

Human-In-The-Loop Machine Learning for health

Julien Martinelli



Bordeaux Population Health - Biostatistics seminar

February 2nd, 2023

Background

- PhD - Inria Saclay & Institut Curie - supervised by François Fages and Annabelle Ballesta (defended february 2022)
 - ▶ **On learning mechanistic models from time series data with applications to personalized chronotherapies**
 - ▶ Designed tools to learn chemical reaction networks (ODEs) from time series data
 - ▶ Mechanistic model of circadian clock & PK-PD of irinotecan, an anticancerous drug
- Prior to that, formation in mathematics, specifically data science & probabilities
- Since february 2022, postdoc at Aalto University, Helsinki
 - ▶ Probabilistic Machine Learning team
 - ▶ **Human-In-The-Loop Machine Learning for drug design**

Modern AI

- Amazing results in classification, regression, generation
- Successfully took the human out of the loop

Price to pay:

- Tremendous amounts of data (ChatGPT, DALL-E...)
- Well-defined task to solve



A painting of a fox sitting in a field at sunrise in the style of Claude Monet

Write a Python function that takes in a list of triangles specified by their 2D vertex coordinates and a per-triangle color and outputs an SVG file that renders the triangles flat shaded in their respective colors.

Sure! Here is a Python function that should do the job:

```
def triangles_to_svg(triangles, colors):
    # Start the SVG document
    svg = '<svg width="100%" height="100%" viewBox="0 0 400 400" '
    xmlns="http://www.w3.org/2000/svg">'

    # Iterate over the triangles and colors
    for triangle, color in zip(triangles, colors):
        # Create a path for the triangle
        points = " ".join(["{x},{y}" for (x, y) in triangle])
        path = f'<path d="M {points} Z" fill="{color}">'
        # Add the path to the SVG document
        svg += path

    # End the SVG document
    svg += '</svg>'

    return svg
```

Modern AI

- Amazing results in classification, regression, generation
- Successfully took the human out of the loop

Price to pay:

- Tremendous amounts of data (ChatGPT, DALL-E...)
- Well-defined task to solve



A painting of a fox sitting in a field at sunrise in the style of Claude Monet

Write a Python function that takes in a list of triangles specified by their 2D vertex coordinates and a per-triangle color and outputs an SVG file that renders the triangles flat shaded in their respective colors.

Sure! Here is a Python function that should do the job:

```
def triangles_to_svg(triangles, colors):
    # Start the SVG document
    svg = '<svg width="100%" height="100%" viewBox="0 0 400 400" '
    xmlns="http://www.w3.org/2000/svg">'

    # Iterate over the triangles and colors
    for triangle, color in zip(triangles, colors):
        # Create a path for the triangle
        points = " ".join([f'{"x":{x} {"y":{y}}}' for (x, y) in triangle])
        path = f'{svg}path d="M {points} Z" fill="{color}">'
        # Add the path to the SVG document
        svg += path

    # End the SVG document
    svg += '</svg>'

    return svg
```

These requirements may not hold in health sciences

Human-In-The-Loop Machine Learning

Field of research concerned with finding ways to **elicitate** and **integrate** expert feedback into ML algorithms

Human-In-The-Loop Machine Learning

Field of research concerned with finding ways to **elicitate** and **integrate** expert feedback into ML algorithms

① Knowledge elicitation

- ▶ What is the most *informative* way to query an expert about a precise topic?
- ▶ Sequential querying strategies: active learning, bayesian experimental design
- ▶ Information theory

Human-In-The-Loop Machine Learning

Field of research concerned with finding ways to **elicitate** and **integrate** expert feedback into ML algorithms

① Knowledge elicitation

- ▶ What is the most *informative* way to query an expert about a precise topic?
- ▶ Sequential querying strategies: active learning, bayesian experimental design
- ▶ Information theory

② Probabilistic expert feedback observation model of...

- ▶ Binary response
- ▶ Human latent utility function
- ▶ Interactions between multiple experts

Human-In-The-Loop Machine Learning

Field of research concerned with finding ways to **elicitate** and **integrate** expert feedback into ML algorithms

① Knowledge elicitation

- ▶ What is the most *informative* way to query an expert about a precise topic?
- ▶ Sequential querying strategies: active learning, bayesian experimental design
- ▶ Information theory

② Probabilistic expert feedback observation model of...

- ▶ Binary response
- ▶ Human latent utility function
- ▶ Interactions between multiple experts

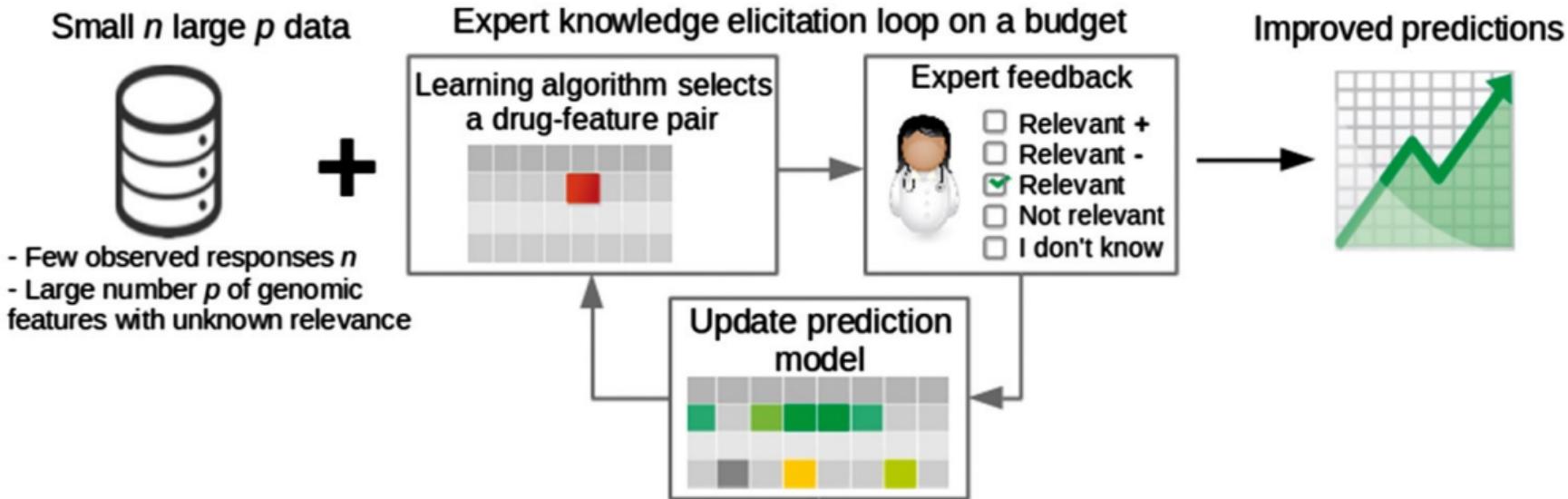
Highly beneficial in the **small data** regime.

Today's talk

- 1 Improving genomics-based predictions for precision medicine through active elicitation of expert knowledge
- 2 Towards a safe integration of expert feedback in Bayesian Optimization
- 3 Human-In-The-Loop Bayesian Optimization for *de novo* drug design

Improving genomics-based predictions for precision medicine through active elicitation of expert knowledge

After Sundin *et al.*, *Bioinformatics*, 2018



Predicting quantitative traits based on genomic features using sparse bayesian linear regression

Input: genomic features $X \in \mathbb{R}^{N \times M}$; output: traits $Y \in \mathbb{R}^{N \times D}$, $M \gg N$

$$y_{n,d} \sim \mathcal{N}(w_d^T x_n, \sigma_d^2)$$

Predicting quantitative traits based on genomic features using sparse bayesian linear regression

Input: genomic features $X \in \mathbb{R}^{N \times M}$; output: traits $Y \in \mathbb{R}^{N \times D}$, $M \gg N$

$$y_{n,d} \sim \mathcal{N}(w_d^T x_n, \sigma_d^2)$$

Sparsity taken into account with a **spike-and-slab** prior:

Predicting quantitative traits based on genomic features using sparse bayesian linear regression

Input: genomic features $X \in \mathbb{R}^{N \times M}$; output: traits $Y \in \mathbb{R}^{N \times D}$, $M \gg N$

$$y_{n,d} \sim \mathcal{N}(w_d^T x_n, \sigma_d^2)$$

Sparsity taken into account with a **spike-and-slab** prior:

$$w_{d,m} \sim \gamma_{d,m} \mathcal{N}(0, \tau_{d,m}^2) + (1 - \gamma_{d,m}) \delta_0$$

Predicting quantitative traits based on genomic features using sparse bayesian linear regression

Input: genomic features $X \in \mathbb{R}^{N \times M}$; output: traits $Y \in \mathbb{R}^{N \times D}$, $M \gg N$

$$y_{n,d} \sim \mathcal{N}(w_d^T x_n, \sigma_d^2)$$

Sparsity taken into account with a **spike-and-slab** prior:

$$w_{d,m} \sim \gamma_{d,m} \mathcal{N}(0, \tau_{d,m}^2) + (1 - \gamma_{d,m}) \delta_0$$

Slab



Predicting quantitative traits based on genomic features using sparse bayesian linear regression

Input: genomic features $X \in \mathbb{R}^{N \times M}$; output: traits $Y \in \mathbb{R}^{N \times D}$, $M \gg N$

$$y_{n,d} \sim \mathcal{N}(w_d^T x_n, \sigma_d^2)$$

Sparsity taken into account with a **spike-and-slab** prior:

$$w_{d,m} \sim \gamma_{d,m} \mathcal{N}(0, \tau_{d,m}^2) + (1 - \gamma_{d,m}) \delta_0$$

Slab

↑ Spike at $w_{d,m} = 0$

Predicting quantitative traits based on genomic features using sparse bayesian linear regression

Input: genomic features $X \in \mathbb{R}^{N \times M}$; output: traits $Y \in \mathbb{R}^{N \times D}$, $M \gg N$

$$y_{n,d} \sim \mathcal{N}(w_d^T x_n, \sigma_d^2)$$

Sparsity taken into account with a **spike-and-slab** prior:

Encodes if m is relevant for d

$$\gamma_{d,m} \sim \text{Bernoulli}(\rho_d)$$

$$w_{d,m} \sim \gamma_{d,m} \mathcal{N}(0, \tau_{d,m}^2) + (1 - \gamma_{d,m}) \delta_0$$

Slab

Spike at $w_{d,m} = 0$

Predicting quantitative traits based on genomic features using sparse bayesian linear regression

Input: genomic features $X \in \mathbb{R}^{N \times M}$; output: traits $Y \in \mathbb{R}^{N \times D}$, $M \gg N$

$$y_{n,d} \sim \mathcal{N}(w_d^T x_n, \sigma_d^2)$$

Sparsity taken into account with a **spike-and-slab** prior:

Encodes if m is relevant for d

$$\gamma_{d,m} \sim \text{Bernoulli}(\rho_d)$$

$$w_{d,m} \sim \begin{cases} \gamma_{d,m} & \text{Slab} \\ \mathcal{N}(0, \tau_{d,m}^2) + (1 - \gamma_{d,m}) \delta_0 & \text{Spike at } w_{d,m} = 0 \end{cases}$$

Hyperpriors:

$$\sigma_d^{-2} \sim \text{Gamma}(\alpha_\sigma, \beta_\sigma)$$

$$\rho_d \sim \text{Beta}(\alpha_p, \beta_p)$$

$$\tau_{d,m} \sim \text{Log-N}(\mu, \omega^2)$$

Posterior distribution of the parameters

$$\theta = (w, \gamma, \rho, \tau^2, \sigma^2) \quad w \in \mathbb{R}^{M \times D}, \gamma \in \mathbb{R}^{M \times D}, \rho \in \mathbb{R}^D, \tau \in \mathbb{R}^{M \times D}, \sigma^2 \in \mathbb{R}^D$$

Bayes rules yields:

$$p(\theta|Y, X) = \frac{p(Y|X, w, \sigma^2)p(w|\gamma, \tau^2)p(\gamma|\rho)p(\rho)p(\tau^2)p(\sigma^2)}{p(Y|X)}$$

Predictive distribution:

$$p(\tilde{y}|Y, X, \tilde{x}) = \int p(\tilde{y}|\tilde{x}, w, \sigma^2)p(\theta|Y, X)d\theta$$

Incorporating expert feedback

An expert is asked to provide a binary answer regarding two questions:

- Is feature m relevant for the prediction of trait d ? $f_{d,m}^{\text{rel}} \in \{0, 1\}$
- Does feature m act positively or negatively on trait d ? $f_{d,m}^{\text{dir}} \in \{0, 1\}$

Incorporating expert feedback

An expert is asked to provide a binary answer regarding two questions:

- Is feature m relevant for the prediction of trait d ? $f_{d,m}^{\text{rel}} \in \{0, 1\}$
- Does feature m act positively or negatively on trait d ? $f_{d,m}^{\text{dir}} \in \{0, 1\}$

Recall that:

$$w_{d,m} \sim \gamma_{d,m} \mathcal{N}(0, \tau_{d,m}^2) + (1 - \gamma_{d,m})\delta_0$$

Incorporating expert feedback

An expert is asked to provide a binary answer regarding two questions:

- Is feature m relevant for the prediction of trait d ? $f_{d,m}^{\text{rel}} \in \{0, 1\}$
- Does feature m act positively or negatively on trait d ? $f_{d,m}^{\text{dir}} \in \{0, 1\}$

Recall that:

$$w_{d,m} \sim \gamma_{d,m} \mathcal{N}(0, \tau_{d,m}^2) + (1 - \gamma_{d,m})\delta_0$$

$$f_{d,m}^{\text{rel}} \sim \gamma_{d,m} \text{Bernoulli}(\pi_d^{\text{rel}}) + (1 - \gamma_{d,m})\text{Bernoulli}(1 - \pi_d^{\text{rel}})$$

Incorporating expert feedback

An expert is asked to provide a binary answer regarding two questions:

- Is feature m relevant for the prediction of trait d ? $f_{d,m}^{\text{rel}} \in \{0, 1\}$
- Does feature m act positively or negatively on trait d ? $f_{d,m}^{\text{dir}} \in \{0, 1\}$

Recall that:

$$w_{d,m} \sim \gamma_{d,m} \mathcal{N}(0, \tau_{d,m}^2) + (1 - \gamma_{d,m})\delta_0$$

Encodes if m

is relevant for d

$$f_{d,m}^{\text{rel}} \sim \gamma_{d,m} \text{Bernoulli}(\pi_d^{\text{rel}}) + (1 - \gamma_{d,m})\text{Bernoulli}(1 - \pi_d^{\text{rel}})$$

Incorporating expert feedback

An expert is asked to provide a binary answer regarding two questions:

- Is feature m relevant for the prediction of trait d ? $f_{d,m}^{\text{rel}} \in \{0, 1\}$
- Does feature m act positively or negatively on trait d ? $f_{d,m}^{\text{dir}} \in \{0, 1\}$

Recall that:

$$w_{d,m} \sim \gamma_{d,m} \mathcal{N}(0, \tau_{d,m}^2) + (1 - \gamma_{d,m})\delta_0$$

Encodes if m

is relevant for d

$$f_{d,m}^{\text{rel}} \sim \gamma_{d,m}$$

$$\text{Bernoulli}(\pi_d^{\text{rel}}) + (1 - \gamma_{d,m})\text{Bernoulli}(1 - \pi_d^{\text{rel}})$$

Probability of the expert being correct

Incorporating expert feedback

An expert is asked to provide a binary answer regarding two questions:

- Is feature m relevant for the prediction of trait d ? $f_{d,m}^{\text{rel}} \in \{0, 1\}$
- Does feature m act positively or negatively on trait d ? $f_{d,m}^{\text{dir}} \in \{0, 1\}$

Recall that:

$$w_{d,m} \sim \gamma_{d,m} \mathcal{N}(0, \tau_{d,m}^2) + (1 - \gamma_{d,m})\delta_0$$

Encodes if m

is relevant for d

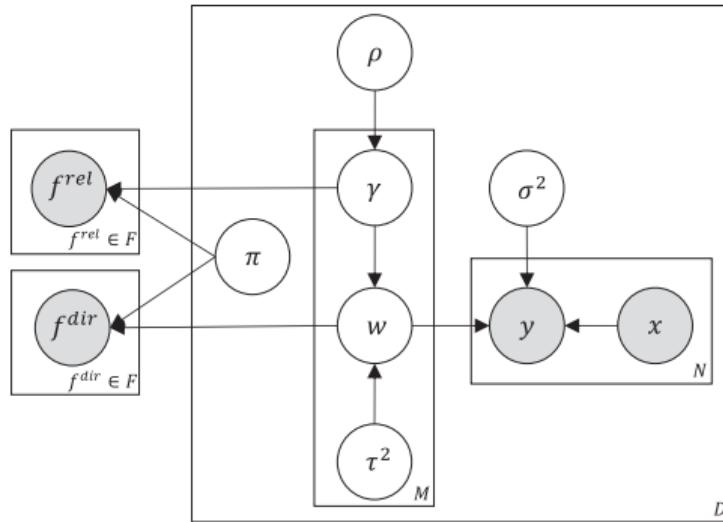
$$f_{d,m}^{\text{rel}} \sim \gamma_{d,m} \text{Bernoulli}(\pi_d^{\text{rel}}) + (1 - \gamma_{d,m})\text{Bernoulli}(1 - \pi_d^{\text{rel}})$$

Probability of the expert being correct

$$f_{d,m}^{\text{dir}} \sim \mathbb{1}_{w_{d,m} > 0} \text{Bernoulli}(\pi_d^{\text{dir}}) + \mathbb{1}_{w_{d,m} < 0} \text{Bernoulli}(1 - \pi_d^{\text{dir}})$$

For simplicity: $\pi_d^{\text{dir}} = \pi_d^{\text{rel}} = \pi_d \sim \text{Beta}(\alpha_\pi, \beta_\pi)$

Final model



$$\theta = (w, \gamma, \rho, \tau^2, \sigma^2, \pi)$$

$$p(\theta|Y, X, F) = \frac{p(Y|X, w, \sigma^2)p(w|\gamma, \tau^2)p(\gamma|\rho)p(\rho)p(\tau^2)p(\sigma^2)}{p(Y|F, X)} \times p(F|\gamma, w, \pi)p(\pi)$$

π controls how strongly the model will change to reflect expert feedback

Experimental design for active expert knowledge elicitation

Experts don't have time to provide feedback about every sample.

Experimental design for active expert knowledge elicitation

Experts don't have time to provide feedback about every sample.

→ Find the most **informative** (trait, feature) candidate pair to show to the expert.

Experimental design for active expert knowledge elicitation

Experts don't have time to provide feedback about every sample.

→ Find the most **informative** (trait, feature) candidate pair to show to the expert.

Informative? Can mean many things 

Here: Informativeness of expert feedback measured by KL divergence between predictive distributions **before** and **after** observing feedback. Let $\mathcal{D}_t = (Y, X, F_t)$,

Experimental design for active expert knowledge elicitation

Experts don't have time to provide feedback about every sample.

→ Find the most **informative** (trait, feature) candidate pair to show to the expert.

Informative? Can mean many things ^_\(^)_/\^

Here: Informativeness of expert feedback measured by KL divergence between predictive distributions **before** and **after** observing feedback. Let $\mathcal{D}_t = (Y, X, F_t)$,

$$u_{n,d,m,t} = \text{KL} \left[p(\tilde{y}_{n,d} | x_n, \mathcal{D}_{t-1}, \tilde{f}_{d,m}^{\text{rel}}, \tilde{f}_{d,m}^{\text{dir}}) \| p(\tilde{y}_{n,d} | x_n, \mathcal{D}_t) \right]$$

Experimental design for active expert knowledge elicitation

Experts don't have time to provide feedback about every sample.

→ Find the most **informative** (trait, feature) candidate pair to show to the expert.

Informative? Can mean many things ^_\(^)_/\^

Here: Informativeness of expert feedback measured by KL divergence between predictive distributions **before** and **after** observing feedback. Let $\mathcal{D}_t = (Y, X, F_t)$,

$$u_{n,d,m,t} = \text{KL} \left[p(\tilde{y}_{n,d} | x_n, \mathcal{D}_{t-1}, \tilde{f}_{d,m}^{\text{rel}}, \tilde{f}_{d,m}^{\text{dir}}) \| p(\tilde{y}_{n,d} | x_n, \mathcal{D}_t) \right]$$

$$(d^*, m^*) = \underset{(d,m) \notin F_{t-1}}{\text{argmax}} \mathbb{E} \left[\tilde{f}_{d,m}^{\text{rel}}, \tilde{f}_{d,m}^{\text{dir}} | \mathcal{D}_{t-1} \right] \left[\sum_{n=1}^N u_{n,d,m,t} \right]$$

↑
predictive distribution of feedbacks

Baselines

- Sequential experimental design
- Targeted sequential experimental design

$$(d^*, m^*) = \operatorname{argmax}_{\substack{(d,m) \notin F_{t-1}}} \mathbb{E}_{\tilde{f}_{d,m}^{\text{rel}}, \tilde{f}_{d,m}^{\text{dir}} | \mathcal{D}_{t-1}} [u_{d,m,t}]$$

with $u_{d,m,t} = \mathbb{KL}[p(\tilde{y}_d | \tilde{x}, \mathcal{D}_{t-1}, \tilde{f}_{d,m}^{\text{rel}}, \tilde{f}_{d,m}^{\text{dir}}) || p(\tilde{y}_d | \tilde{x}, \mathcal{D}_{t-1})]$

→ Focus on improving the prediction **over the current sample \tilde{y}_d** , not globally.

- Random uniform sampling of the next pair not yet queried $(d, m) \notin F_{t-1}$

Metabolite concentration prediction and simulated experts

Predicting concentration of $D = 4$

standard metabolites:

HDL-C, LDL-C, TC, TG

$N = 3918$ individuals, $M = 3107$ SNPs

Metabolite concentration prediction and simulated experts

Predicting concentration of $D = 4$

standard metabolites:

HDL-C, LDL-C, TC, TG

$N = 3918$ individuals, $M = 3107$ SNPs

Simulated expert feedback using GWAS
meta-analysis over 24925 individuals.

- relevant SNPs per output metabolite:
 $p < 2.3 \times 10^{-9} \implies 13, 46, 39, 11$
- irrelevant:
 $p > 0.9 \implies 1010, 859, 620, 628.$
- dir. feedback: regression weights
from meta-analysis

12428 possible queries (3109×4)

Metabolite concentration prediction and simulated experts

Predicting concentration of $D = 4$

standard metabolites:

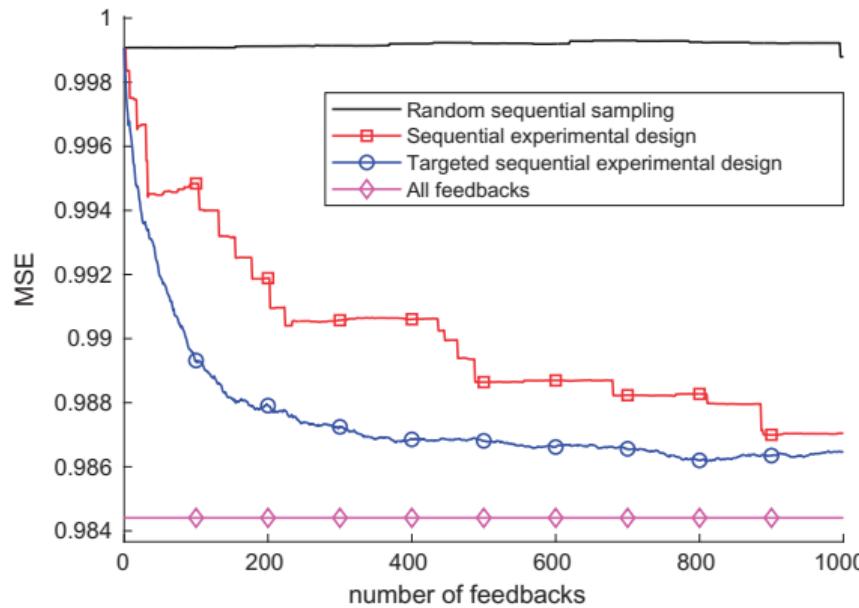
HDL-C, LDL-C, TC, TG

$N = 3918$ individuals, $M = 3107$ SNPs

Simulated expert feedback using GWAS meta-analysis over 24925 individuals.

