# Multi-fidelity Bayesian optimization structure search

An open-source Python library for accelerated optimization in computational materials science

Aalto University
School of Science

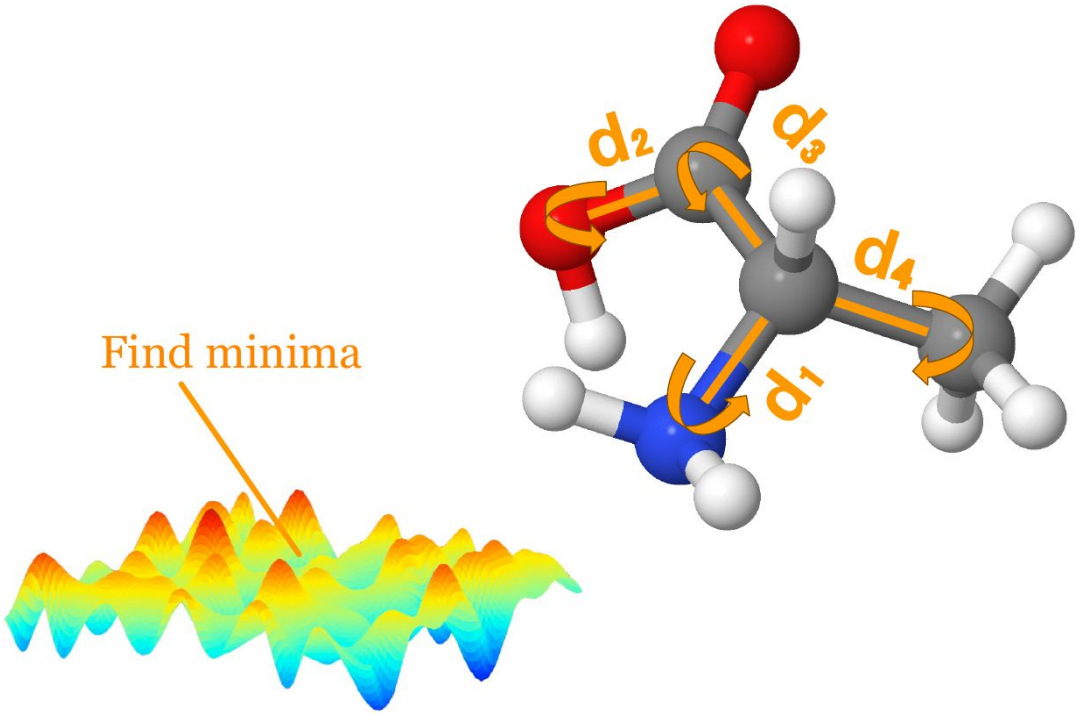
University of Stuttgart
Germany

# Motivation

**Goal**: Find stable configurations on materials' energy landscape with Bayesian optimization

**Challenge:** Quantum mechanical calculations to construct energy landscape are costly

**Alanine conformer search**



# Contents

**Theory and Methodology**
- Bayesian optimization structure search (BOSS)
- Multi-fidelity BOSS

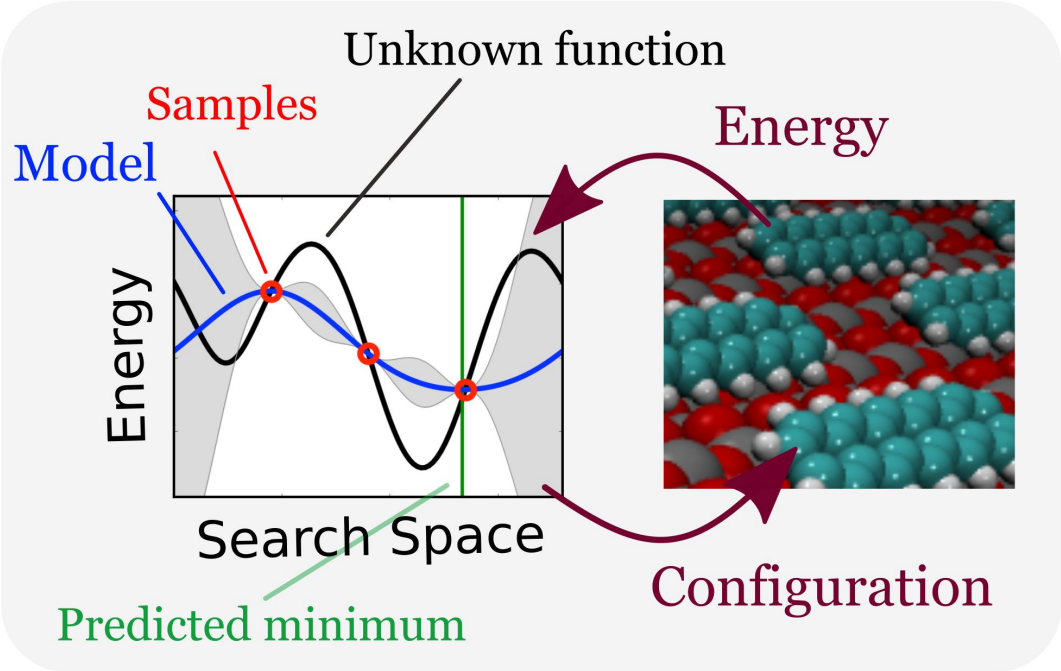
**Computational Implementation**
- Extension of multi-fidelity BOSS
- Code validation

**Experiments and Results**
- Test case: Alanine conformer search
- Transfer learning
- Multi-task learning strategies
- Discussion and Conclusion

# Theory and Methodology

# Bayesian optimization structure search

- BOSS* applies sample-efficient active learning for global minimum inference based on Gaussian process regression
- Returns a probabilistic model of the potential energy surface



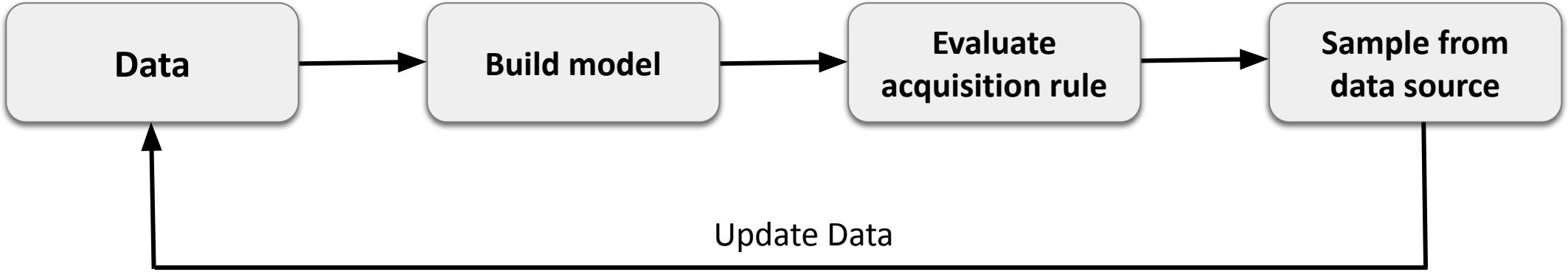

GitLab

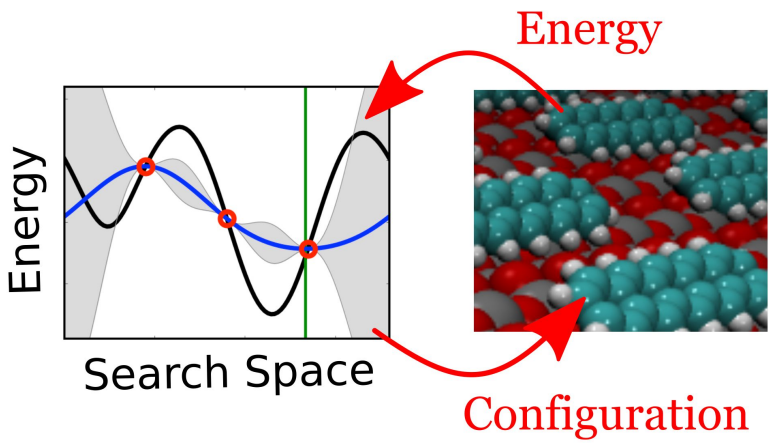
python Package Index



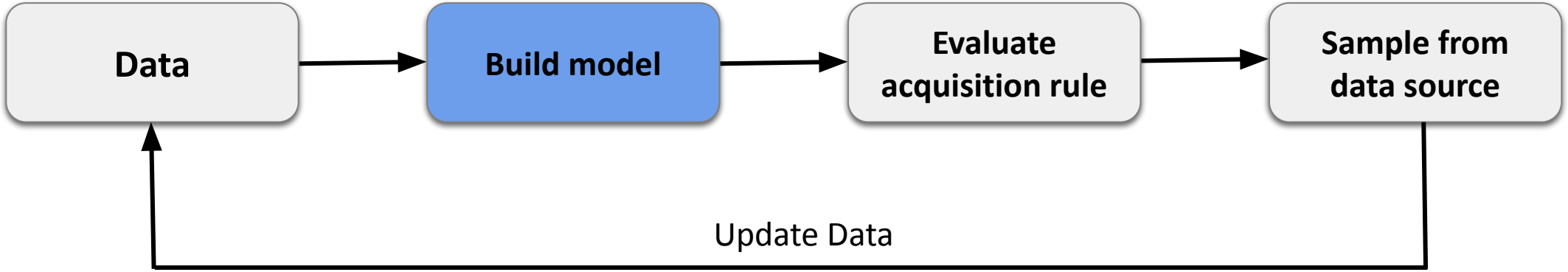
*M. Todorović, M.U. Gutmann, J. Corander and P. Rinke,
*Bayesian inference of atomistic structure in functional materials* (2019)

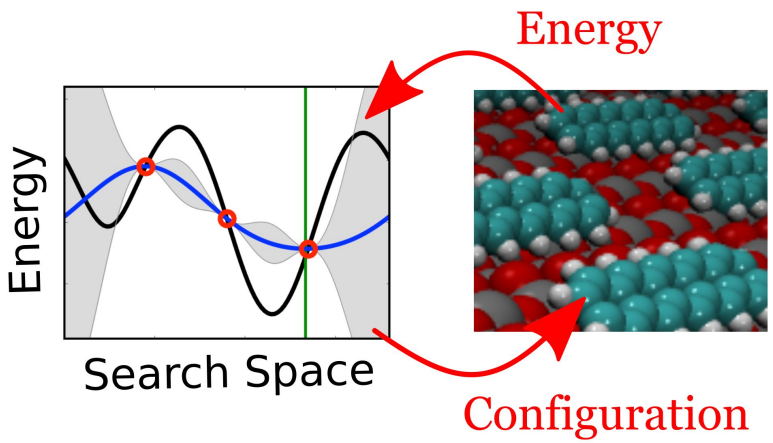
# Bayesian optimization structure search



# Bayesian optimization structure search



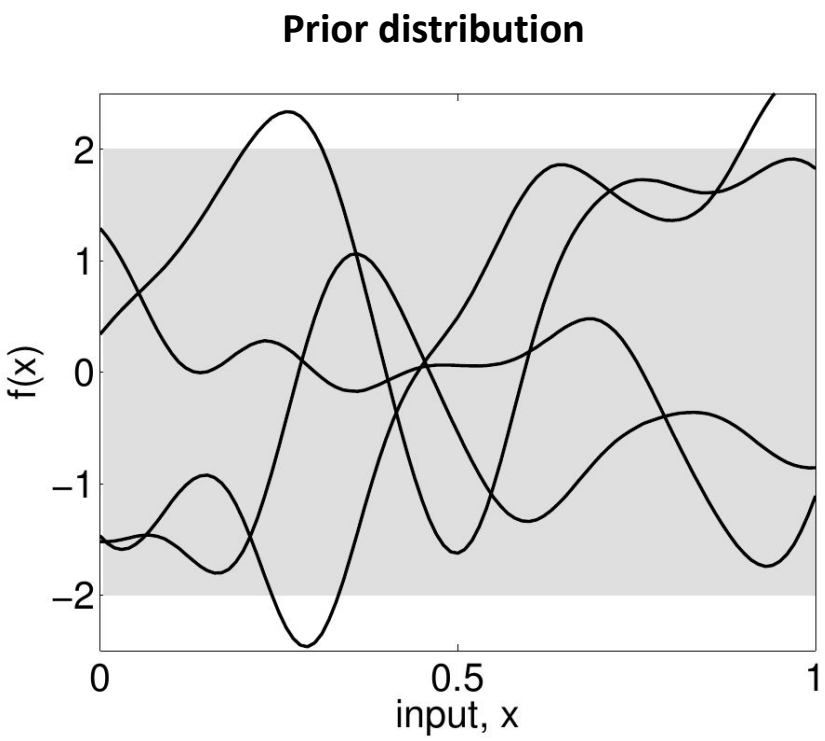



# Probabilistic modelling with Gaussian process regression

**Gaussian process**: Distribution over functions defined by mean $\mu(x)$ and covariance function $k(x, x')$

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

$$k_{\mathrm{RBF}}(x, x') = \sigma^2 e^{-\frac{(x-x')^2}{2L^2}}$$




C.E. Rasmussen et al.,
*Gaussian Processes for Machine Learning* (2008)

# Probabilistic modelling with Gaussian process regression

**Gaussian process**: Distribution over functions defined by mean $\mu(x)$ and covariance function $k(x, x')$

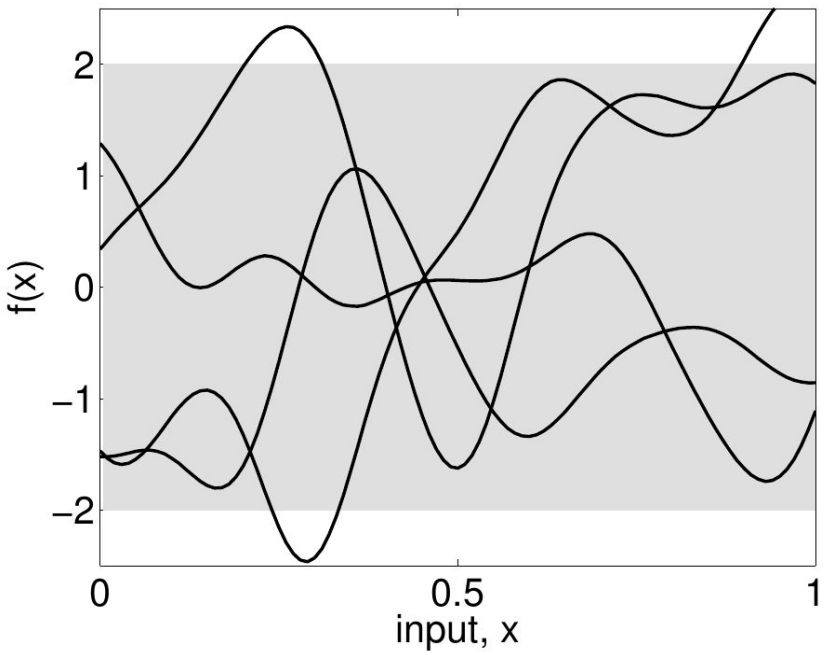
$$f(x) \sim \mathcal{GP}(\mu(x), k(x, x'))$$

$$k_{\mathrm{RBF}}(x, x') = \sigma^2 e^{-\frac{(x-x')^2}{2L^2}}$$

Sample function over finite domain $X$:

$$f(X) \sim \mathcal{N}(\mu(X), K(X, X))$$

**Prior distribution**




C.E. Rasmussen et al.,
*Gaussian Processes for Machine Learning* (2008)

# Probabilistic modelling with Gaussian process regression

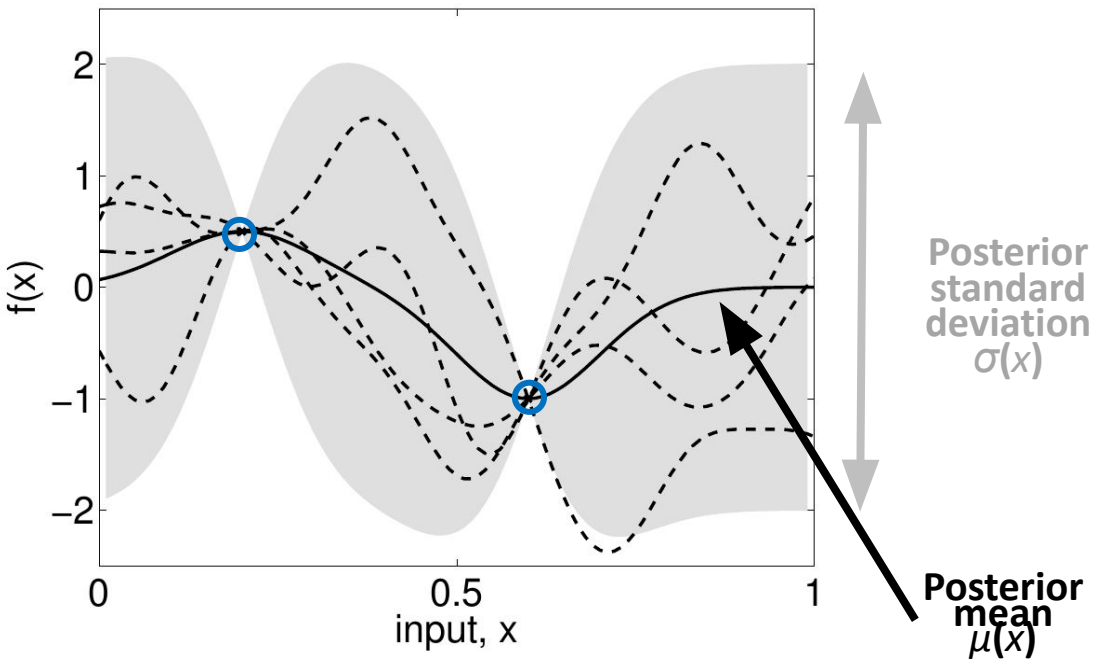
Observed data:

$$X = [x_1, \ldots, x_N]^T \quad \boldsymbol{y} = [y_1, \ldots, y_N]^T$$

Predict at $x_*$, condition the distribution on observed samples $\{X, \boldsymbol{y}\}$:

$$y_* \sim \mathcal{N}(\mu(x_*|X, \boldsymbol{y}), \sigma^2(x_*|X, \boldsymbol{y}))$$

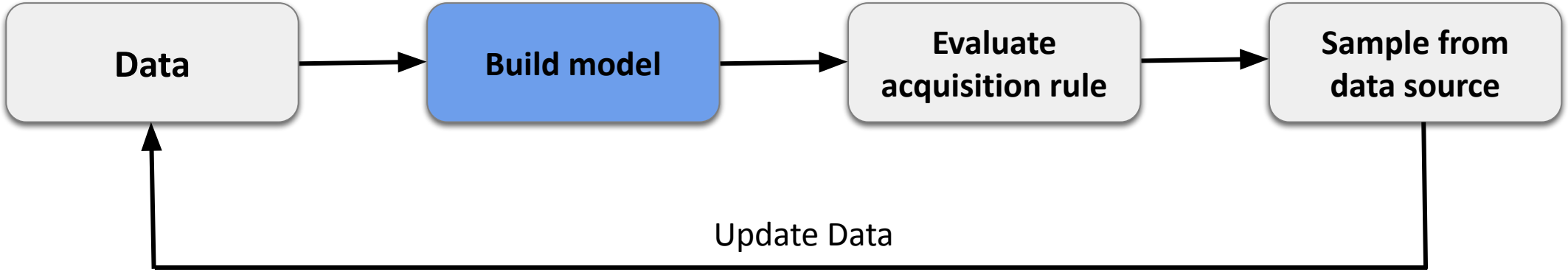
**Posterior distribution**

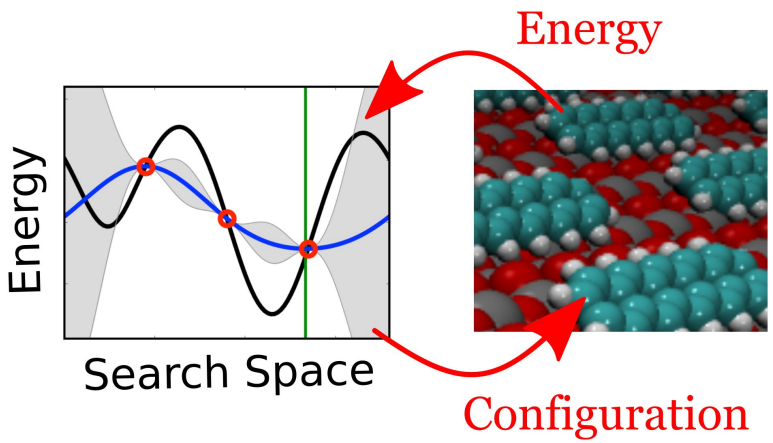




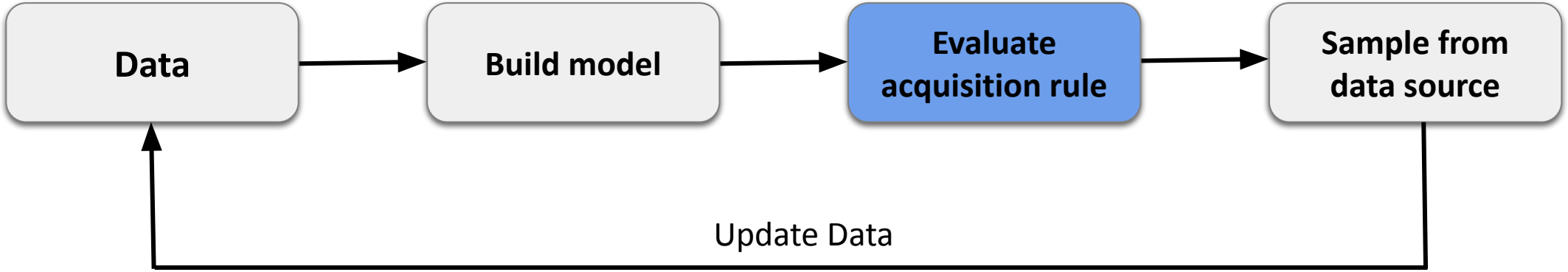

C.E. Rasmussen et al.,
*Gaussian Processes for Machine Learning* (2008)

