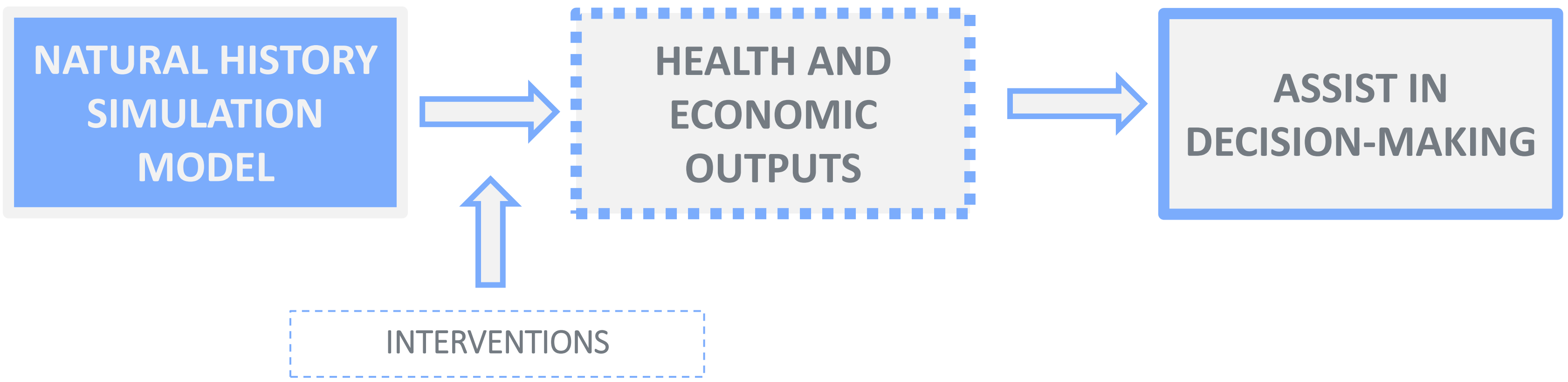
BAYESIAN OPTIMIZATION WITH ADDITIVE KERNELS FOR THE CALIBRATION OF SIMULATION MODELS TO PERFORM COST-EFFECTIVENESS ANALYSIS

David Gómez, Mireia Diaz, Josep Lluis Arcos, Jesus Cerquides

BACKGROUND

- Cost-Effectiveness Analyses in healthcare are a type of economic evaluation that COMPARE the COSTS and BENEFITS of different health interventions (e.g., diagnosis, screening, treatment,…).
- The objective is to MAXIMIZE the health benefits of the population while MINIMIZING costs.
- These analyses can be performed in clinical studies or within the framework of disease **simulation models.**

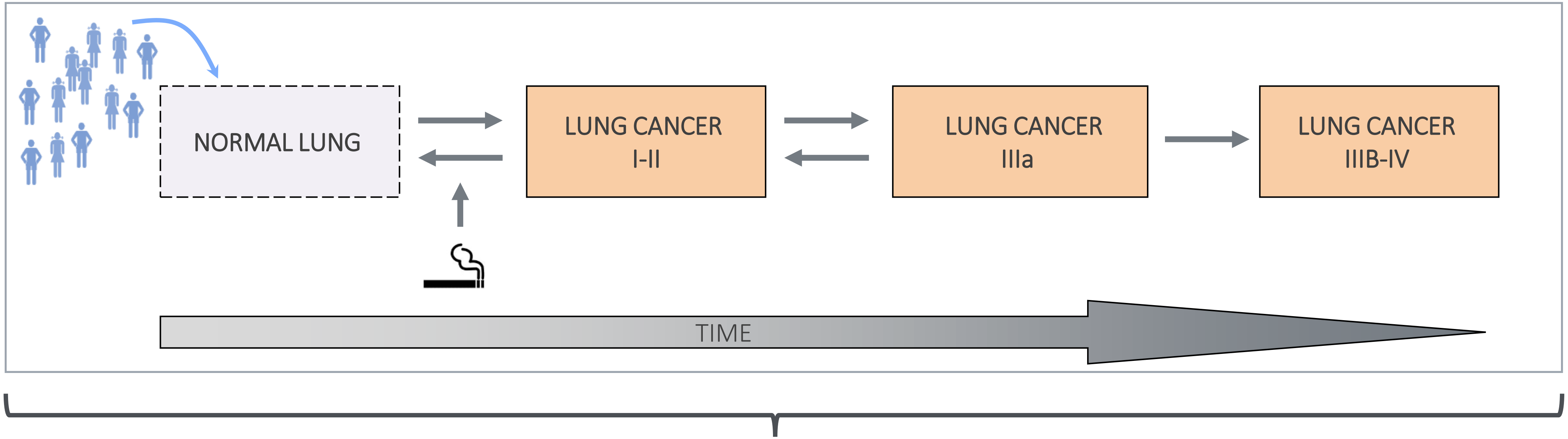


- They can **help decision-making** and allocate resources in complex and uncertain situations.

SIMULATION MODELS IN HEALTHCARE

A mathematical simulation model in healthcare tries to reproduce the dynamics of a disease over time.

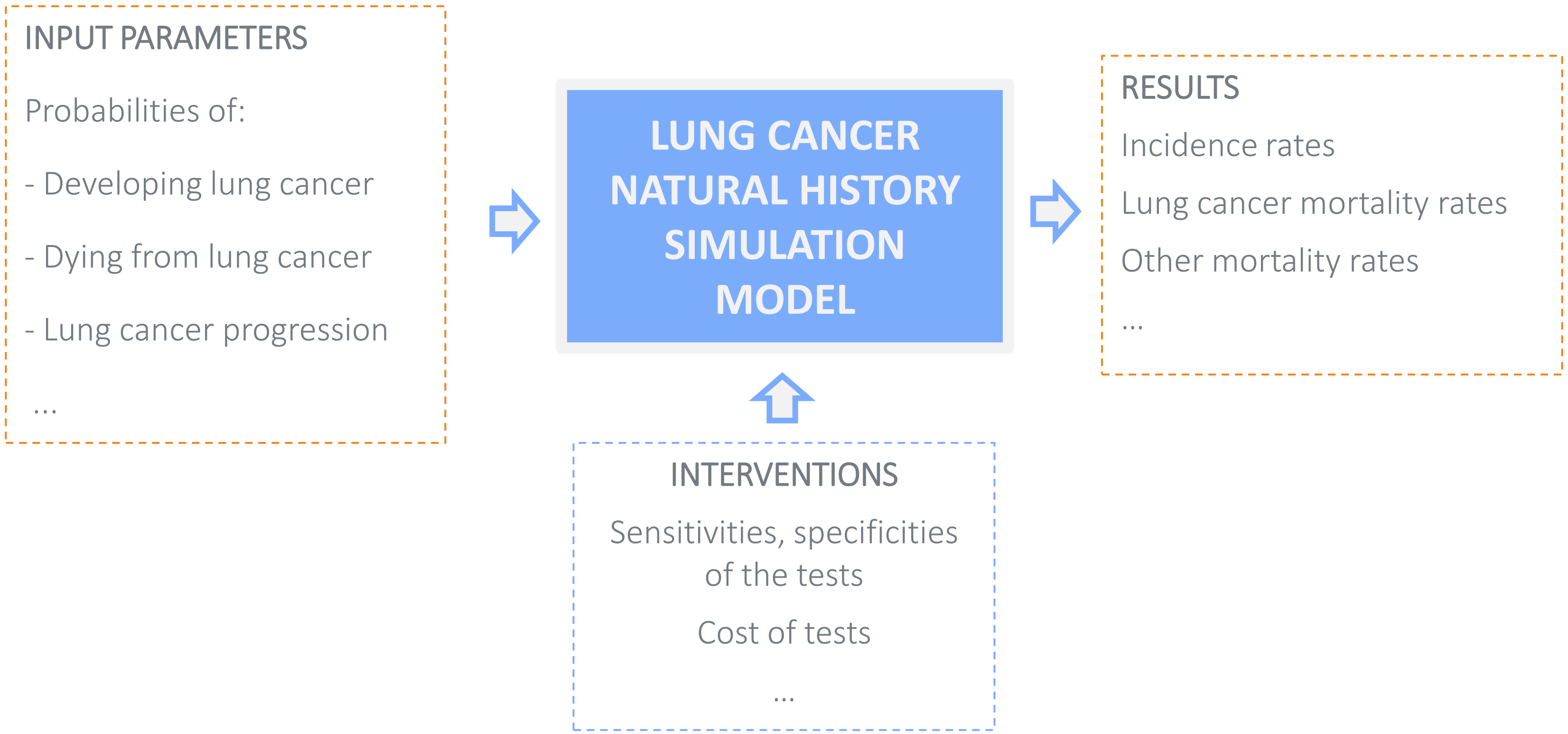
Example: Simulation of the natural history of lung cancer



HEALTH & ECONOMIC IMPACT

SIMULATION MODELS IN HEALTHCARE

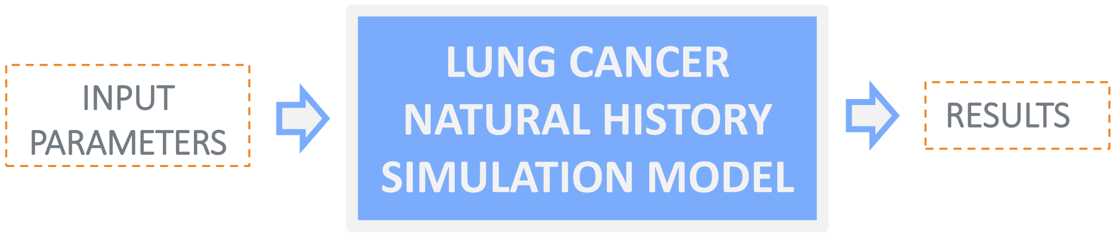
Simulation model depends not only on the logic and structure of the modeling system but also on the quality of the input data.



