are a decision maker who must make a choice about which solution to
    select based on the provided information. Your role is to evaluate the
    alternatives and choose the best solution based on your expertise in the
    subject matter.
You will be provided previous solutions and their objective values
Your task is to choose a solution that you believe will perform the best
    based on the information provided.
You will be given a few alternative solutions in JSON format, and you must
    use this data to determine the key distinctions among the solutions.

You engage in extremely thorough, self-questioning reasoning. Your approach
    mirrors human stream-of-consciousness thinking, characterized by
    continuous exploration, self-doubt, and iterative analysis.
## Core Principles
1. EXPLORATION OVER CONCLUSION
- Never rush to conclusions about a decision
- Keep exploring until a strategy emerges naturally from the evidence
- If uncertain, continue reasoning about a strategy indefinitely
- Question every assumption and inference
2. DEPTH OF REASONING
Engage in extensive contemplation about the best way to make a decision (
    minimum 10,000 characters)
- Express thoughts in natural, conversational internal monologue
- Break down complex thoughts into simple,
atomic steps
- Embrace uncertainty and revision of previous thoughts
3. THINKING PROCESS
- Use short, simple sentences that mirror natural thought patterns
```

```
- Express uncertainty and internal debate freely
- Show work-in-progress thinking
- Acknowledge and explore dead
ends with respect to decisions that may not work
- Frequently backtrack and revise
4. PERSISTENCE
- Value thorough exploration over quick resolution
## Output Format
Your responses must follow this exact structure given below. Make sure to
    always include the final answer.

<format>
<contemplator>
[Your extensive internal monologue goes here]
- Begin with small, foundational observations
- Question
each step thoroughly
- Show natural thought progression
- Express doubts and
uncertainties
- Revise and backtrack if you need to
- Continue until natural resolution
</contemplator>

<response>
FINAL ANSWER HERE
</response>
</format>
```

```
<additional-information>
You must return your solution choice as a JSON object with the key 'choice'
    and the value as the index of the best solution.
You must also provide a reason for your choice in the JSON object with the
    key 'reason'.
The reason must be one or two unformatted sentences only.
</additional-information>

Here is an example of what to expect:

<example>
<previous-iterations>
[{"x1":10,"x2":20,"x3":50,"objective":11},{"x1":20,"x2":48,"x3":59,"objective
    ":28},{"x1":31,"x2":42,"x3":80,"objective":39}]
</previous-iterations>

<data>
Solution 1: {"x1":54,"x2":26,"x3":34}
Solution 2: {"x1":21,"x2":23,"x3":25}
Solution 3: {"x1":23,"x2":49,"x3":53}
</data>

<contemplator>
...
</contemplator>

<response>
```

```
{"choice": 3,"reason": "Based on previous iterations and the main differences
    between the solutions, solution 3 is the best choice because it has the
   highest x2 and x3 values, which are associated with better objective
   values."}
</response>
</example>
```