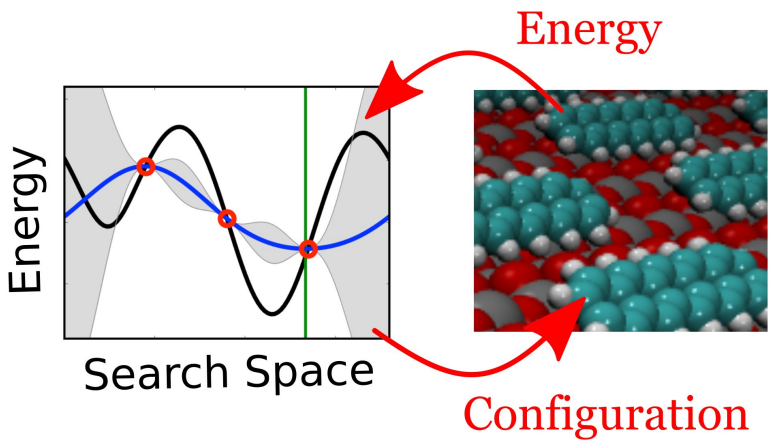
# Bayesian optimization structure search



# Bayesian optimization structure search



# Acquisition rule

- In BO: Minimize acquisition function $\alpha(x)$ to obtain next sample location
- $\alpha(x)$ uses model mean $\mu(x)$ and variance function $\sigma(x)$

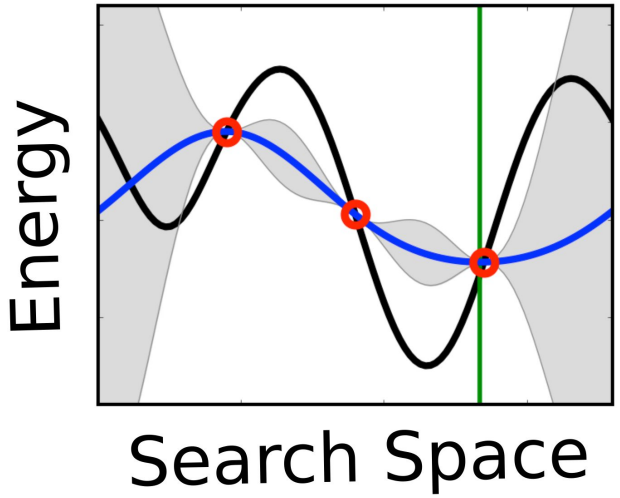
# Acquisition rule

- In BO: Minimize acquisition function $\alpha(x)$ to obtain next sample location
- $\alpha(x)$ uses model mean $\mu(x)$ and variance function $\sigma(x)$
- Many heuristic approaches, e.g. (Exploratory) lower confidence bound (ELCB):

$$\alpha_{\mathrm{LCB}}(x) = \mu(x) - \kappa\sigma(x)$$

Exploitation term

Exploration term






N. Srinivas et al.,
*Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design* (2010)

# Entropy-search acquisition function

- Treat the next sample $\{x, y\}$ and global minimum location $x_*$ as random variables
- Next sample should reduce uncertainty about global minimum location as much as possible
- Mutual information $I(X, Y)$ : Information obtained about a random variable by observing another random variable



P. Hennig and C.J. Schuler, *Entropy Search for Information-Efficient Global Optimization* (2011)
Z. Wang and S. Jegelka, *Max-value Entropy Search for Efficient Bayesian Optimization* (2017)

# Entropy-search acquisition function

- Treat the next sample $\{x, y\}$ and global minimum location $x_*$ as random variables
- Next sample should reduce uncertainty about global minimum location as much as possible
- Mutual information $I(X, Y)$ : Information obtained about a random variable by observing another random variable

**Entropy Search**

$$\begin{aligned}\alpha_{\mathrm{ES}}(x) &= I(\{x, y\}; x_* | \mathcal{D}) \\ &= \mathbb{H}(p(x_* | \mathcal{D})) - \mathbb{E}[\mathbb{H}(p(x_* | \{x, y\} \cup \mathcal{D}))]\end{aligned}$$



P. Hennig and C.J. Schuler, *Entropy Search for Information-Efficient Global Optimization* (2011)
Z. Wang and S. Jegelka, *Max-value Entropy Search for Efficient Bayesian Optimization* (2017)

# Entropy-search acquisition function

- Treat the next sample $\{x, y\}$ and global minimum location $x_*$ as random variables
- Next sample should reduce uncertainty about global minimum location as much as possible
- Mutual information $I(X, Y)$ : Information obtained about a random variable by observing another random variable

**Entropy Search**

$$\begin{aligned}\alpha_{\mathrm{ES}}(x) &= I(\{x, y\}; x_* | \mathcal{D}) \\ &= \mathbb{H}(p(x_* | \mathcal{D})) - \mathbb{E}[\mathbb{H}(p(x_* | \{x, y\} \cup \mathcal{D}))]\end{aligned}$$

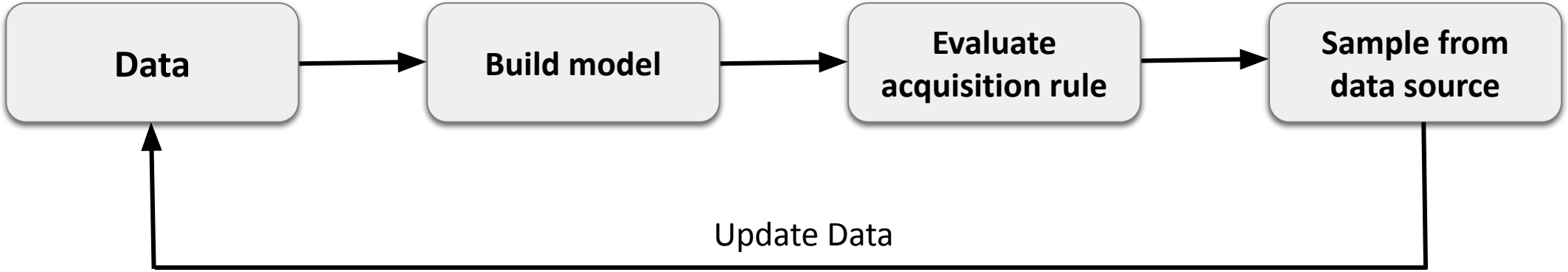
**Max-value Entropy Search (MES)**

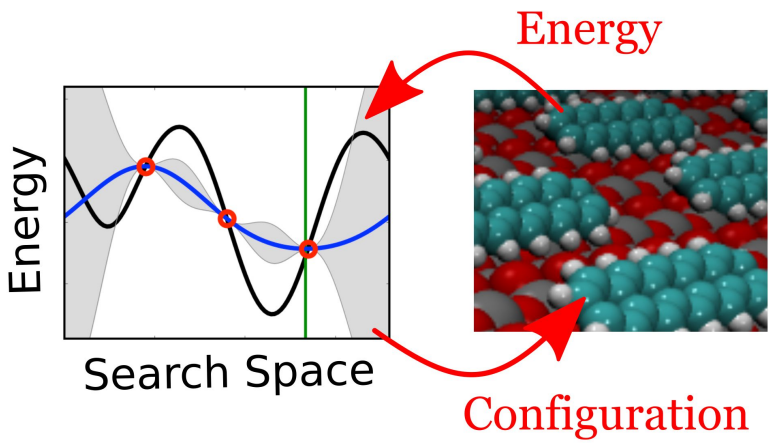
$$\alpha_{\mathrm{MES}}(x) = \mathbb{H}(p(y | x, \mathcal{D})) - \mathbb{E}[\mathbb{H}(p(y | x, \mathcal{D}, y_*))]$$



P. Hennig and C.J. Schuler, *Entropy Search for Information-Efficient Global Optimization* (2011)
Z. Wang and S. Jegelka, *Max-value Entropy Search for Efficient Bayesian Optimization* (2017)

# Bayesian optimization structure search



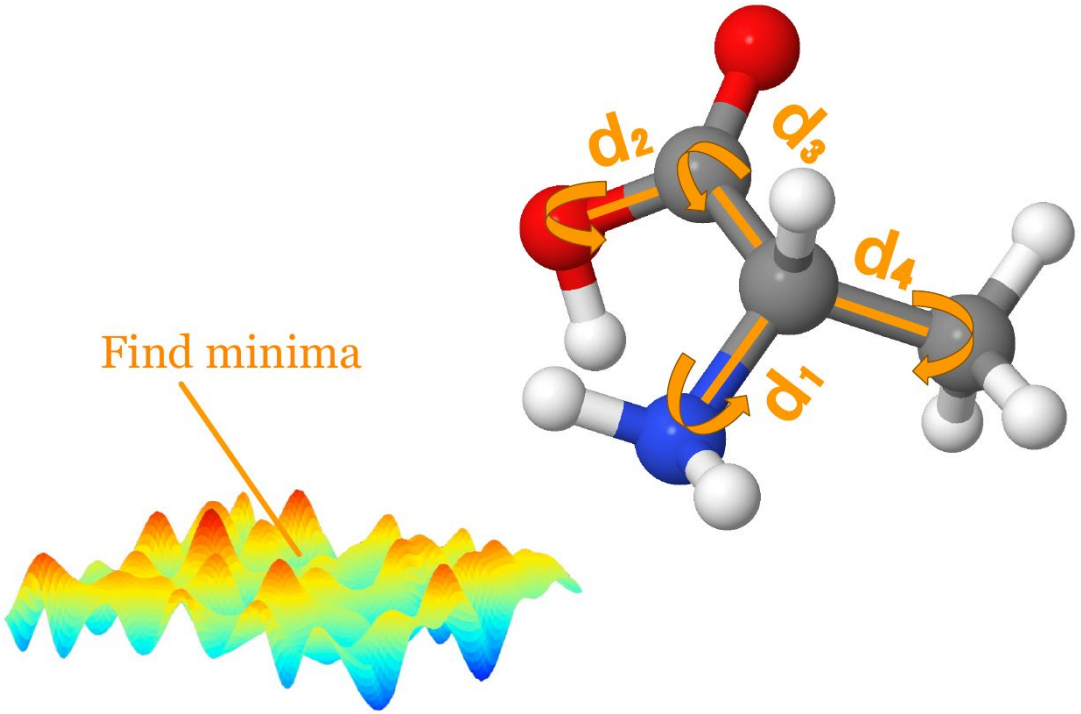
# Objective

**Goal**: Find stable configurations on materials' energy landscape with Bayesian optimization

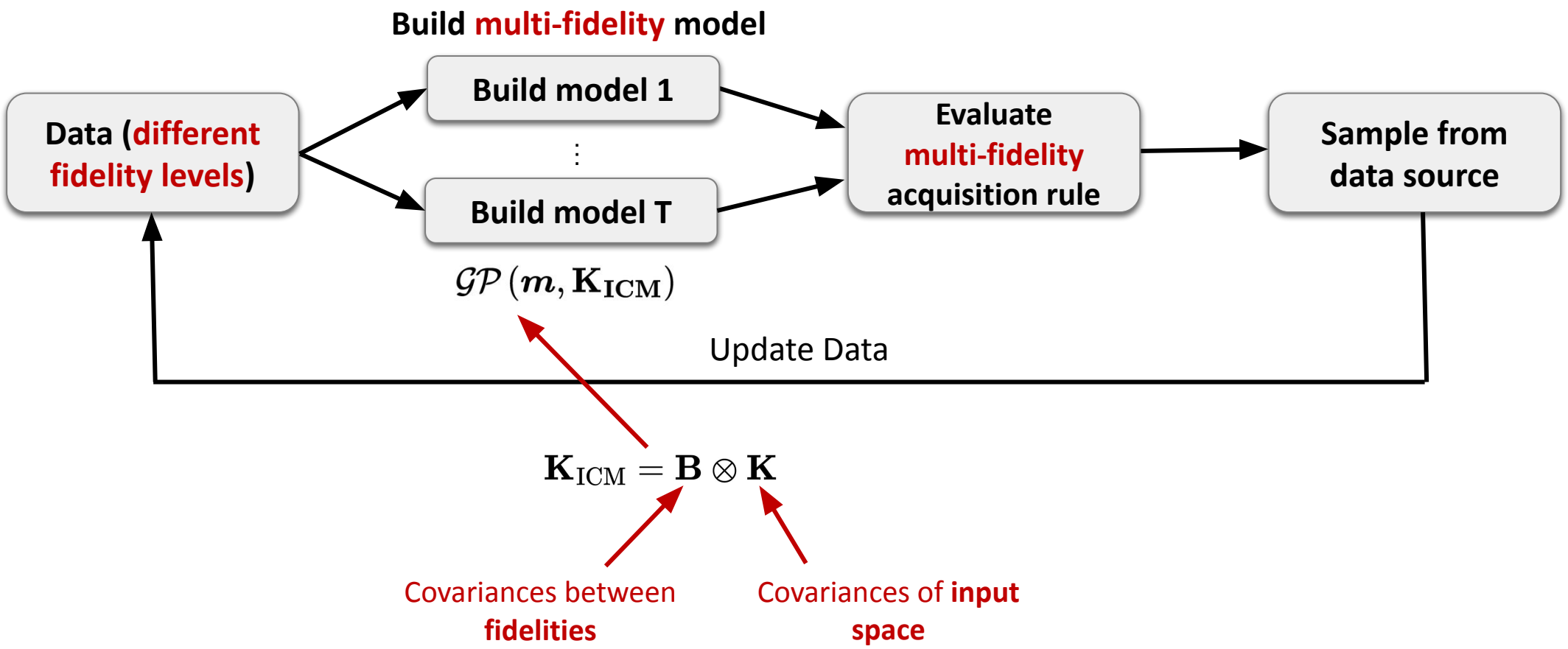
**Challenge:** Quantum mechanical calculations to construct energy landscape are costly

**Approach:** Develop multi-fidelity BO to incorporate information from *inexpensive*, *less accurate* calculations to accelerate optimization

**Alanine conformer search**



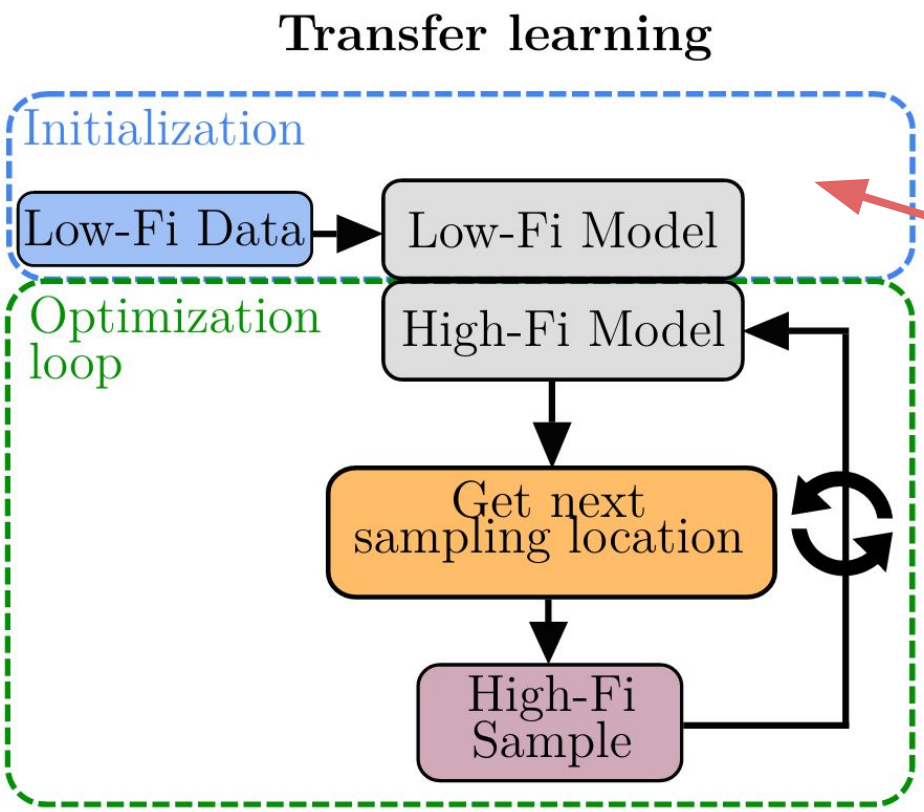
# Multi-fidelity Bayesian optimization structure search