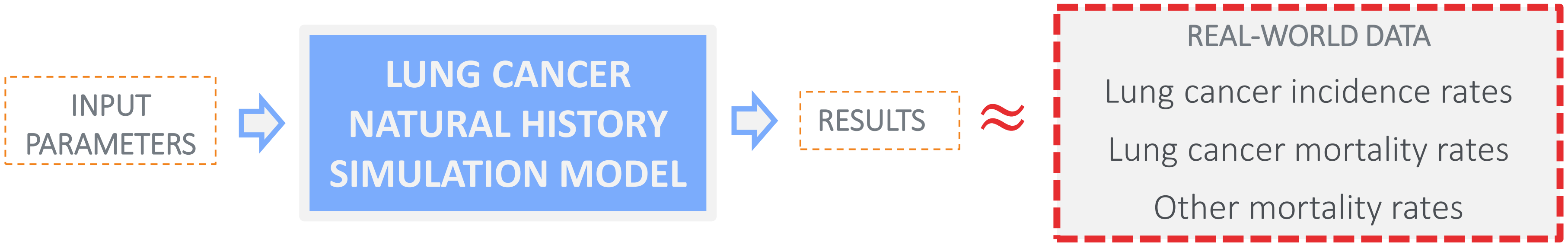
MODEL CALIBRATION

- These parameters are usually obtained from scientific literature or expert consensus which are usually subject to some degree of uncertainty.
- This can lead to unusable results and poor conclusions.
- To mitigate the impact of the uncertainty of this data, it is necessary to establish a procedure that compares the output data generated by the simulation model with the knowledge of the natural history of the disease (calibration).