- relevant SNPs per output metabolite:
 $p < 2.3 \times 10^{-9} \implies 13, 46, 39, 11$
- irrelevant:
 $p > 0.9 \implies 1010, 859, 620, 628.$
- dir. feedback: regression weights from meta-analysis

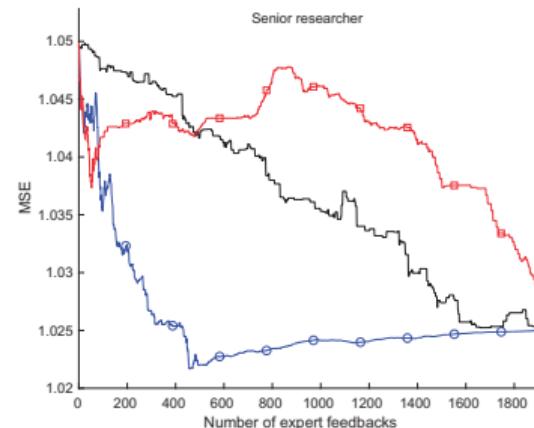
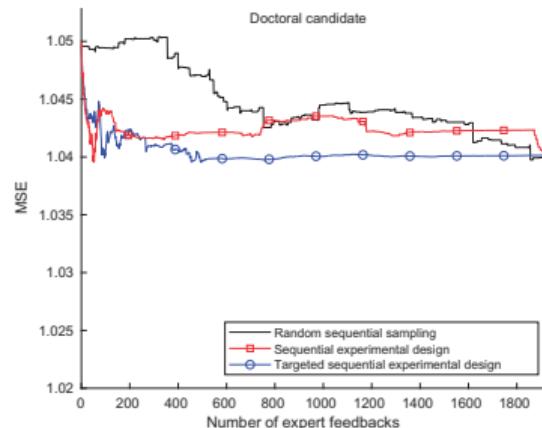
12428 possible queries (3109×4)



Drug sensitivity prediction for multiple myeloma patients

Real expert feedback

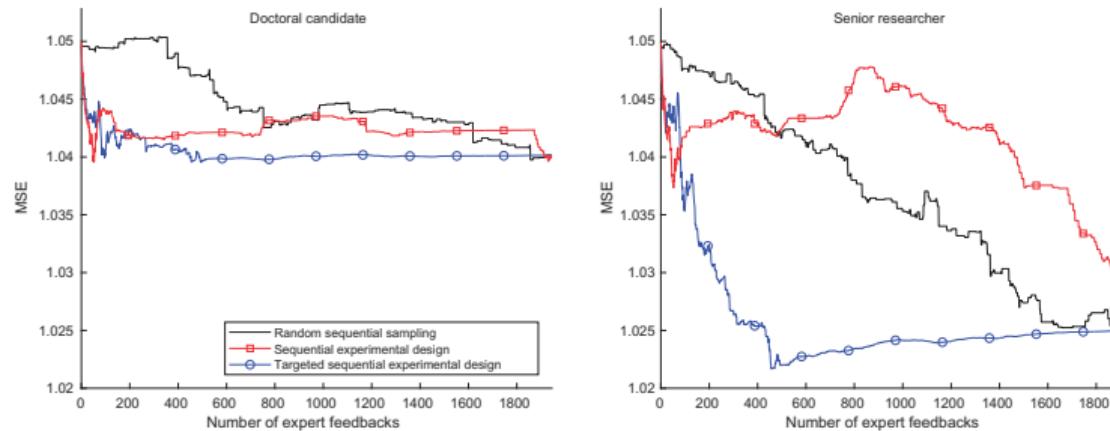
- $N = 44$ patient samples, $M = 2942$ genomic features, $D = 12$ drugs.
- Experts feedback on the relevance of genomic features for drug sensitivity.
- Feedback only collected on 162 gene mutations causally involved in cancer.



Drug sensitivity prediction for multiple myeloma patients

Real expert feedback

- $N = 44$ patient samples, $M = 2942$ genomic features, $D = 12$ drugs.
- Experts feedback on the relevance of genomic features for drug sensitivity.
- Feedback only collected on 162 gene mutations causally involved in cancer.



Nondecreasing MSE w.r.t. number of expert feedbacks? 🤔

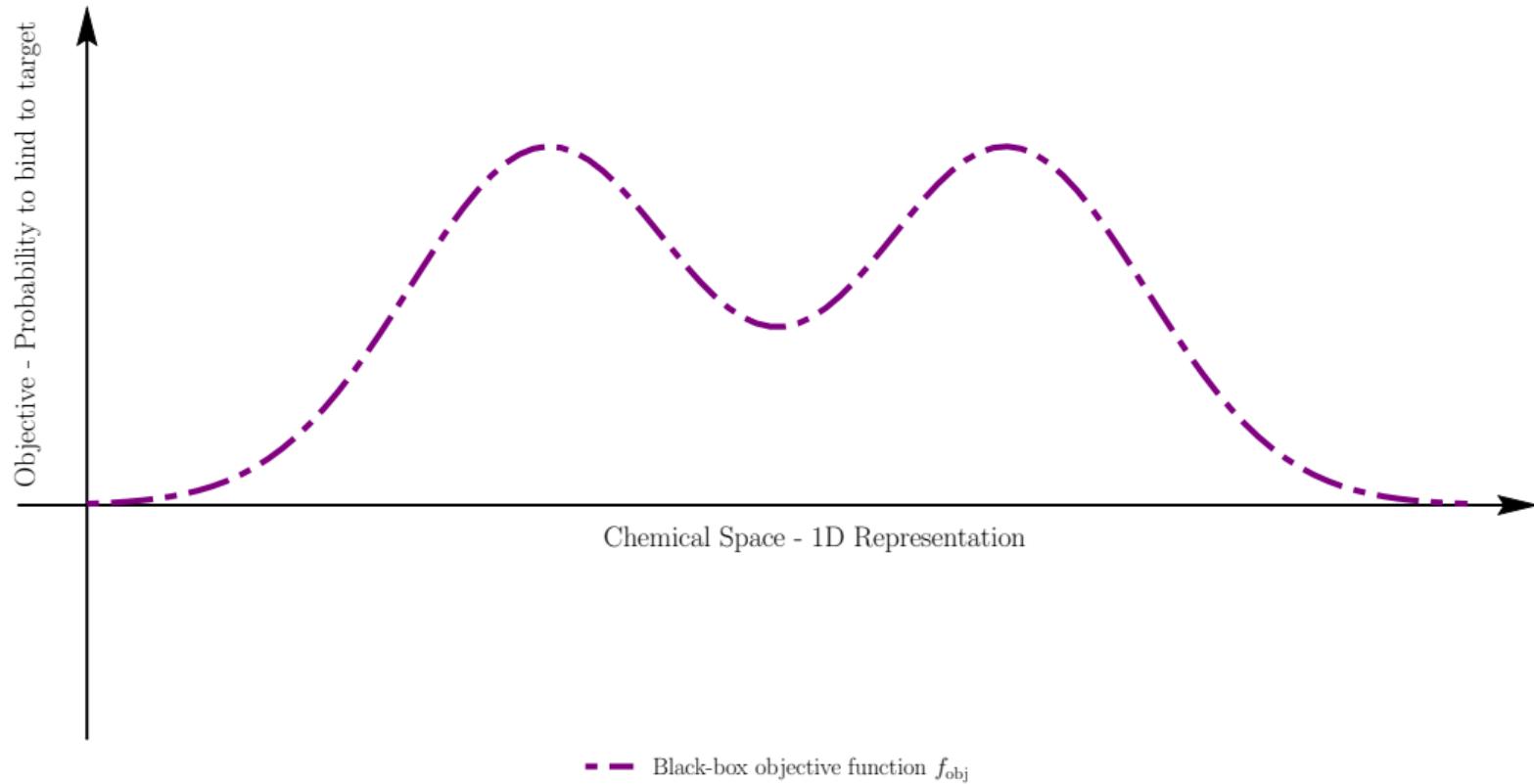
Expert were assumed to be right on 19 out of 20 feedbacks: $\pi \sim \text{Beta}(19, 1)$

Towards a safe integration of expert feedback in Bayesian Optimization

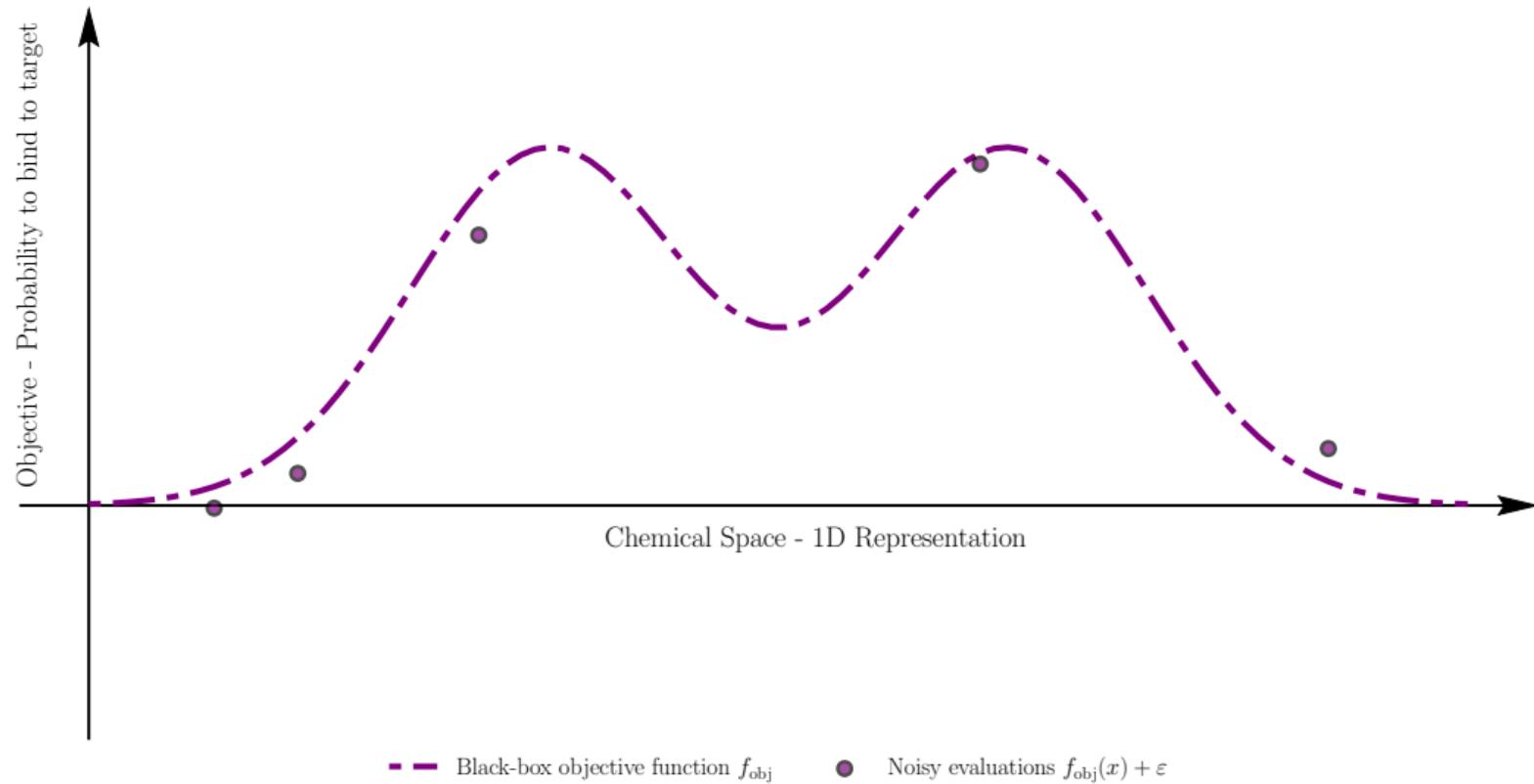
Petrus Mikkola, Julien Martinelli, Louis Filstroff, Samuel Kaski,
accepted for publication at AISTATS2023.