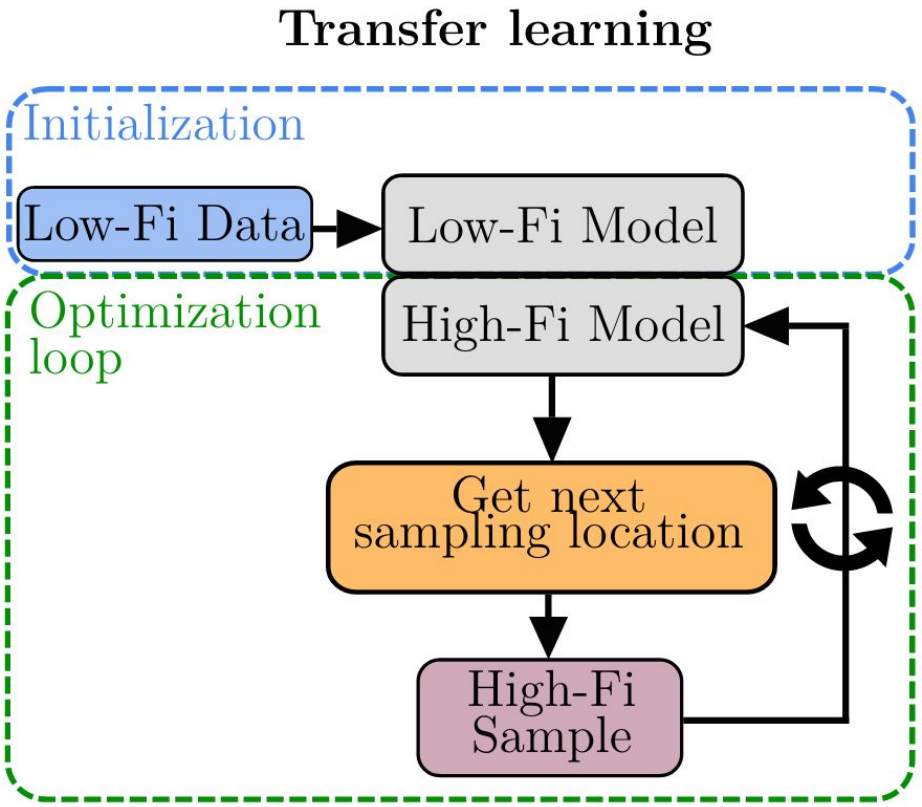

M. A. Alvarez et al., *Kernels for Vector-Valued Functions: a Review* (2011)

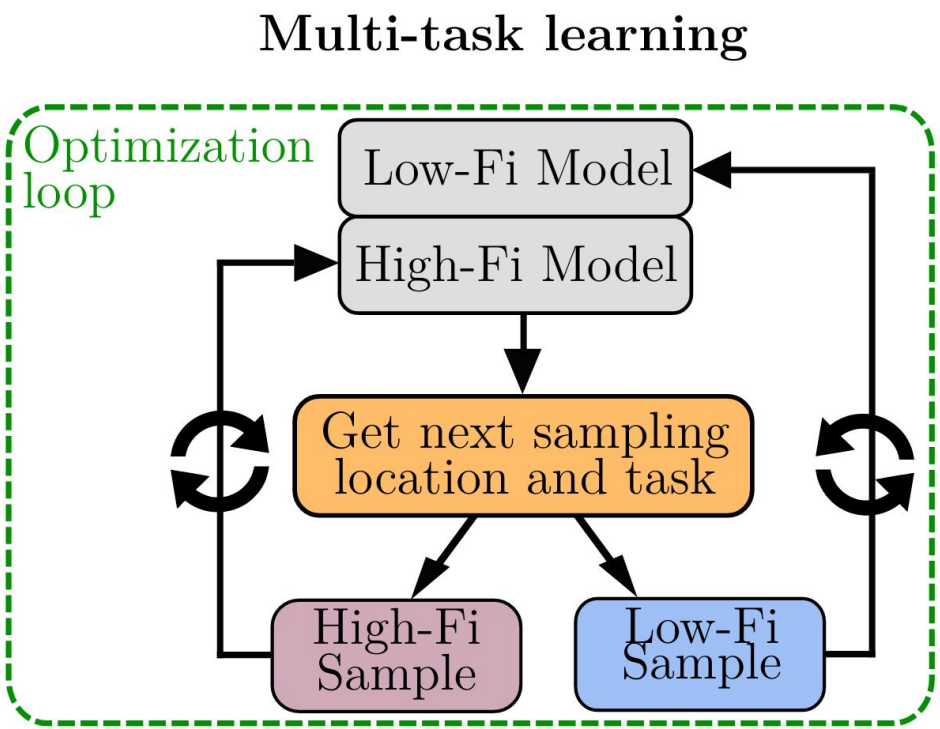
# Using multiple data sources



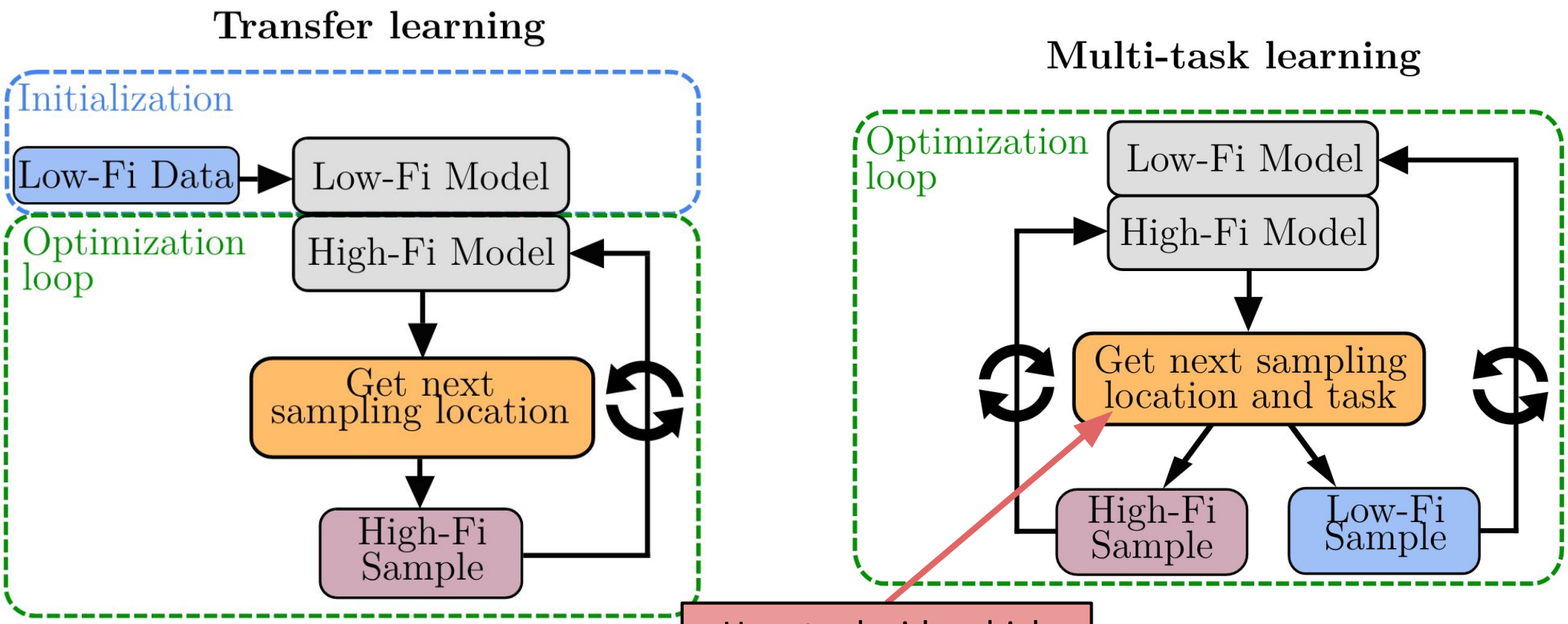

How many samples should be used for the initialization?

# Using multiple data sources



# Using multiple data sources




How to decide which fidelity level to sample?

# Research questions

1. Is global minimum found? How many resources are required with respect to single-fidelity BO?

2. Does the ICM model reduce both, BO iterations and CPU time?

3. What fidelity combinations are most efficient? Is this quantifiable by correlation and cost difference between simulators?

4. Transfer learning: How does efficiency change with the number of initialization points?

# Research objectives

1. **Literature review** on multi-task Bayesian optimization
   - Multi-task regression models
   - Multi-task acquisition strategies

2. **Computational implementation**: Extension of available code for multi-fidelity Bayesian optimization

3. **Design experiments** to answer research questions

# Computational Implementation

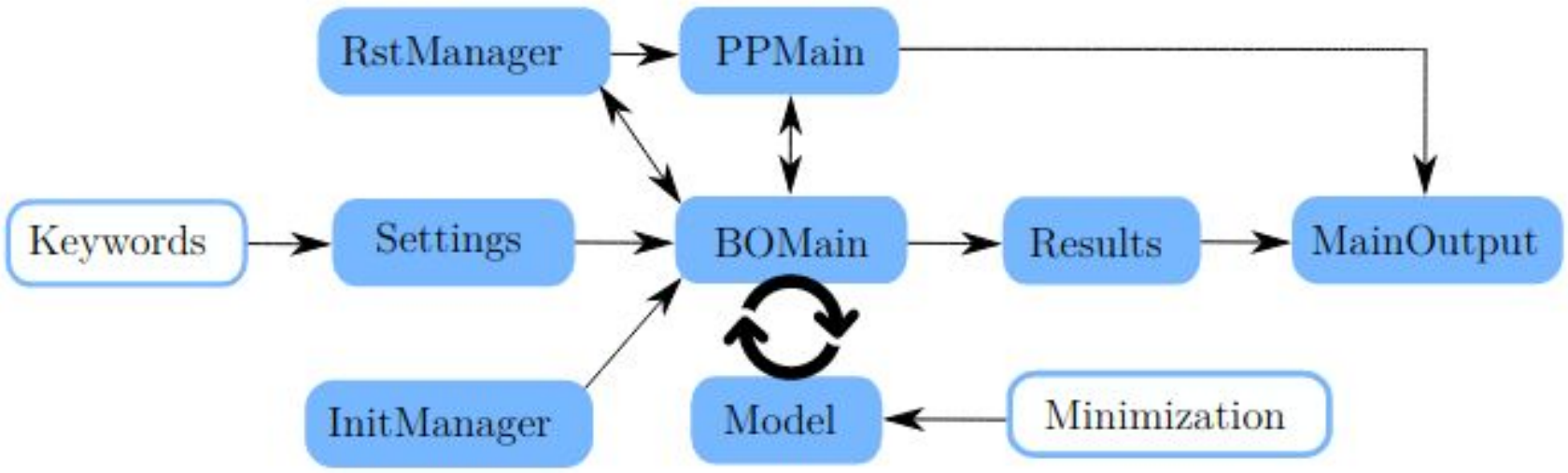
# Implementation of BOSS

- Builds on top of GPy library
  - GPy: Python framework for GPR methods

- Constructs GP models from input data and attributes defined in settings file

- BOSS extends GP regression with BO functionalities and integrates with materials simulation codes

# BOSS framework



# Extension of multi-fidelity BOSS

Extending BOSS framework:

- Unify acquisition functions to package
- Add acquisition function base template to consolidate functionalities
- Add class for multi-task acquisition approaches

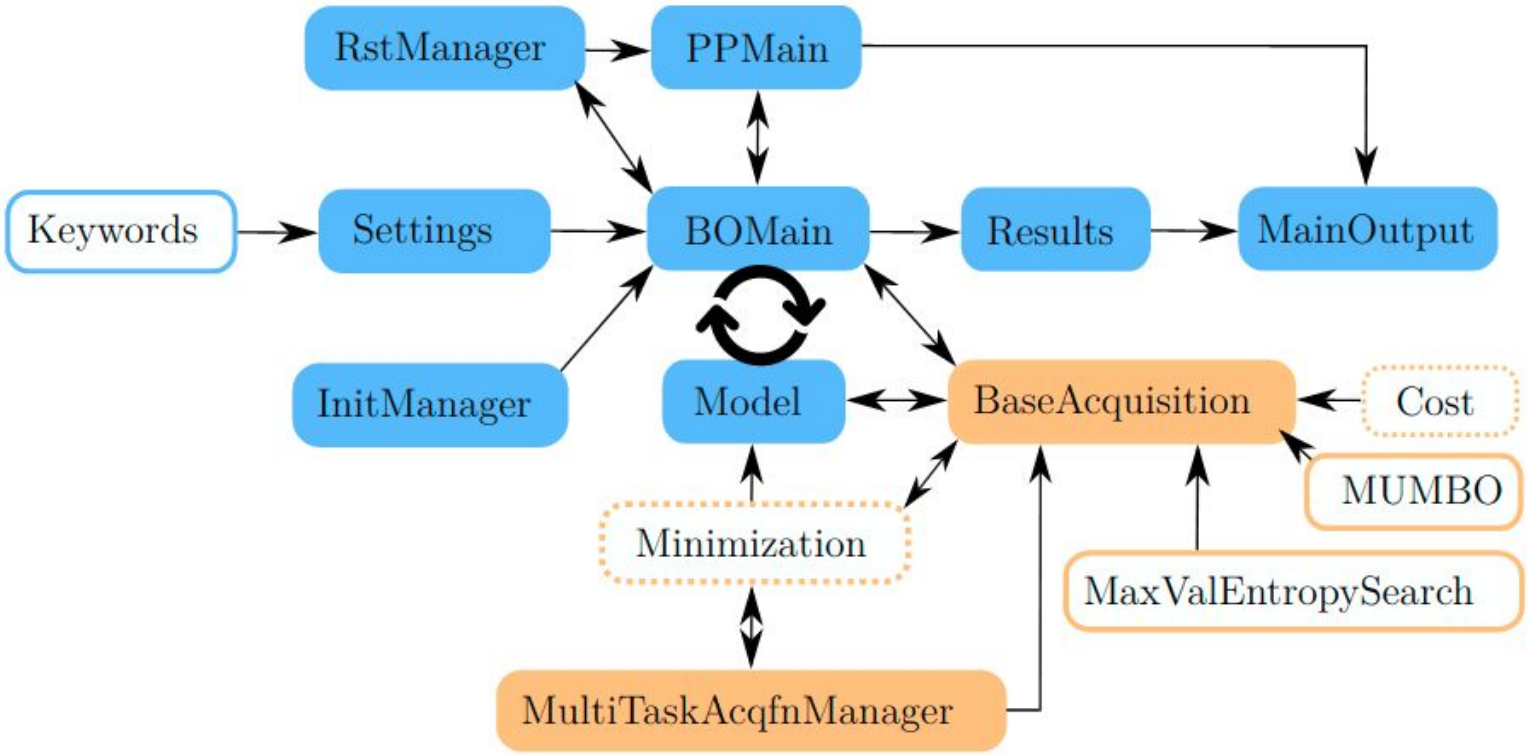
New acquisition functions:

- Add MES and MUMBO (multi-task version of MES)
- New initialization for minimizer (used to minimize acquisition function and posterior model)

Code validation:

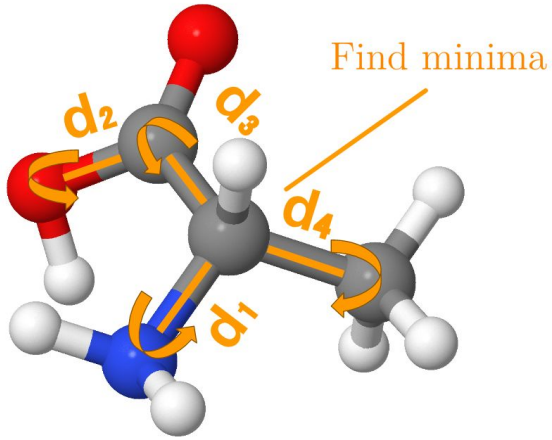
- Unit tests to validate individual parts of the implementation

# Contributions to existing code



# Experiments and Results

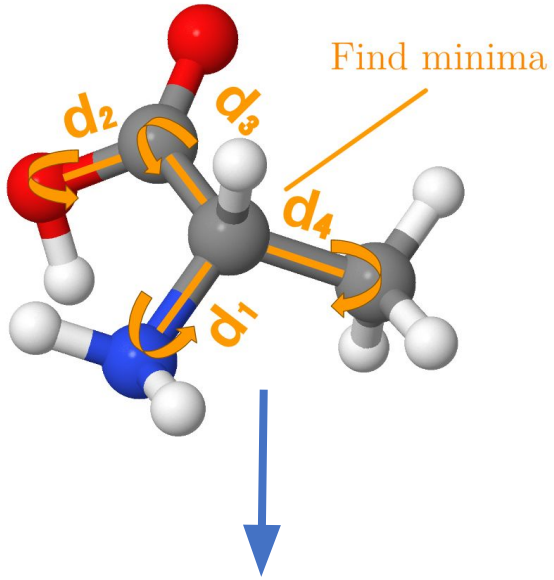
# Test case: alanine conformer search




Find energy minimum of alanine model for

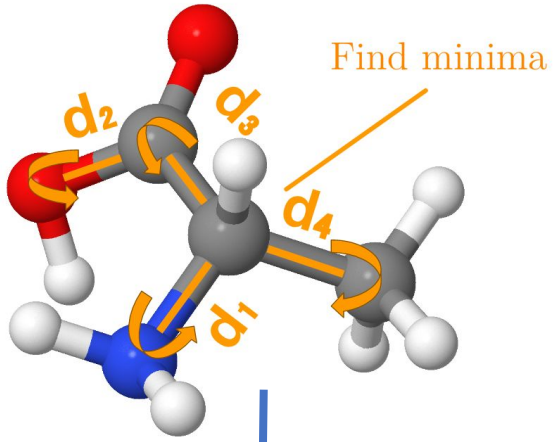
1. 2D search space (keep dihedrals 3 and 4 fixed)
2. 4D search space

# Test case: alanine conformer search

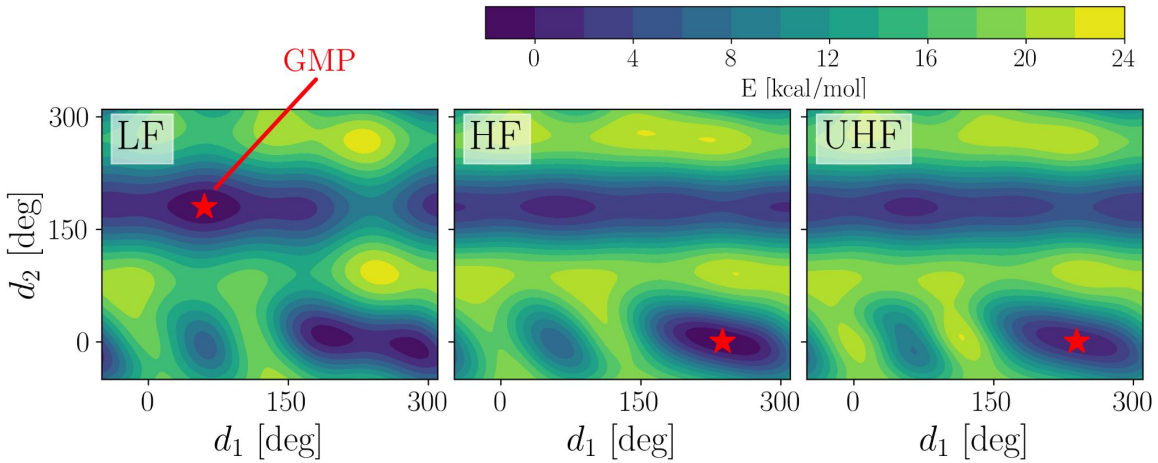



- **Low (LF)**: Force fields (AMBER18)
- **High (HF)**: Density functional theory (FHI-AIMS, using PBE-exchange correlation functional)
- **Ultra high (UHF)**: Quantum chemistry (Gaussian16, using CCSD(T))

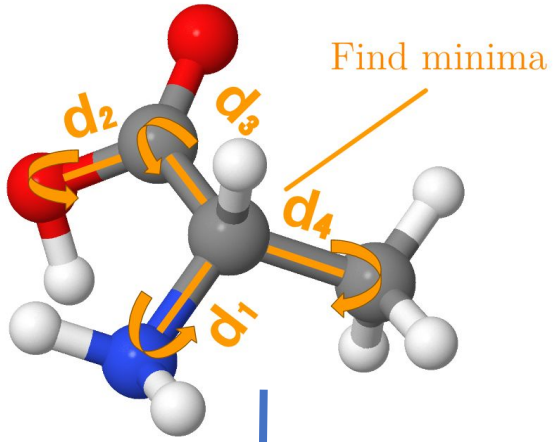
# Test case: alanine conformer search



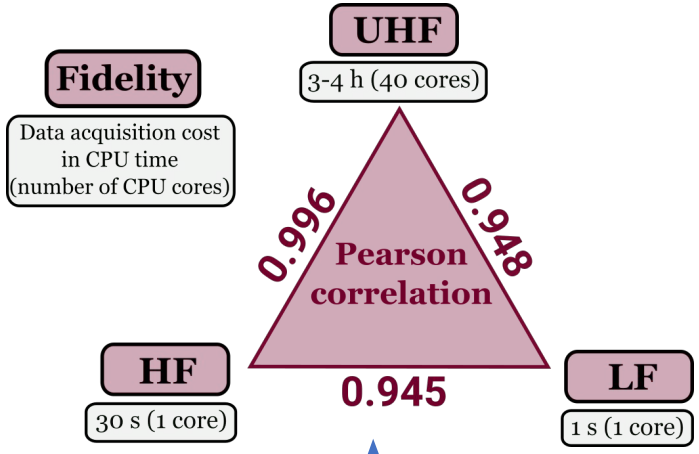

- **Low (LF)**: Force fields (AMBER18)
- **High (HF)**: Density functional theory (FHI-AIMS, using PBE-exchange correlation functional)
- **Ultra high (UHF)**: Quantum chemistry (Gaussian16, using CCSD(T))