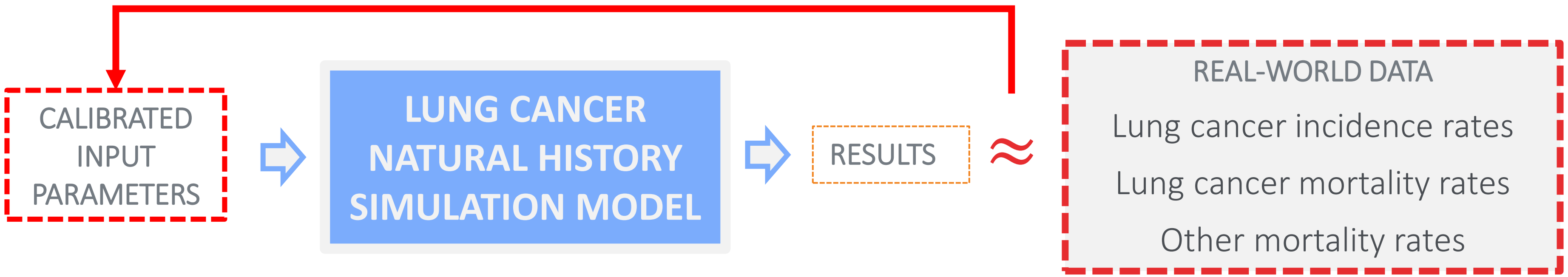
MODEL CALIBRATION



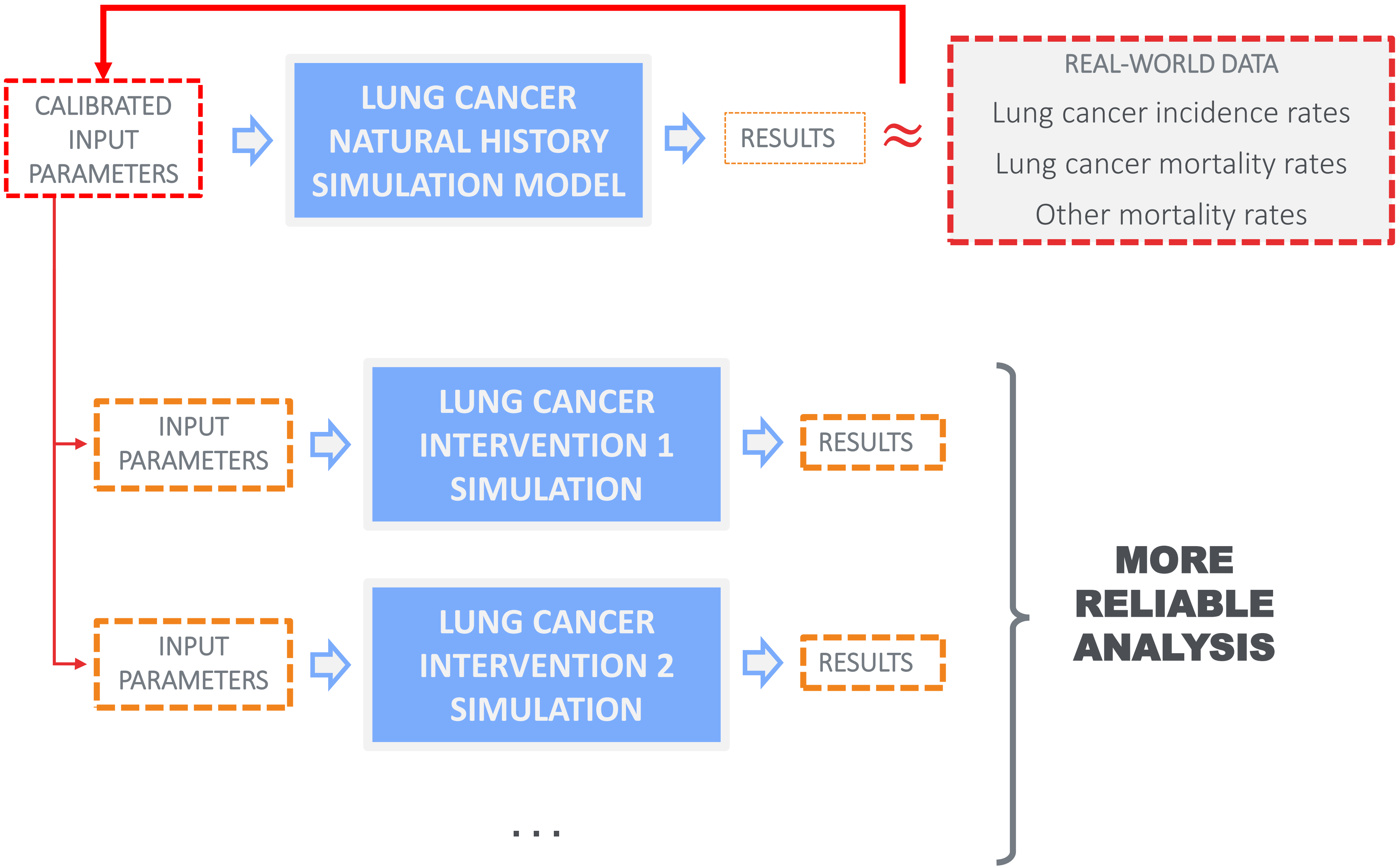
MODEL CALIBRATION



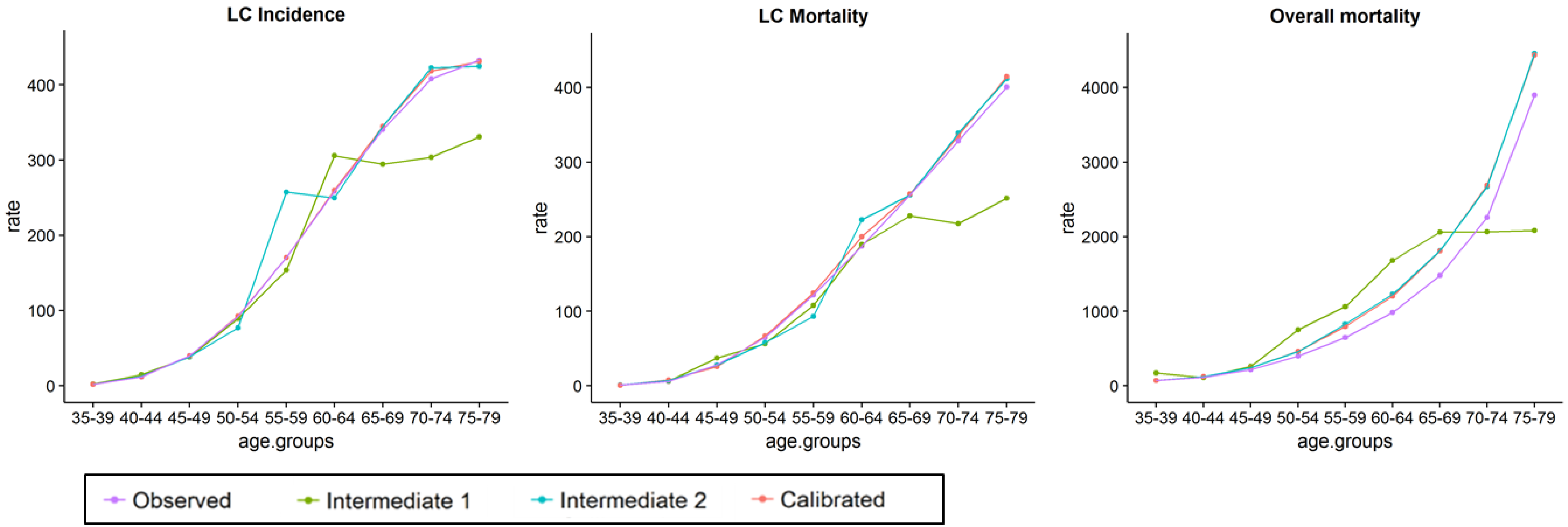
MODEL CALIBRATION



MODEL CALIBRATION



EXAMPLE OF LUNG CANCER MODEL CALIBRATION

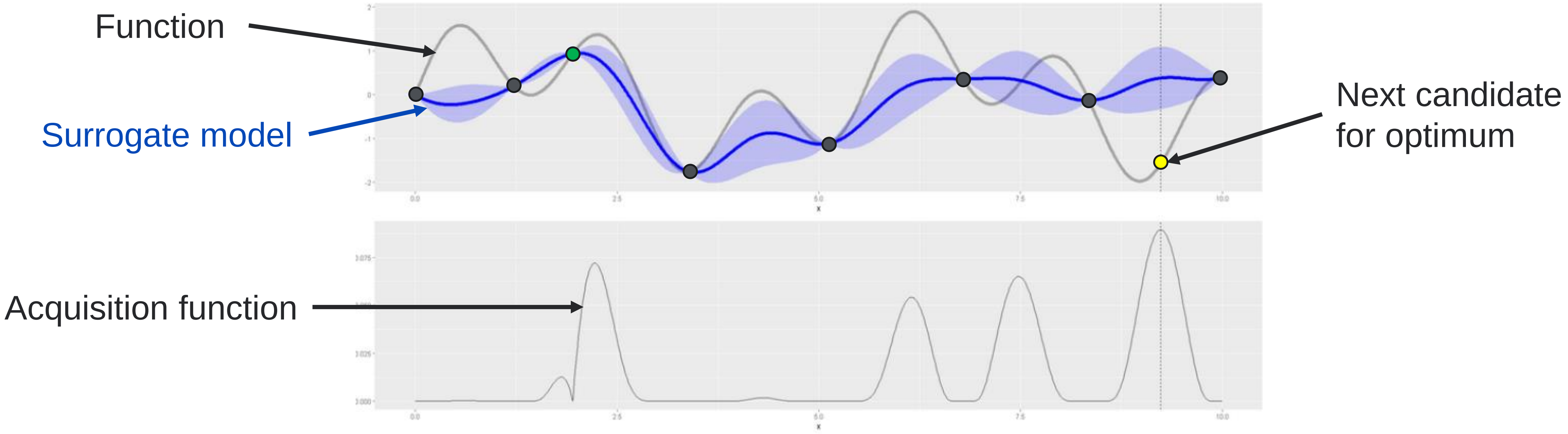


WHY BAYESIAN OPTIMIZATION?

- Model calibration is an optimization problem of a function with some particularities:
  - Knowledge of the logic between the inputs and the outputs
  - Computationally expensive
  - High-dimensional
  - Highly-constrained
  - …
- Traditional methods used (Nelder-Mead, Simulated Annealing, ...) would require a lot of simulations and can take a long time, with no guarantee of finding valid solutions.
- We explored a more efficient approach to calibrate these simulation models by incorporating domain knowledge into the process using Bayesian Optimization (BO).

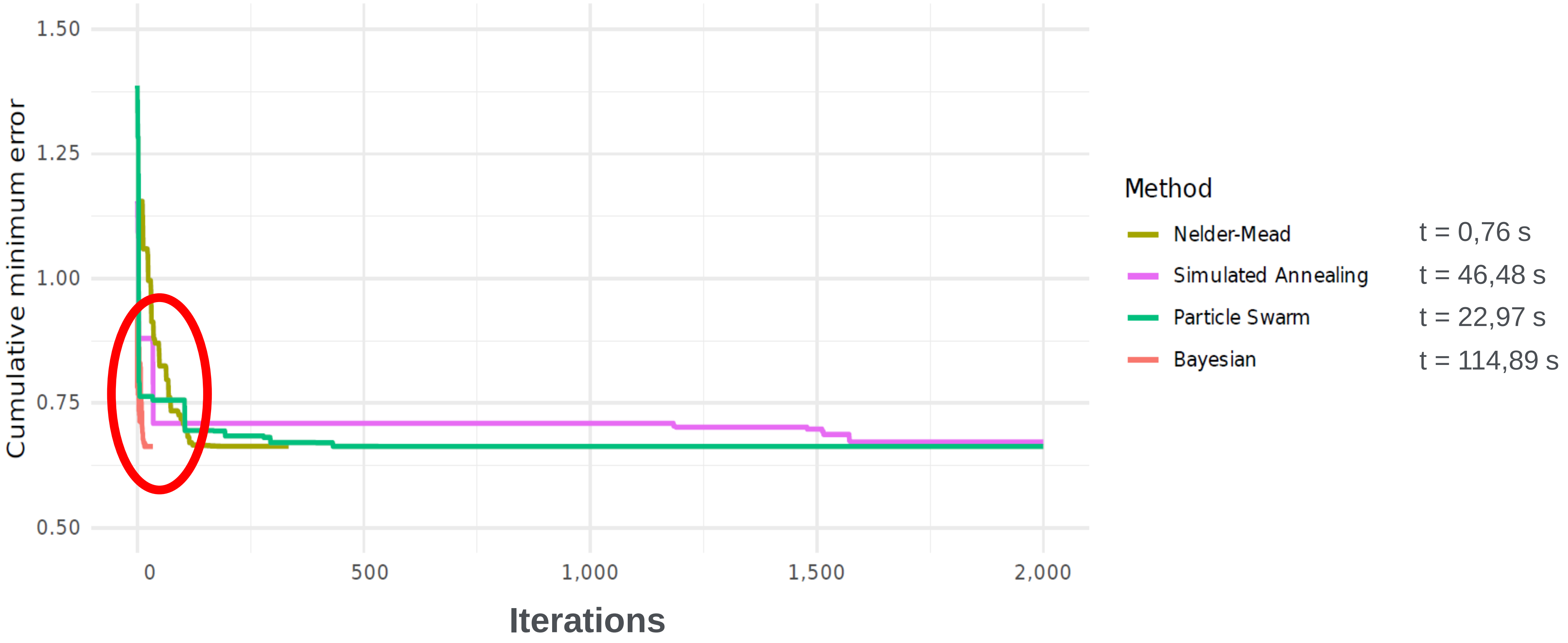
BAYESIAN OPTIMIZATION

- Bayesian optimization (BO) is a state-of-the-art optimization method for expensive functions.
- These functions are represented by **surrogate models**, such as Gaussian Processes (GPs).
- The search for global optima is guided by an **acquisition function**, informed by the surrogate model.



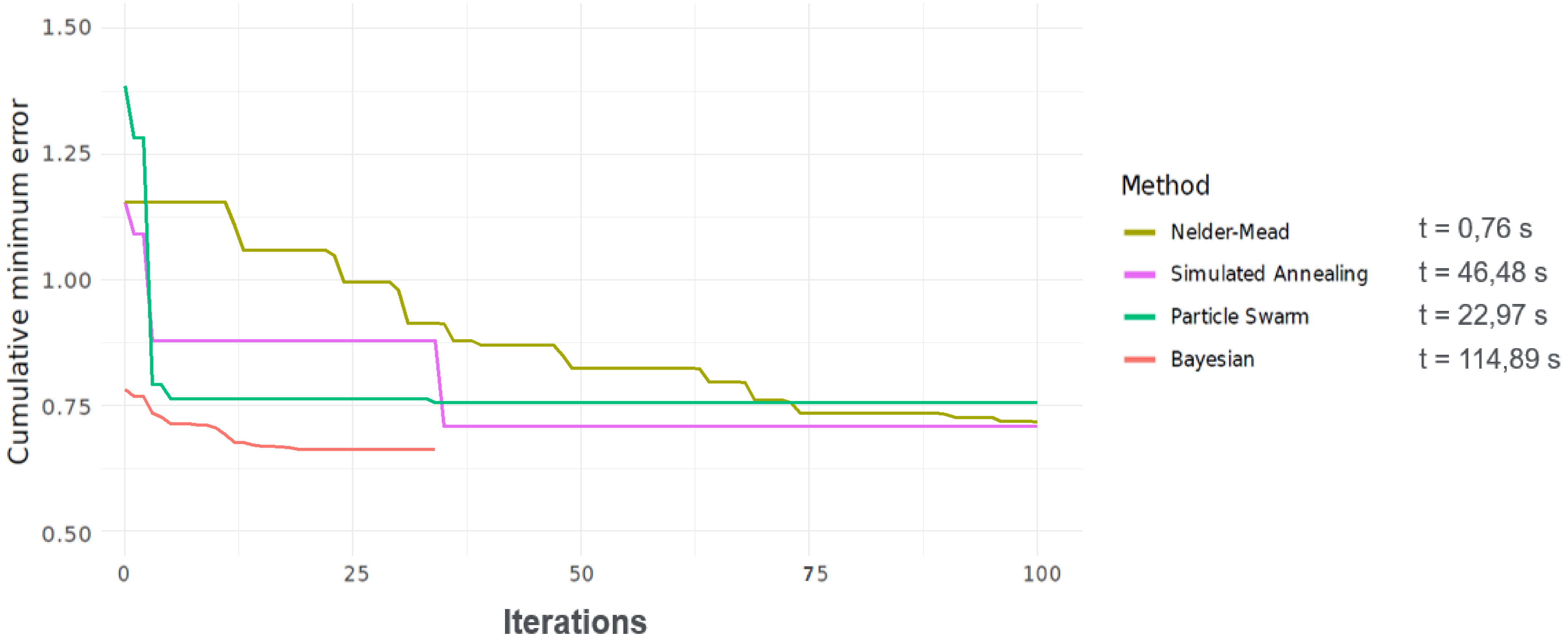
RESULTS

- BO reaches good solutions using much fewer iterations than traditional methods.