Bayesian Optimization 101

Bayesian Optimization 101

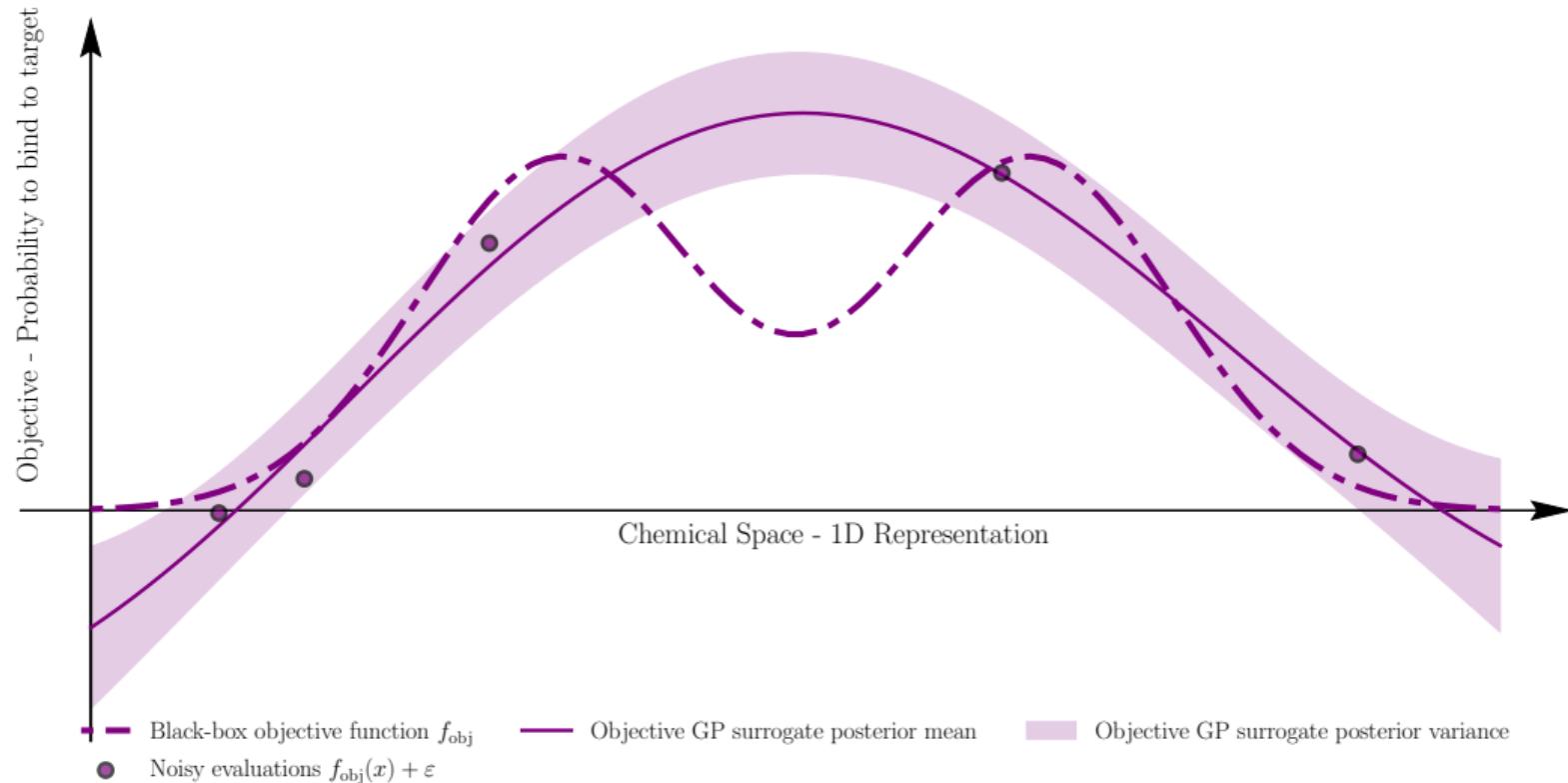


Bayesian Optimization 101



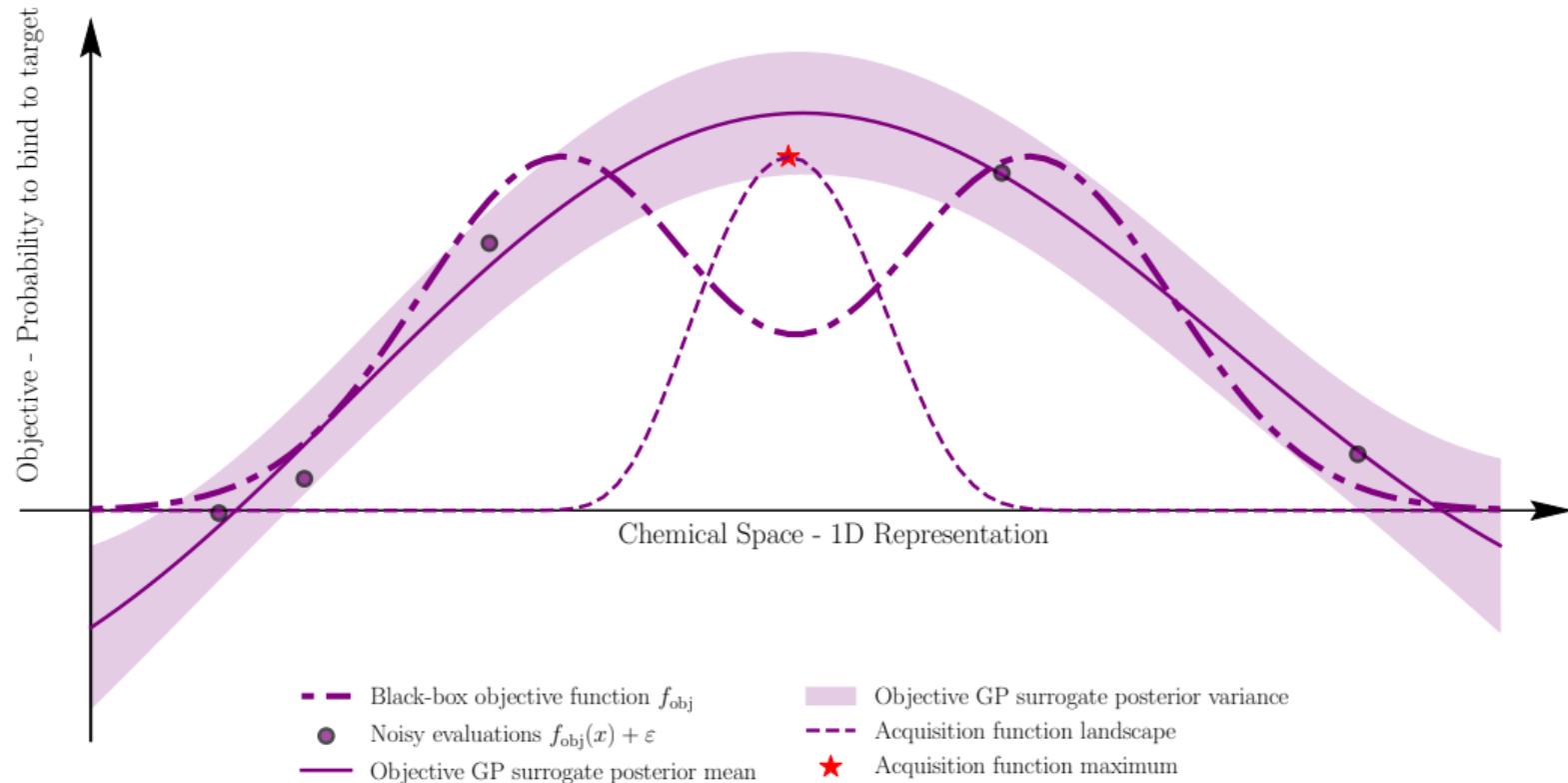
Bayesian Optimization 101

Budget = 20



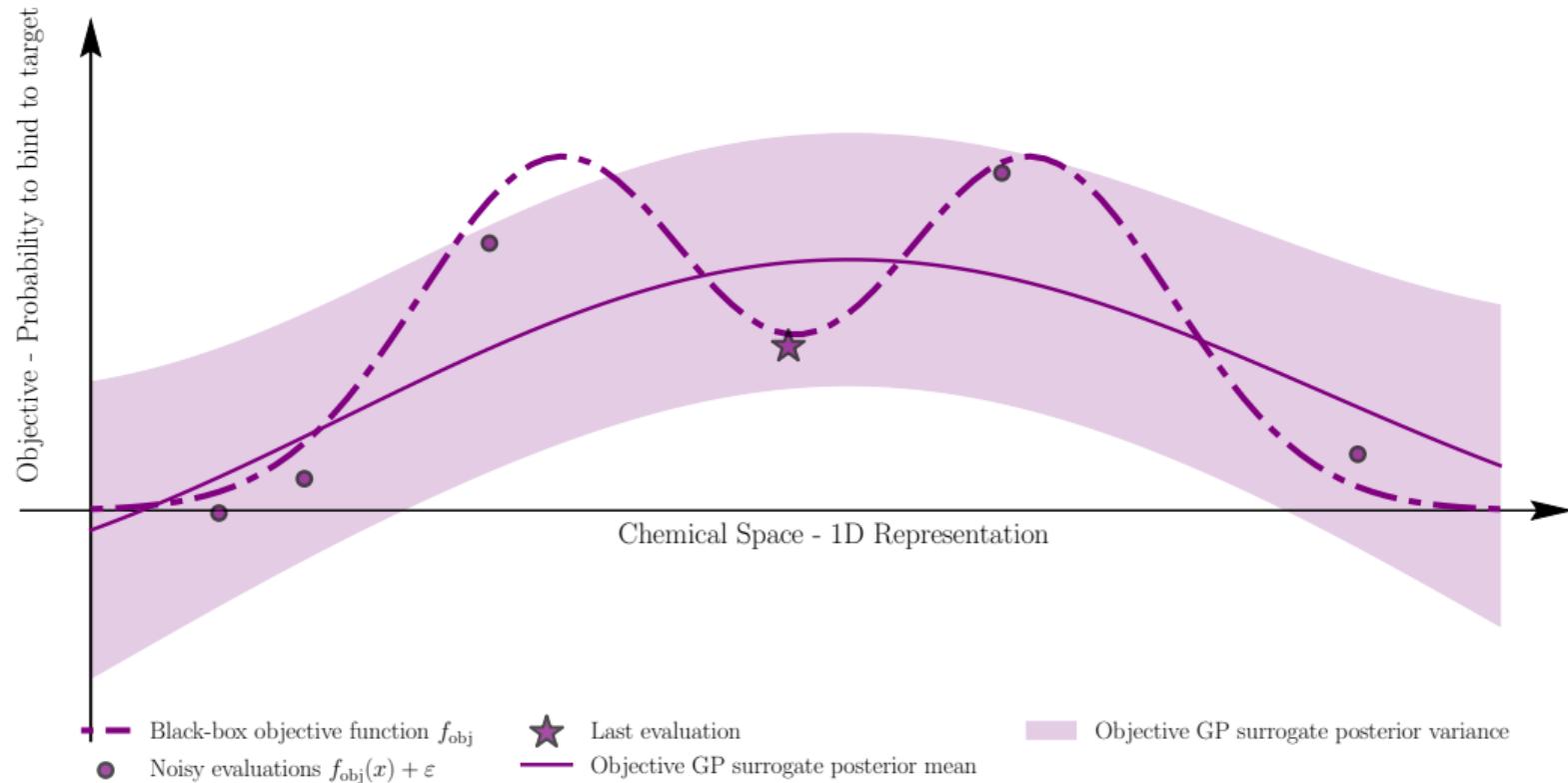
Bayesian Optimization 101

Budget = 20



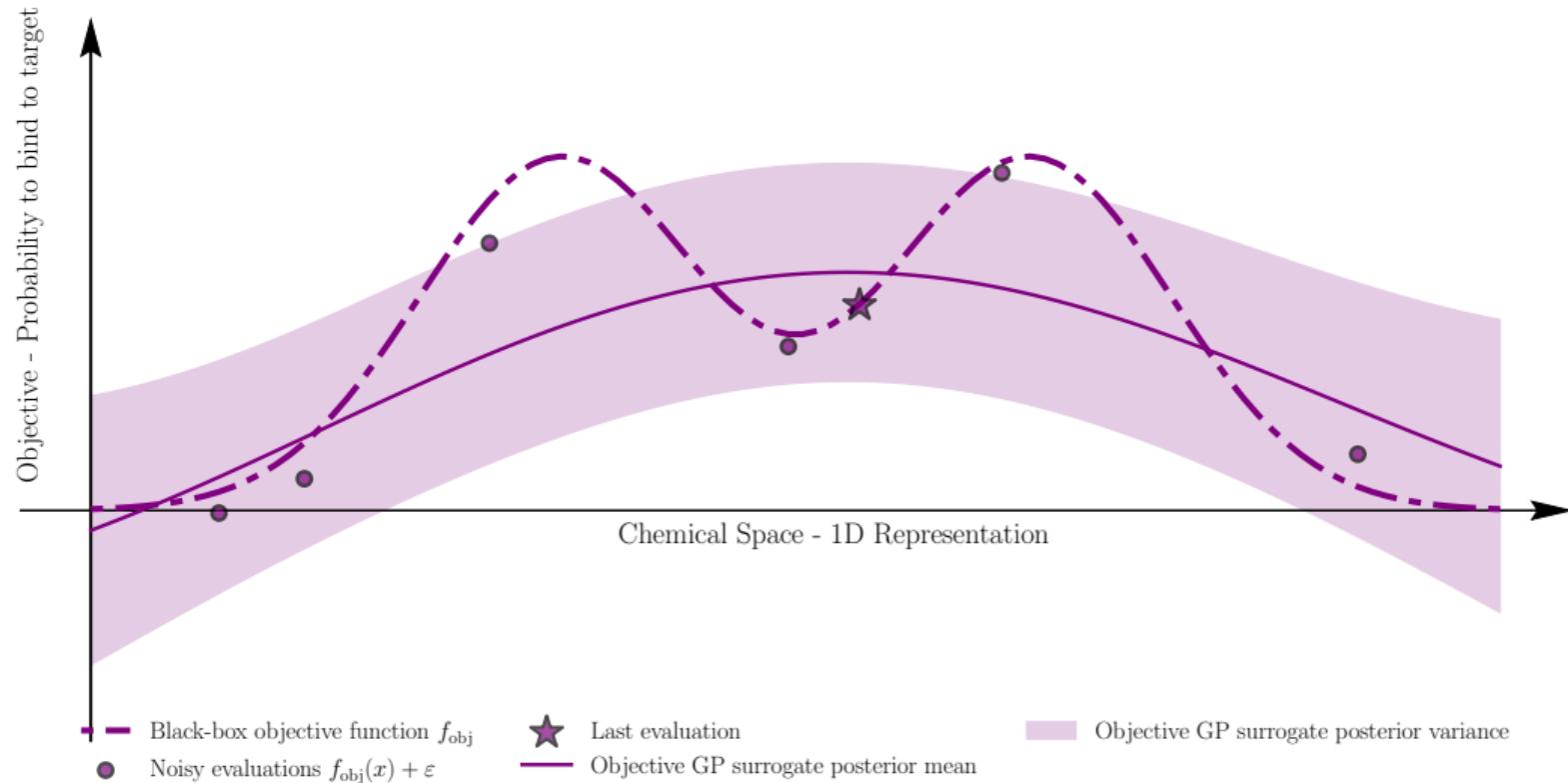
Bayesian Optimization 101

Budget = 19



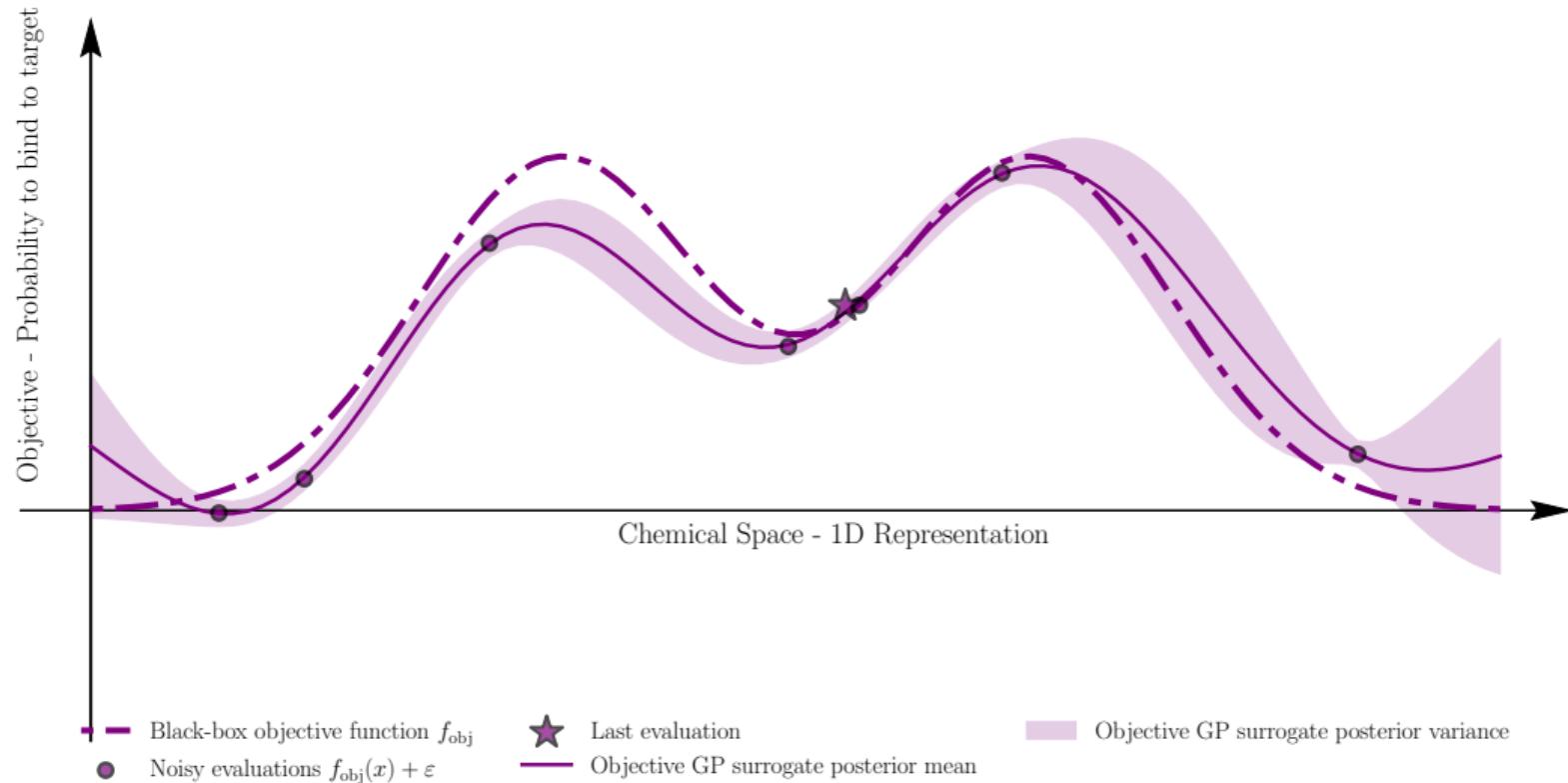
Bayesian Optimization 101

Budget = 18



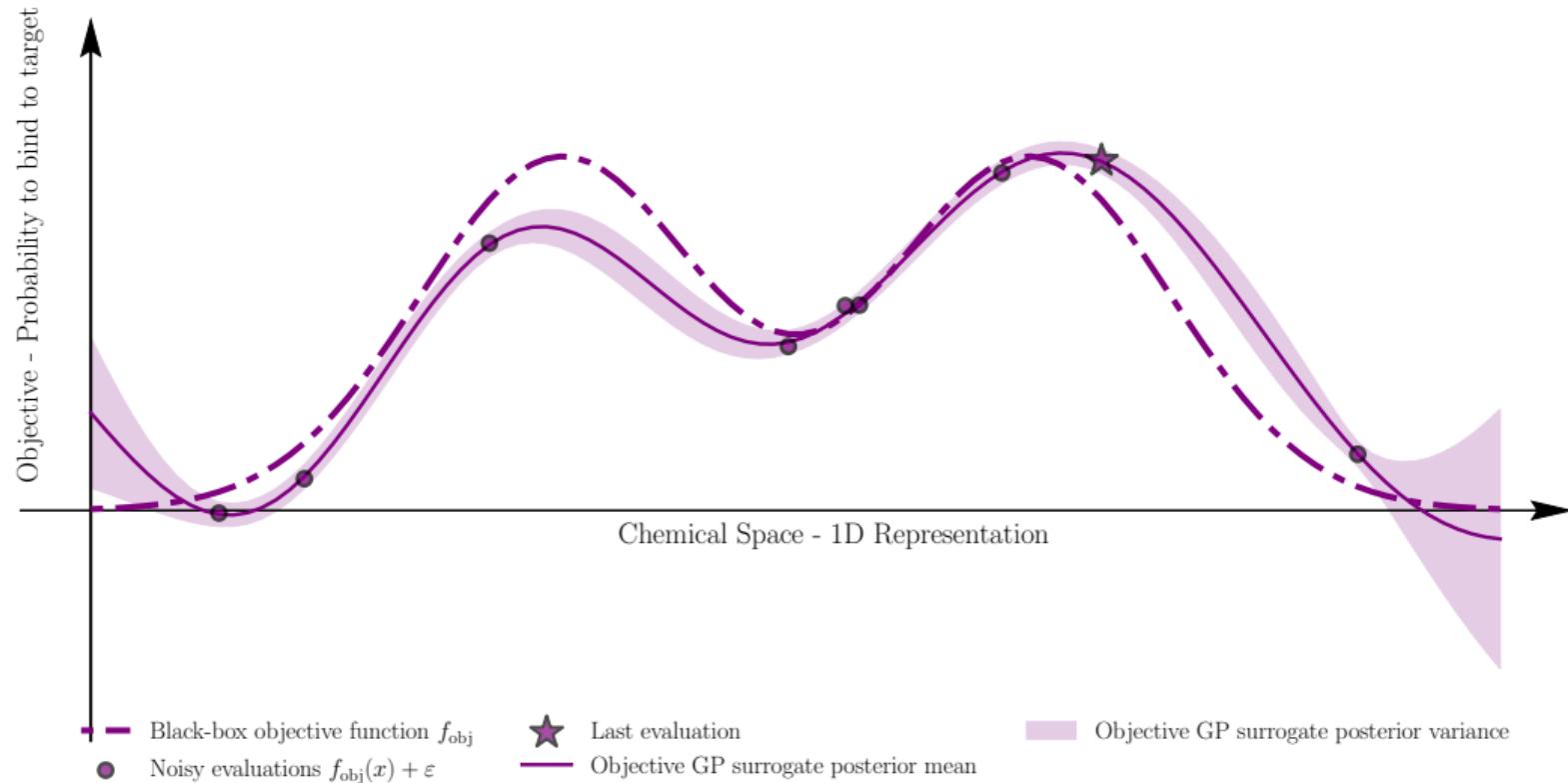
Bayesian Optimization 101

Budget = 17



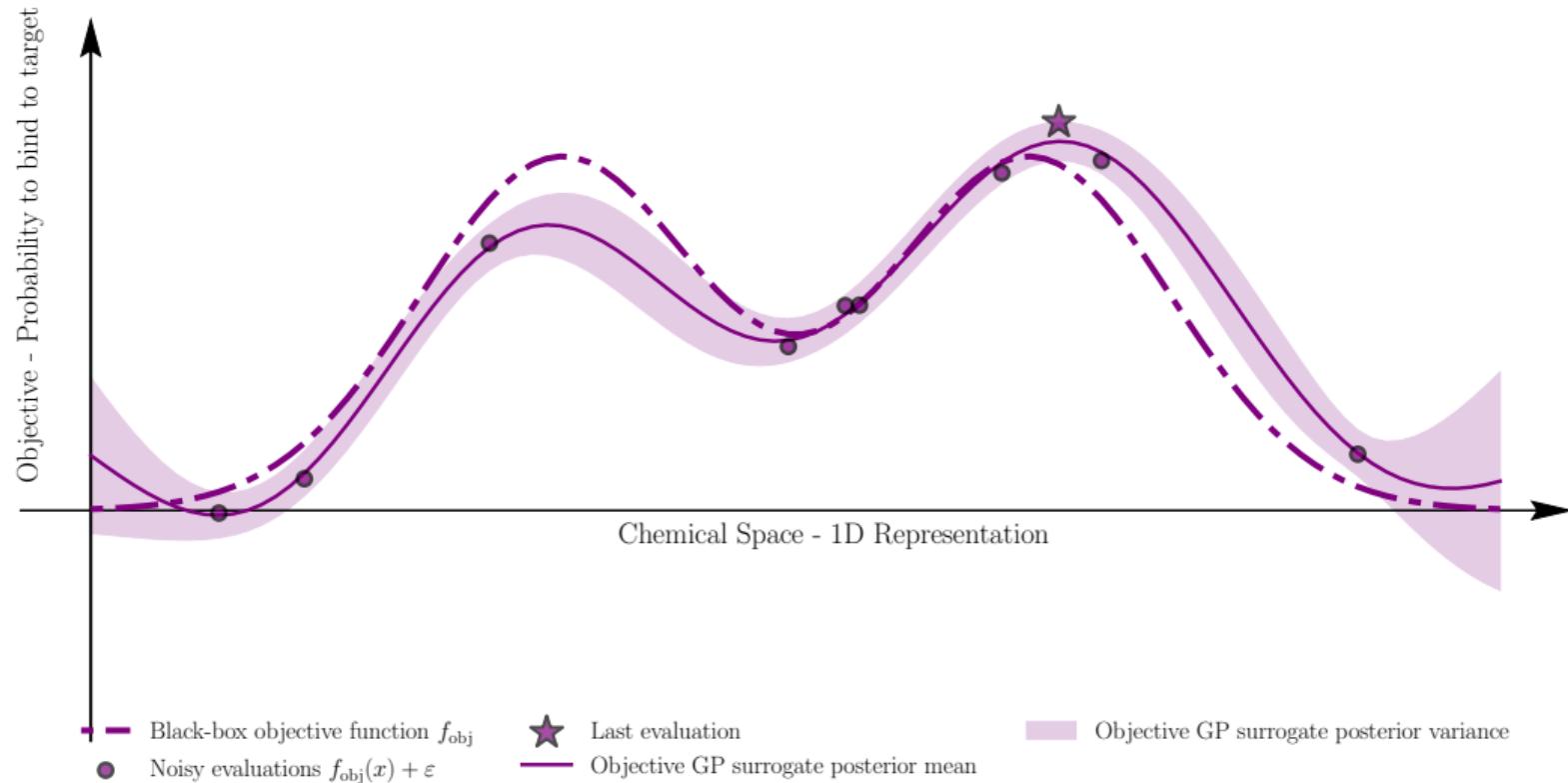
Bayesian Optimization 101

Budget = 16



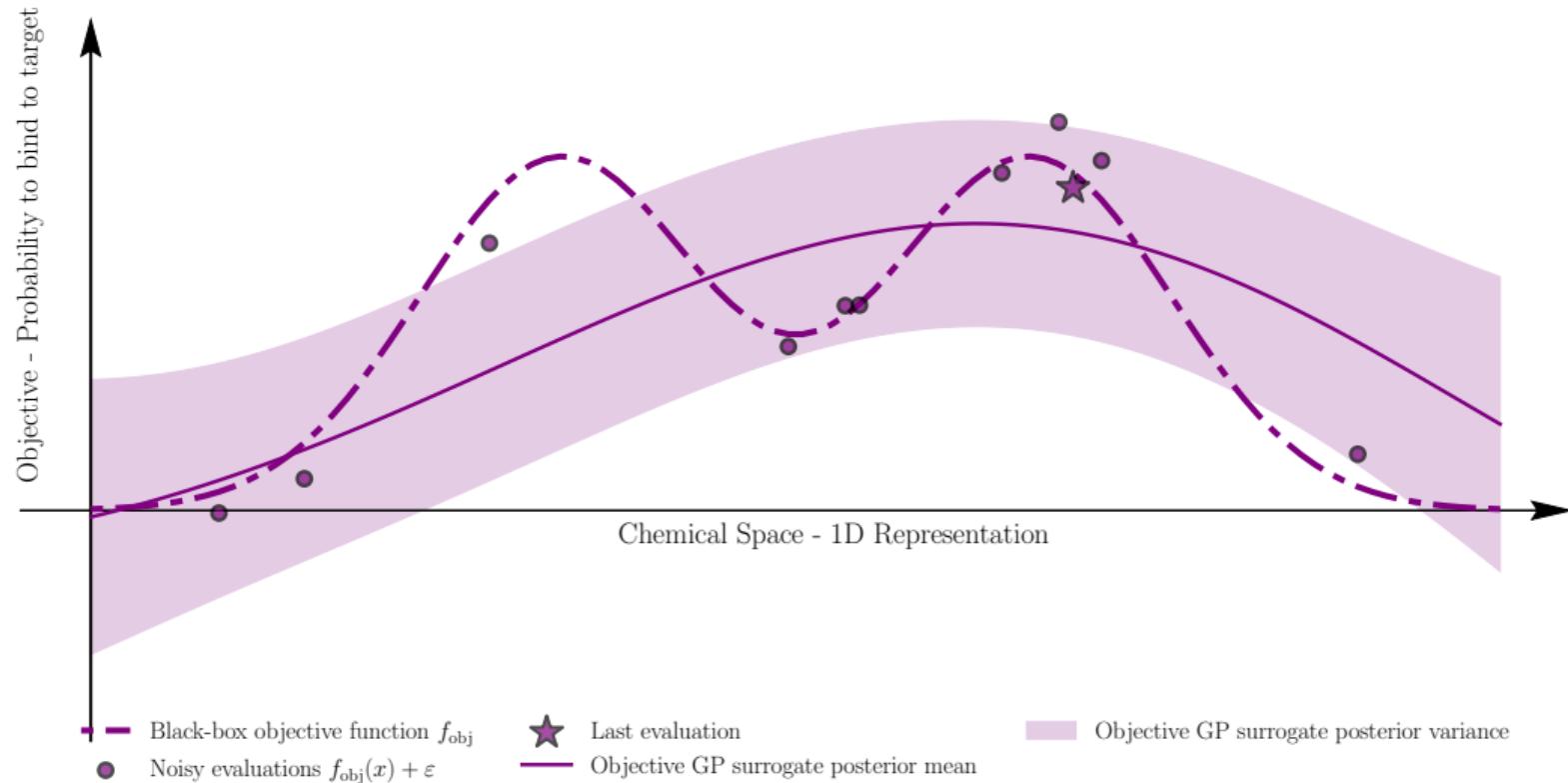
Bayesian Optimization 101

Budget = 15



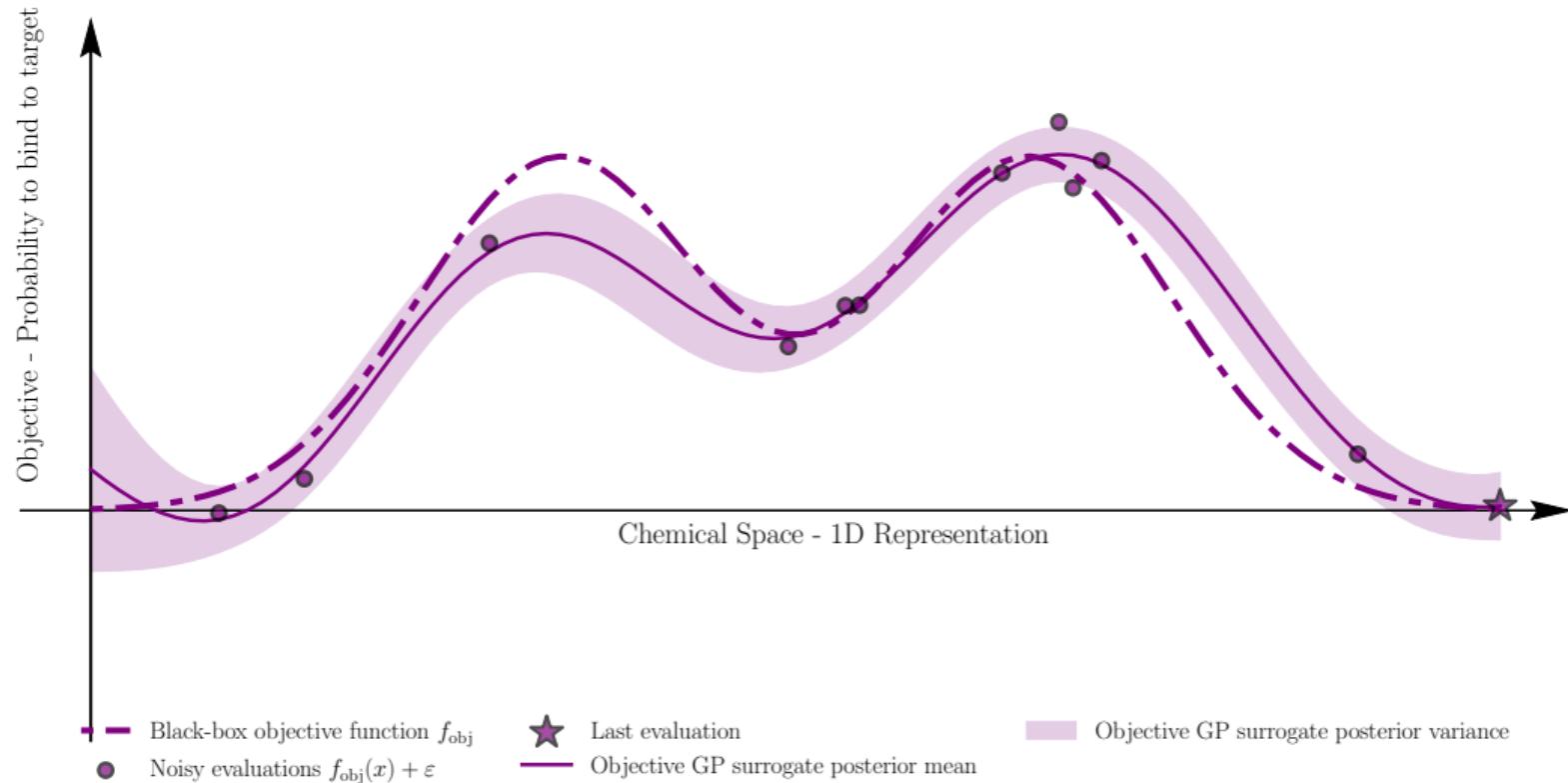