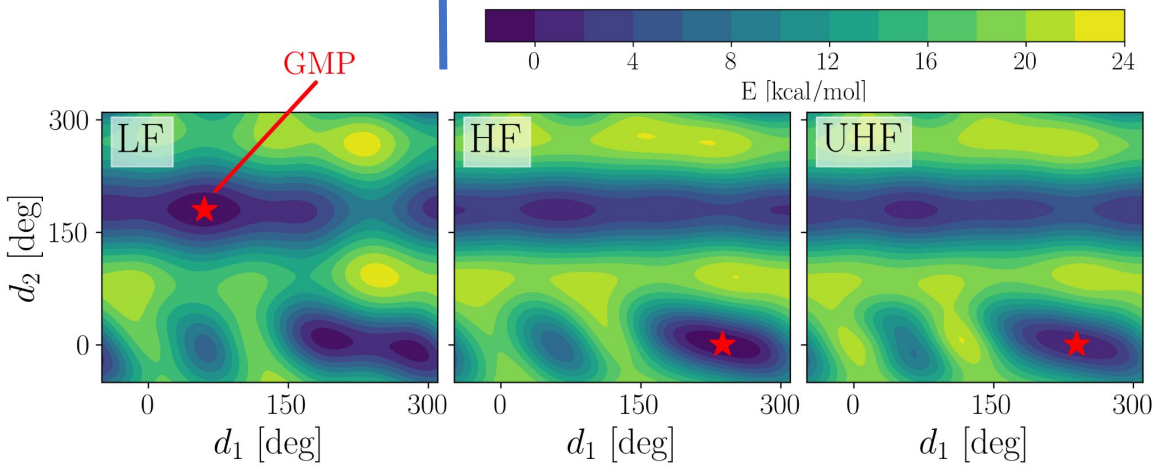


# Test case: alanine conformer search




- **Low (LF)**: Force fields (AMBER18)
- **High (HF)**: Density functional theory (FHI-AIMS, using PBE-exchange correlation functional)
- **Ultra high (UHF)**: Quantum chemistry (Gaussian16, using CCSD(T))



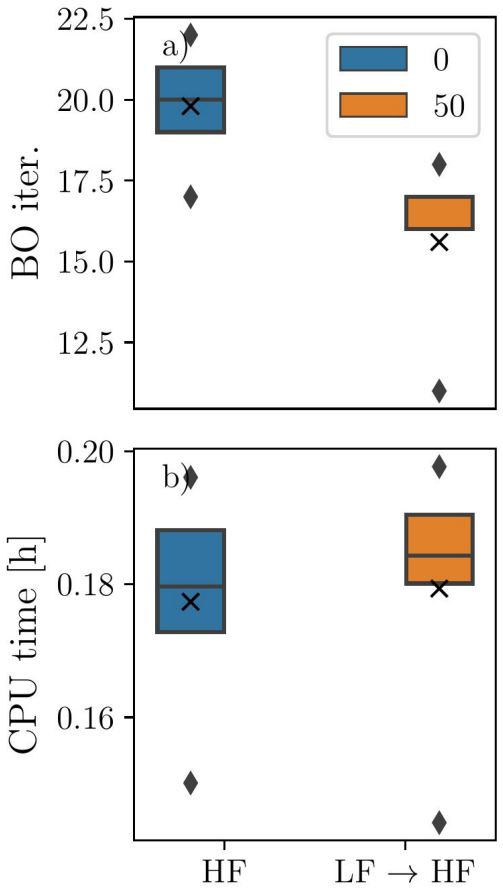



# Measuring efficiency of approaches

- **Accuracy**: Global minimum found?
- **Efficiency**: How fast is global minimum found?
- **Repeat 5 BOSS cycles**
  - 2 randomly chosen initial points
  - 100 (300) BO iterations in 2D (4D)
- Convergence tolerance: 0.01 eV (0.23 kcal/mol)

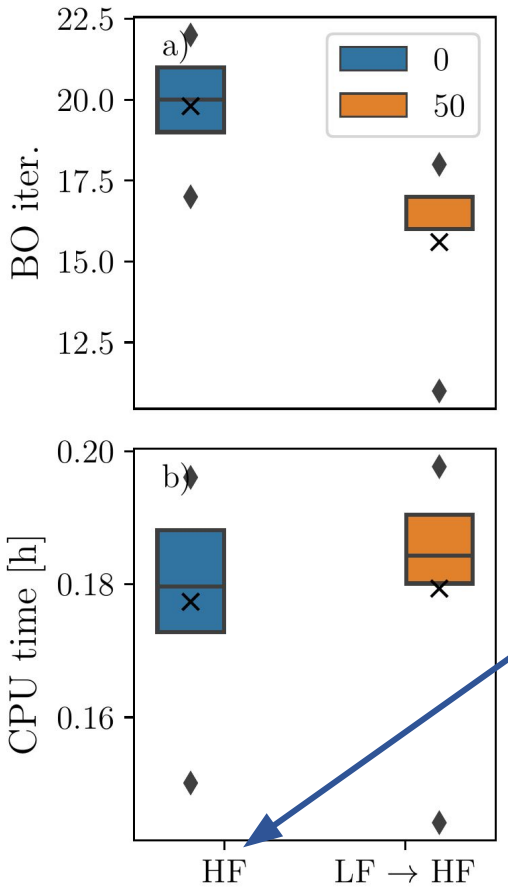
# Measuring efficiency of approaches

- **Accuracy**: Global minimum found?
- **Efficiency**: How fast is global minimum found?
- **Repeat 5 BOSS cycles**
  - 2 randomly chosen initial points
  - 100 (300) BO iterations in 2D (4D)
- Convergence tolerance: 0.01 eV (0.23 kcal/mol)



# Measuring efficiency of approaches

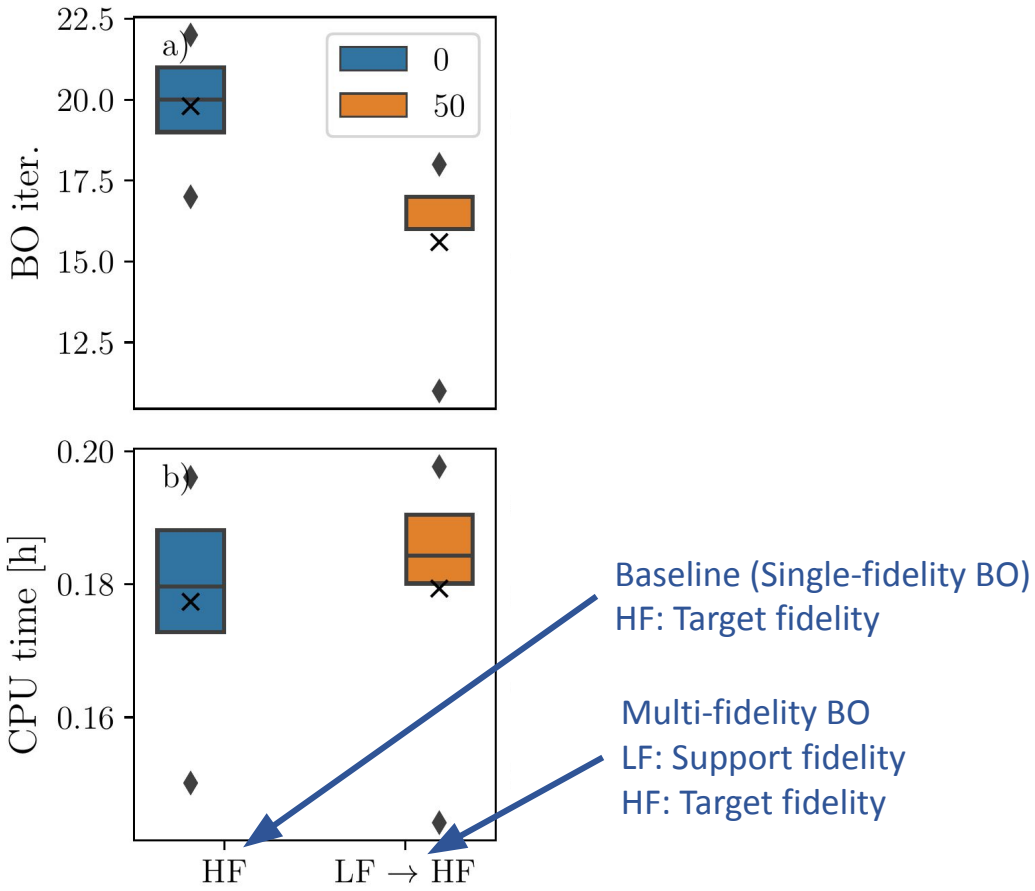
- **Accuracy**: Global minimum found?
- **Efficiency**: How fast is global minimum found?
- **Repeat 5 BOSS cycles**
  - 2 randomly chosen initial points
  - 100 (300) BO iterations in 2D (4D)
- Convergence tolerance: 0.01 eV (0.23 kcal/mol)



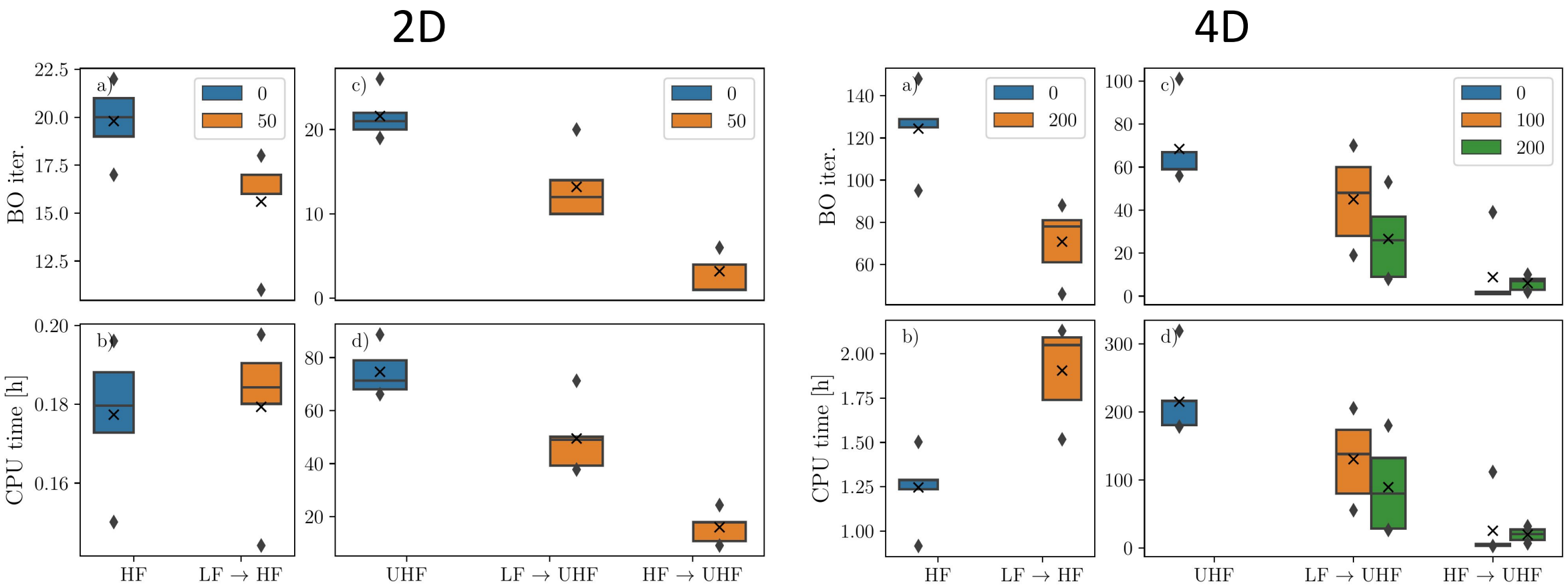

Baseline (Single-fidelity BO)
HF: Target fidelity

# Measuring efficiency of approaches

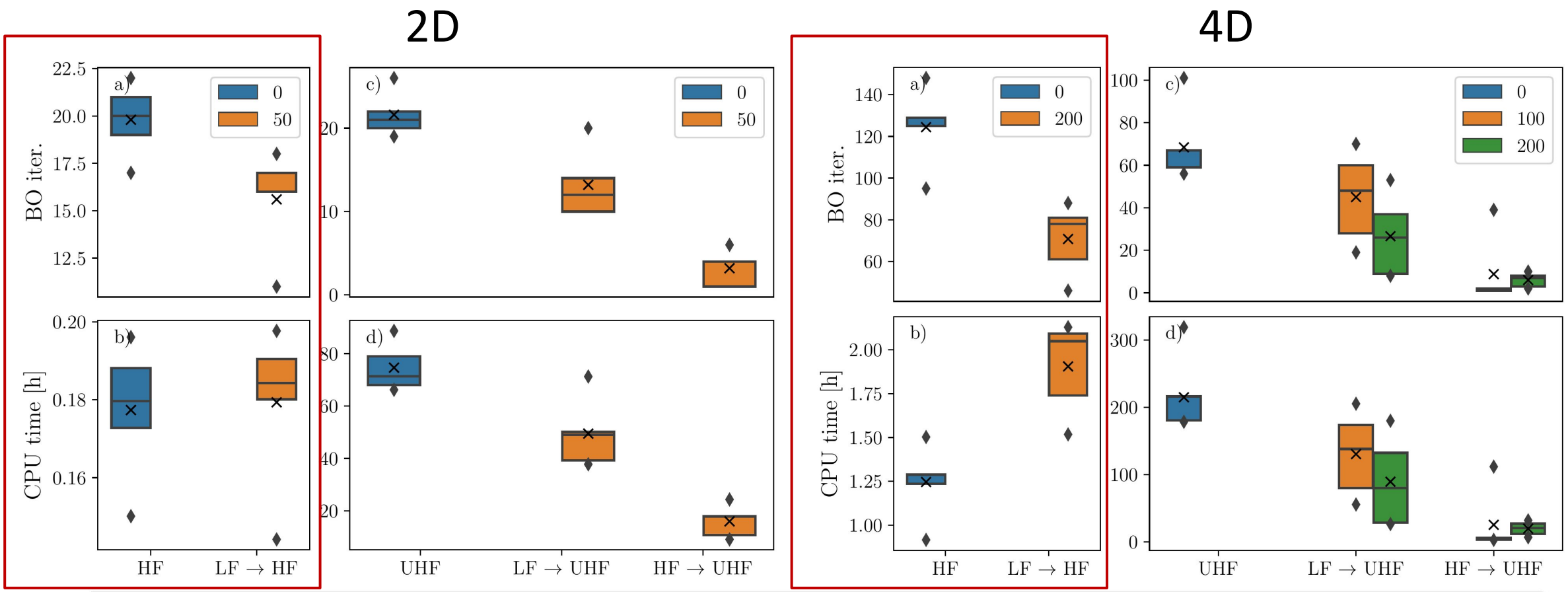
- **Accuracy**: Global minimum found?
- **Efficiency**: How fast is global minimum found?
- **Repeat 5 BOSS cycles**
  - 2 randomly chosen initial points
  - 100 (300) BO iterations in 2D (4D)
- Convergence tolerance: 0.01 eV (0.23 kcal/mol)



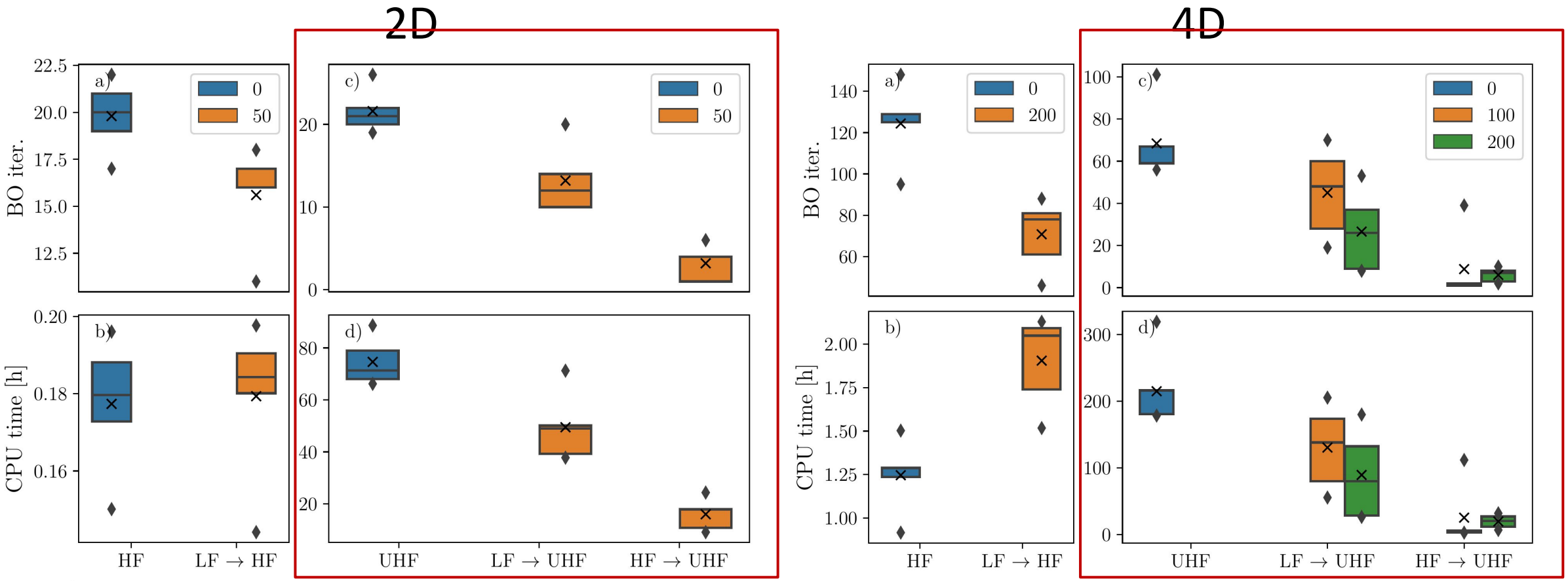
# Transfer learning results



# Transfer learning results



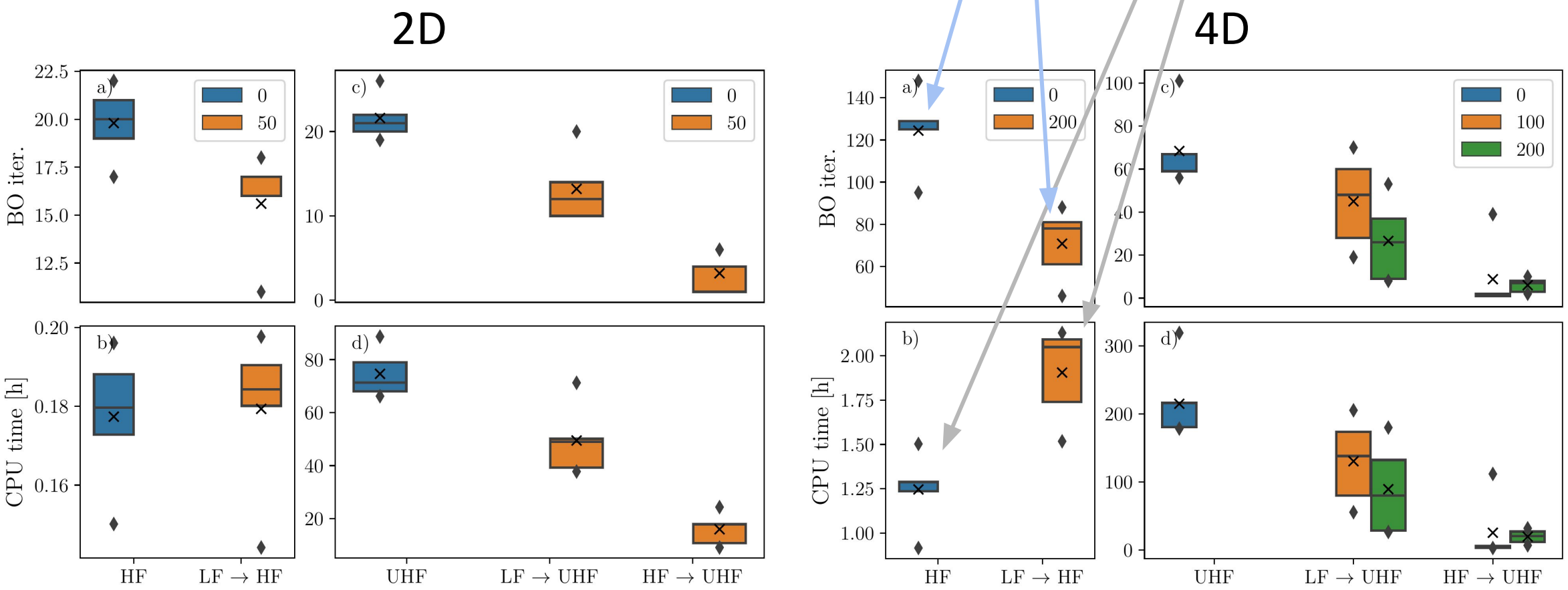
# Transfer learning results



# Transfer learning results

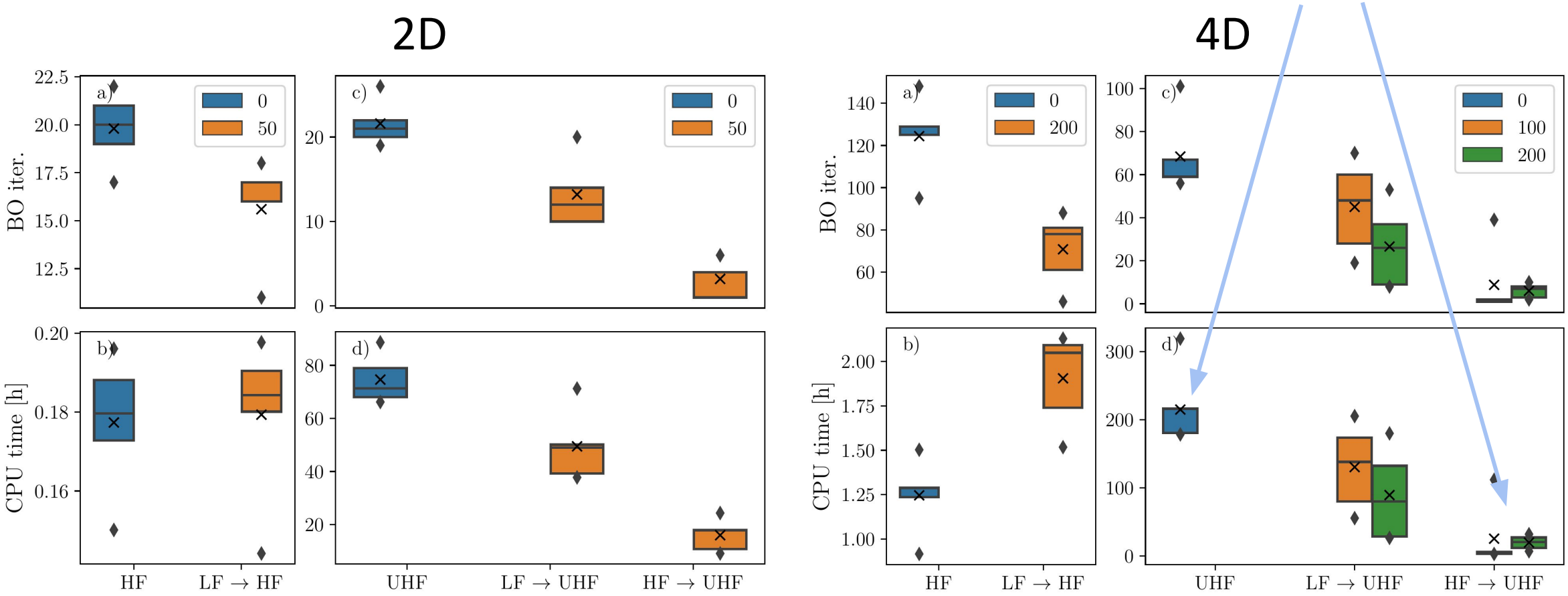
Reduction in BO Iterations

Increase in the computation time!