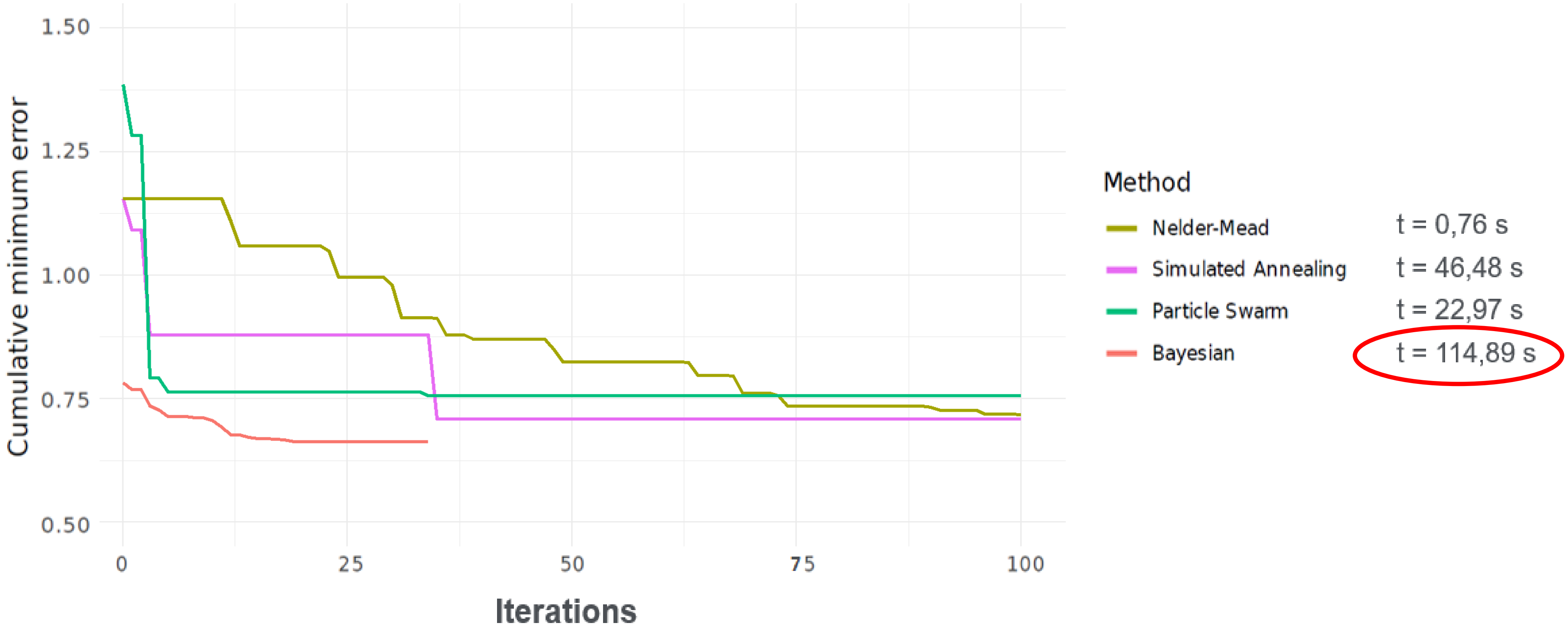
RESULTS

- BO reaches good solutions using much fewer iterations than traditional methods.



RESULTS

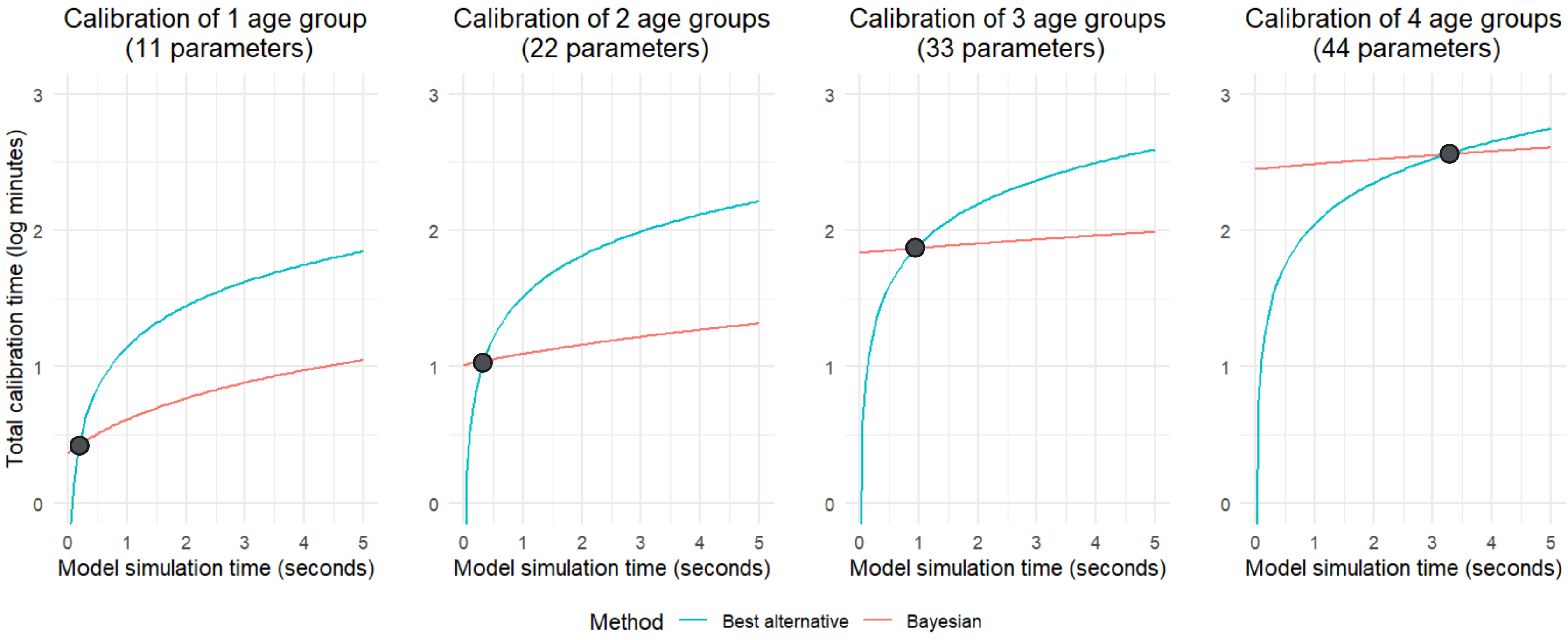
- BO reaches good solutions using much fewer iterations than traditional methods.



RESULTS

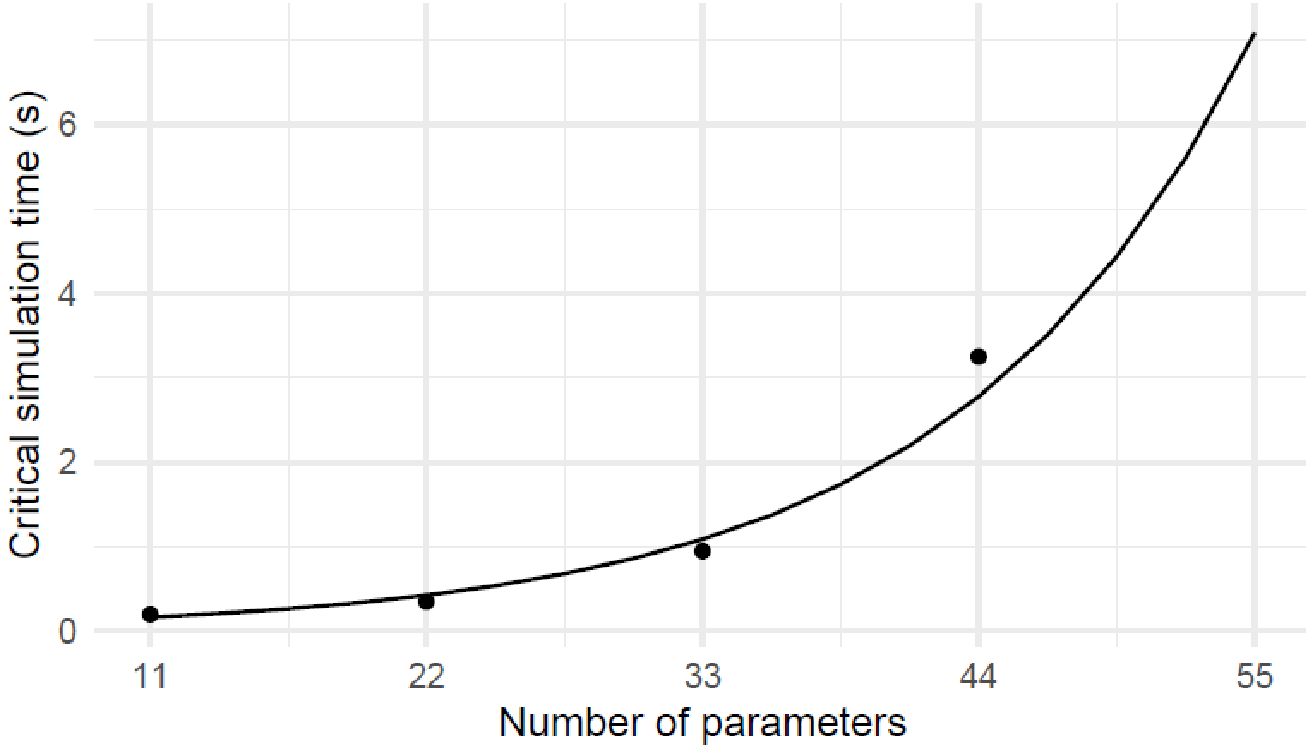
- Problem: Even with fewer iterations, BO takes more time to calibrate then the other methods **unless the simulation model is expensive enough** (critical simulation time: t_c).