Bayesian Optimization 101

Budget = 14



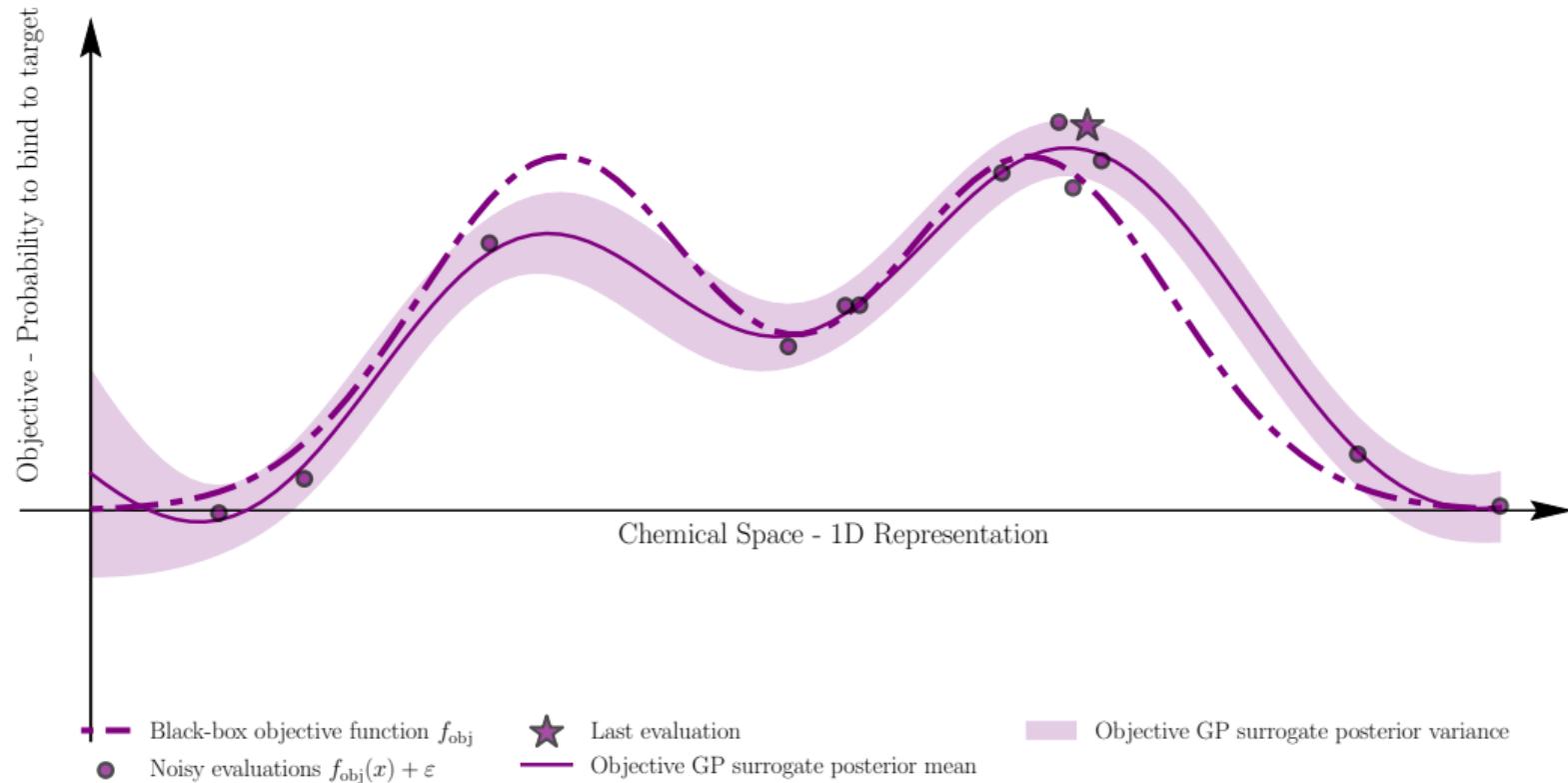
Bayesian Optimization 101

Budget = 13

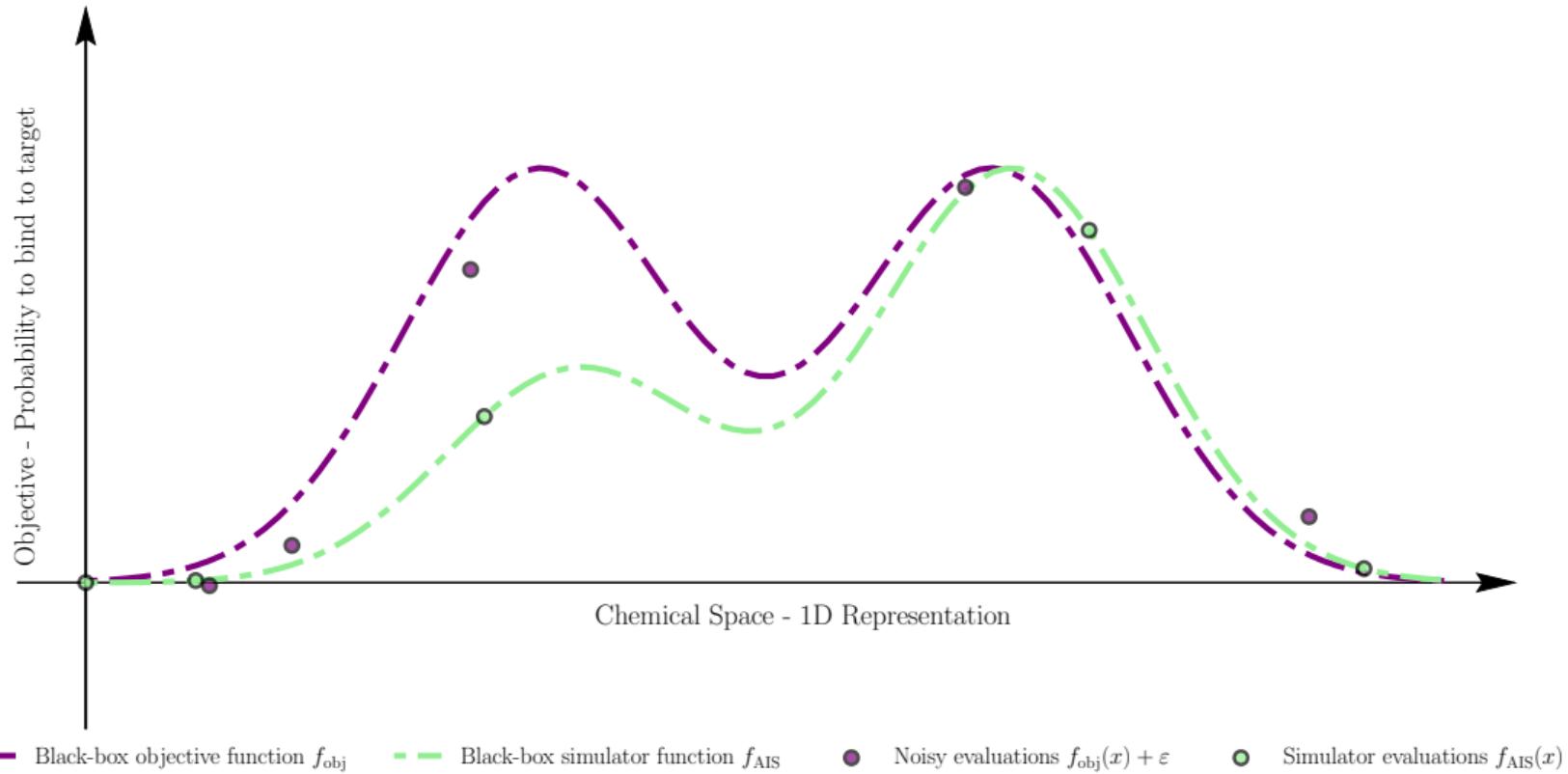


Bayesian Optimization 101

Budget = 12

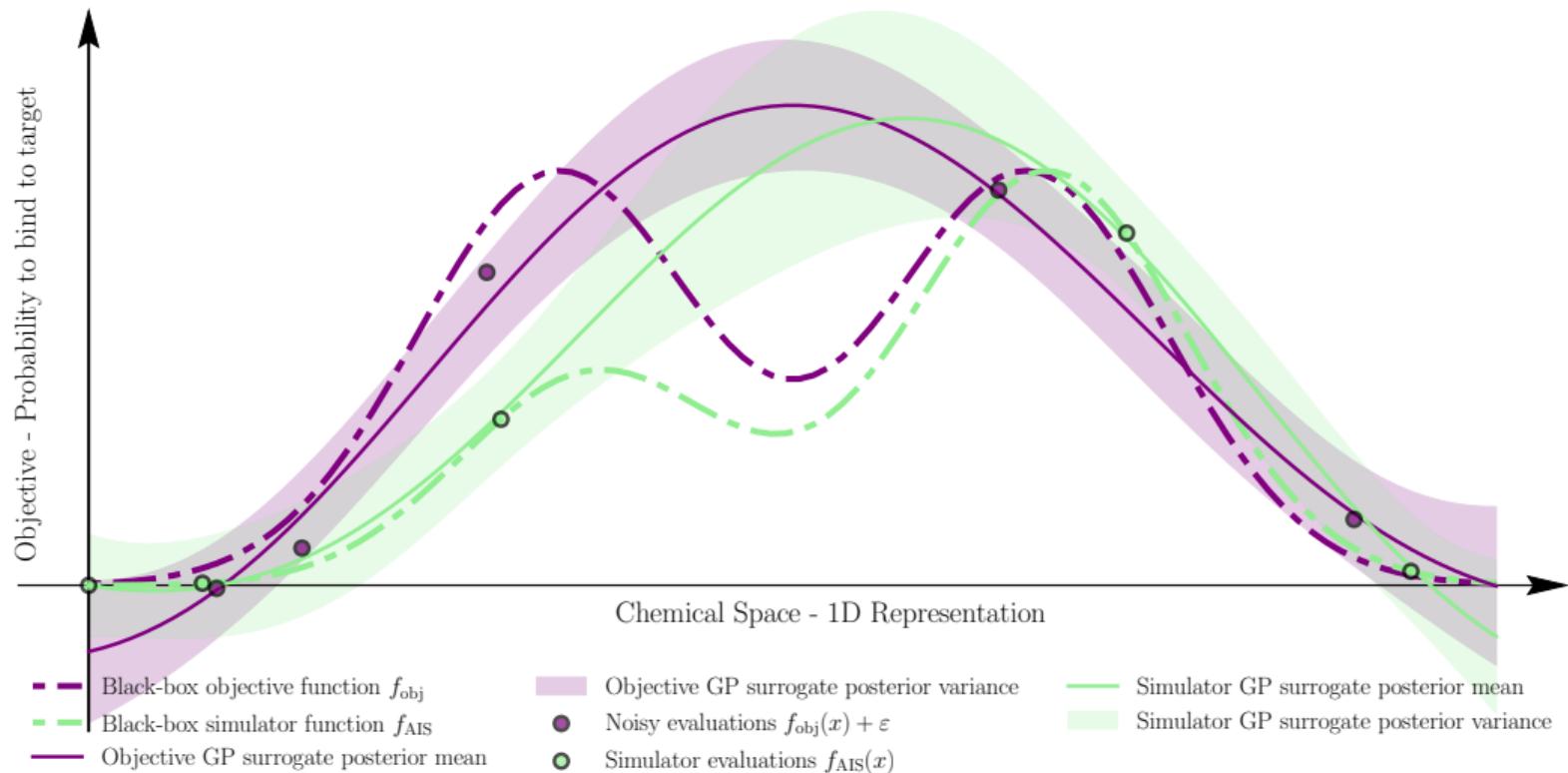


Multi Fidelity Bayesian Optimization 101



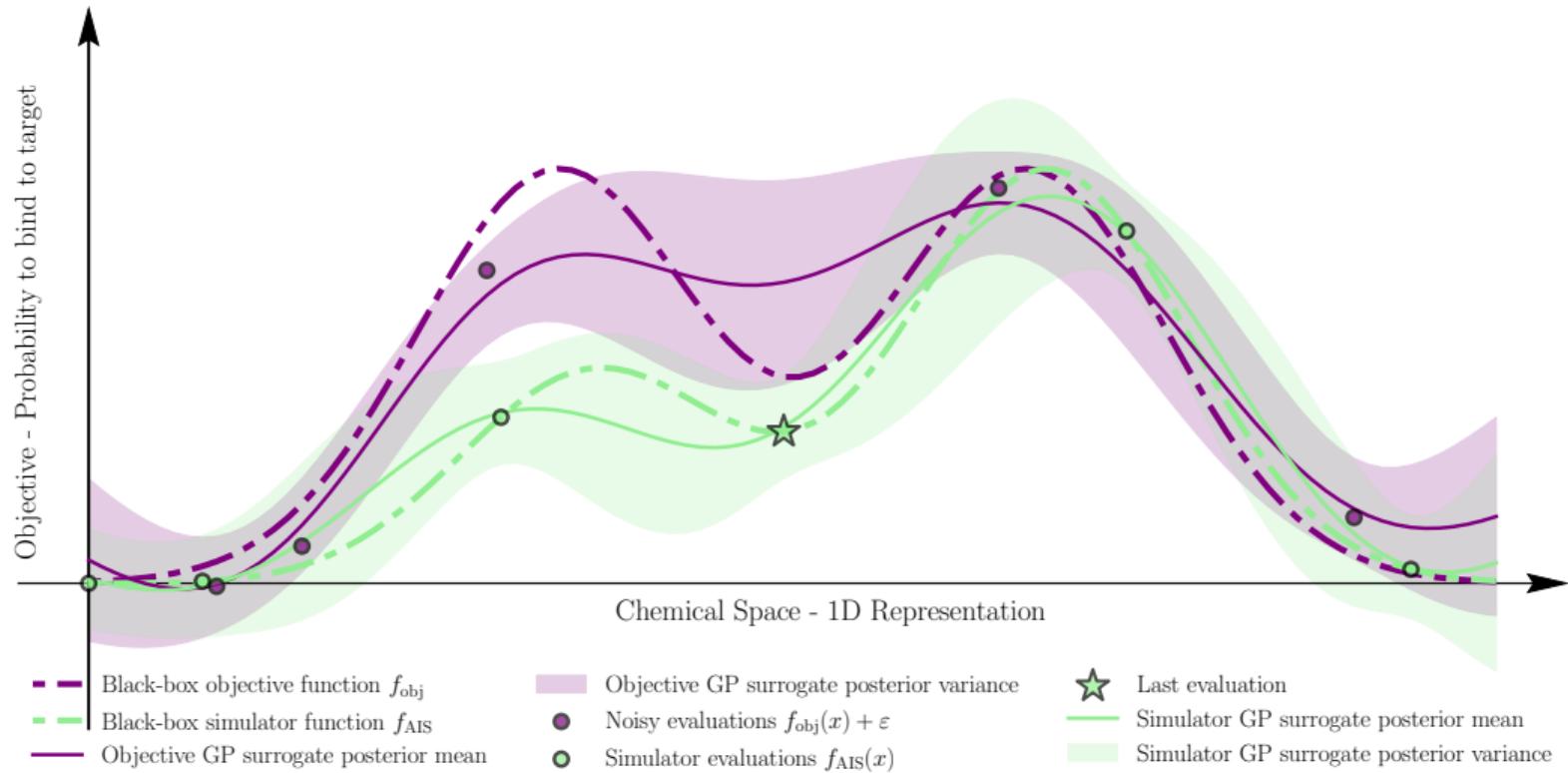
Multi Fidelity Bayesian Optimization 101

Budget = 20



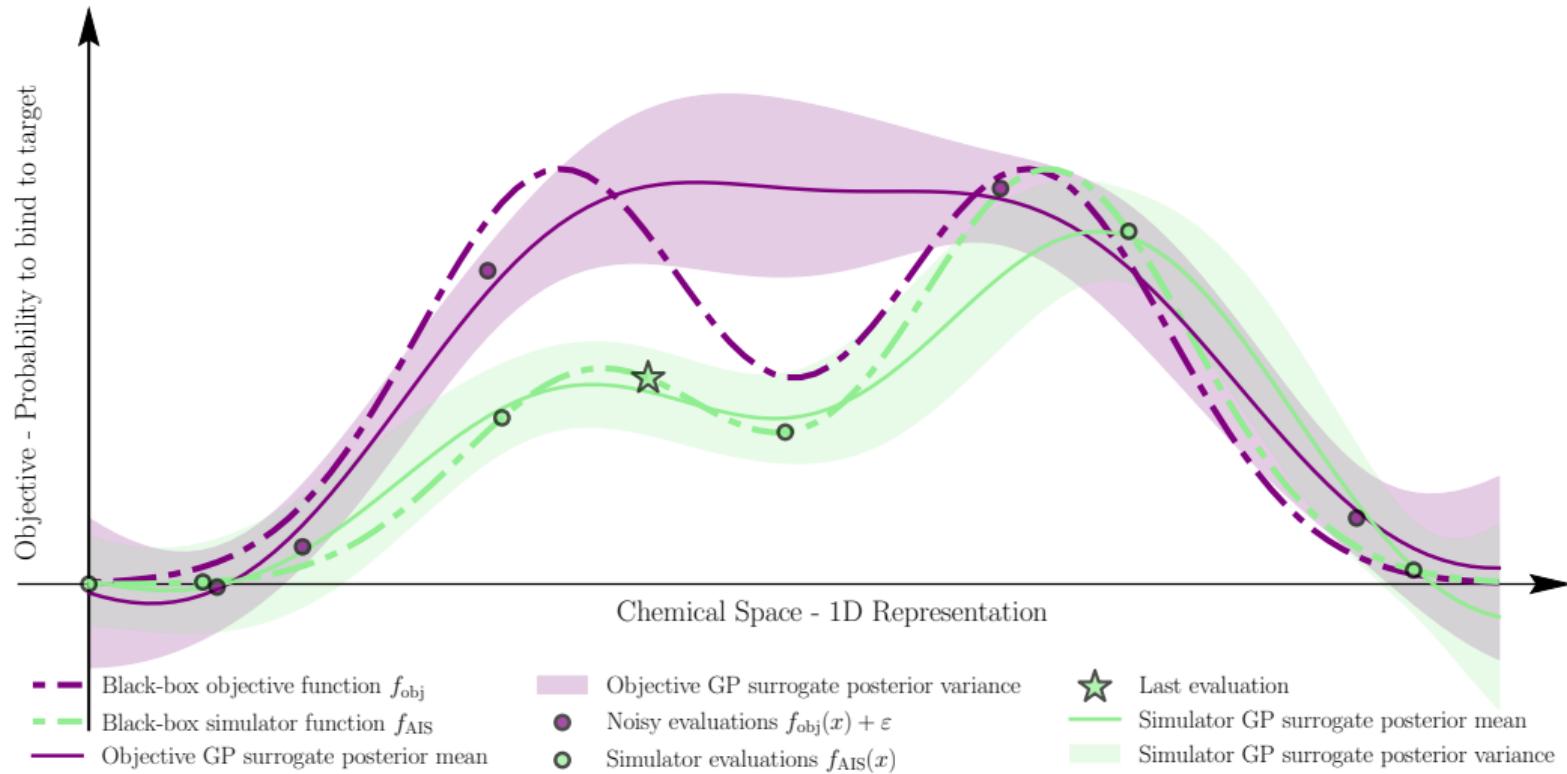
Multi Fidelity Bayesian Optimization 101

Budget = 19.8



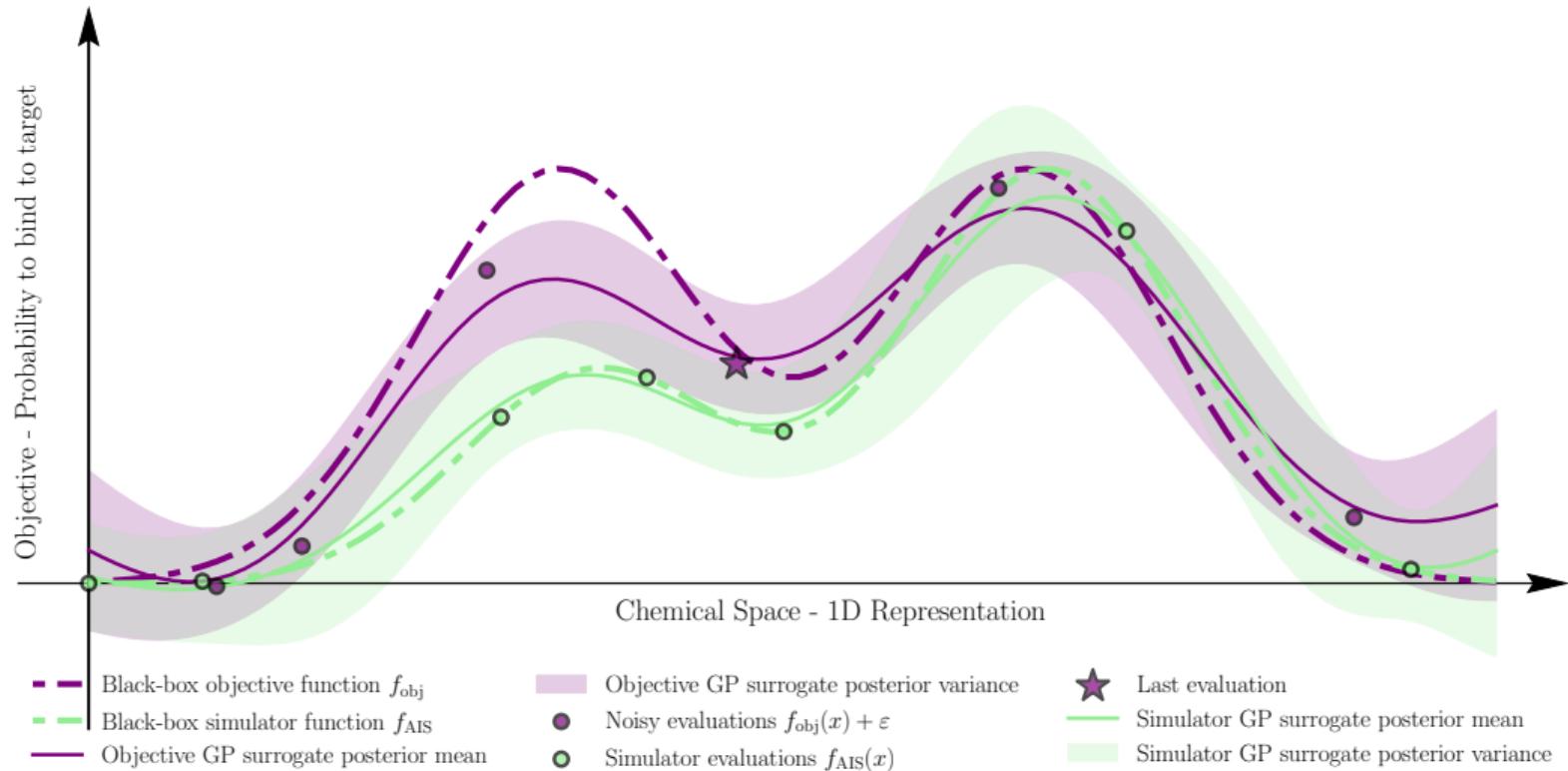
Multi Fidelity Bayesian Optimization 101

Budget = 19.6



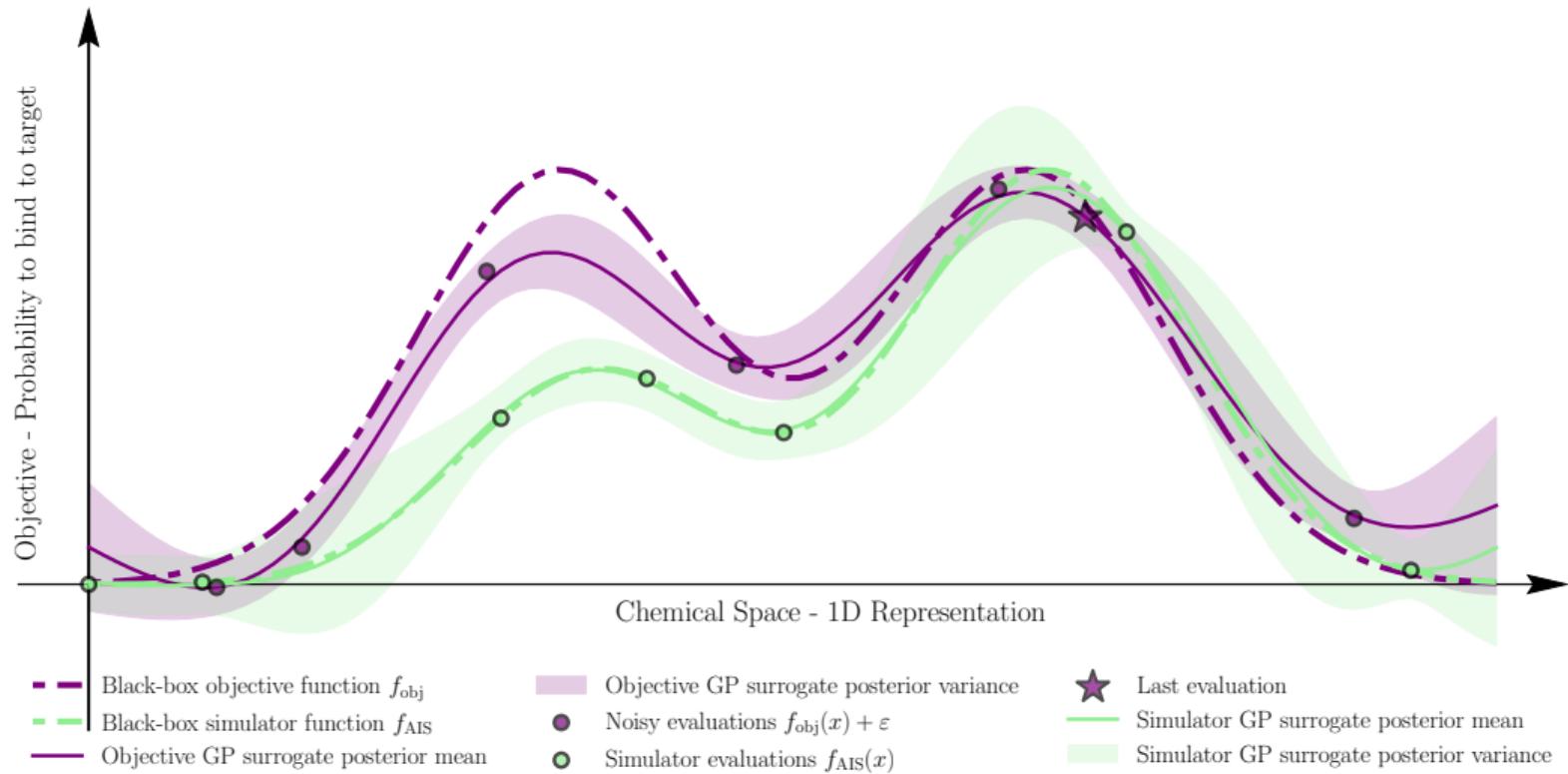
Multi Fidelity Bayesian Optimization 101

Budget = 18.6



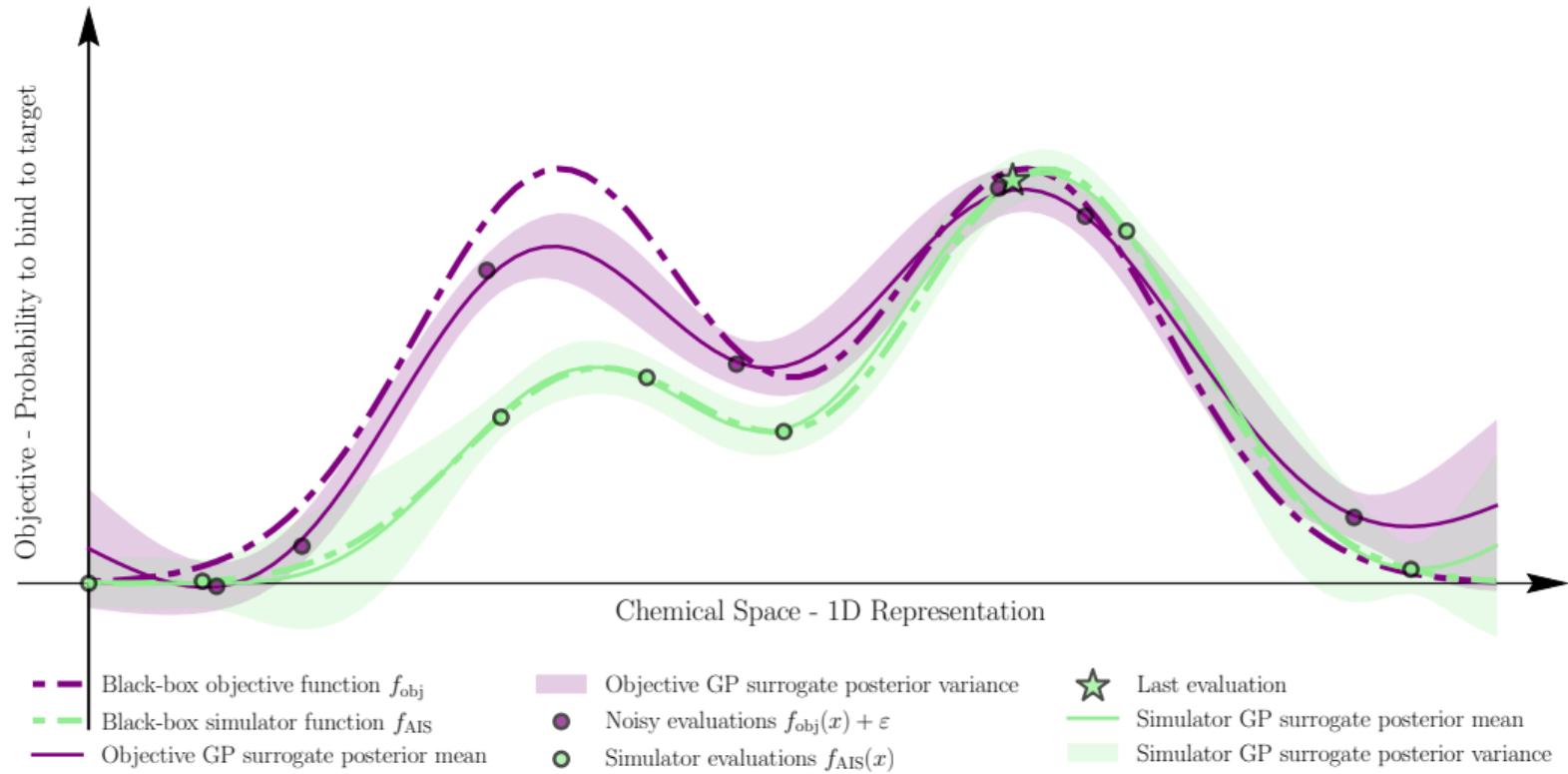
Multi Fidelity Bayesian Optimization 101