# Transfer learning results

Reduction in computation time by 90%!



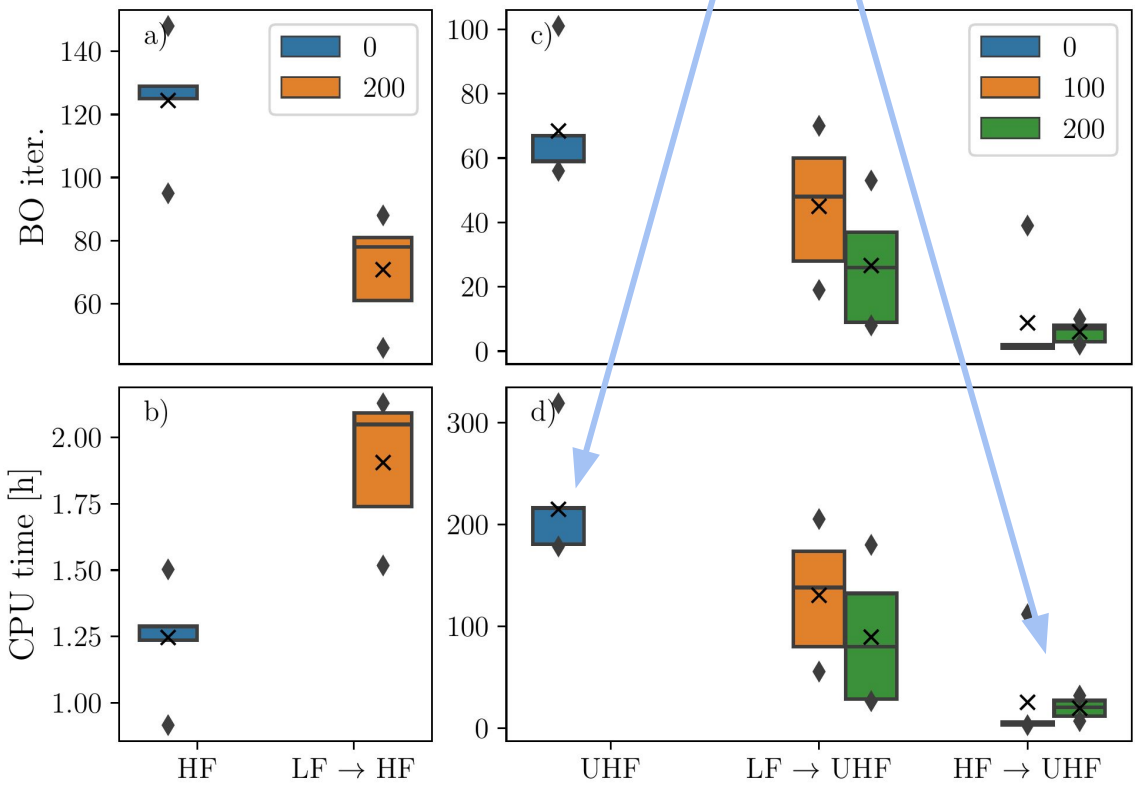
# Transfer learning results

Transfer learning savings depend on

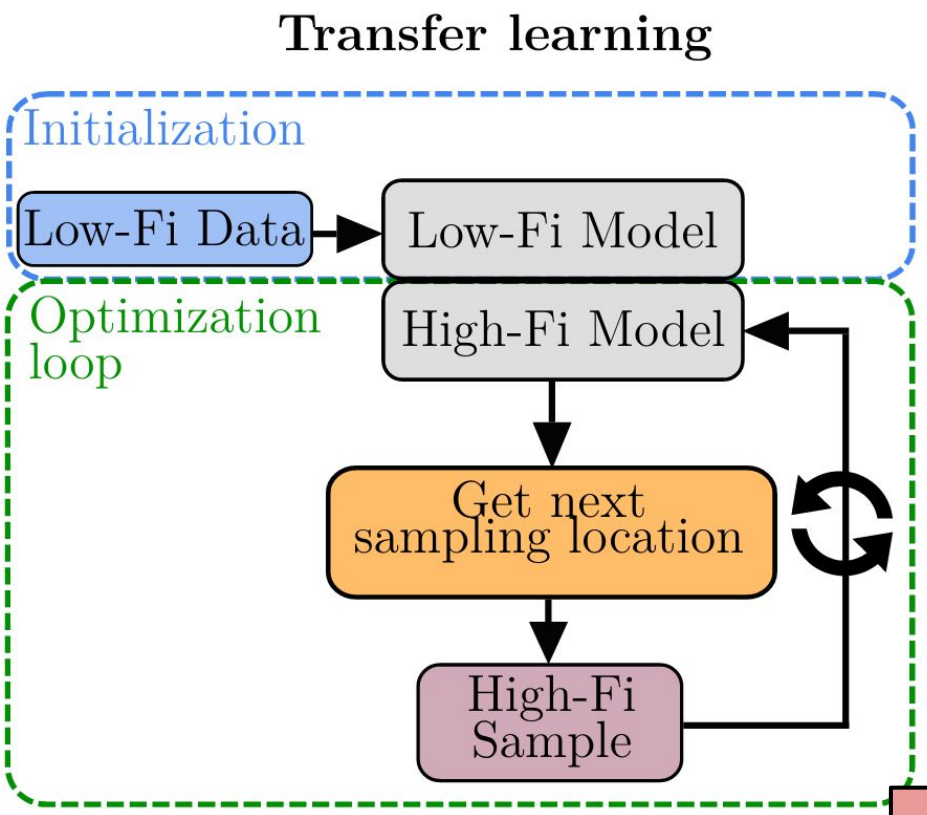
- Correlation between the fidelities (x-axis)
- Support fidelity data points
- Simulator evaluation time (< 1 min vs 3 h)

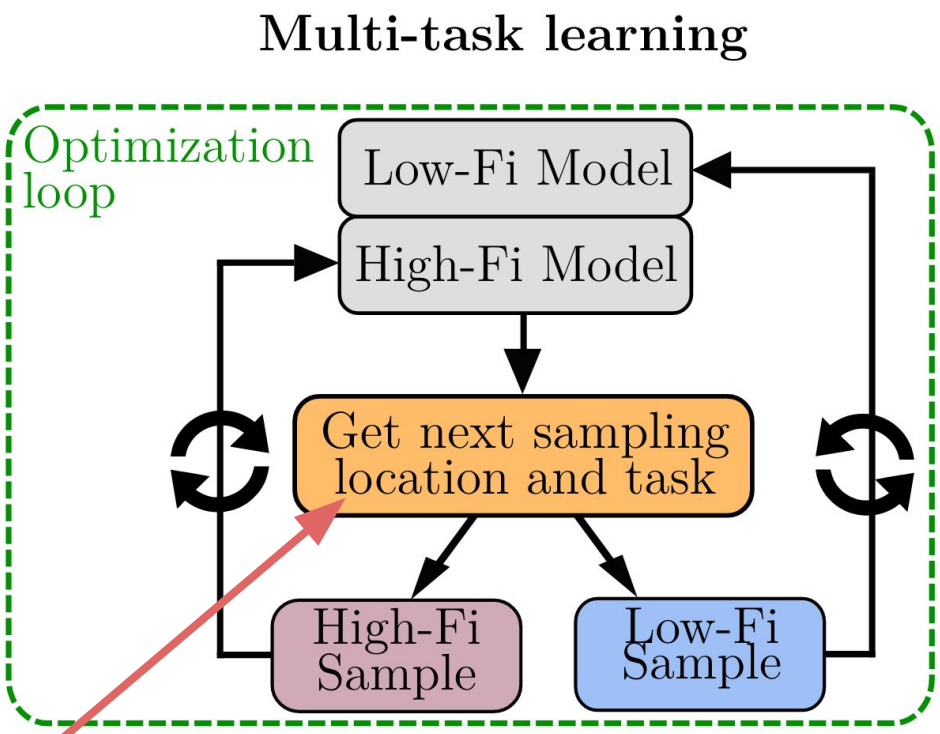
Reduction in computation time by 90%!

4D



# How to make use of multiple data sources?




How to decide which fidelity level to sample?

# Multi-task learning: Information gain per cost

Objective: Construct a
multi-fidelity acquisition rule

Fidelity selection should be based
on **correlation** and **simulator cost**
of the fidelities

# Multi-task learning: Information gain per cost

Objective: Construct a multi-fidelity acquisition rule

Fidelity selection should be based on **correlation** and **simulator cost** of the fidelities

**Maximize 'information gain per cost'** $g$ at $x_*$

$$g_*(t) = \frac{\sigma_*(t,t')^2}{\sigma_*(t,t)c(t)}$$

$t'$ : Target fidelity

$\sigma_*(t,t')$ : Covariance at $x_*$ between fidelity level $t$ and $t'$

$\sigma_*(t,t)$ : Variance of fidelity level $t$ at $x_*$

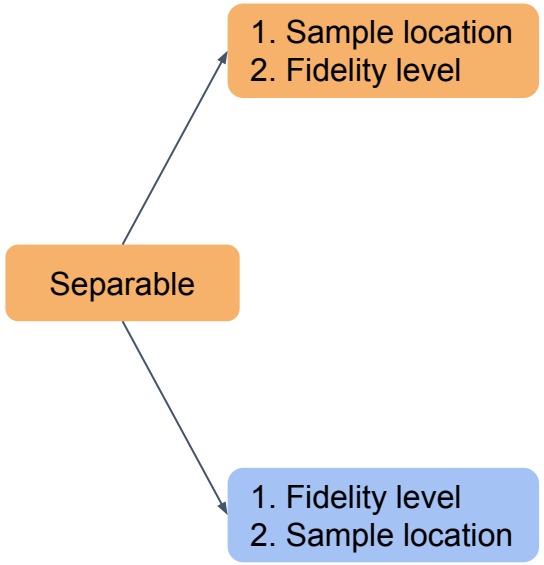
$c(t)$ : Evaluation time of fidelity level *t* in CPU time



R. Lam et al., *Multifidelity Optimization using Statistical Surrogate Modeling for Non-Hierarchical Information Sources* (2015)
U. Remes et al., *Multi-task Bayesian Optimization in Multi-fidelity Atomistic Structure Search (in preparation)*

# Separable multi-task learning approaches

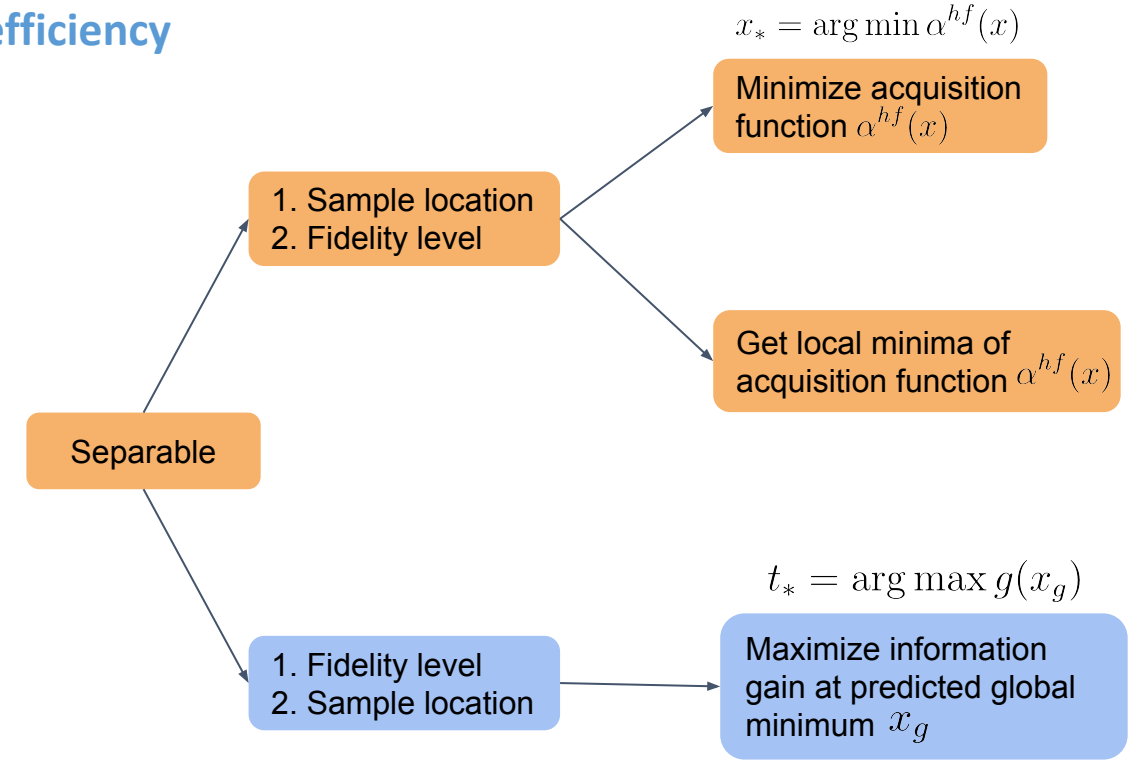
**Very good efficiency**
**Good efficiency**



# Separable multi-task learning approaches

**Very good efficiency**

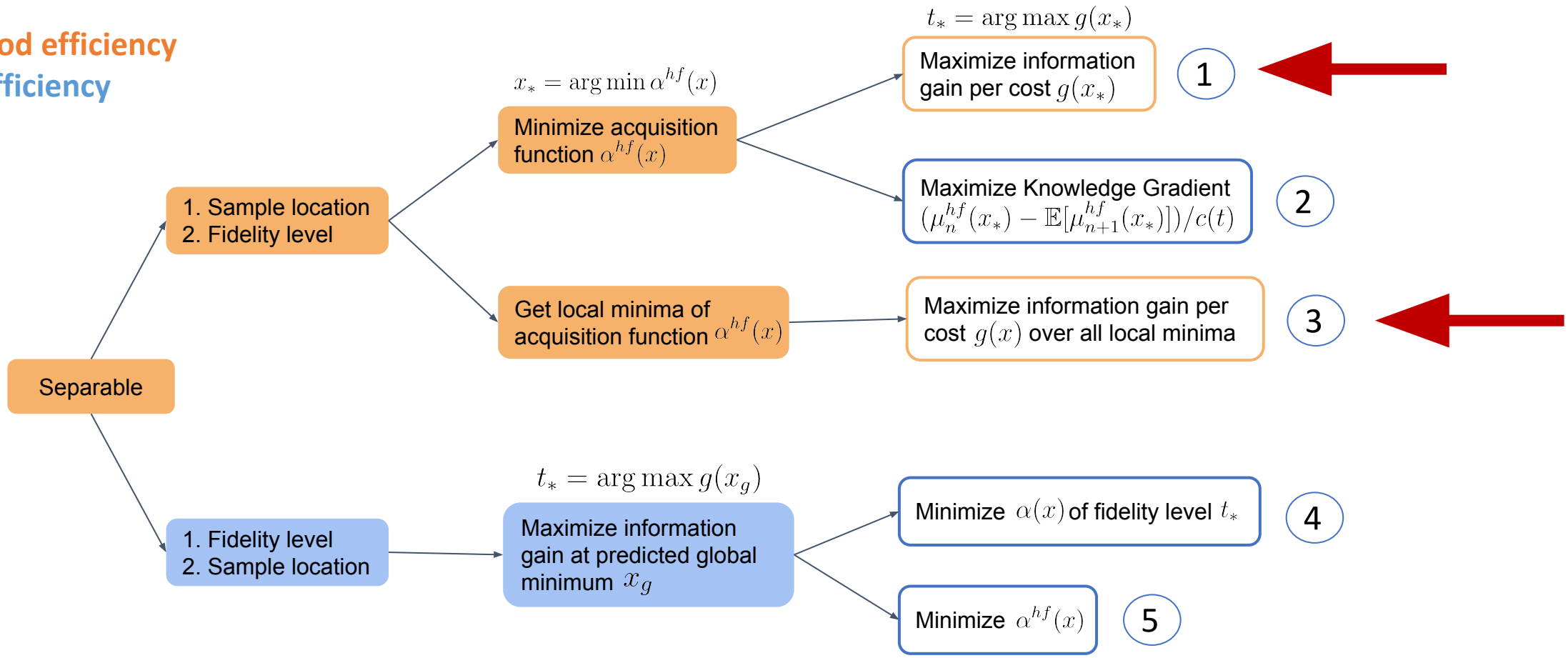
**Good efficiency**



# Separable multi-task learning approaches

**Very good efficiency**

**Good efficiency**



# Inseparable multi-task learning approaches

**Very good efficiency**
**Good efficiency**

Inseparable

Minimize acquisition function over sample location and fidelity level:
$$x_*, t_* = \arg\min_{x,t}[\alpha(x,t)]$$

6

Maximize information gain per cost over sample location and fidelity level:
$$x_*, t_* = \arg\max_{x,t}[g(x,t,t')]$$