- BO calibrates faster than traditional methods even for relatively inexpensive simulation models (t_c ≈1-3 seconds).



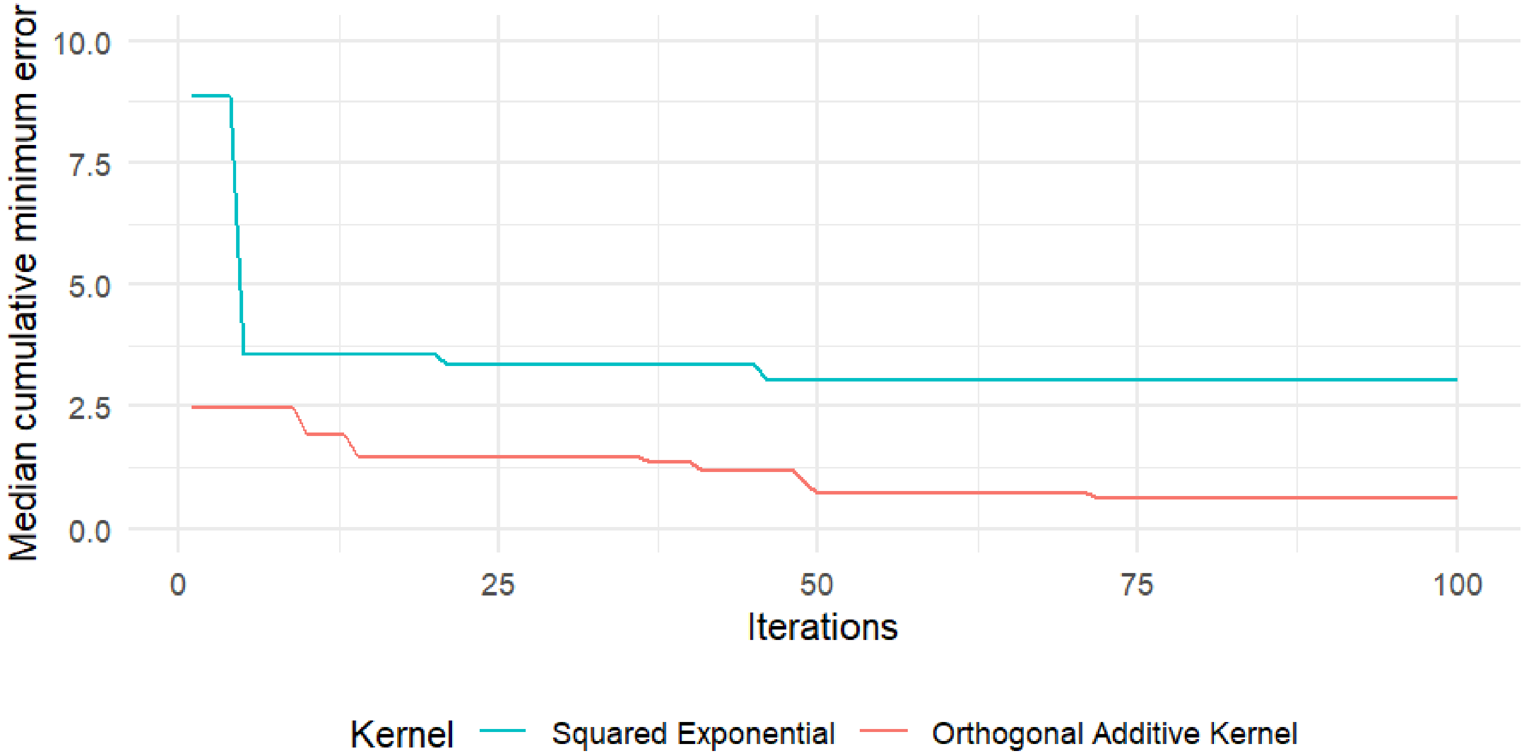
RESULTS

- The critical simulation time scales exponentially with the number of parameters and it can become a problem for high dimensionalities.



RESULTS

- GPs with additive kernels (Duvenaud 2013, Lu 2022) can explore the high-dimensional space of our simulation models more efficiently and **decrease the error in fewer iterations** compared to non-additive kernels.



FUTURE WORK

- Multiple constraint handling
  - Gardner JR et al. (2014). Bayesian optimization with inequality constraints
  - Zhang S. et al. (2023). Dependence in constrained Bayesian optimization
- Improved GP regression techniques
  - Moss et al (2023). Inducing Point Allocation for Sparse Gaussian Processes in High-Throughput Bayesian Optimisation
  - Binois et al (2022). A Survey on High-dimensional Gaussian Process Modeling with Application to Bayesian Optimization
- Improved GP hyperparameter tuning
  - Manzhos et al (2022). On the optimization of hyperparameters in Gaussian process regression with the help of low-order high-dimensional model representation
- Parallelization/Batch processing
  - Wang et al (2019). Parallel Bayesian Global Optimization of Expensive Functions
  - Liu et al (2021). Batch Bayesian optimization via adaptive local search
- …

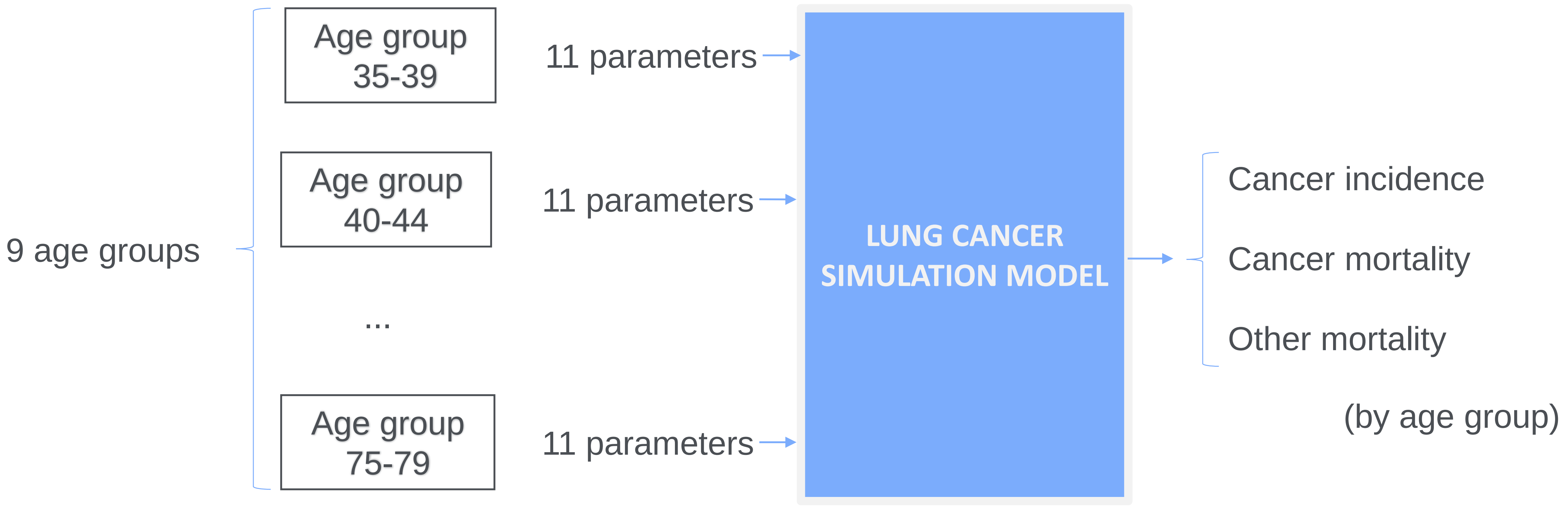
CONCLUSIONS

- BO shows promising performance even for lightweight simulation models.
- BO would be even more efficient for more time-consuming models, but scalability issues may appear in higher dimensionalities.
- BO with additive kernels in a high-dimensional setting can further decrease the error in fewer iterations.
- Bayesian Optimization can help calibrate simulation models for Cost-Effectiveness Analysis more efficiently than traditional methods by exploiting domain knowledge about their structure.

EXTRA SLIDES

METHODOLOGY

- Calibration of lung cancer simulation model using different optimization methods
  - Single cohort, simulating from 35 years old up to 79 years old (earlier for some tests, e.g. 39 years old when simulating one age group).