Budget = 17.6



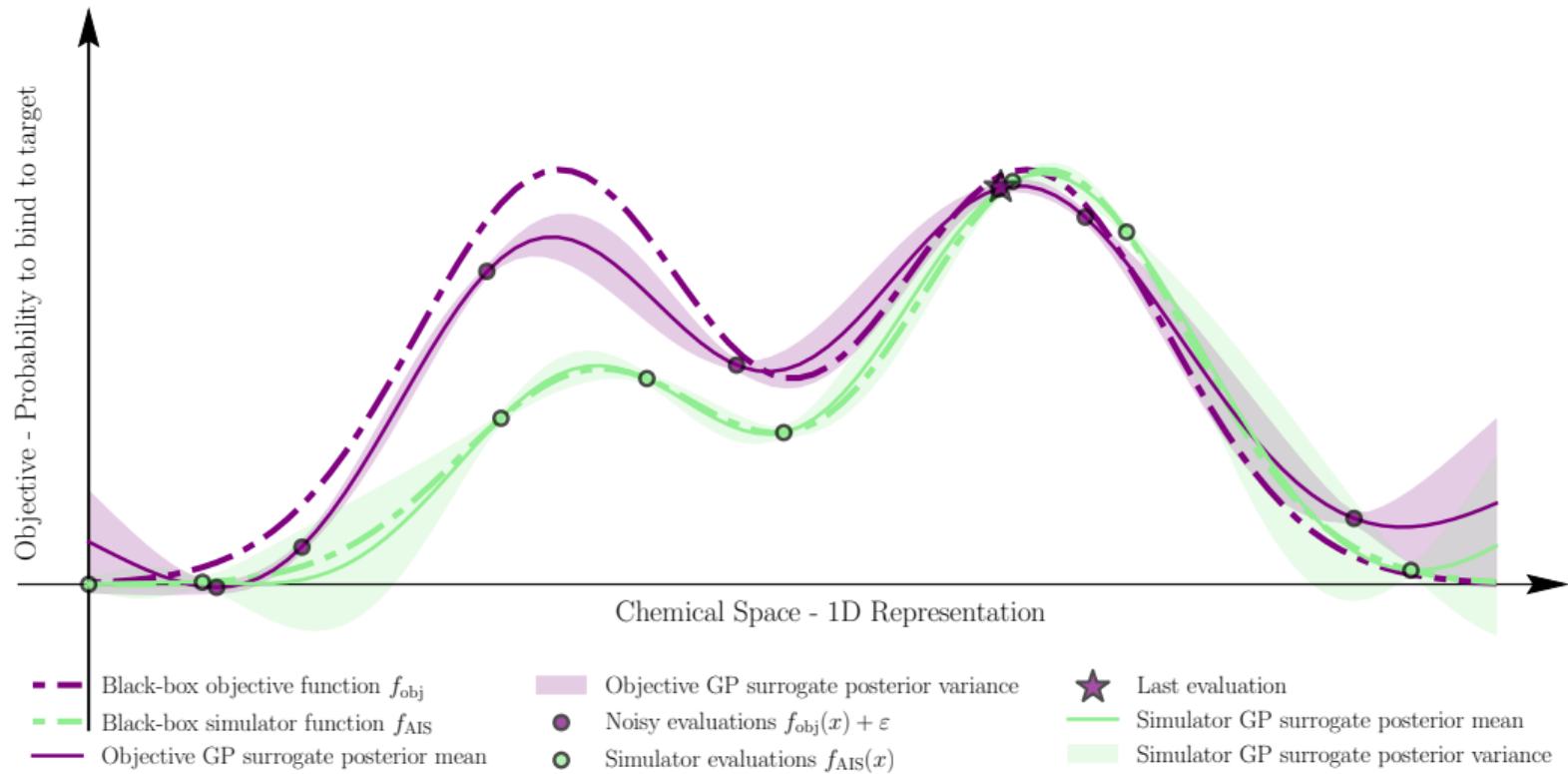
Multi Fidelity Bayesian Optimization 101

Budget = 17.4



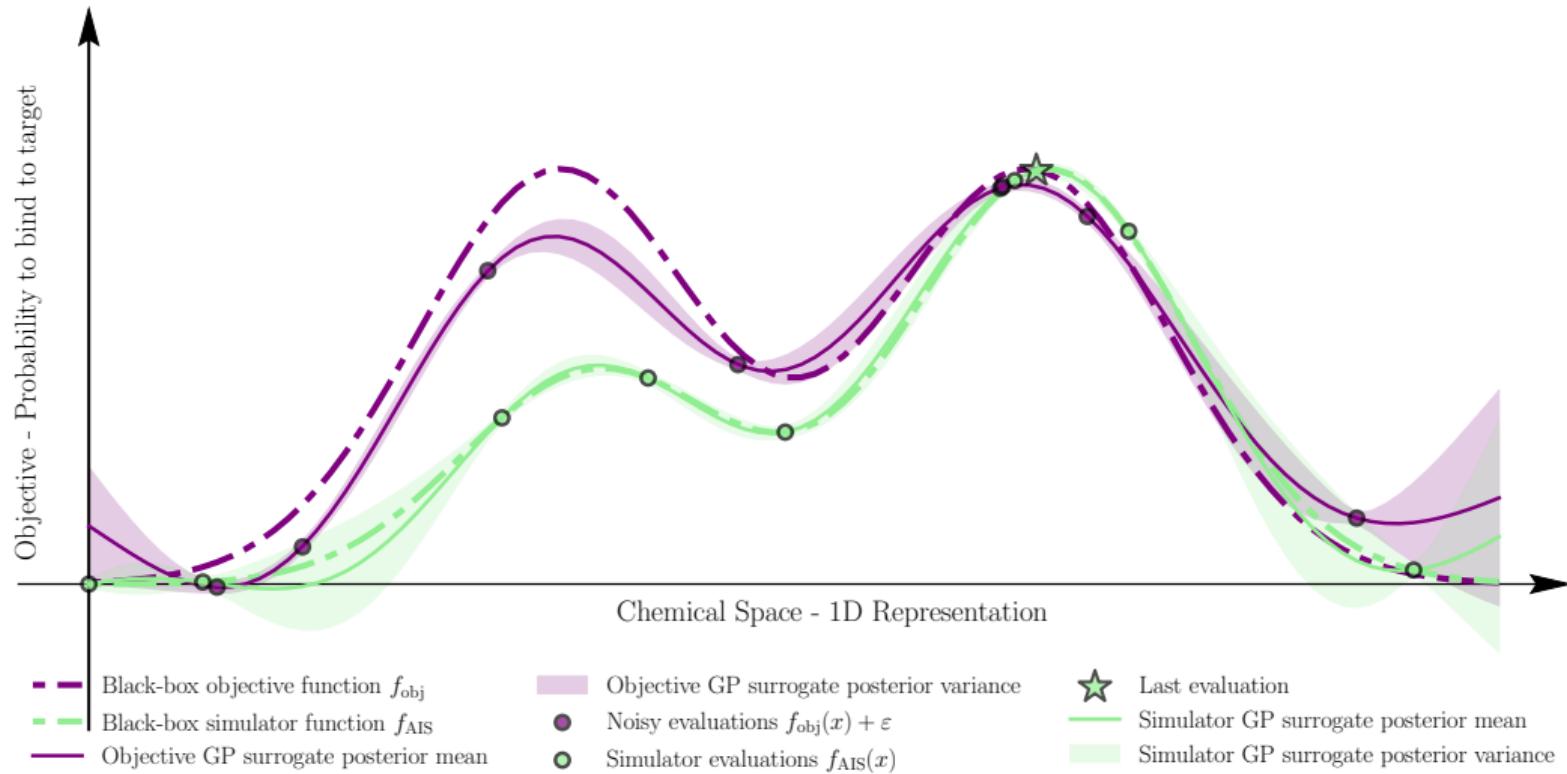
Multi Fidelity Bayesian Optimization 101

Budget = 16.4



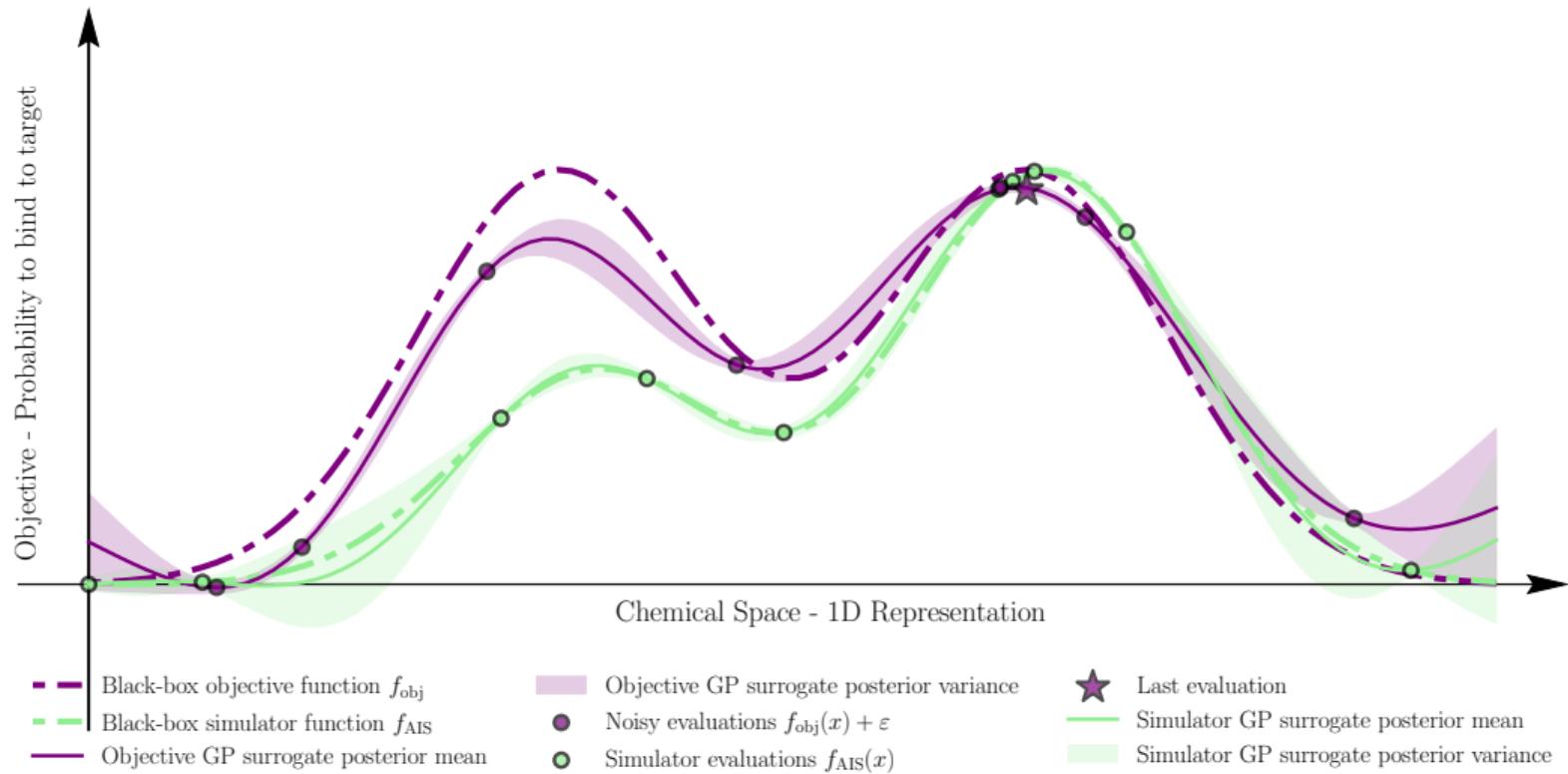
Multi Fidelity Bayesian Optimization 101

Budget = 16.2

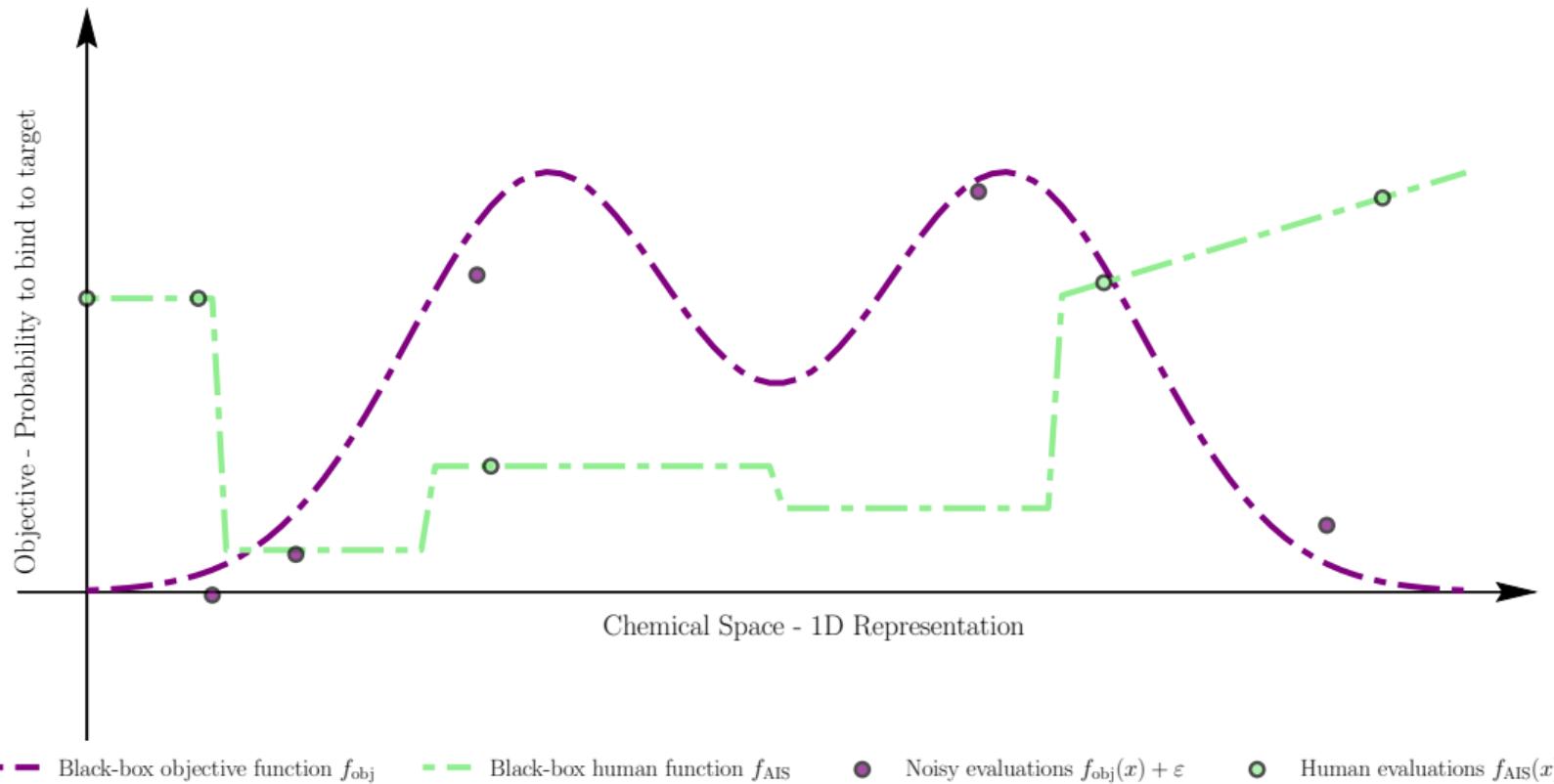


Multi Fidelity Bayesian Optimization 101

Budget = 15.2

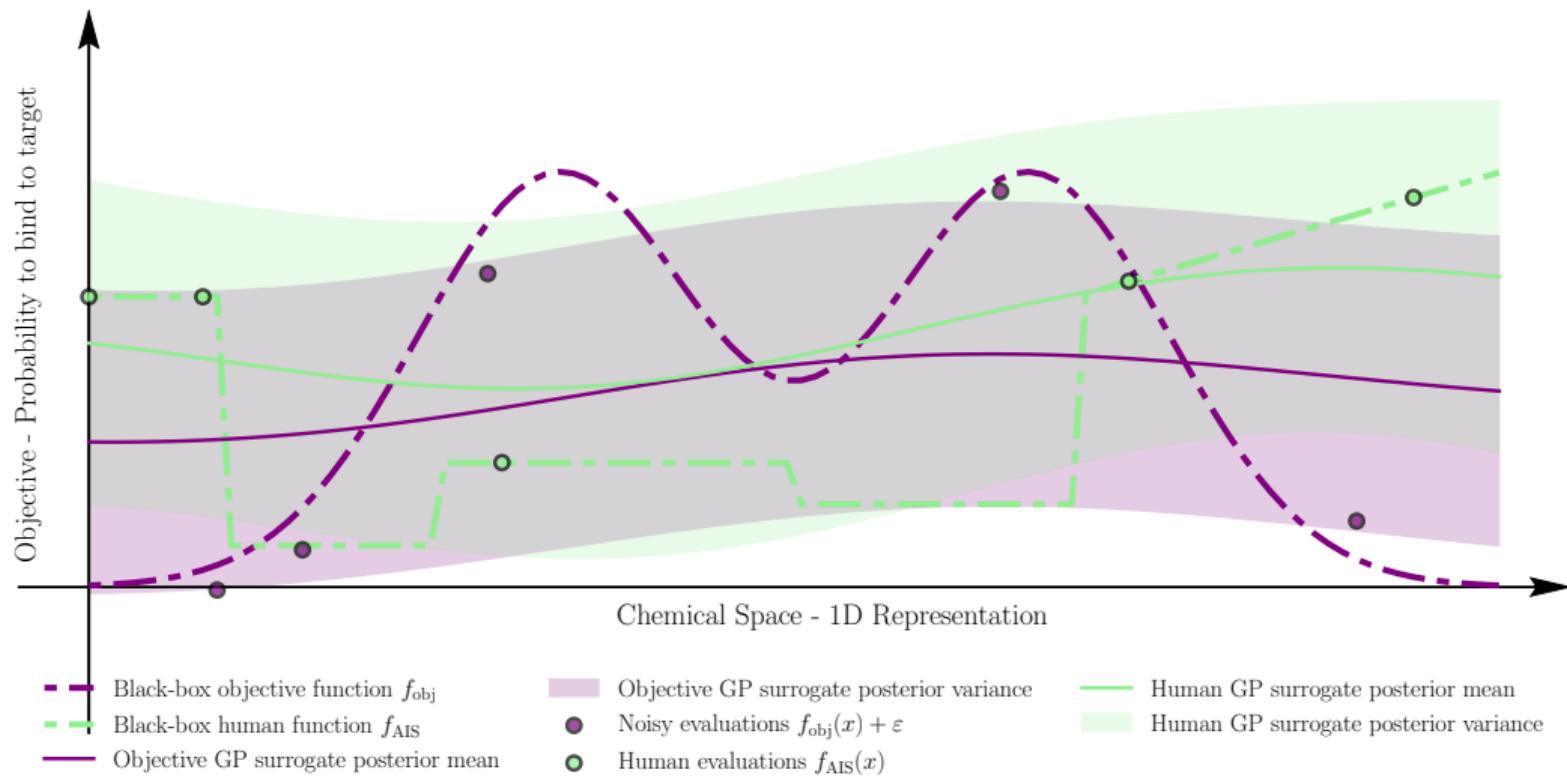


Multi Fidelity Bayesian Optimization with Unreliable Sources



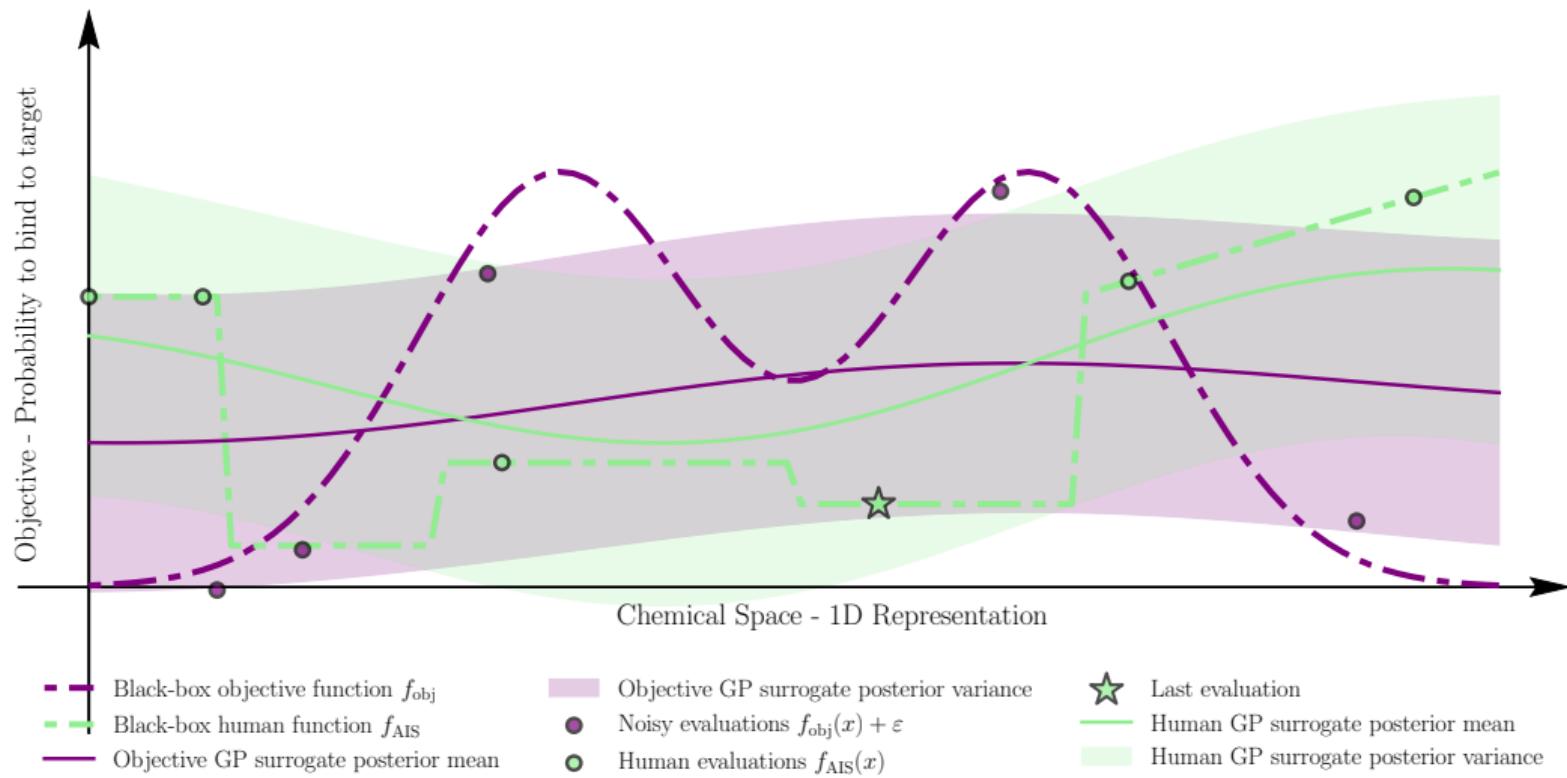
Multi Fidelity Bayesian Optimization with Unreliable Sources