7

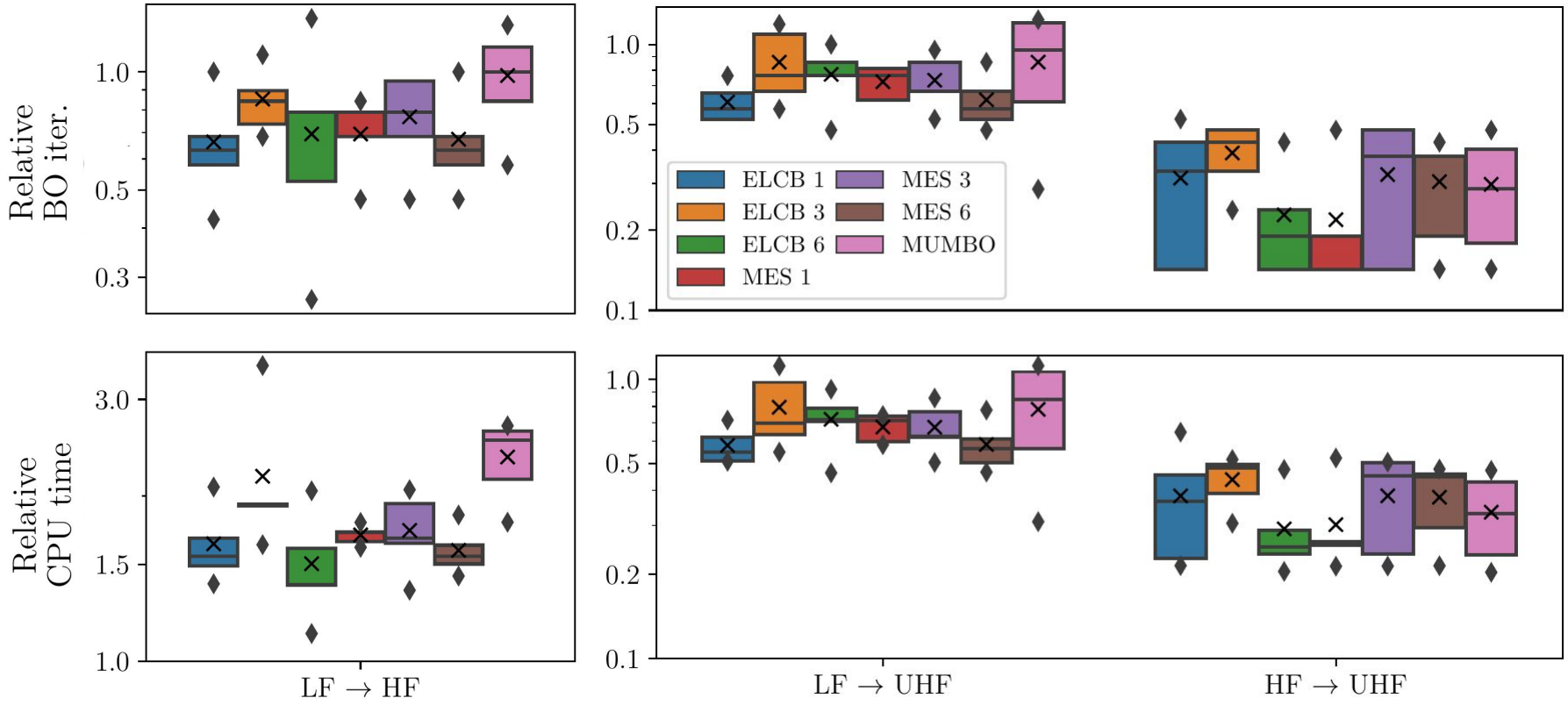
MUlti-task Max-value Bayesian Optimization (MUMBO)

8

*Eight* multi-task approaches ⟶ Reduce number using pre-screening experiments

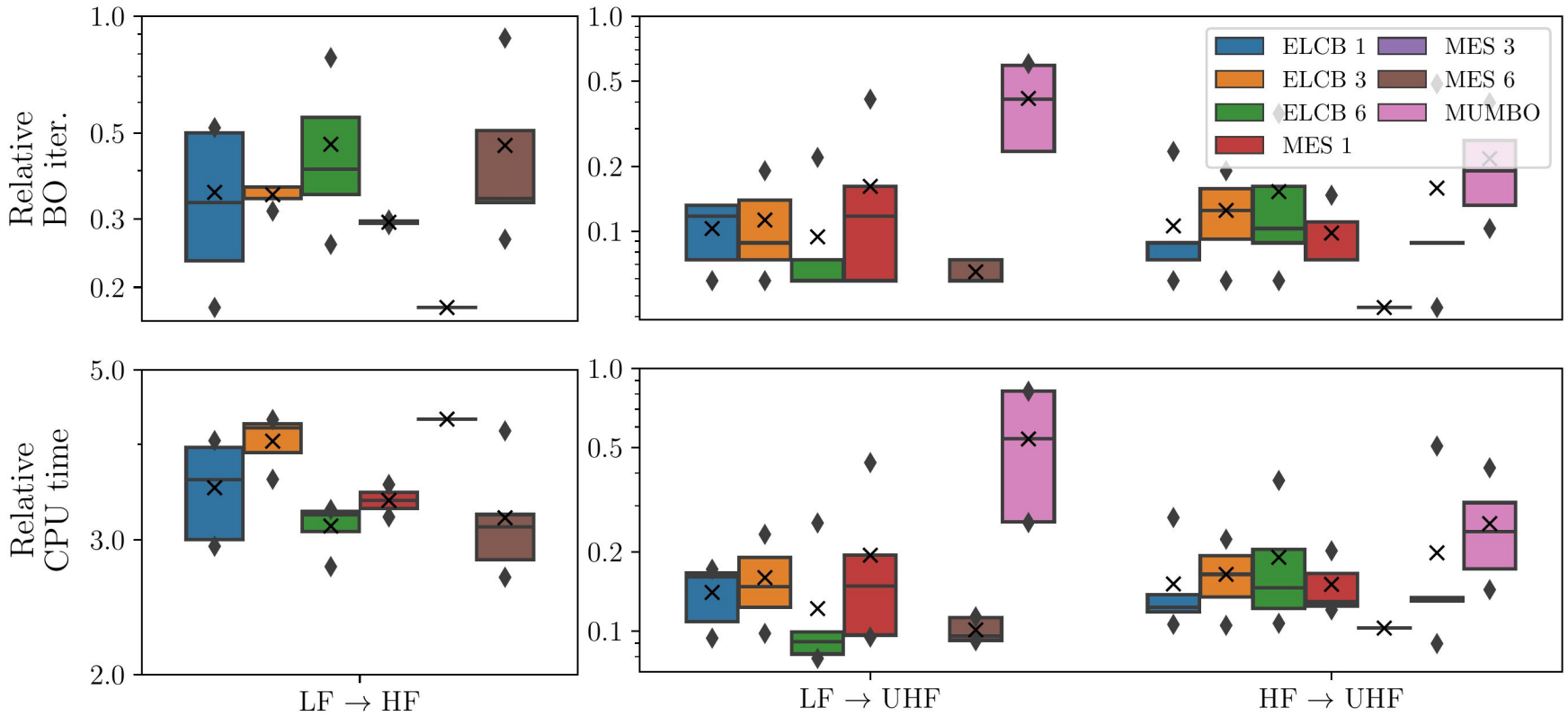
# Multi-task learning results

**Cost relative to baseline (2D optimization)**



# Multi-task learning results

**Cost relative to baseline (4D optimization)**



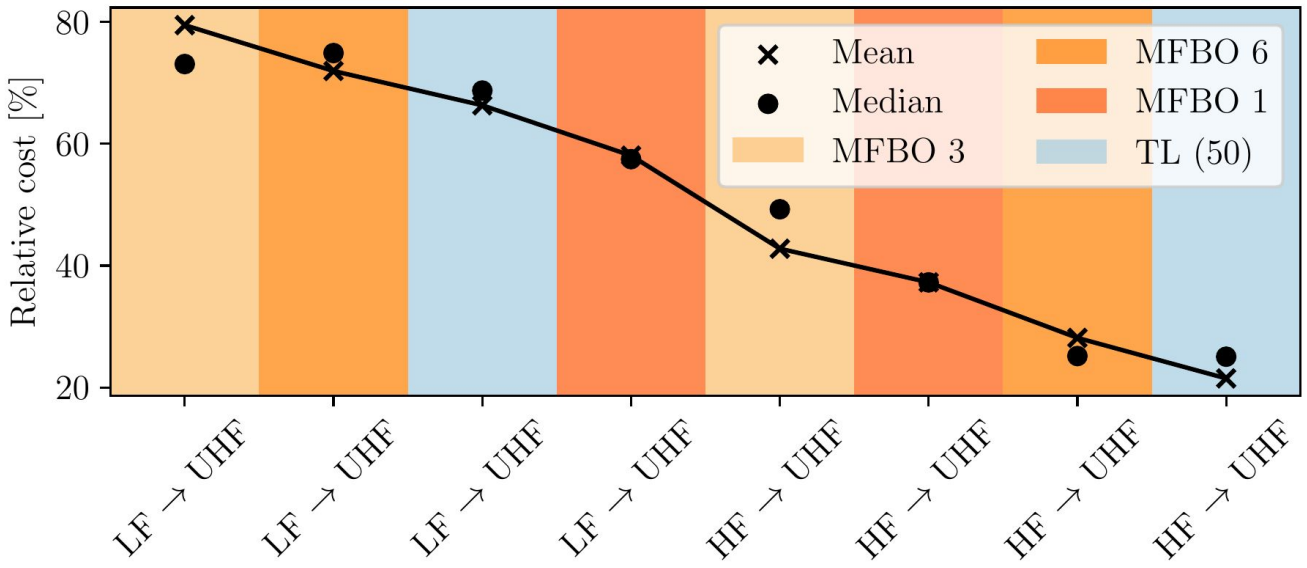
# Comparing transfer and multi-task learning

- HF always better as support fidelity

**Best performance:**

- MFBO 6
- TL with HF as support fidelity

**Relative CPU cost to single-fidelity BO (2D)**



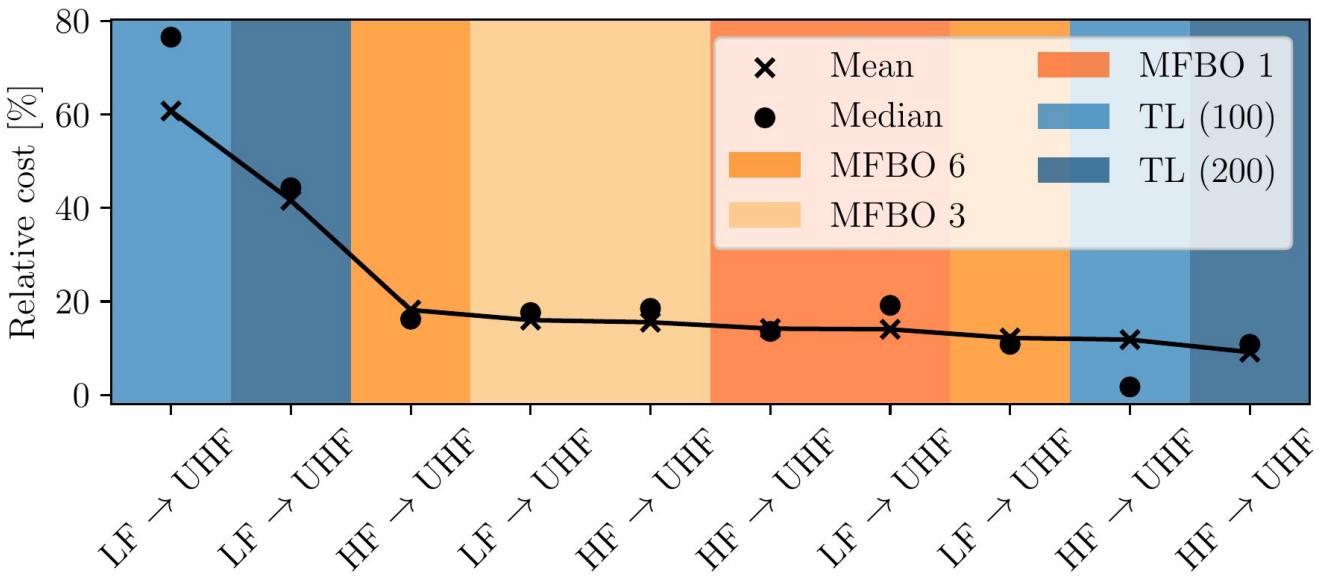
# Comparing transfer and multi-task learning

- LF enables huge savings for MT learning, HF for both TL and MT

**Best performance:**

- MFBO 1 & 6
- TL with HF as support fidelity

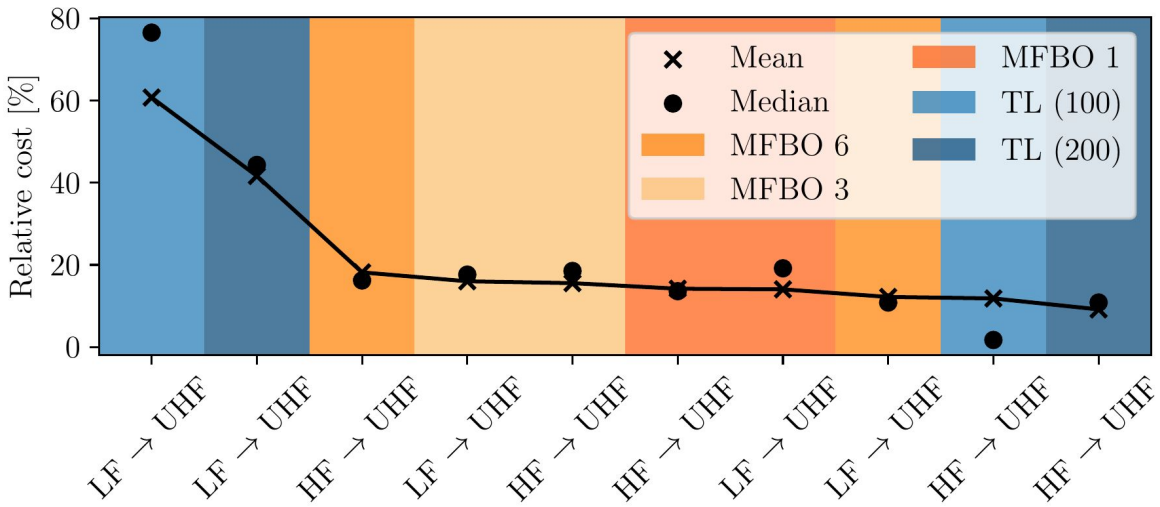
**Relative CPU cost to single-fidelity BO (4D)**



# Summary

- **Cost reduction:** Multi-fidelity BO has **saved 90% CPU time** in comparison to standard BO

- **Usage:** MT and TL both enable large savings. MT does not require setting the number of support data



# Information gain per cost

- Information gain at x* corresponds to variance reduction at x*
- Conditional distribution of target task, when observing f(x*,t) is again normal distribution with variance

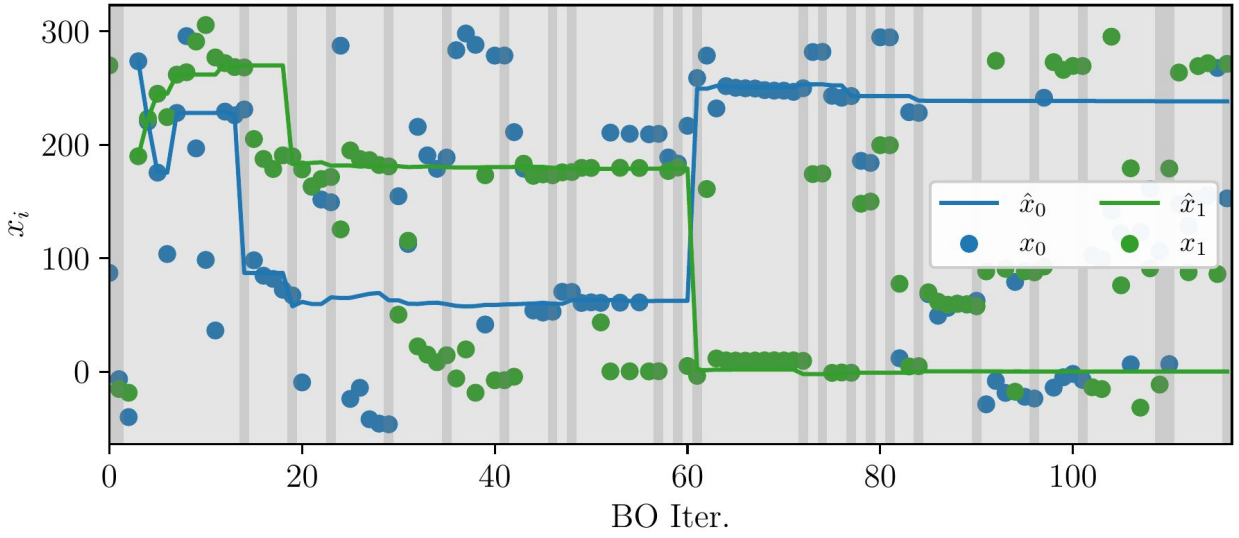
$$\sigma_*(t,t) - \frac{\sigma_*(t,t')^2}{\sigma_*(t,t)}$$

Dividing the variance reduction term by cost c(t):

$$g_*(t) = \frac{\sigma_*(t,t')^2}{\sigma_*(t)c(t)}$$

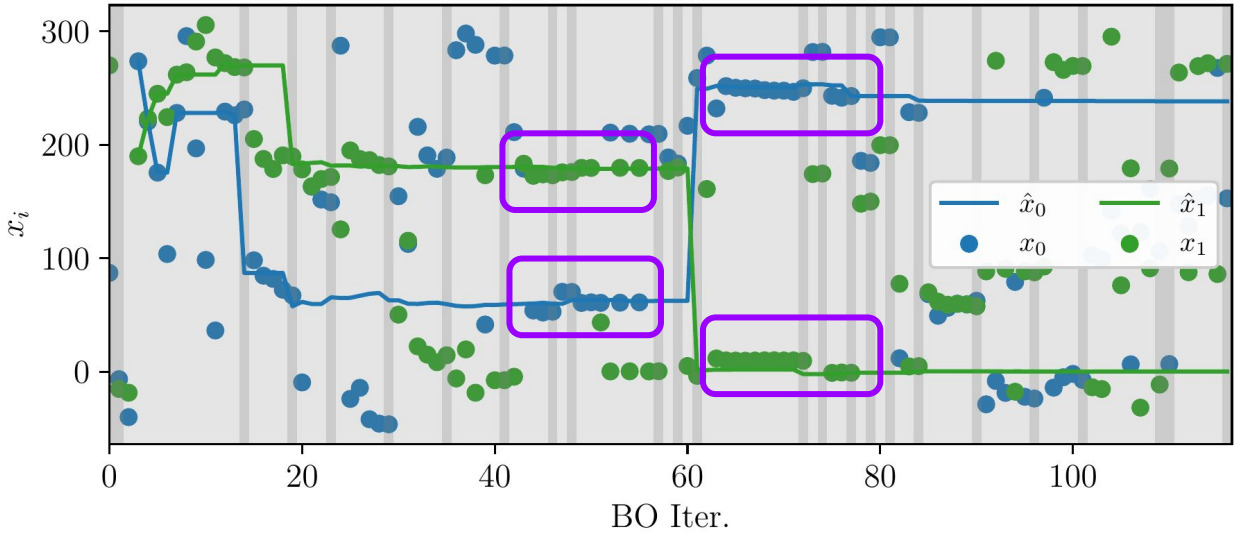
# Multi-task approach 1

- Solid line: Global minimum prediction for each coordinate
- Light gray background: Sample from LF simulator
- Dark gray background: Sample from UHF simulator



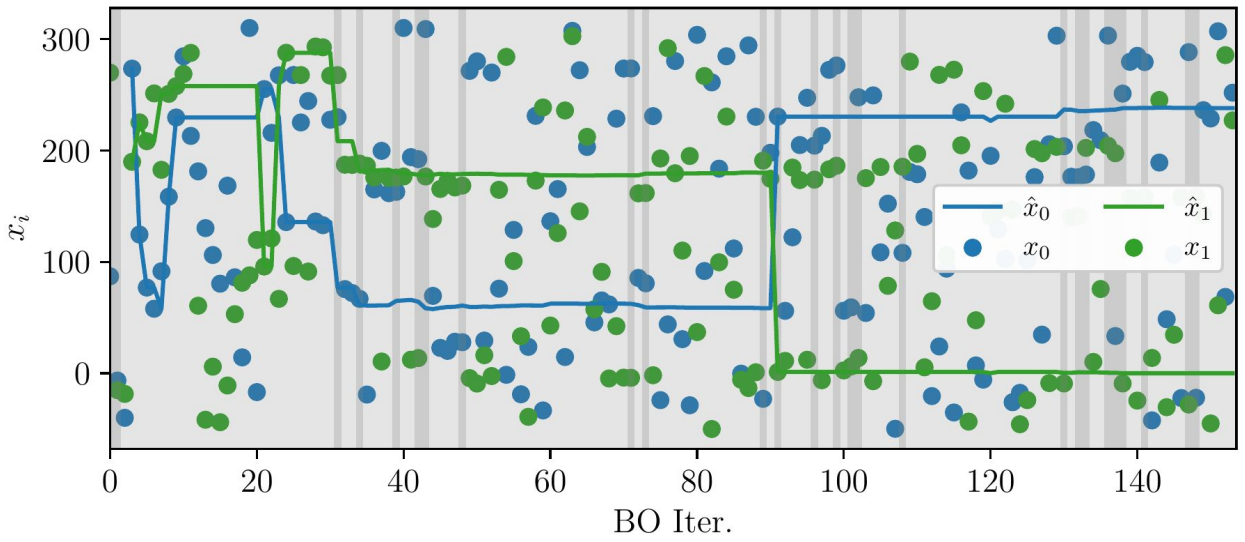
# Multi-task approach 1

- Convergence of global minimum prediction after about 60 BO iterations
- Exploitation of minimum regions