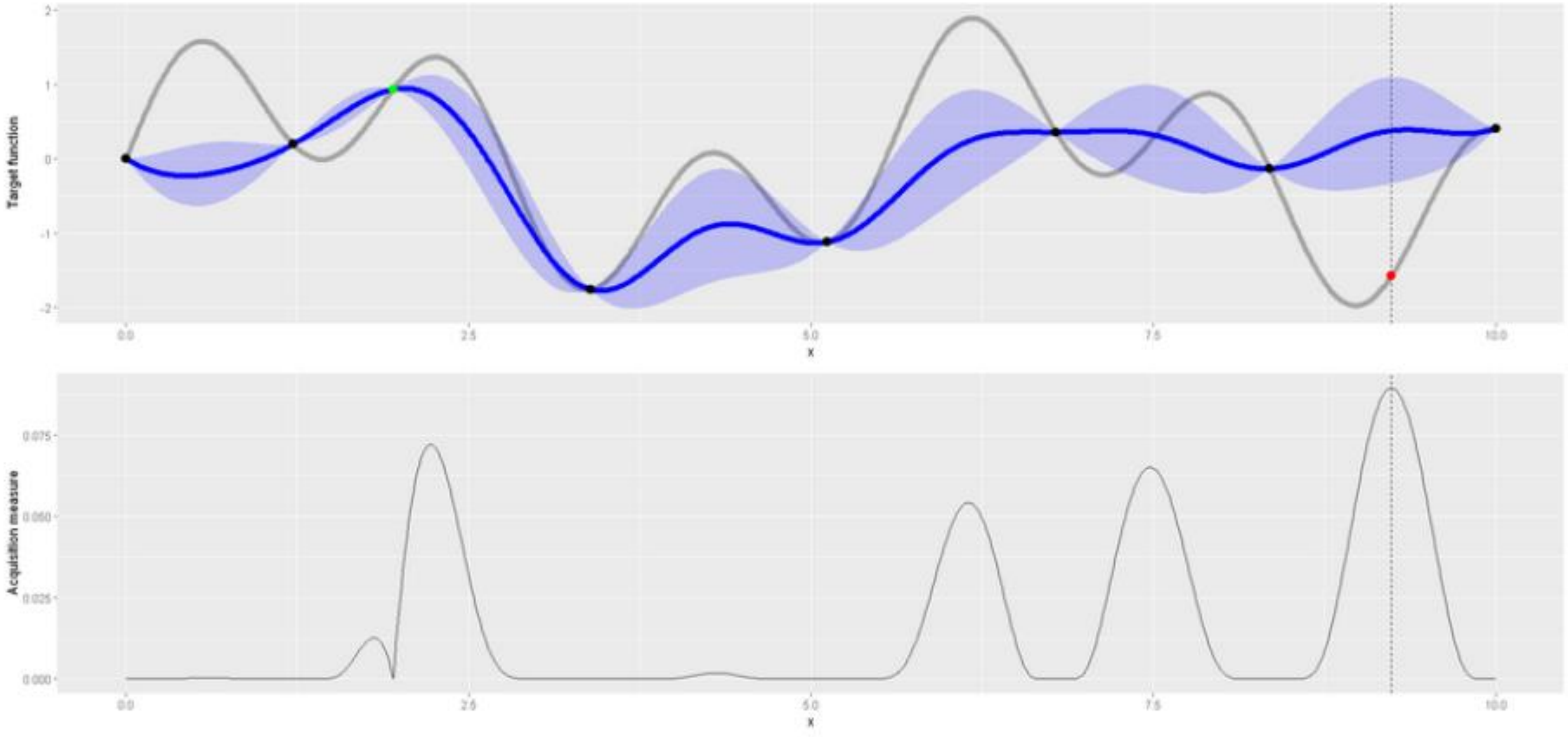


STATE OF THE ART

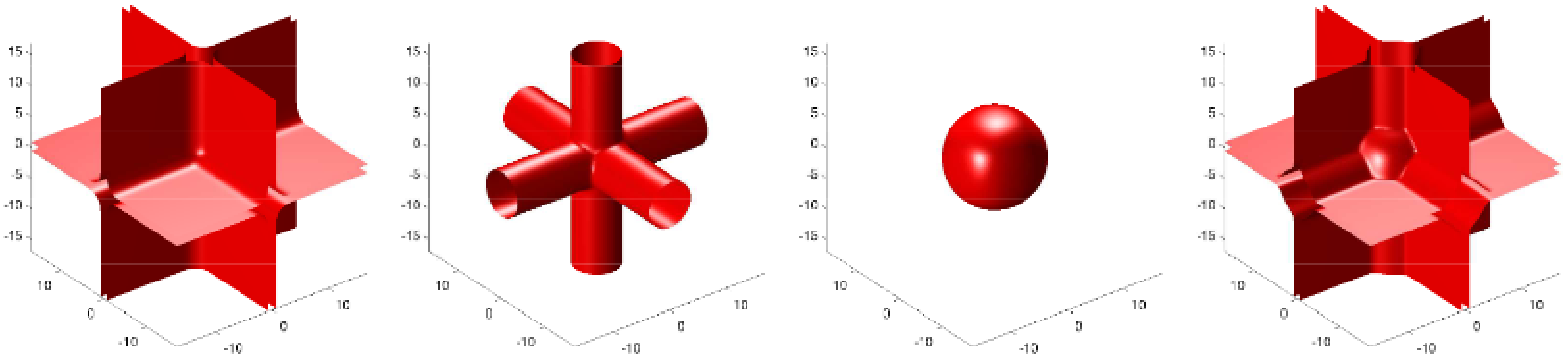
- Bayesian optimization (BO) is a state-of-the-art optimization method for expensive systems such as hyperparameter tuning for machine learning models.
- BO uses a surrogate model to represent the system, such as Gaussian Processes.
- Gaussian Process Regression has scaling issues with the number of observations (n).

$$\bar{f}^* = \mu^* + K(X^*, X)[\underbrace{K(X, X)}_{\text{n x n matrix}} + \sigma_n^2 I]^{-1}(y - \mu)$$

BAYESIAN OPTIMIZATION



SQUARED EXPONENTIAL KERNEL
VS
ADDITIVE KERNEL



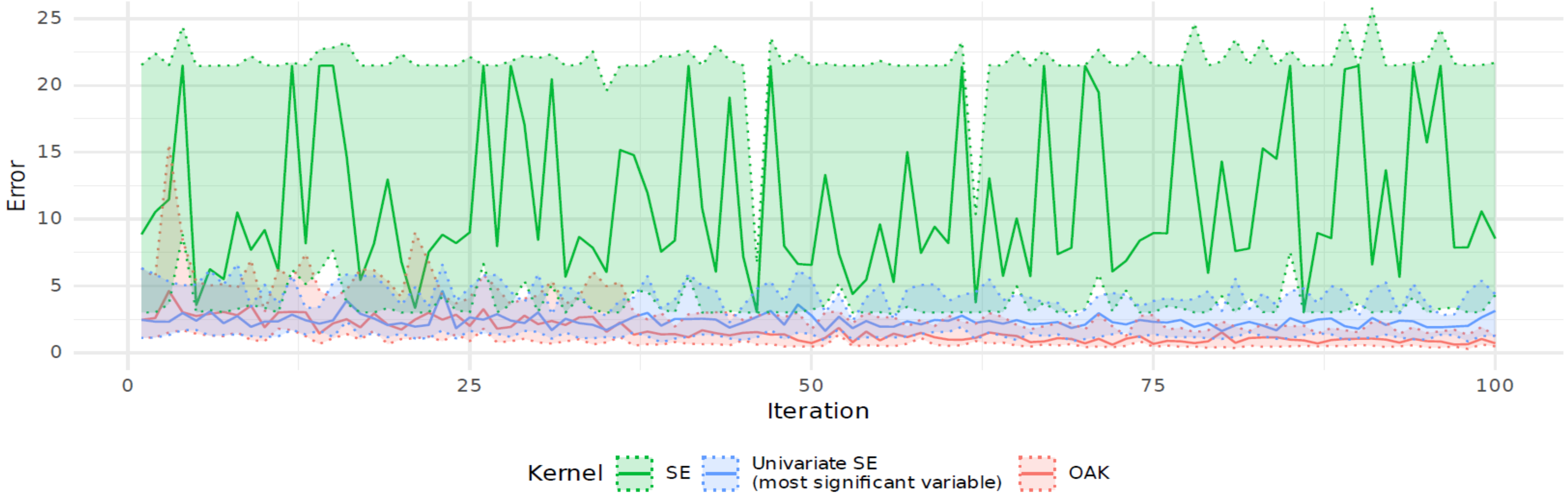
1st order interactions	2nd order interactions	3rd order interactions	All interactions
$k_1 + k_2 + k_3$	$k_1k_2 + k_2k_3 + k_1k_3$	$k_1k_2k_3$	
		(Squared-exp kernel)	(Additive kernel)

Figure 4: Isocontours of additive kernels in 3 dimensions. The third-order kernel only considers nearby points relevant, while the lower-order kernels allow the output to depend on distant points, as long as they share one or more input value.