Budget = 20



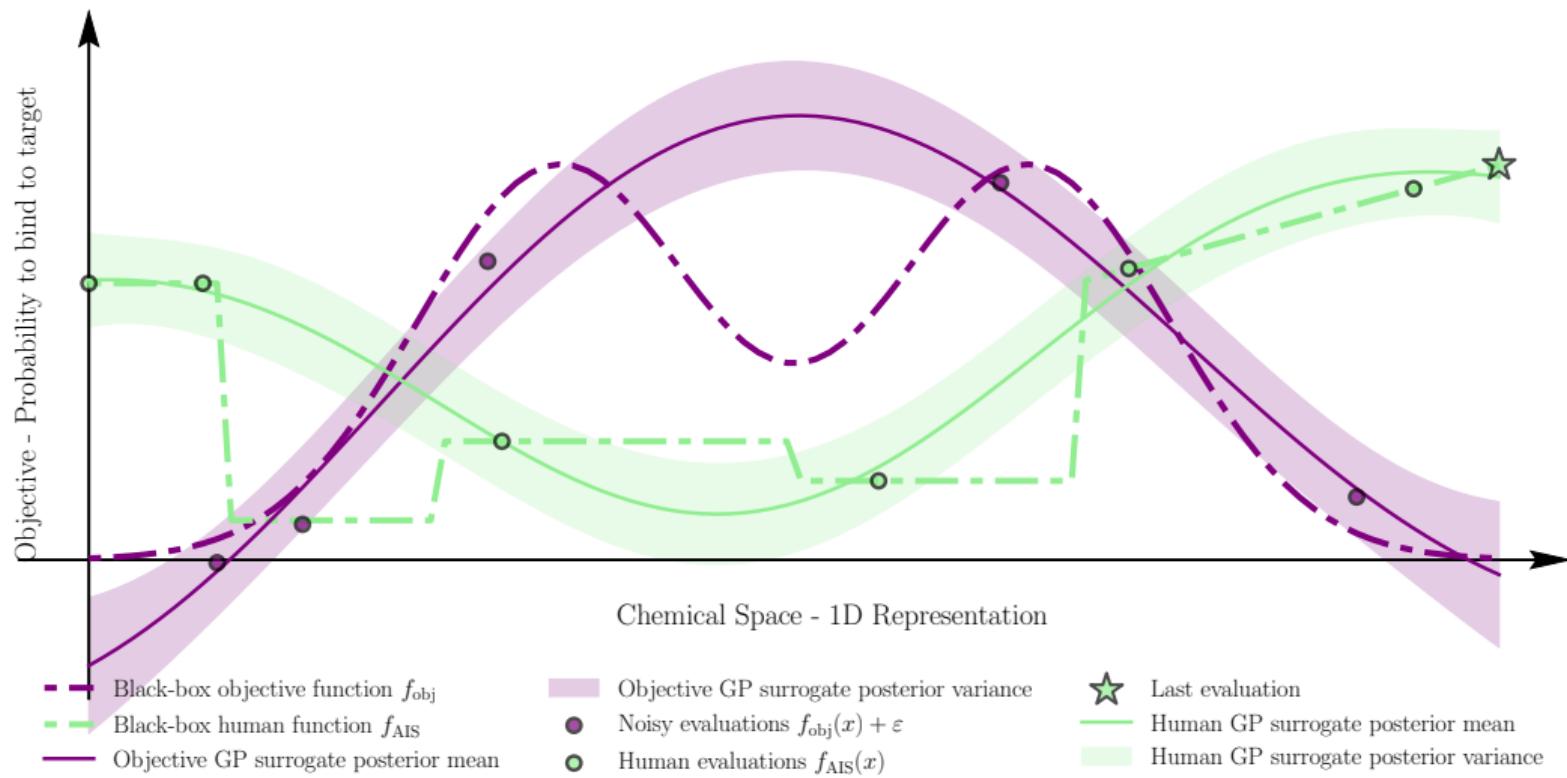
Multi Fidelity Bayesian Optimization with Unreliable Sources

Budget = 19.9



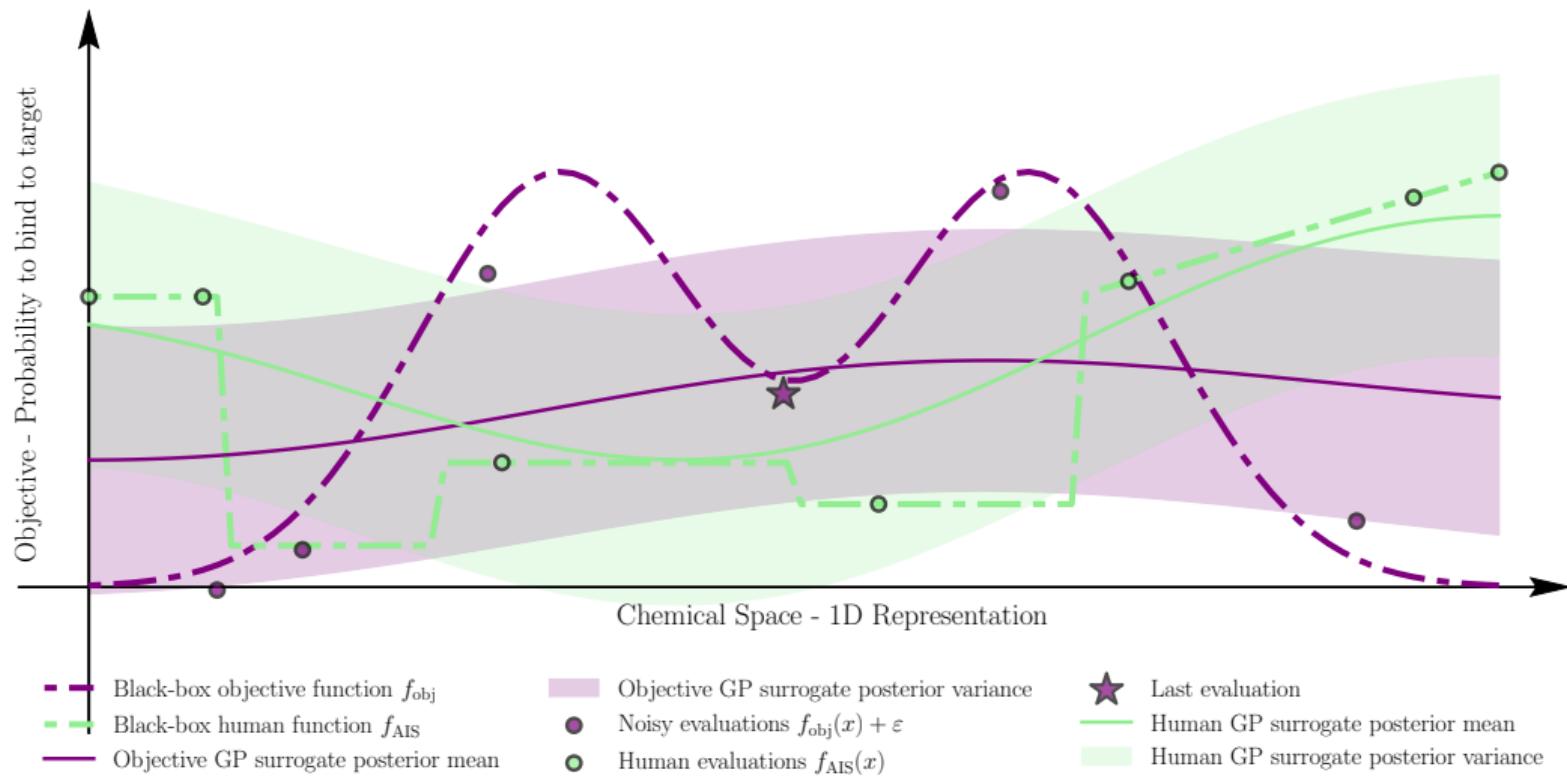
Multi Fidelity Bayesian Optimization with Unreliable Sources

Budget = 19.8



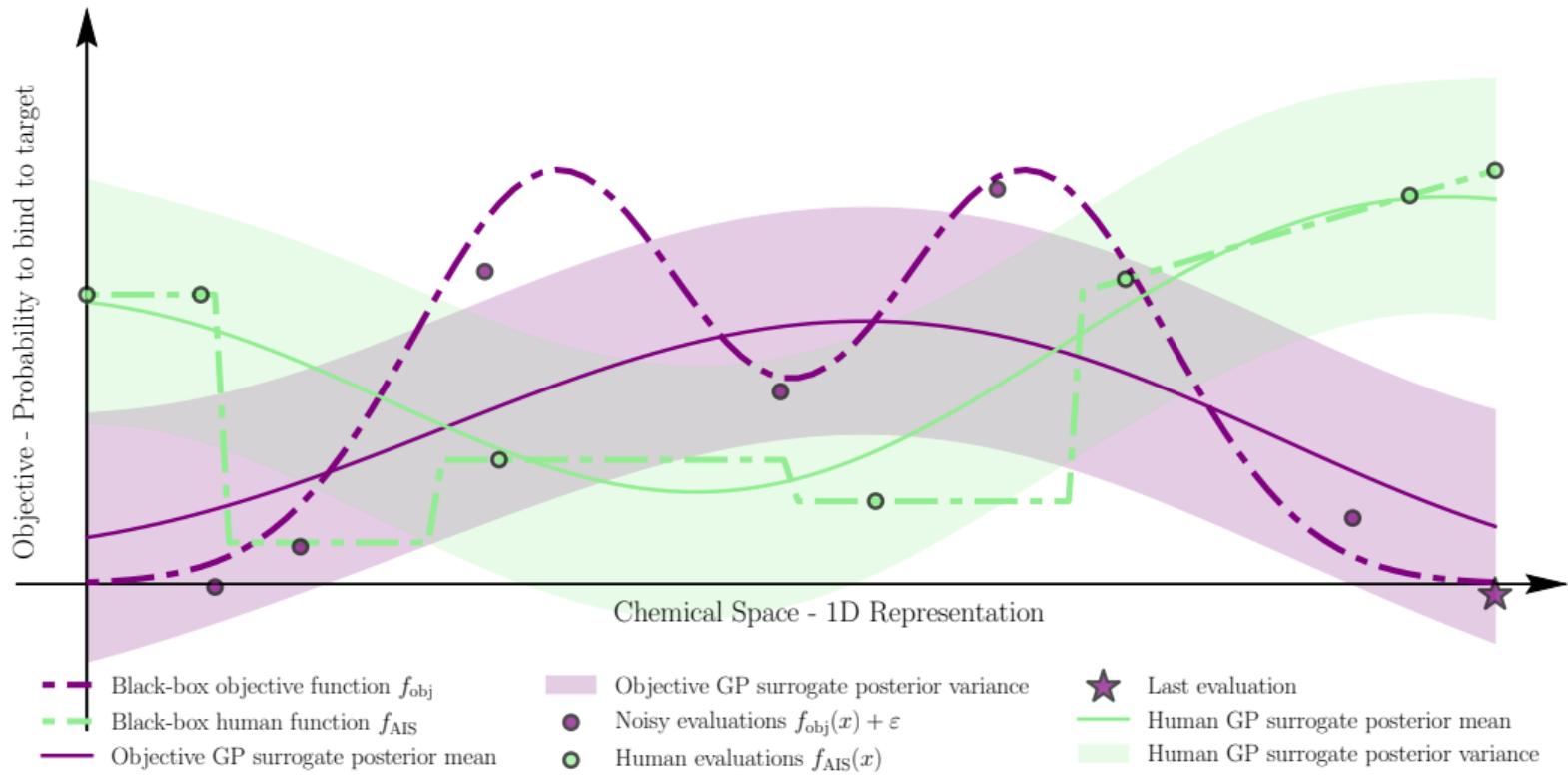
Multi Fidelity Bayesian Optimization with Unreliable Sources

Budget = 18.8



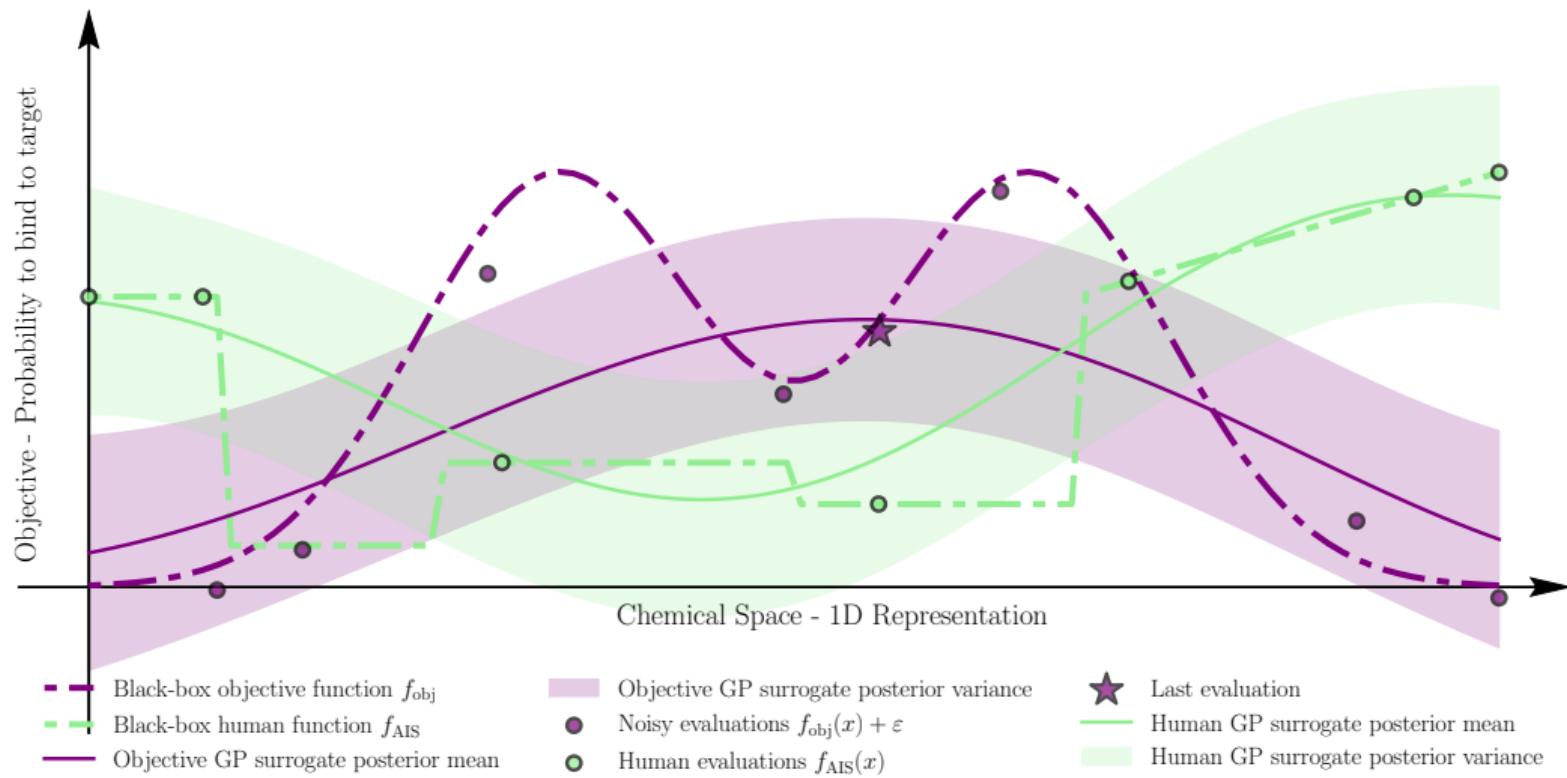
Multi Fidelity Bayesian Optimization with Unreliable Sources

Budget = 17.8



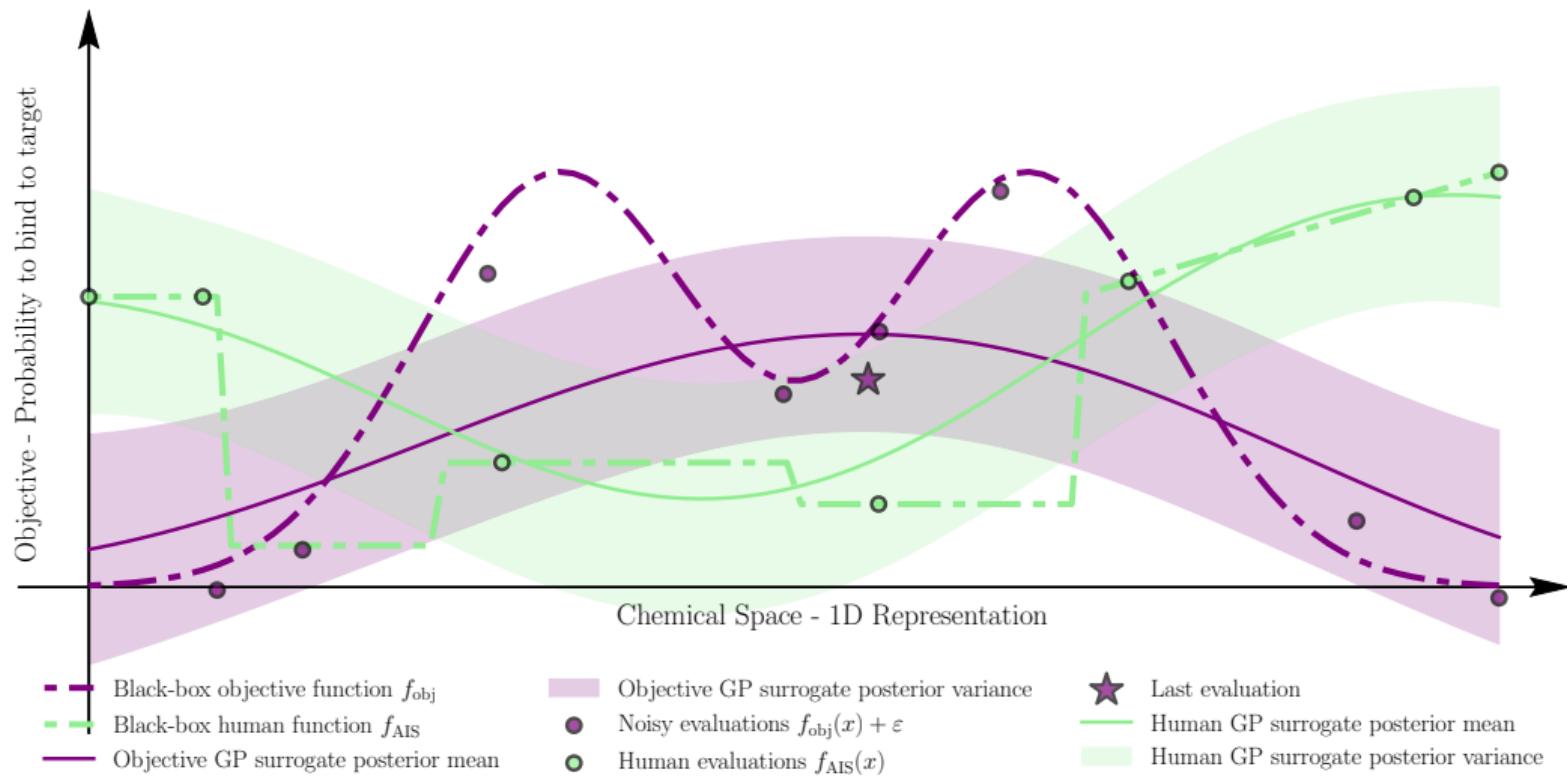
Multi Fidelity Bayesian Optimization with Unreliable Sources

Budget = 16.8



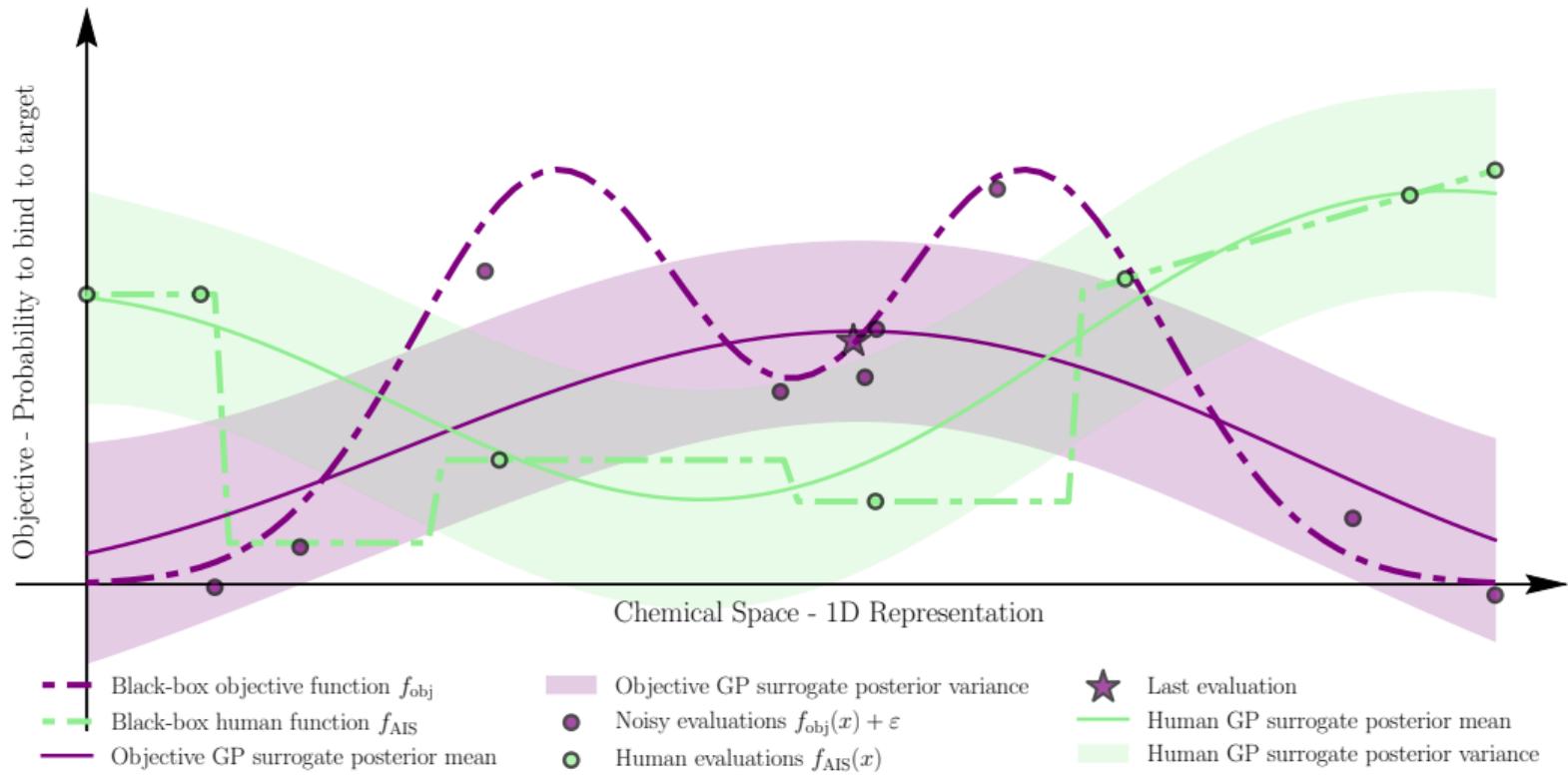
Multi Fidelity Bayesian Optimization with Unreliable Sources

Budget = 15.8



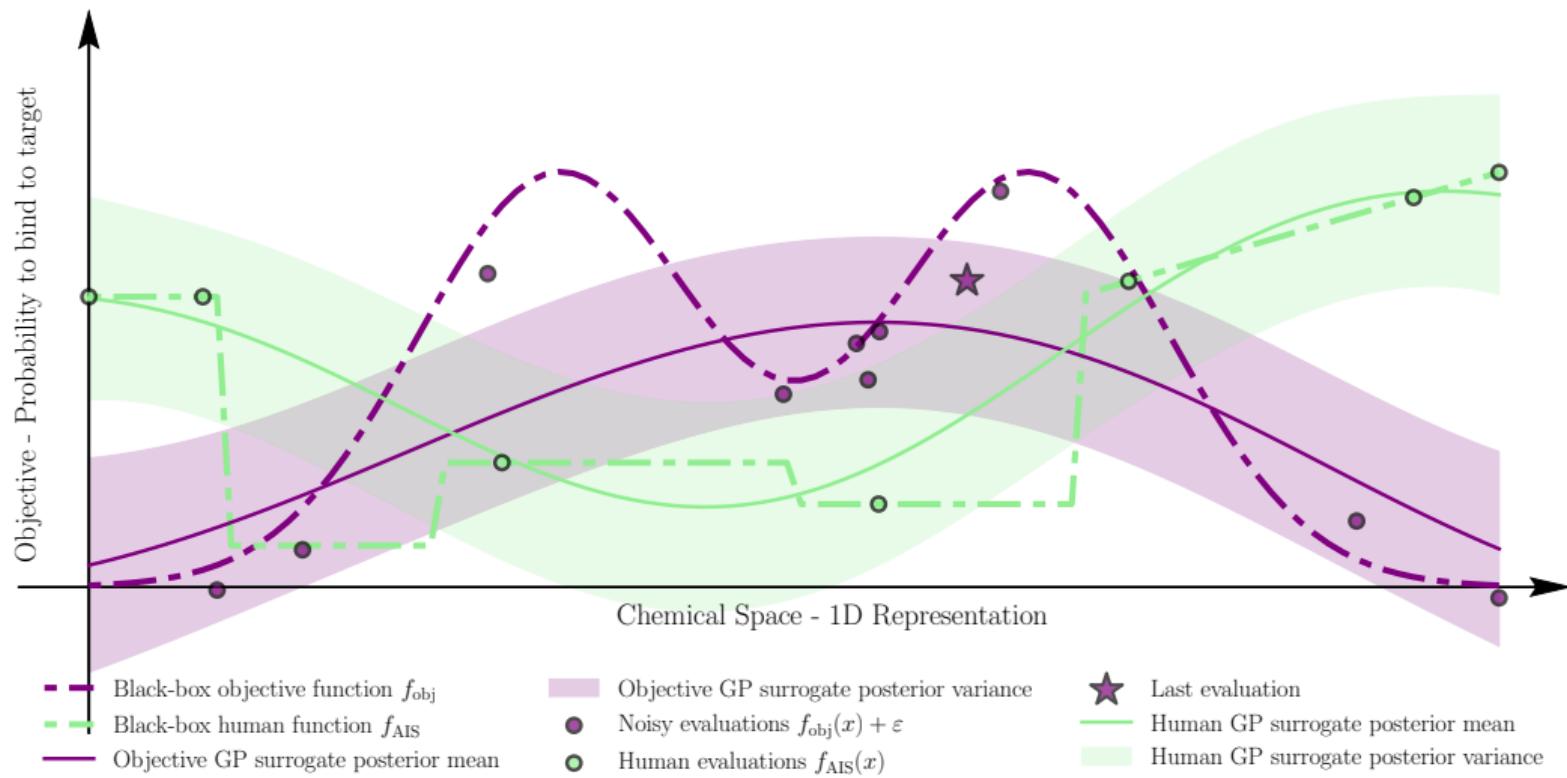
Multi Fidelity Bayesian Optimization with Unreliable Sources