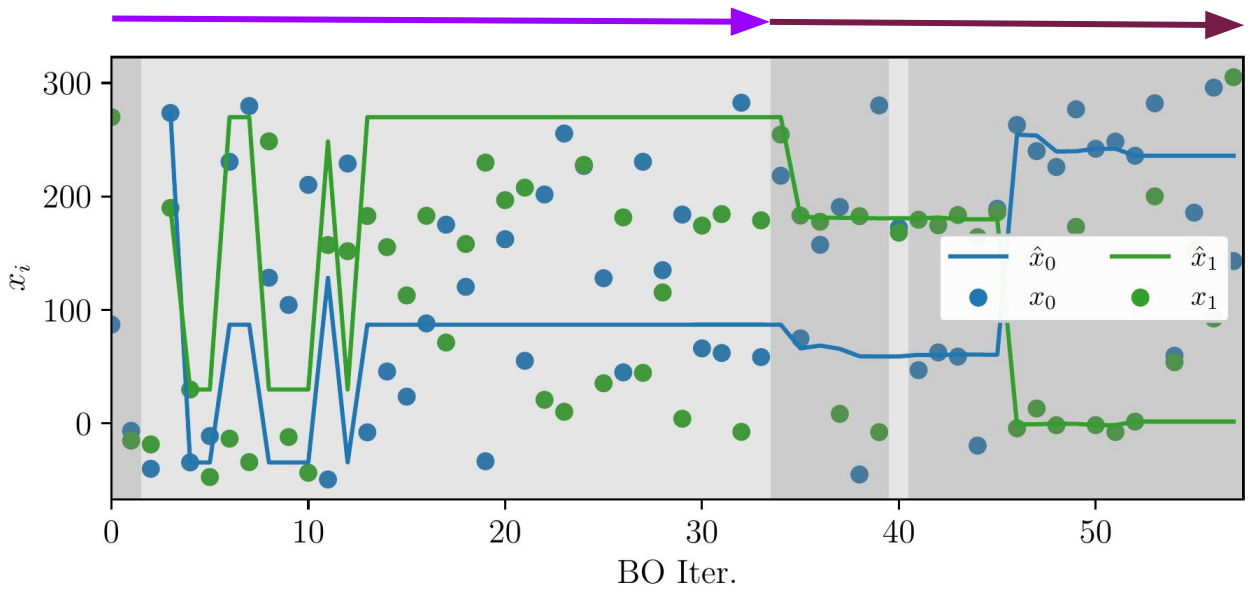
# Multi-task approach 3

- Very short exploitation phases
- Much more exploration of the search space



# Multi-task approach 6

Sampling strategy like in transfer learning:
First only support fidelity samples, then only target fidelity samples



# Prescreening multi-task approaches

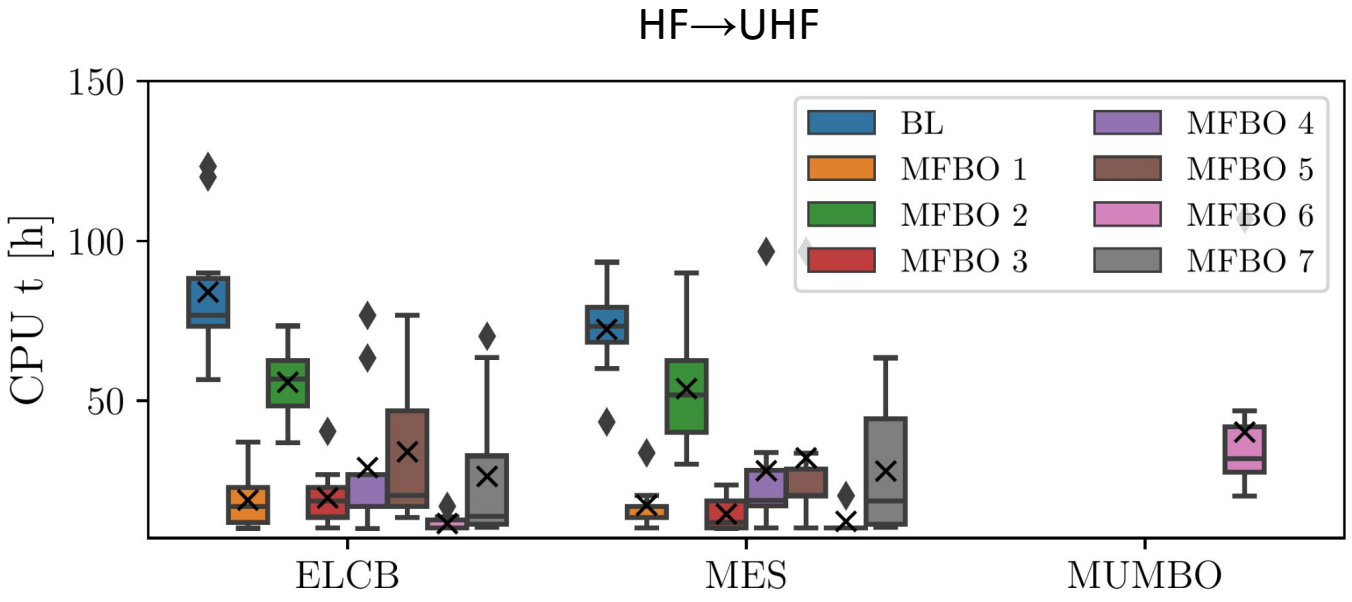
**Atomistic simulator emulator**: GP models, trained on simulator data
→ No simulator evaluation time

- 10 BOSS cycles
- Test acquisition functions: ELCB, MES & MUMBO

# Prescreening multi-task approaches

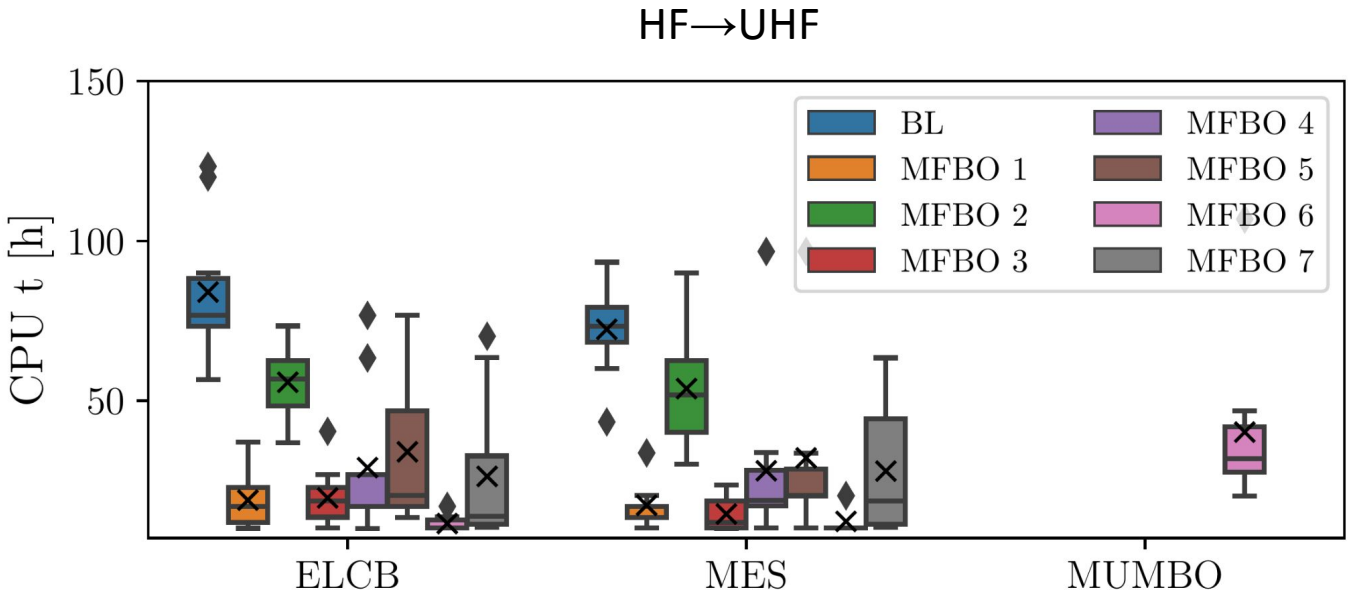
**Atomistic simulator emulator**: GP models, trained on simulator data
⟶ No simulator evaluation time

- 10 BOSS cycles
- Test acquisition functions: ELCB, MES & MUMBO



# Prescreening multi-task approaches

- Approach 1, 3 & 6 have largest cost reduction
- Approach 4 & 5 tend to fail finding global minimum
- ELCB and MES very similar, MUMBO more expensive



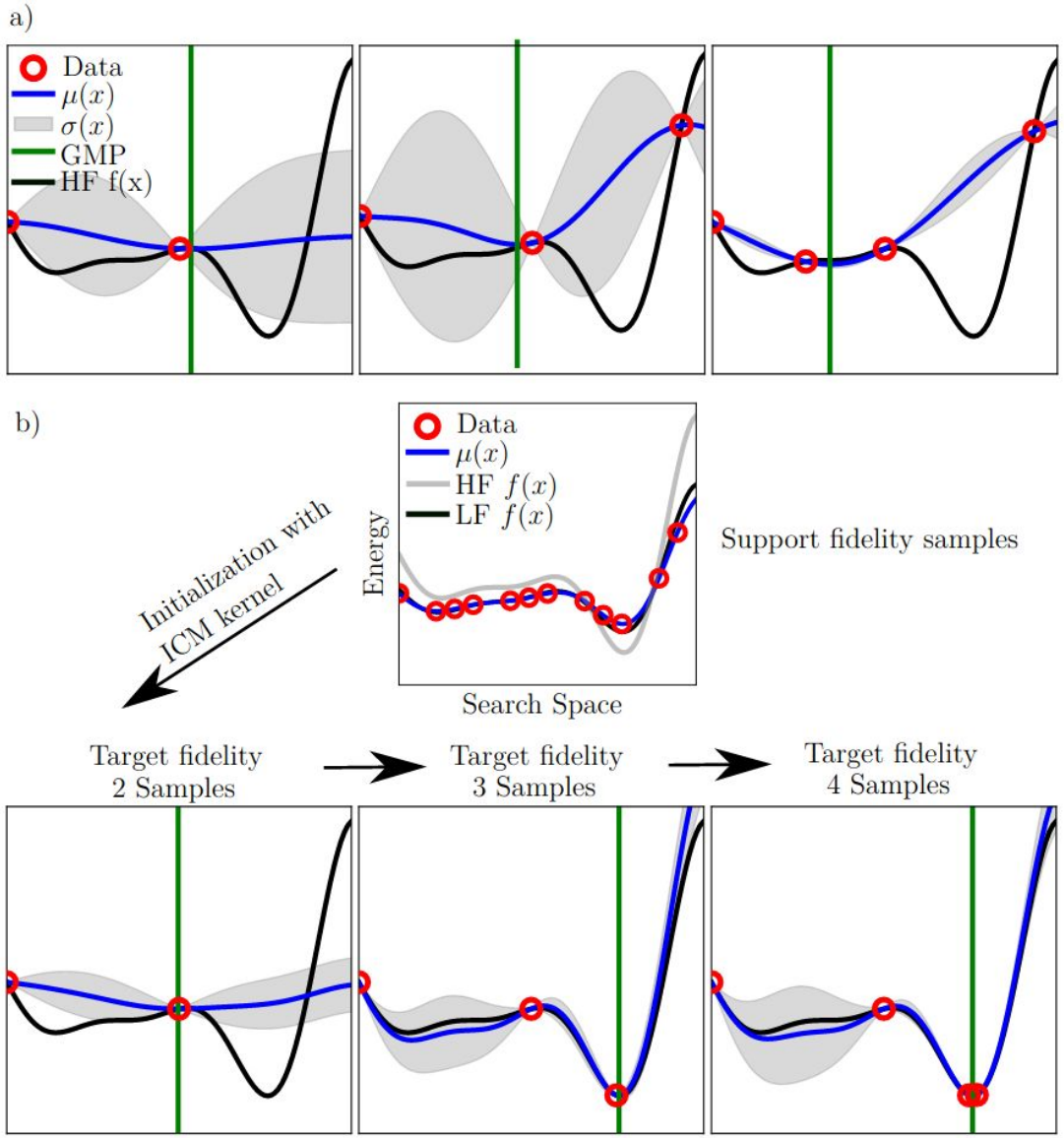

**Conclusion:** Continue calculations with real simulators for approaches 1, 3 &6

# TL example

Top row: Standard BO

Middle row: Support fidelity samples used to initialize model

Bottom row: Multi-fidelity BO



# Probabilistic modelling with Gaussian process regression

**Gaussian process**: Distribution over functions defined by mean $\mu(x)$ and covariance function $k(x, x')$

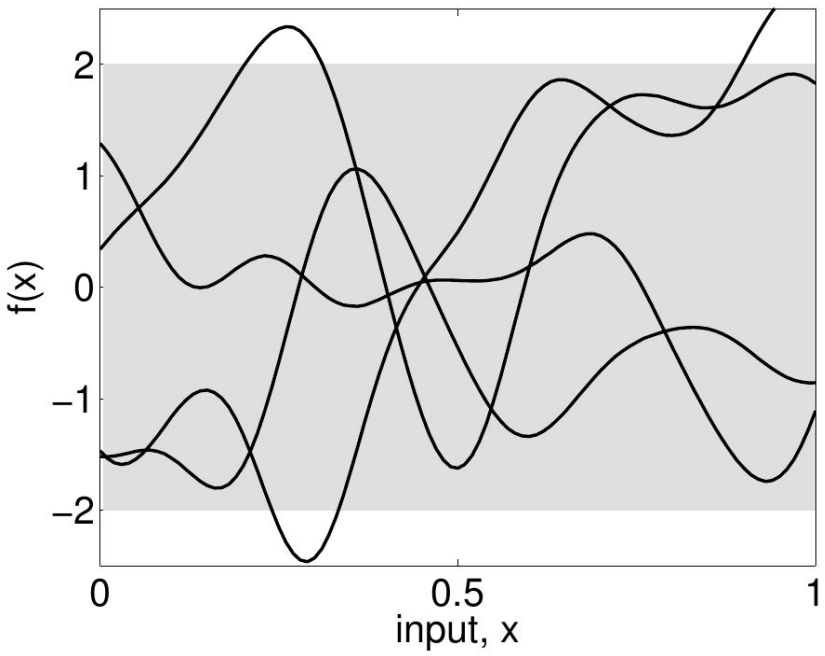
$$f(x) \sim \mathcal{GP}(\mu(x), k(x, x'))$$

$$k_{\mathrm{RBF}}(x, x') = \sigma^2 e^{-\frac{(x-x')^2}{2L^2}}$$

Finite set of random variables $X$ is following a **multivariate Gaussian distribution**:

$$f(X) \sim \mathcal{N}(\mu(X), K(X, X))$$

**Prior distribution**




C.E. Rasmussen et al.,
*Gaussian Processes for Machine Learning* (2008)

# Gaussian Process Regression

- Gaussian Processes (GP) can be used to model functions
- GP's are distributions over functions described by a mean *m*(*x*) and a positive definite covariance function *k*(*x,x'*)

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

- Finite set of random variables is following a multivariate Gaussian distribution:

$$f(X) \sim \mathcal{N}(m(X), k(X, X))$$

# Probabilistic modelling with GPR

Observed data:

$$X = [x_1, \ldots, x_N]^T \quad \boldsymbol{y} = [y_1, \ldots, y_N]^T$$

Predict at $x_*$, condition the distribution on observed samples $\{X, \boldsymbol{y}\}$:

$$y_* \sim \mathcal{N}(\mu(x_*|X, \boldsymbol{y}), \sigma^2(x_*|X, \boldsymbol{y}))$$

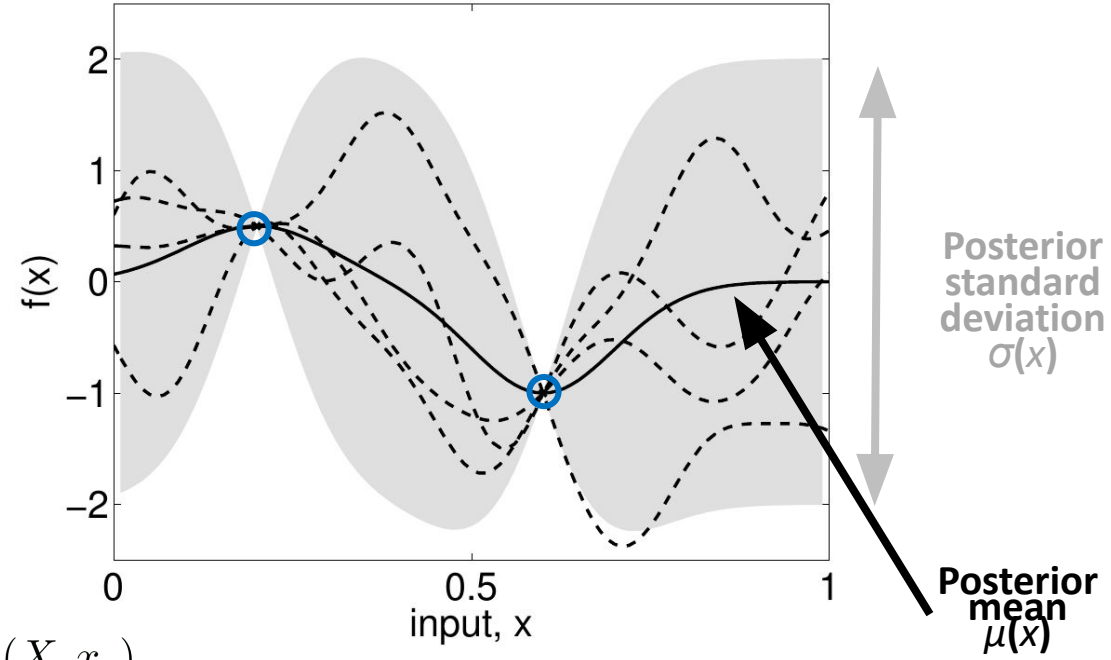
Posterior mean and variance:

$$\mu(x_*|X, \boldsymbol{y}) = K(x_*, X)[K(X, X) + \mathrm{diag}(\sigma_\varepsilon^2)]^{-1}\boldsymbol{y}$$
$$\sigma^2(x_*|X) = k(x_*, x_*) - K(x_*, X)[K(X, X) + \mathrm{diag}(\sigma_\varepsilon^2)]^{-1}K(X, x_*)$$

Log marginal likelihood:

$$\log p(\hat{\boldsymbol{y}}|\hat{X}) = -\tfrac{1}{2}\hat{\boldsymbol{y}}^T[K(\hat{X}, \hat{X}) + \mathrm{diag}(\sigma_\varepsilon^2)]^{-1}\hat{\boldsymbol{y}} - \tfrac{1}{2}\log|K(\hat{X}, \hat{X}) + \mathrm{diag}(\sigma_\varepsilon^2)| - \tfrac{n}{2}\log 2\pi$$

**Posterior distribution**




C.E. Rasmussen et al.,
*Gaussian Processes for Machine Learning* (2008)

# Acquisition rule

Next sampling location obtained by minimizing so-called acquisition function, examples:

**(Exploratory) lower confidence bound**

$$\alpha_{\mathrm{LCB}(\mathbf{x})} = \mu(\mathbf{x}) - \kappa\sigma(\mathbf{x})$$