Duvenaud et al (2013)

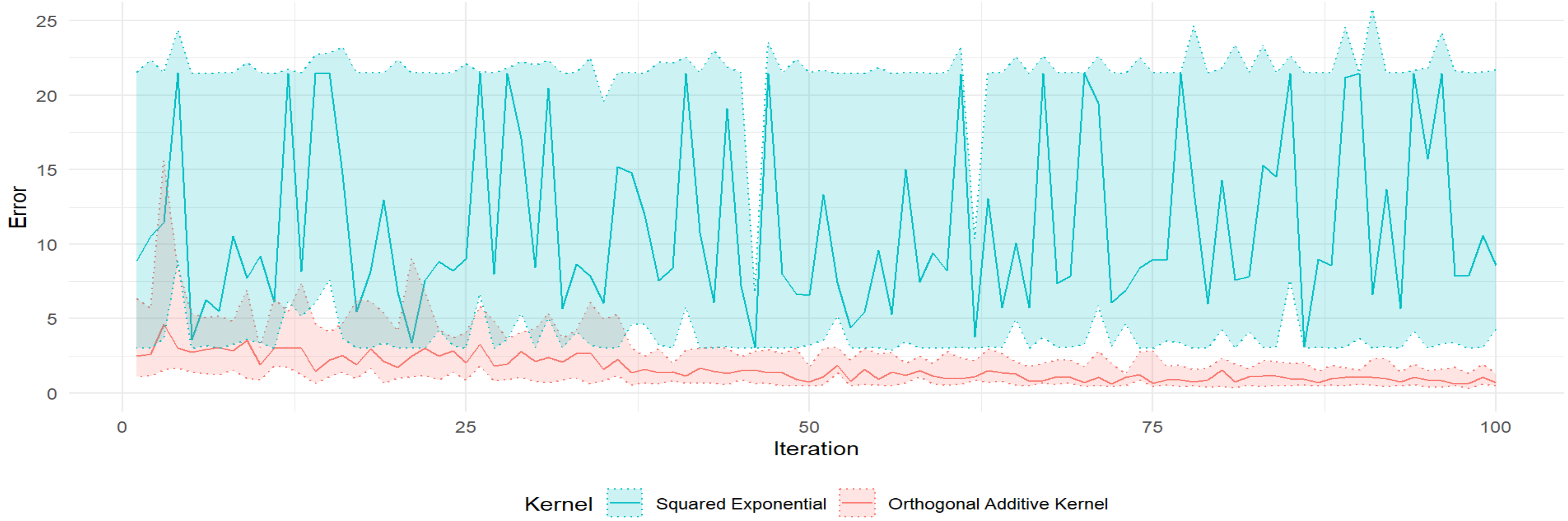
PRELIMINARY RESULTS

- BO with additive kernels improves the solutions in fewer iterations compared to non-additive kernels.



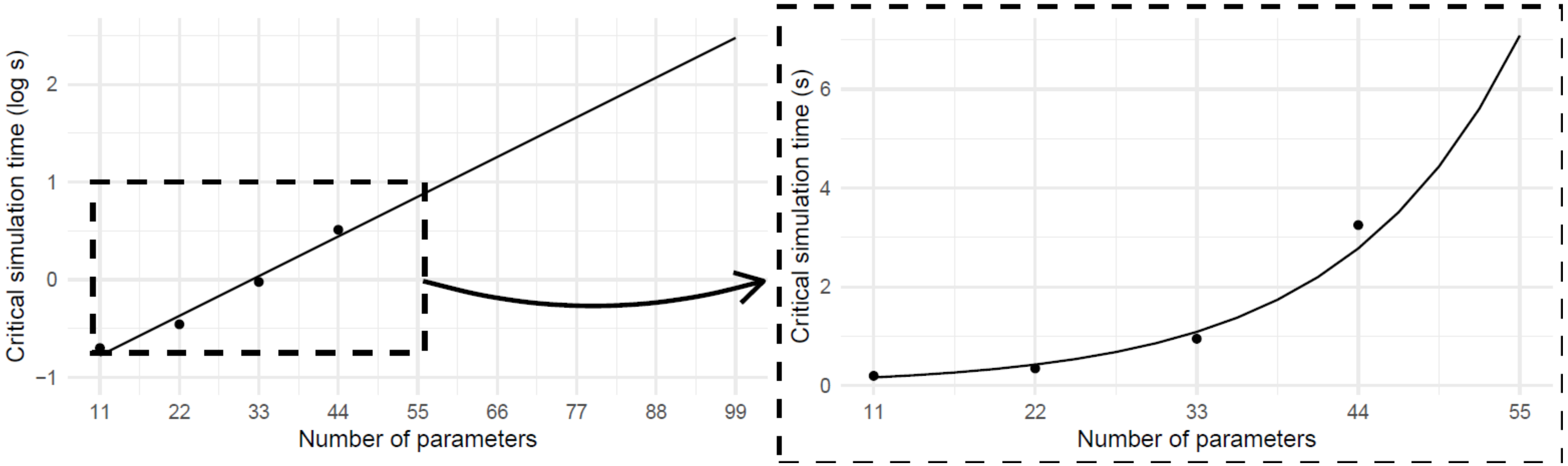
RESULTS

- BO with additive kernels (Duvenaud 2013, Lu 2022) decreases the error in fewer iterations compared to non-additive kernels.

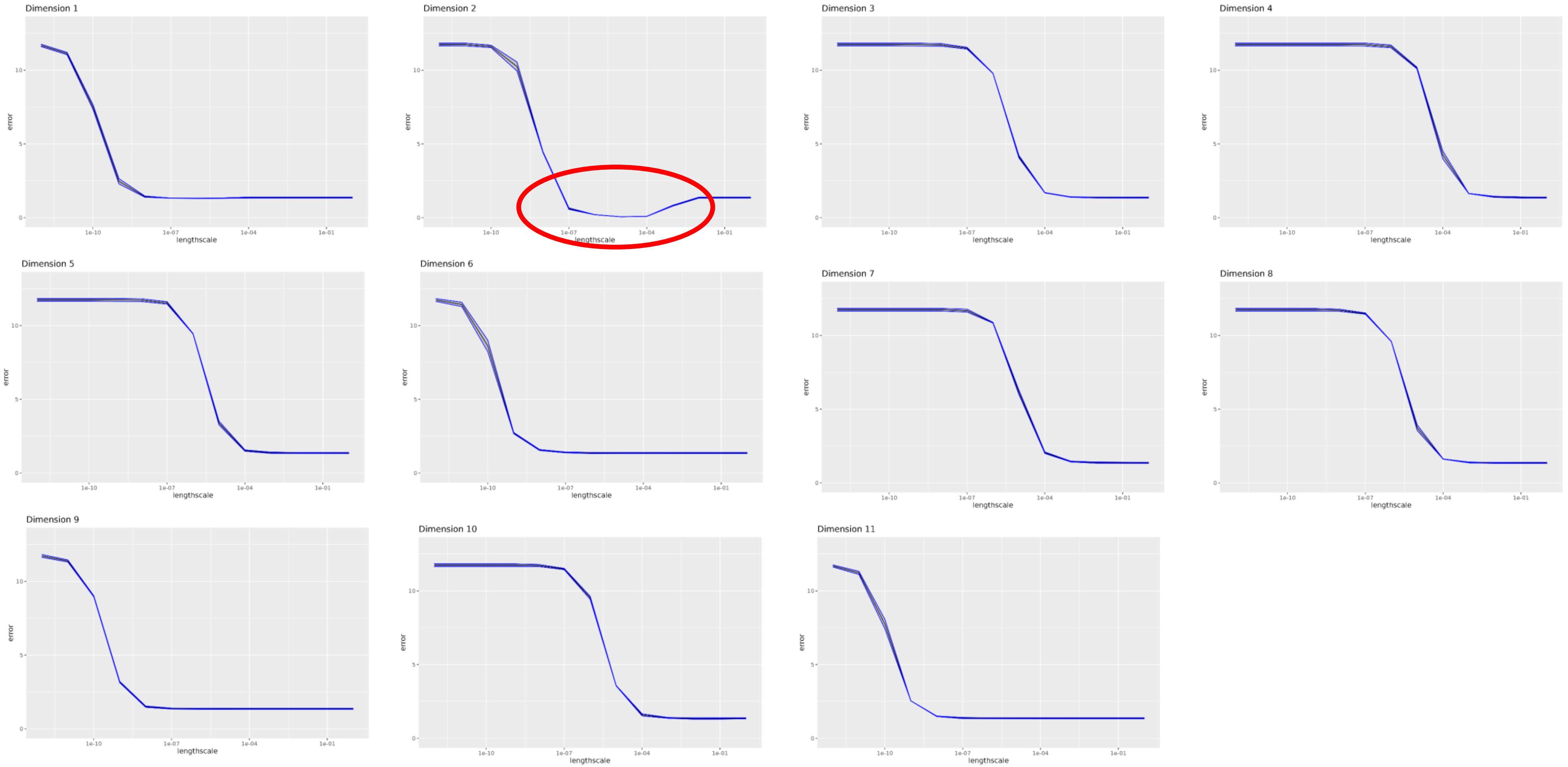


RESULTS

- The minimum simulation time required before BO becomes the fastest optimization method **shows an exponential trend over the number of parameters, with a projected minimum simulation time of 5 minutes for 99 parameters.**



HYPERPARAMETER TUNING (LENGTHSCALE MSE)