Budget = 14.8



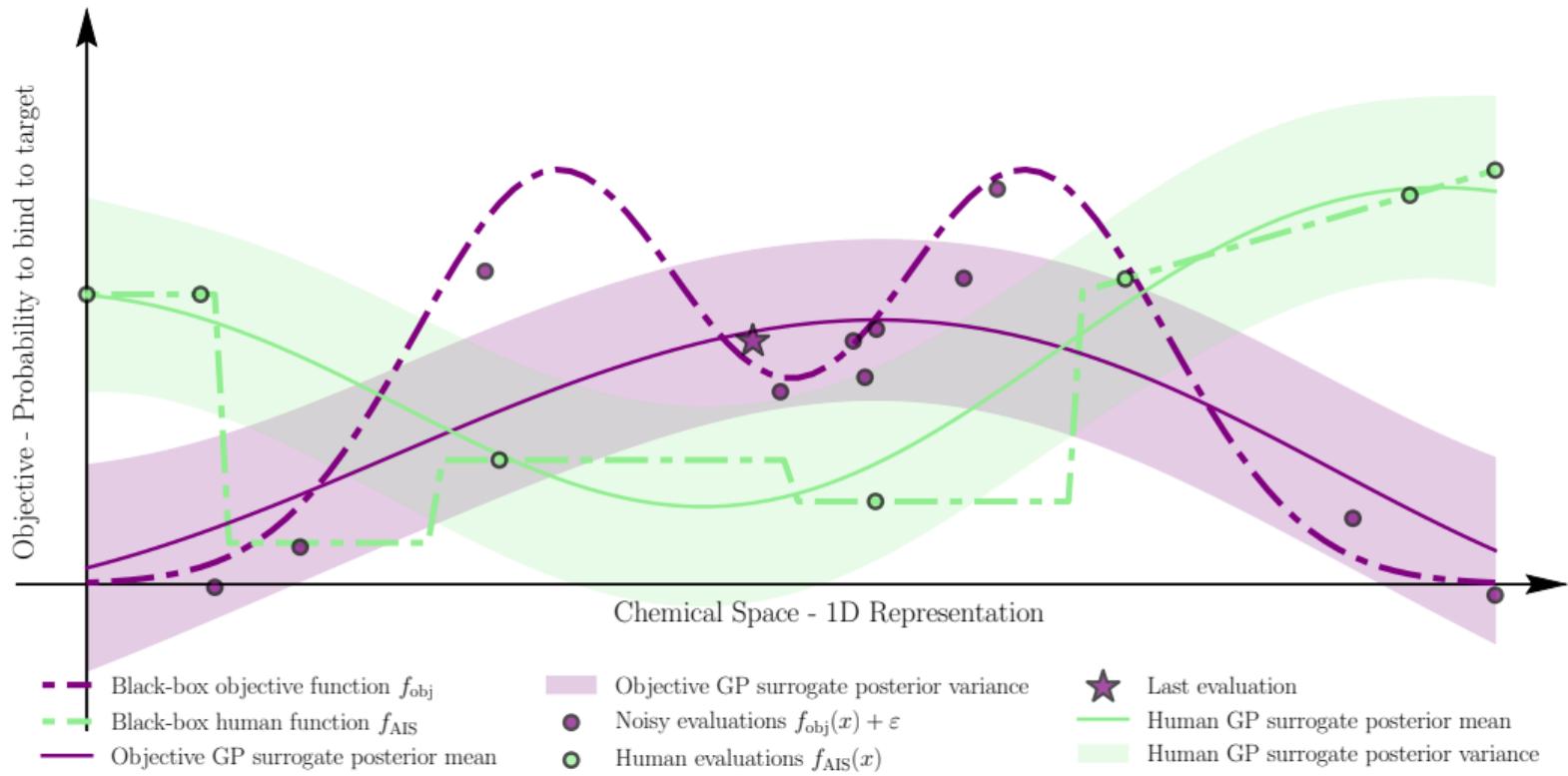
Multi Fidelity Bayesian Optimization with Unreliable Sources

Budget = 13.8



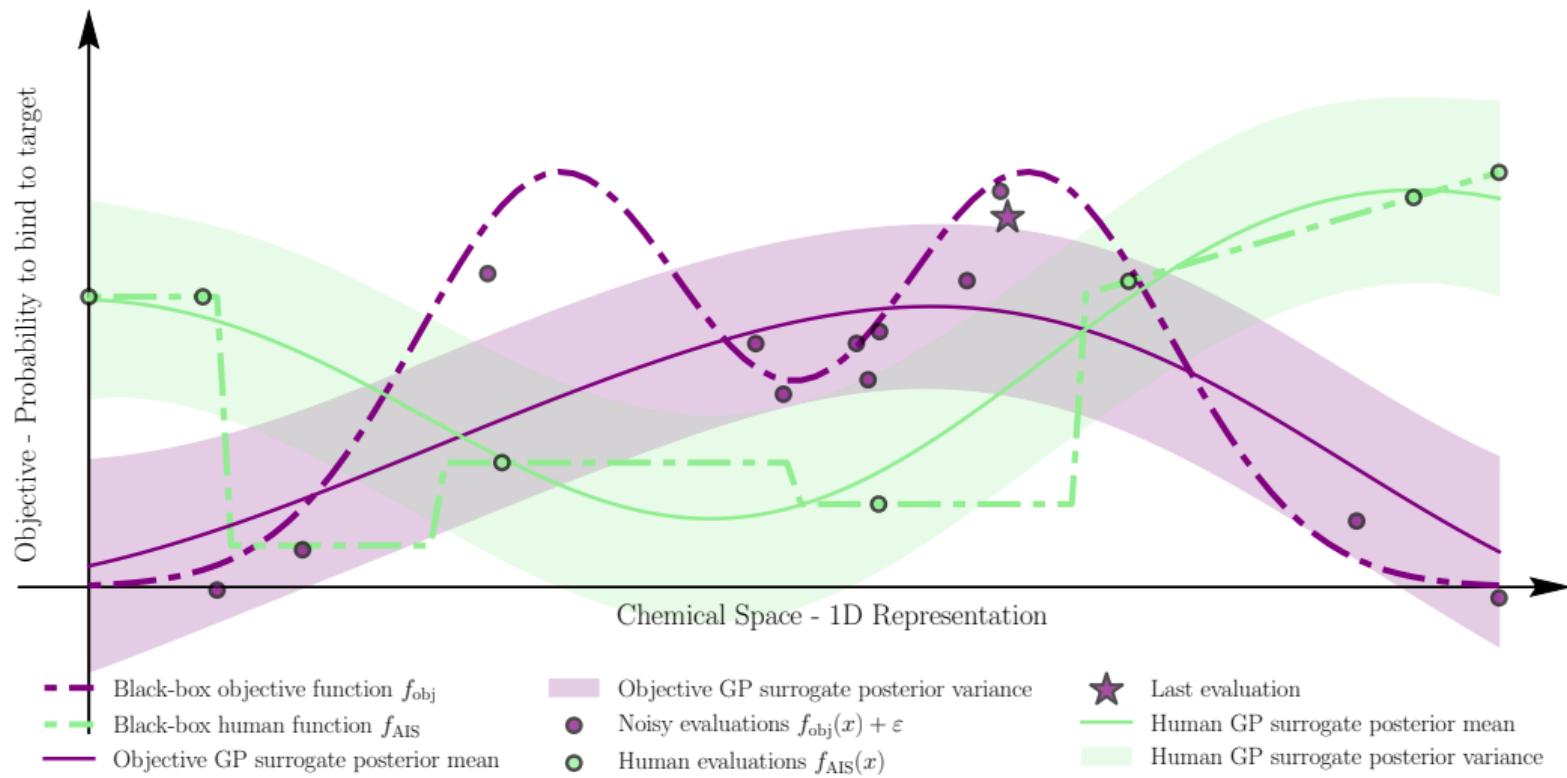
Multi Fidelity Bayesian Optimization with Unreliable Sources

Budget = 12.8



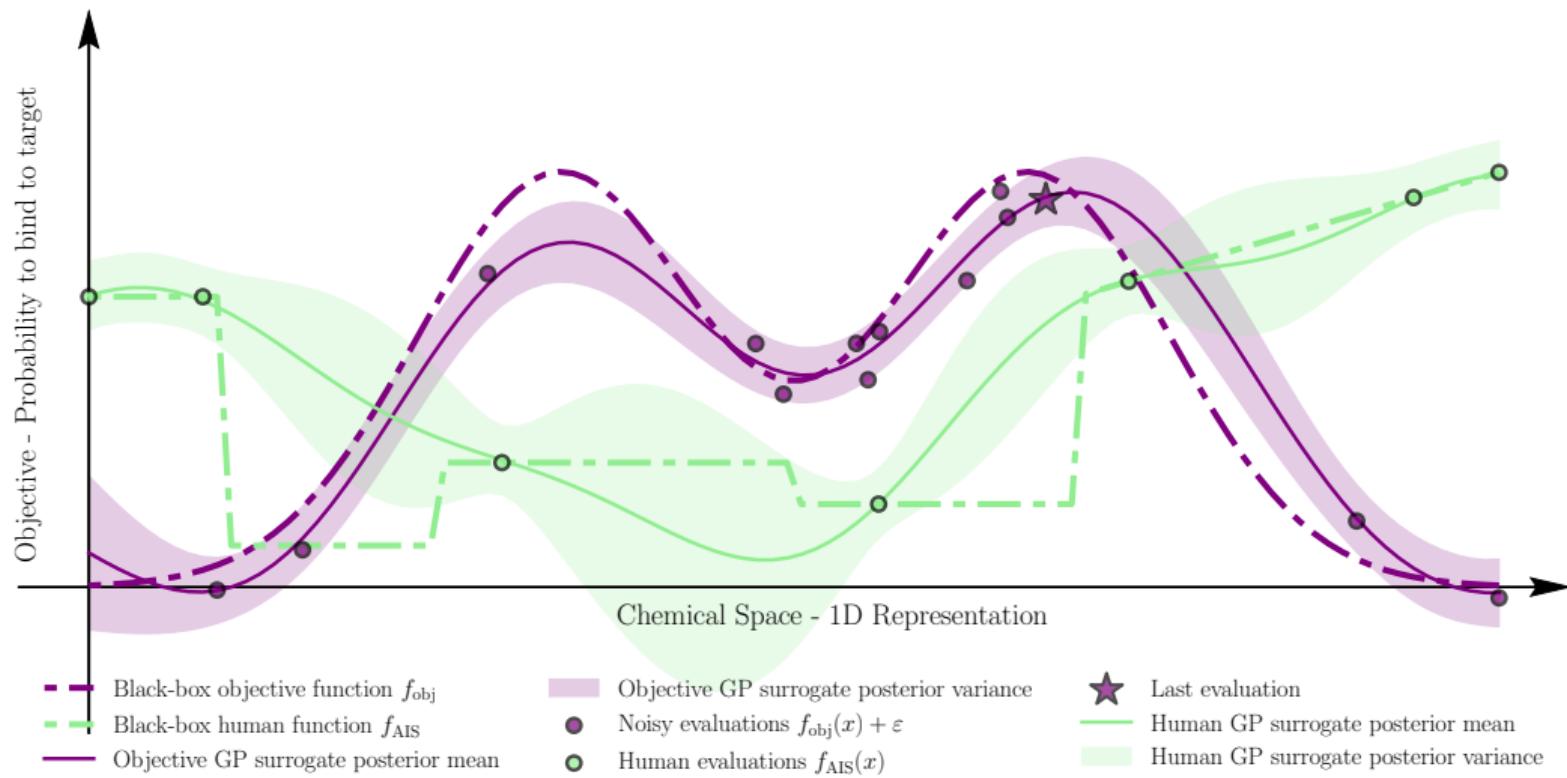
Multi Fidelity Bayesian Optimization with Unreliable Sources

Budget = 11.8



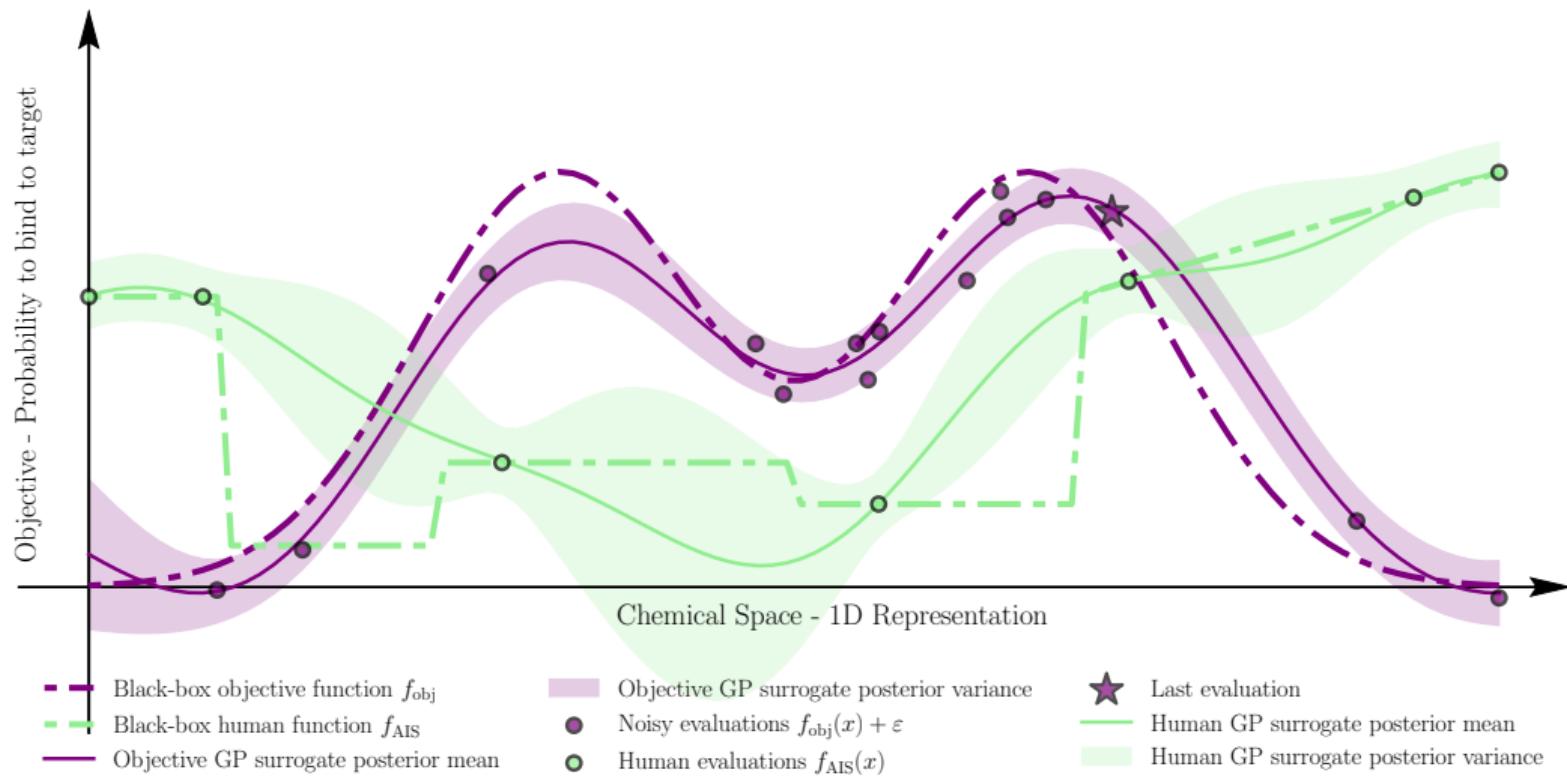
Multi Fidelity Bayesian Optimization with Unreliable Sources

Budget = 10.8

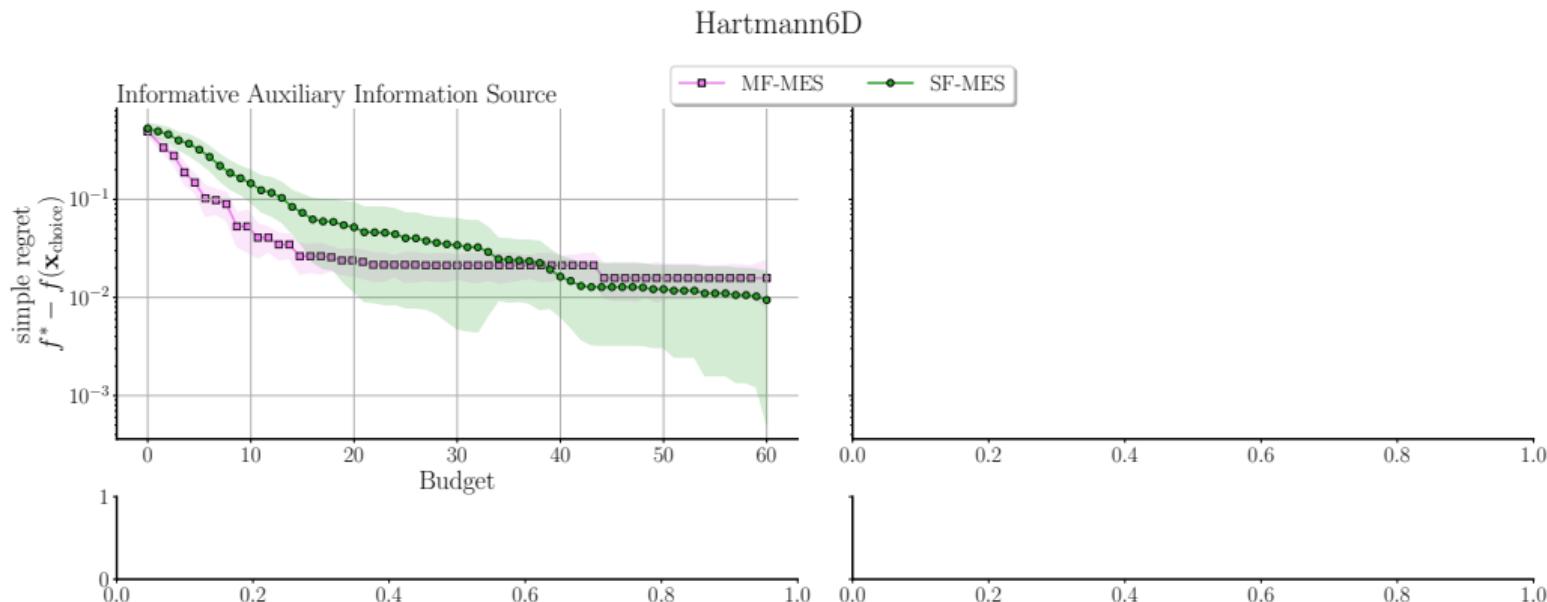


Multi Fidelity Bayesian Optimization with Unreliable Sources

Budget = 9.8



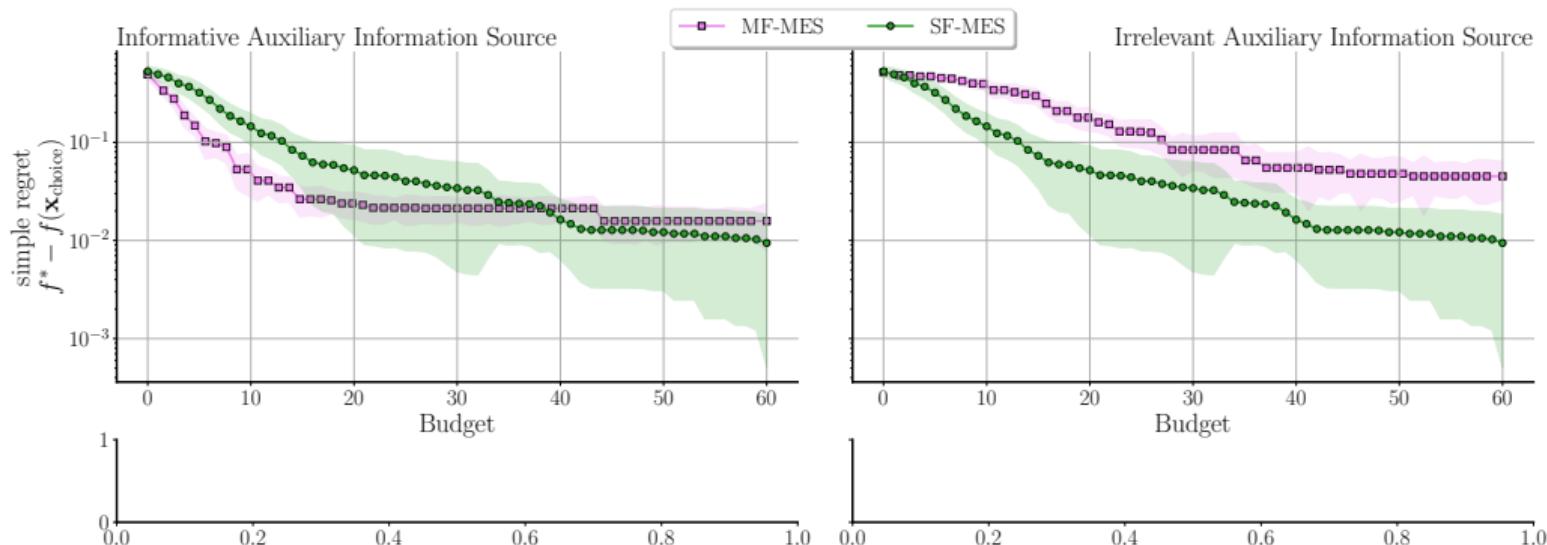
Multi-Fidelity BO is not robust to unreliable Information Sources



- Both relevant and irrelevant AIS have a cost 5x cheaper than objective.
- SF-MES: Single-Fidelity BO w/ Maximum Entropy Search Acquisition Function
- MF-MES: Multi-Fidelity BO w/ Maximum Entropy Search Acquisition Function

Multi-Fidelity BO is not robust to unreliable Information Sources

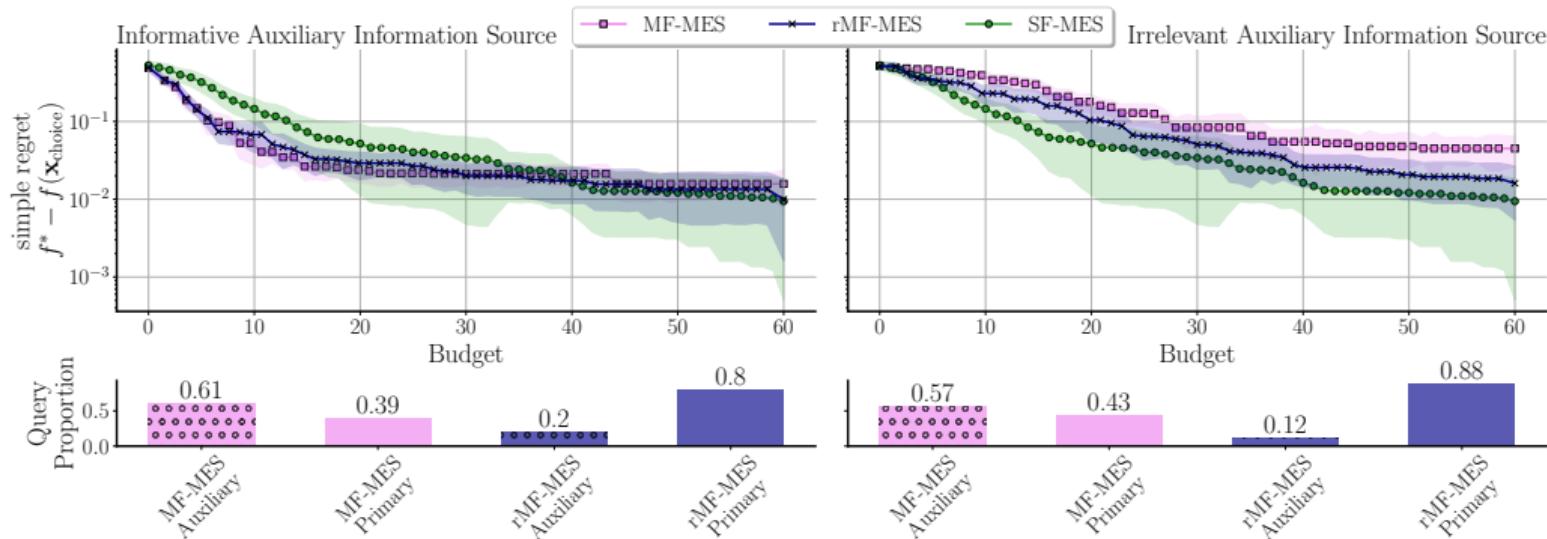
Hartmann6D



- Both relevant and irrelevant AIS have a cost 5x cheaper than objective.
- SF-MES: Single-Fidelity BO w/ Maximum Entropy Search Acquisition Function
- MF-MES: Multi-Fidelity BO w/ Maximum Entropy Search Acquisition Function

Multi-Fidelity BO is not robust to unreliable Information Sources

Hartmann6D



- Main aim of our contribution: **robustness** to irrelevant AIS...
- ...While still **accelerating** convergence for relevant AIS (otherwise, just do BO)

Introducing robust MFBO (rMFBO), defensive acquisition strategy

We modify the BO loop with a **building block added on top of any MFBO method.**

Introducing robust MFBO (rMFBO), defensive acquisition strategy

We modify the BO loop with a **building block added on top of any MFBO method.**

- Two separate GPs: MOGP $\mu_{MF}, \sigma_{MF} | \mathcal{D}^{MF}$ and a GP μ_{SF}, σ_{SF} trained using a *pseudo* dataset of objective queries only \mathcal{D}^{pSF}

Introducing robust MFBO (rMFBO), defensive acquisition strategy

We modify the BO loop with a **building block added on top of any MFBO method.**

- Two separate GPs: MOGP $\mu_{MF}, \sigma_{MF} | \mathcal{D}^{MF}$ and a GP μ_{SF}, σ_{SF} trained using a *pseudo* dataset of objective queries only \mathcal{D}^{pSF}

$$(x_t^{MF}, \ell_t) = \underset{x \in \mathcal{X}, \ell \in \{\text{obj, AIS}\}}{\operatorname{argmax}} \alpha(x, \ell | \mu_{MF}, \sigma_{MF}, \mathcal{D}^{MF})$$

$$(x_t^{pSF}, \text{obj}) = \underset{x \in \mathcal{X}}{\operatorname{argmax}} \alpha(x | \mu_{SF}, \sigma_{SF}, \mathcal{D}^{pSF})$$

Introducing robust MFBO (rMFBO), defensive acquisition strategy

We modify the BO loop with a **building block added on top of any MFBO method**.