**Improvement**

$$I(\mathbf{x}) = \max(y_* - Y, 0)$$

**Expected Improvement**

$$\begin{aligned}
\alpha_{\mathrm{EI}}(\mathbf{x}) &= \mathbb{E}_{y\sim\mathcal{N}(\mu,\sigma)}\left[I(\mathbf{x})\right] \\
&= \ldots \\
\alpha_{\mathrm{EI}}(\mathbf{x}) &= (y_* - \mu(\mathbf{x}))\Phi\left(\gamma_*(\mathbf{x})\right) + \sigma(\mathbf{x})\phi\left(\gamma_*(\mathbf{x})\right) \\
\text{with} \quad \gamma_*(\mathbf{x}) &= \frac{y_* - \mu(\mathbf{x})}{\sigma(\mathbf{x})}
\end{aligned}$$

# Entropy-based acquisition function

**Entropy Search**

$$\alpha_{\mathrm{ES}}(\mathbf{x}) = I(\{\mathbf{x}, y\}; \mathbf{x}^* | \mathcal{D})$$
$$= \mathbb{H}(p(\mathbf{x}^* | \mathcal{D})) - \mathbb{E}[\mathbb{H}(p(\mathbf{x}^* | \{\mathbf{x}, y\} \cup \mathcal{D}))]$$

**x***: Unknown minimum location

*I(x,y)*: Mutual information between two random variables *x,y*

H[p(**x**)]: Entropy

# Entropy-based acquisition function

**Entropy Search**

$$\alpha_{\mathrm{ES}}(\mathbf{x}) = I(\{\mathbf{x}, y\}; \mathbf{x}^* | \mathcal{D})$$
$$= \mathbb{H}(p(\mathbf{x}^* | \mathcal{D})) - \mathbb{E}[\mathbb{H}(p(\mathbf{x}^* | \{\mathbf{x}, y\} \cup \mathcal{D}))]$$

**Predictive entropy search**

$$\alpha_{\mathrm{PES}}(\mathbf{x}) = \mathbb{H}(p(y | \mathcal{D}, \mathbf{x})) - \mathbb{E}[\mathbb{H}(p(y | \mathcal{D}, \mathbf{x}, \mathbf{x}^*))]$$

**x***: Unknown minimum location

*I(x,y)*: Mutual information between two random variables *x,y*

H[p(**x**)]: Entropy

# Entropy-based acquisition function

**Entropy Search**

$$\alpha_{\mathrm{ES}}(\mathbf{x}) = I(\{\mathbf{x}, y\}; \mathbf{x}^* | \mathcal{D})$$
$$= \mathbb{H}(p(\mathbf{x}^* | \mathcal{D})) - \mathbb{E}[\mathbb{H}(p(\mathbf{x}^* | \{\mathbf{x}, y\} \cup \mathcal{D}))]$$

**Predictive entropy search**

$$\alpha_{\mathrm{PES}}(\mathbf{x}) = \mathbb{H}(p(y | \mathcal{D}, \mathbf{x})) - \mathbb{E}[\mathbb{H}(p(y | \mathcal{D}, \mathbf{x}, \mathbf{x}^*))]$$

**x***: Unknown minimum location

*I(x,y)*: Mutual information between two random variables *x,y*

H[p(**x**)]: Entropy

Problem: Some expectation values and entropy integrals are expensive to evaluate for high dimensional systems

p(y|D, **x**)

p(**x***|D_t)

Empirical tests have shown that using y* instead of x* (minimum value instead of minimum location) gives almost similar results → Use p(y*|D) instead of p(x*|D) → Only 1D Integral to calculate

# Entropy-based acquisition function

**Max-value entropy search**

$$\alpha_{\mathrm{MES}}(\mathbf{x}) = \mathbb{H}(p(y|\mathcal{D},\mathbf{x})) - \mathbb{E}[\mathbb{H}(p(y|\mathcal{D},\mathbf{x},y^*))]$$
$$\approx \frac{1}{N}\sum_{y^*\in Y^*}\left[\frac{\gamma_{y^*}(\mathbf{x})\phi(\gamma_{y^*}(\mathbf{x}))}{2\Phi(\gamma_{y^*}(\mathbf{x}))} - \log(\Phi(\gamma_{y^*}(\mathbf{x})))\right]$$

**Multi-task max-value entropy search (MUMBO)**

$$\alpha_{\mathrm{MUMBO}}(\mathbf{x},t) = \mathbb{H}(p(y,t|\mathcal{D},\mathbf{x}) - \mathbb{E}_{g^*}[\mathbb{H}(p(y,t|g^*,\mathcal{D},\mathbf{x})]$$
$$= \frac{1}{N}\sum_{g^*\in G}\left[\rho(\mathbf{x},\mathbf{z})^2\frac{\gamma_{g^*}(\mathbf{x})\phi(\gamma_{g^*}(\mathbf{x}))}{2\Phi(\gamma_{g^*}(\mathbf{x}))} - \log(\Phi(\gamma_{g^*}(\mathbf{x})))\right.$$
$$\left. + \mathbb{E}_{\theta\sim Z_{g^*}(\mathbf{x},\mathbf{z})}\left[\log\left(\Phi\left\{\frac{\gamma_{g^*}(\mathbf{x}) - \rho(\mathbf{x},\mathbf{z})\theta}{\sqrt{1-\rho^2(\mathbf{x},\mathbf{z})}}\right\}\right)\right]\right],$$

**x***: Unknown minimum location

*I(x,y)*: Mutual information between two random variables *x,y*

H[p(**x**)]: Entropy

$$\begin{pmatrix} g \\ y \end{pmatrix} \sim N\left[\begin{pmatrix} \mu_g \\ \mu_f \end{pmatrix}, \begin{pmatrix} \sigma_g^2 & \Sigma \\ \Sigma & \sigma_f^2 + \sigma^2 \end{pmatrix}\right]$$

# Intrinsic model of coregionalization (ICM)

Kernel:

$$k_{\mathrm{ICM}}((\mathbf{x}, t), (\mathbf{x}', t')) = B_{t,t'} k_I(\mathbf{x}, \mathbf{x}')$$

Model:

$$f_{\mathrm{HF}}(\mathbf{x}) = f_1(\mathbf{x}) = b_{11} u_1(\mathbf{x}) + b_{12} u_2(\mathbf{x})$$
$$f_{\mathrm{LF}}(\mathbf{x}) = f_2(\mathbf{x}) = b_{21} u_1(\mathbf{x}) + b_{22} u_2(\mathbf{x})$$

$$f_{\mathrm{HF}}(\mathbf{x}) = f_{\mathrm{LF}}(\mathbf{x}) + u'(\mathbf{x})\,, \text{ with}$$
$$u'(\mathbf{x}) = (b_{21} - b_{11}) u_1(\mathbf{x}) + (b_{22} - b_{12}) u_2(\mathbf{x})$$

# Entropy Search

- Information-theoretic acquisition functions: maximize the information gain (= minimizing the uncertainty) about the global optimum **x*** (or f(**x***) = y*)
- Uncertainty is quantified as differential entropy: Calculate Entropy *H* for the current data and calculate Entropy *H* for current data + new sample (**x**,y).
- argmax_x (alpha) leads to **x** with largest entropy reduction, corresponding to the sample that maximally reduces the uncertainty about our global optimum

$$\alpha_{\mathrm{ES}}(\mathbf{x}) = \mathbb{H}[p(\mathbf{x}_\star \mid \mathcal{D})] - \mathbb{E}_{p(y|\mathcal{D},\mathbf{x})}\left[\mathbb{H}\left[p(\mathbf{x}_\star \mid \mathcal{D} \cup \{\mathbf{x}, y\})\right]\right].$$

Entropy of the minimum for the current posterior

Expectation of the entropy of the minimum for the current posterior, given new sample (**x**,y)

# Entropy Search

$$\alpha_t(x) = I(\{\boldsymbol{x}, y\}; \boldsymbol{x}_* \mid D_t) \quad (1)$$

Entropy Search → $$= H\left(p(\boldsymbol{x}_* \mid D_t)\right) - \boxed{\mathbb{E}}\left[H\left(p(\boldsymbol{x}_* \mid D_t \cup \{\boldsymbol{x}, y\})\right)\right] \quad (2)$$

Predictive Entropy Search → $$= H\left(p(y \mid D_t, \boldsymbol{x})\right) - \boxed{\mathbb{E}}\left[H\left(p(y \mid D_t, \boldsymbol{x}, \boldsymbol{x}_*)\right)\right]. \quad (3)$$

MaxVal Entropy Search → $$H\left(p(y \mid D_t, \boldsymbol{x})\right) - \boxed{\mathbb{E}}\left[H\left(p(y \mid D_t, \boldsymbol{x}, y_*)\right)\right] \quad (4)$$

p(y|D_t, **x**)

p(**x***|D_t)

p(y*|D_t)

# Entropy Search

$$\alpha_t(x) = I(\{\boldsymbol{x}, y\}; \boldsymbol{x}_* \mid D_t) \quad (1)$$

Entropy Search → $$= H\left(p(\boldsymbol{x}_* \mid D_t)\right) - \mathbb{E}\left[H\left(p(\boldsymbol{x}_* \mid D_t \cup \{\boldsymbol{x}, y\})\right)\right] \quad (2)$$

Predictive Entropy Search → $$= H(p(y \mid D_t, \boldsymbol{x})) - \mathbb{E}\left[H(p(y \mid D_t, \boldsymbol{x}, \boldsymbol{x}_*))\right]. \quad (3)$$

MaxVal Entropy Search → $$H(p(y \mid D_t, \boldsymbol{x})) - \mathbb{E}\left[H(p(y \mid D_t, \boldsymbol{x}, y_*))\right] \quad (4)$$

Expensive to approximate

p(y|D_t, **x**)

p(**x***|D_t)

p(y*|D_t)

MaxVal Entropy Search: Empirical results show that using value y* instead of location **x*** is as good as Entropy Search and Predictive Entropy Search

# MaxVal Entropy Search

MaxVal Entropy Search

(1) (2)

$$H(p(y \mid D_t, \boldsymbol{x})) - \boxed{\mathbb{E}}[H(p(y \mid D_t, \boldsymbol{x}, y_*))]$$

$$\approx \frac{1}{K} \sum_{y_* \in Y_*} \left[ \frac{\gamma_{y_*}(\boldsymbol{x})\psi(\gamma_{y_*}(\boldsymbol{x}))}{2\Psi(\gamma_{y_*}(\boldsymbol{x}))} - \log(\Psi(\gamma_{y_*}(\boldsymbol{x}))) \right]$$

Closed forms for the entropy of Gaussian (**first term**) and entropy of truncated Gaussian (conditioned on y* < y, **second term**) exist.

Expectation over p(y_*|D_t, **x**) is only 1D, can be cheaply approximated

# MUMBO

$$\begin{pmatrix} g \\ y \end{pmatrix} \sim N\left[\begin{pmatrix} \mu_g \\ \mu_f \end{pmatrix}, \begin{pmatrix} \sigma_g^2 & \Sigma \\ \Sigma & \sigma_f^2 + \sigma^2 \end{pmatrix}\right]$$

**(1)** **(2)**

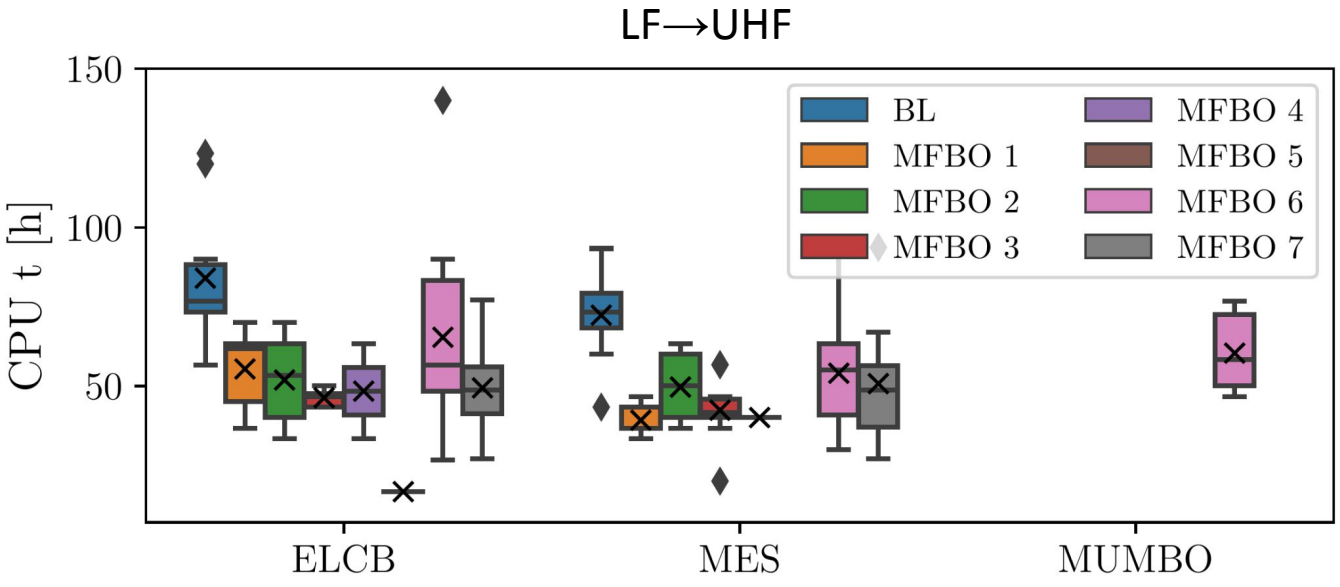
$$\begin{aligned}
\alpha_n^{MUMBO}(\mathbf{x}, \mathbf{z}) =& H(y(\mathbf{x}, \mathbf{z}) \mid D_n) - \boxed{\mathbb{E}_{g^*}}[H(y(\mathbf{x}, \mathbf{z}) \mid g^*, D_n)], \\
=& \frac{1}{N} \sum_{g^* \in G} \Bigg[ \rho(\mathbf{x}, \mathbf{z})^2 \frac{\gamma_{g^*}(\mathbf{x}) \phi(\gamma_{g^*}(\mathbf{x}))}{2\Phi(\gamma_{g^*}(\mathbf{x}))} - \log(\Phi(\gamma_{g^*}(\mathbf{x}))) \\
&+ \mathbb{E}_{\theta \sim Z_{g^*}(\mathbf{x}, \mathbf{z})} \Bigg[ \log \Bigg( \Phi \Bigg\{ \frac{\gamma_{g^*}(\mathbf{x}) - \rho(\mathbf{x}, \mathbf{z})\theta}{\sqrt{1 - \rho^2(\mathbf{x}, \mathbf{z})}} \Bigg\} \Bigg) \Bigg] \Bigg],
\end{aligned}$$

Closed forms for the entropy of Gaussian (**first term**) exists. Distribution in second term follows an extended-skew Gaussian and has no closed form. Entropy for **second term** does not have closed form.

Expectation over p(y_*|D_t, **x**) is only 1D, can be cheaply approximated

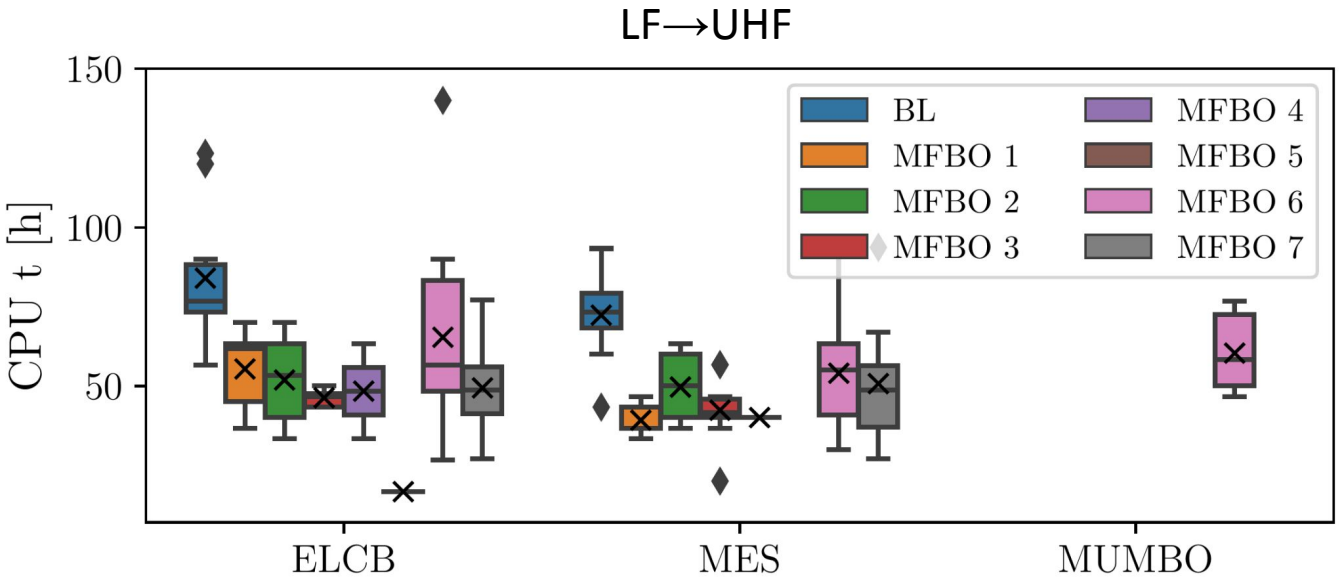
# Prescreening multi-task approaches

- Approaches 1, 3 and 6 still with largest cost reduction
- Approach 4 & 5 tend to fail finding global minimum
- ELCB and MES very similar, MUMBO more expensive



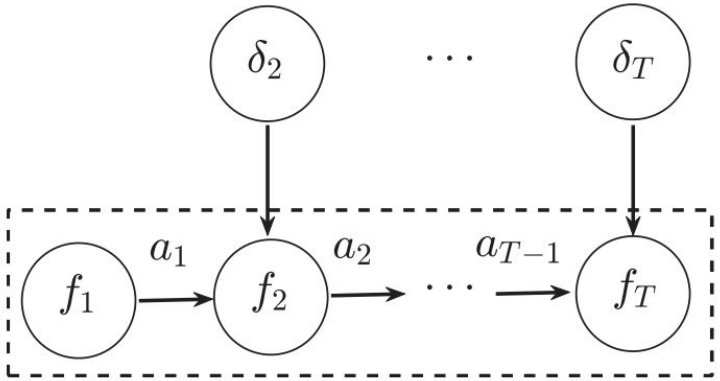
# Prescreening multi-task approaches

- Approaches 1 & 3 with largest cost reduction
- Approach 4 & 5 tend to fail finding global minimum
- ELCB and MES very similar, MUMBO more expensive



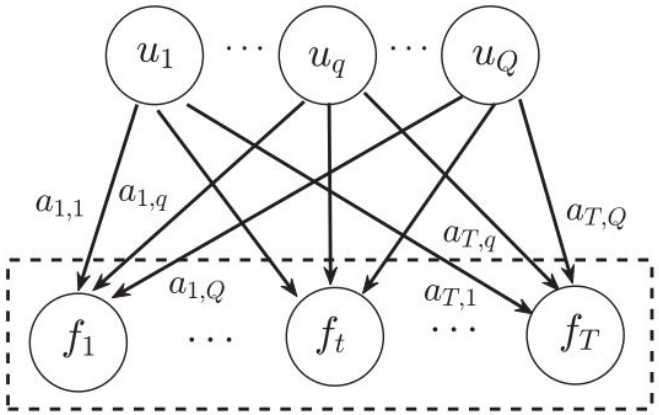

**Conclusion:** Continue calculations with real simulators for approaches 1, 3 &6

# Autoregressive scheme (Asymmetric)




$$f_t(\mathbf{x}) = a_{t-1} f_{t-1}(\mathbf{x}) + \delta_t(\mathbf{x}), \quad 2 \leq t \leq T$$

# LMC/ICM (Symmetric)




$$f_1(\mathbf{x}) = a_{11} u_1(\mathbf{x}) + a_{12} u_2(\mathbf{x}),$$
$$f_2(\mathbf{x}) = a_{21} u_1(\mathbf{x}) + a_{22} u_2(\mathbf{x}),$$

$$\mathrm{Cov}[f_d(\mathbf{x}), f_{d'}(\mathbf{x}')] = B_{d,d'} \kappa(\mathbf{x}, \mathbf{x}')$$



Fig. from Liu et al (2017): Remarks on multi-output Gaussian process regression