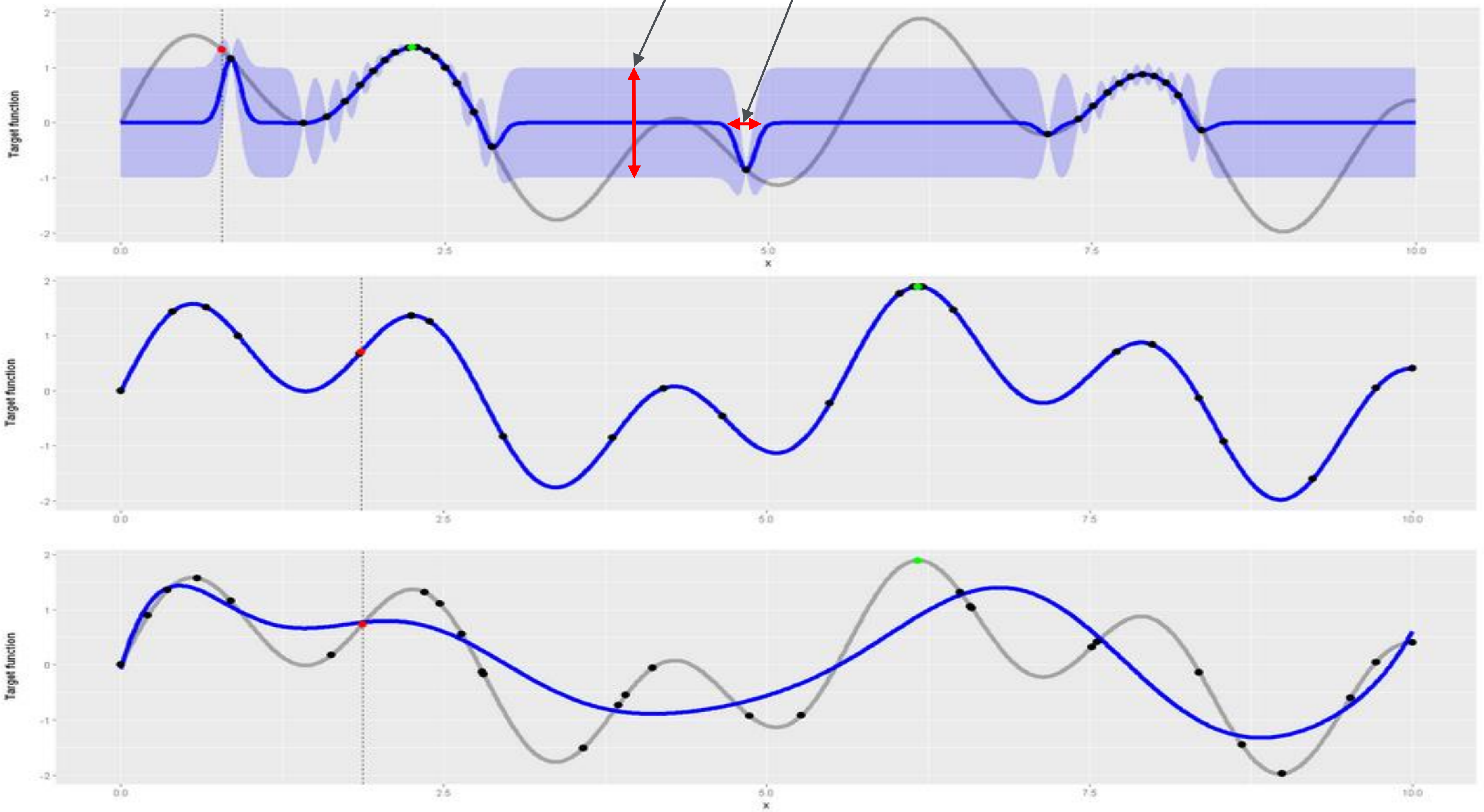
Squared exponential kernel

$$k(\boldsymbol{x},\boldsymbol{x}') = \boldsymbol{\sigma}^{\mathbf{2}} \mathrm{e}^{-\frac{1}{2\boldsymbol{l}^2}\|\boldsymbol{x}-\boldsymbol{x}'\|^2}$$

Too small lengthscale

Appropriate lengthscale

Too large lengthscale



Additive kernel

$$k_{add}(\boldsymbol{x},\boldsymbol{x}') = \sum_{j=1}^{n} \boldsymbol{\sigma_j^2}\, k_{add_j}(\boldsymbol{x},\boldsymbol{x}')$$

One kernel per dimension

$$k_i(x_i, x_i') = e^{-\frac{1}{2\boldsymbol{l_i^2}}(x_i - x_i')^2}$$

Let $k_i \coloneqq k_i(x_i, x_i')$

One additive kernel per order of interaction

$$k_{add_1}(\boldsymbol{x},\boldsymbol{x}') = k_1 + k_2 + \ldots + k_n$$

$$k_{add_2}(\boldsymbol{x},\boldsymbol{x}') = k_1k_2 + k_1k_3 + \cdots + k_1k_n + \\ + k_2k_3 + \ldots + k_2k_n + \\ + \cdots + k_{n-1}k_n$$

$$k_{add_3}(\boldsymbol{x},\boldsymbol{x}') = k_1k_2k_3 + \cdots + \\ + \cdots + \\ + \cdots + k_{n-2}k_{n-1}k_n$$

…

$$k_{add_n}(\boldsymbol{x},\boldsymbol{x}') = k_1k_2 \ldots k_n$$

O(n^2) using Newton-Girard formulae

Example n=4

$$k_1, k_2, k_3, k_4$$

$$k_{add_1}(\boldsymbol{x},\boldsymbol{x}') = k_1 + k_2 + k_3 + k_4$$

$$k_{add_2}(\boldsymbol{x},\boldsymbol{x}') = k_1k_2 + k_1k_3 + k_1k_4 + \\ + k_2k_3 + k_2k_4 + k_3k_4$$

$$k_{add_3}(\boldsymbol{x},\boldsymbol{x}') = k_1k_2k_3 + k_1k_2k_4 + \\ + k_1k_3k_4 + k_2k_3k_4$$

$$k_{add_4}(\boldsymbol{x},\boldsymbol{x}') = k_1k_2k_3k_4$$

$$k_{add}(\boldsymbol{x},\boldsymbol{x}') = \boldsymbol{\sigma_1^2}\, k_{add_1}(\boldsymbol{x},\boldsymbol{x}') + \\ + \boldsymbol{\sigma_2^2}\, k_{add_2}(\boldsymbol{x},\boldsymbol{x}') + \\ + \boldsymbol{\sigma_3^2}\, k_{add_3}(\boldsymbol{x},\boldsymbol{x}') + \\ + \boldsymbol{\sigma_4^2}\, k_{add_4}(\boldsymbol{x},\boldsymbol{x}')$$

Orthogonal additive kernel

$$k_{OAK}(\boldsymbol{x}, \boldsymbol{x}') = \sum_{j=0}^{n} \boldsymbol{\sigma_j^2}\, k_{add_j}(\boldsymbol{x}, \boldsymbol{x}')$$

One kernel per dimension

Assuming each $x_i \sim N(\mu_i, \delta_i^2)$

For $x_i \sim Beta(\alpha, \beta)$?
For $x_i \sim U(a, b)$?

$$k_i(x_i, x_i') = e^{-\frac{1}{2l_i^2}(x_i - x_i')^2} - \frac{l\sqrt{l_i^2 + 2\delta_i^2}}{l_i^2 + \delta_i^2} e^{-\frac{(x_i - \mu_i)^2 + (x_i' - \mu_i)^2}{2(l_i^2 + \delta_i^2)}}$$

Let $k_i \coloneqq k_i(x_i, x_i')$

One additive kernel per order of interaction + 1 (constant)

$$\begin{cases} k_{add_0}(\boldsymbol{x}, \boldsymbol{x}') = 1 \\ k_{add_1}(\boldsymbol{x}, \boldsymbol{x}') = k_1 + k_2 + \ldots + k_n \\ k_{add_2}(\boldsymbol{x}, \boldsymbol{x}') = k_1 k_2 + k_1 k_3 + \cdots + k_1 k_n + \\ \quad + k_2 k_3 + \ldots + k_2 k_n + \\ \quad + \cdots + k_{n-1} k_n \\ k_{add_3}(\boldsymbol{x}, \boldsymbol{x}') = k_1 k_2 k_3 + \cdots + \\ \quad + \cdots + \\ \quad + \cdots + k_{n-2} k_{n-1} k_n \\ \ldots \\ k_{add_n}(\boldsymbol{x}, \boldsymbol{x}') = k_1 k_2 \ldots k_n \end{cases}$$

O(n^2) using Newton-Girard formulae

Example n=4

k_1, k_2, k_3, k_4

$$k_{add_0}(\boldsymbol{x}, \boldsymbol{x}') = 1$$

$$k_{add_1}(\boldsymbol{x}, \boldsymbol{x}') = k_1 + k_2 + k_3 + k_4$$

$$k_{add_2}(\boldsymbol{x}, \boldsymbol{x}') = k_1 k_2 + k_1 k_3 + k_1 k_4 + \\ + k_2 k_3 + k_2 k_4 + k_3 k_4$$

$$k_{add_3}(\boldsymbol{x}, \boldsymbol{x}') = k_1 k_2 k_3 + k_1 k_2 k_4 + \\ + k_1 k_3 k_4 + k_2 k_3 k_4$$

$$k_{add_4}(\boldsymbol{x}, \boldsymbol{x}') = k_1 k_2 k_3 k_4$$

$$k_{OAK}(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\sigma_0^2}\, k_{add_0}(\boldsymbol{x}, \boldsymbol{x}') + \\ + \boldsymbol{\sigma_1^2}\, k_{add_1}(\boldsymbol{x}, \boldsymbol{x}') + \\ + \boldsymbol{\sigma_2^2}\, k_{add_2}(\boldsymbol{x}, \boldsymbol{x}') + \\ + \boldsymbol{\sigma_3^2}\, k_{add_3}(\boldsymbol{x}, \boldsymbol{x}') + \\ + \boldsymbol{\sigma_4^2}\, k_{add_4}(\boldsymbol{x}, \boldsymbol{x}')$$