- Two separate GPs: MOGP $\mu_{MF}, \sigma_{MF} | \mathcal{D}^{MF}$ and a GP μ_{SF}, σ_{SF} trained using a *pseudo* dataset of objective queries only \mathcal{D}^{pSF}

$$(x_t^{MF}, \ell_t) = \operatorname{argmax}_{x \in \mathcal{X}, \ell \in \{\text{obj, AIS}\}} \alpha(x, \ell | \mu_{MF}, \sigma_{MF}, \mathcal{D}^{MF})$$

$$(x_t^{pSF}, \text{obj}) = \operatorname{argmax}_{x \in \mathcal{X}} \alpha(x | \mu_{SF}, \sigma_{SF}, \mathcal{D}^{pSF})$$

- $\sigma_{MF}(x_t^{pSF}, \text{obj}) \leq c_1$ → Do I trust my joint model at the objective?
- $s(x_t^{MF}, \ell_t) \geq c_2$ → Is my joint model suggestion informative enough?

Introducing robust MFBO (rMFBO), defensive acquisition strategy

We modify the BO loop with a **building block added on top of any MFBO method**.

- Two separate GPs: MOGP $\mu_{MF}, \sigma_{MF} | \mathcal{D}^{MF}$ and a GP μ_{SF}, σ_{SF} trained using a *pseudo* dataset of objective queries only \mathcal{D}^{pSF}

$$(x_t^{MF}, \ell_t) = \underset{x \in \mathcal{X}, \ell \in \{\text{obj, AIS}\}}{\operatorname{argmax}} \alpha(x, \ell | \mu_{MF}, \sigma_{MF}, \mathcal{D}^{MF})$$

$$(x_t^{pSF}, \text{obj}) = \underset{x \in \mathcal{X}}{\operatorname{argmax}} \alpha(x | \mu_{SF}, \sigma_{SF}, \mathcal{D}^{pSF})$$

- $\sigma_{MF}(x_t^{pSF}, \text{obj}) \leq c_1$ → Do I trust my joint model at the objective?
- $s(x_t^{MF}, \ell_t) \geq c_2$ → Is my joint model suggestion informative enough?
- Upon satisfaction: query (x_t^{MF}, ℓ_t) and add *pseudo-observation* of objective:
 $\mathcal{D}^{pSF} \leftarrow (x_t^{pSF}, \mu_{MF}(x_t^{pSF}, \text{obj}))$ → What if we had queried the objective?
- Otherwise, query (x_t^{pSF}, obj)

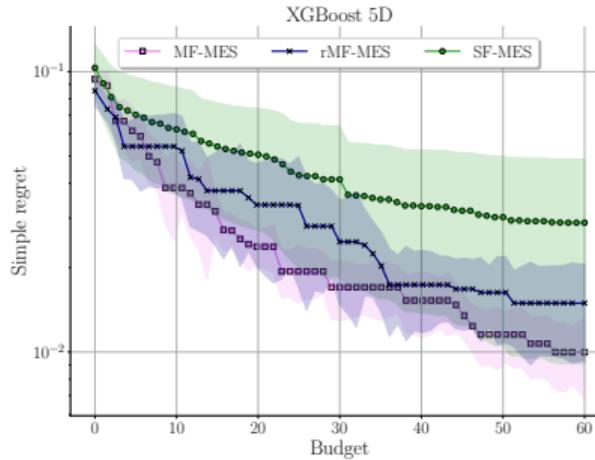
Theorem

Upon classical assumptions for deriving regret bounds in the BO litterature:

For any auxiliary information source, the difference in regrets achieved by SFBO and rMFBO can be bounded with a high, controllable probability.

Results

XGBoost hyperparameter tuning - relevant AIS | Rosenbrock 2D - irrelevant AIS

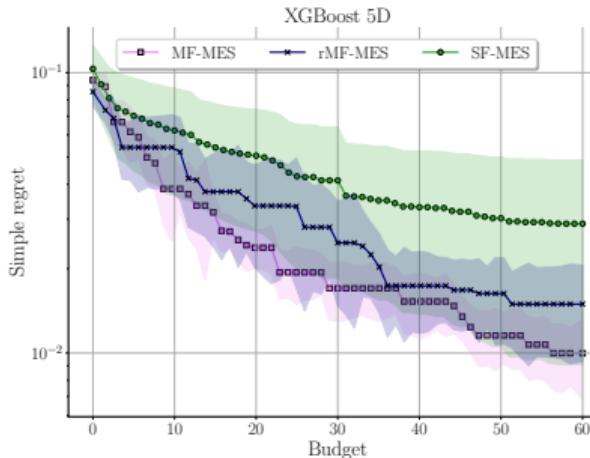


Tuning 5 hyperparameters for a regression task on a Diabetes dataset

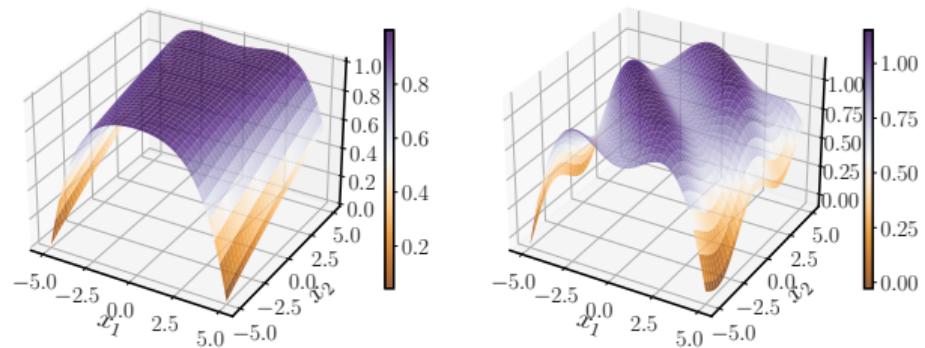
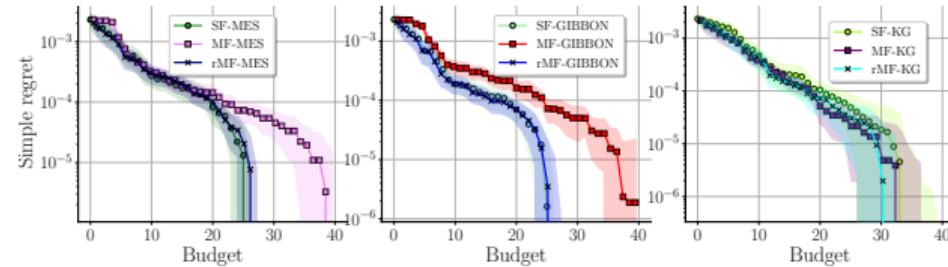
target: 100 decisions trees; ais: 10 decision trees

Results

XGBoost hyperparameter tuning - relevant AIS | Rosenbrock 2D - irrelevant AIS

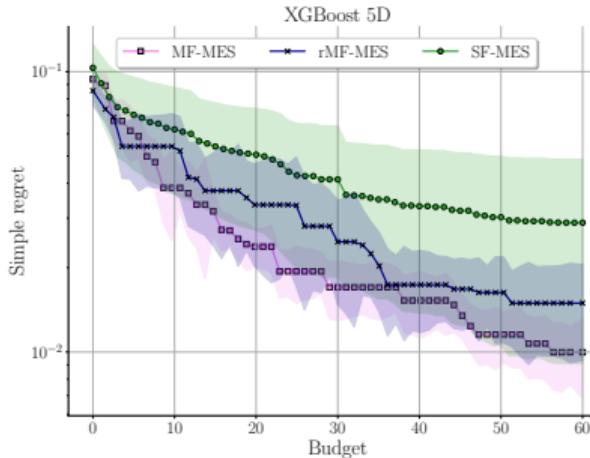


Tuning 5 hyperparameters for a regression task on a Diabetes dataset
target: 100 decisions trees; ais: 10 decision trees

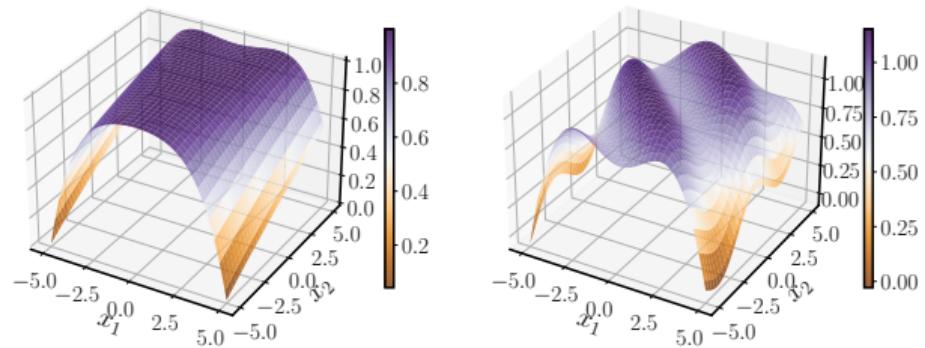
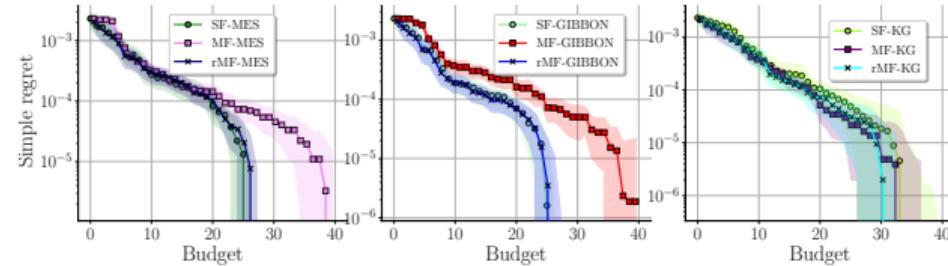


Results

XGBoost hyperparameter tuning - relevant AIS | Rosenbrock 2D - irrelevant AIS

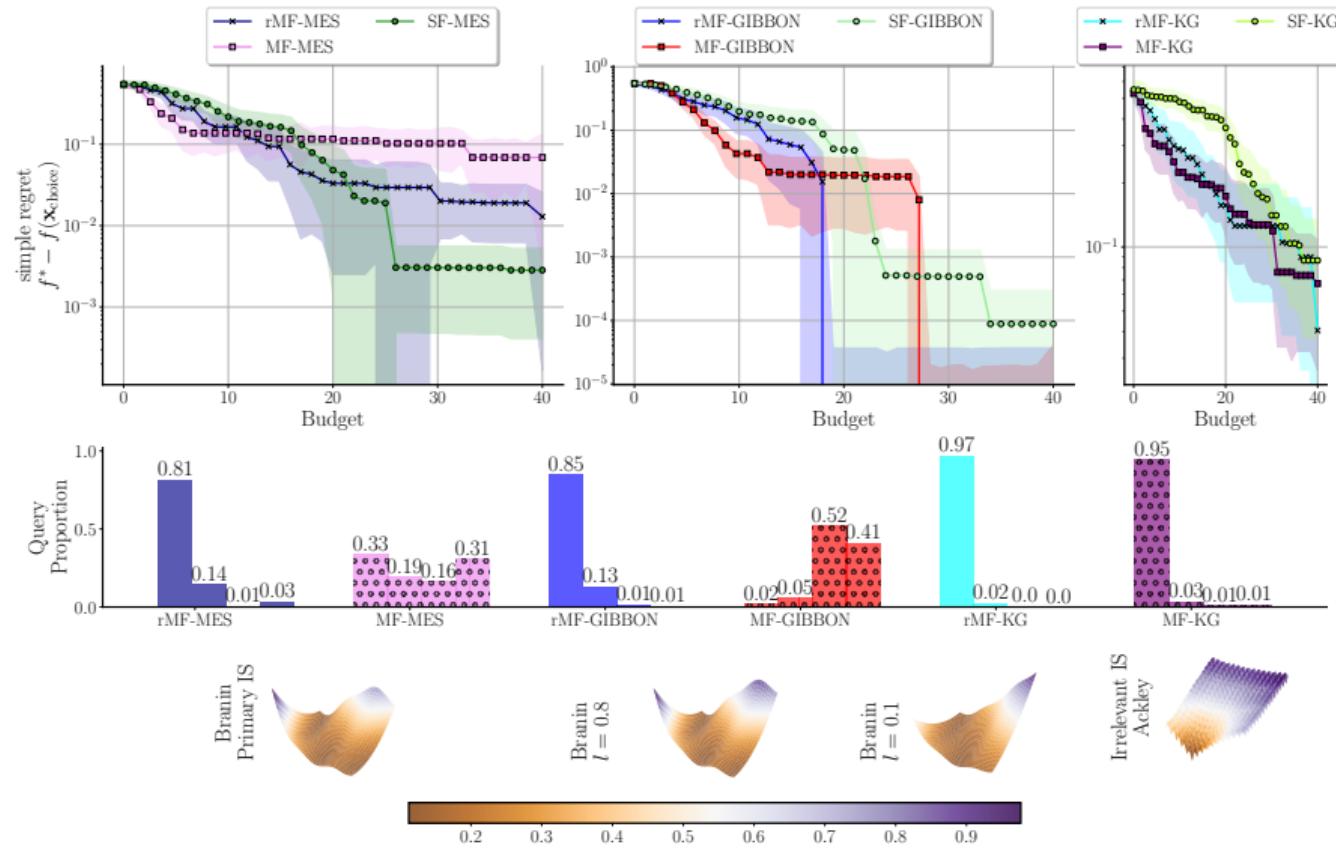


Tuning 5 hyperparameters for a regression task on a Diabetes dataset
target: 100 decisions trees; ais: 10 decision trees



We use the same settings in our algorithm for these two cases!

Multiple Information Sources of varying relevance - 2D case



Human-In-The-Loop Bayesian Optimization for *de novo* drug design

Work In Progress ^_(ツ)_/^-

De novo drug design

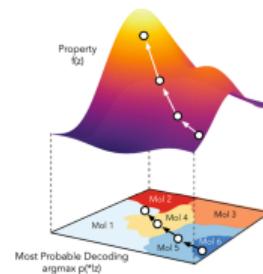
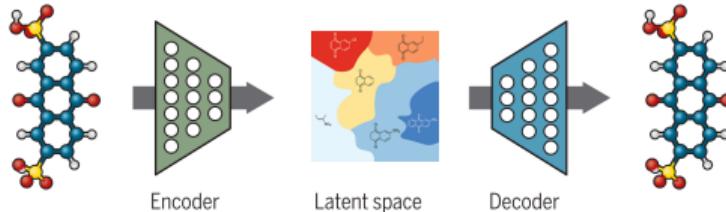
Inverse problem: find a molecule x^* that maximizes a given property f_0

$$x^* = \operatorname{argmax}_{x \in \mathcal{X}} f_0(x)$$

- Black-box, expensive-to-evaluate, function optimization problem.
- Search over the whole molecular space \mathcal{X} a large discrete space ($\approx 10^{60}$).
- Usually we only have access to a database $\mathcal{X}_u \subset \mathcal{X}$.
- It might not hold that $x^* \in \mathcal{X}_u$, it might even not have been synthesized yet!

→ Solution: **Deep generative models.**

Latent space optimization using Variational Auto Encoders (VAEs)

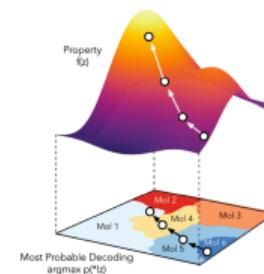
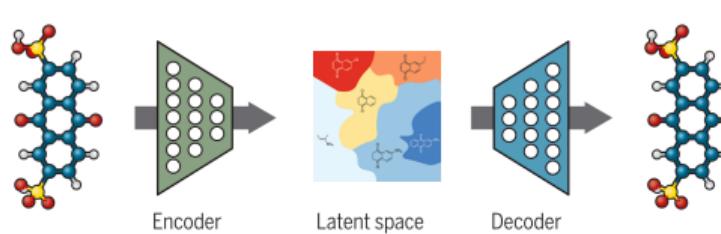


Gómez-Bombarelli *et al.*, 2018

Starting from an unsupervised database \mathcal{X}_u , a VAE learns:

- ① A probabilistic encoder from x into a latent code $z \sim q_\phi(\cdot|x)$, $z \in \mathcal{Z} \subset \mathbb{R}^d$.
- ② A probabilistic decoder from z to $x \sim p_\theta(\cdot|z)$.

Latent space optimization using Variational Auto Encoders (VAEs)



Gómez-Bombarelli *et al.*, 2018

Starting from an unsupervised database \mathcal{X}_u , a VAE learns:

- ① A probabilistic encoder from x into a latent code $z \sim q_{\phi}(\cdot|x)$, $z \in \mathcal{Z} \subset \mathbb{R}^d$.
- ② A probabilistic decoder from z to $x \sim p_{\theta}(\cdot|z)$.

Property optimization in latent space corresponds to

$$z^* = \underset{z \in \mathcal{Z}}{\operatorname{argmax}} g_{\theta}(z) := \mathbb{E}_{x \sim p_{\theta}(\cdot|z)}[f_0(x)] \quad \text{continuous optimisation!}$$

Generative approach: $g_{\theta}(z^*)$ is highly likely not to belong to \mathcal{X}_u .

Current challenges in Latent Bayesian Optimization

- ① **Lack of "functional smoothness":** The latent space doesn't account for the objective f_0 , thus f_0 can be highly non-smooth over \mathcal{Z} .

Current challenges in Latent Bayesian Optimization

- ① **Lack of "functional smoothness":** The latent space doesn't account for the objective f_0 , thus f_0 can be highly non-smooth over \mathcal{Z} .
- ② **Expert are not integrated in the optimization process:** querying the function f_0 can be very expensive. Experts can help uncover f_0 .

Current challenges in Latent Bayesian Optimization

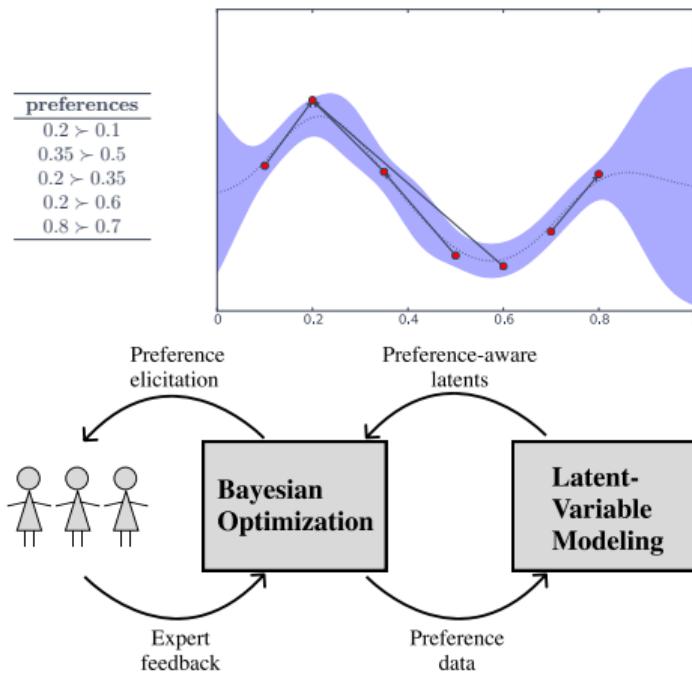
- ① **Lack of "functional smoothness":** The latent space doesn't account for the objective f_0 , thus f_0 can be highly non-smooth over \mathcal{Z} .
- ② **Expert are not integrated in the optimization process:** querying the function f_0 can be very expensive. Experts can help uncover f_0 .

We propose to kill two birds with one stone using **preferential elicitation**.

Experts f_1, \dots, f_d have different goals and areas of expertise → need a model defining the correlation structure between f_0, f_1, \dots, f_d .

Preferential expert elicitation

Dataset $\mathcal{D} = \{x_i > x'_i\}_{i=1}^n$. $x_i > x'_i$ is a R.V. with **value 1** when x_i is preferred to x'_i , o.w. 0.



Ways to model them: probit likelihood, **sign constraints on the derivatives**.

Preferences as gradient sign observations

Preferential queries for f_j can be thought of as observations $\text{sgn}(\partial_z f_j(z))$. These can be modeled and inform us on f_j .

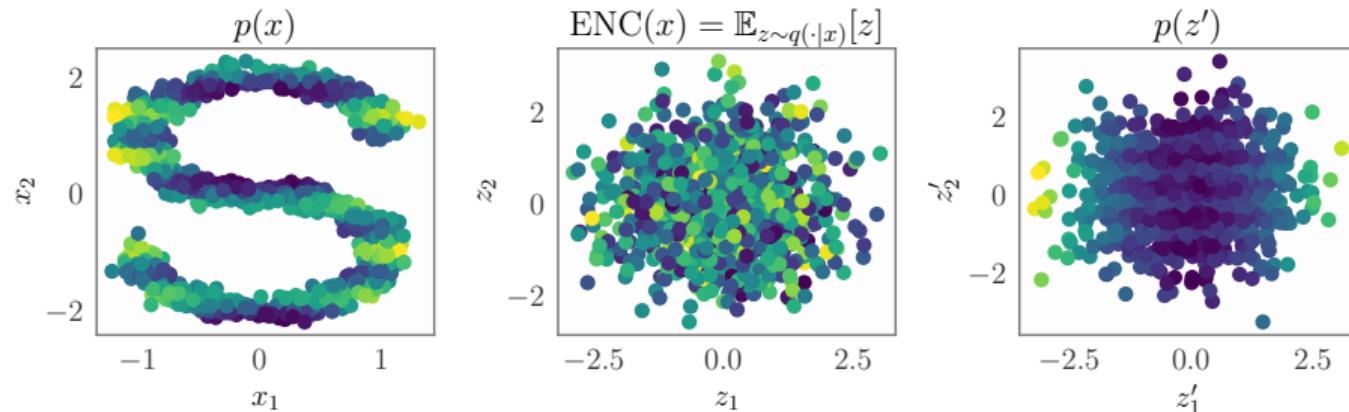
$$\begin{bmatrix} f_1(z) \\ \vdots \\ f_m(z) \\ \partial_z f_1(z) \\ \vdots \\ \partial_z f_m(z) \end{bmatrix} \sim \mathcal{GP}\left(0, \begin{pmatrix} \mathbf{K}(z, z') & \partial_{z'} \mathbf{K}(z, z') \\ \hline \partial_z \mathbf{K}(z, z') & \partial_{zz'}^2 \mathbf{K}(z, z') \end{pmatrix}\right)$$

Where $\mathbf{K}(z, z') = (\text{cov}[f_i(z), f_j(z')])_{1 \leq i, j \leq m} \in \mathbb{R}^{m \times m}$: "low-rank" correlation structure between experts.

⚠ We do not observe $\partial_z f_j(z)$ but $\text{sgn}(\partial_z f_j(z))$.

Property-aware latent space

Use preferential queries to enforce functional smoothness over latent space:



An idea would be to learn a mapping $h_\psi : \mathcal{Z} \rightarrow \mathcal{Z}'$ as an invertible neural net s.t. $z' = h_\psi(\text{ENC}(x))$. ψ would be learned using expert feedback.

Conclusion

- Expert knowledge can be integrated through **probabilistic modeling** and advanced **query strategies**.
- One cornerstone of HITL-ML is BO, which we made **robust to unreliable information sources**, thus paving the way for integration of human feedback.
- Currently tackling the case of multiple, correlated experts, to enhance latent space optimization.

Conclusion

- Expert knowledge can be integrated through **probabilistic modeling** and advanced **query strategies**.
- One cornerstone of HITL-ML is BO, which we made **robust to unreliable information sources**, thus paving the way for integration of human feedback.
- Currently tackling the case of multiple, correlated experts, to enhance latent space optimization.

One perspective: take BO to real-world applications using HITL

Classical issue: effect of **confounders** (e.g. temperature, light...) on objective

- BO with a list of m confounders each with a different query cost
- Human-In-The-Loop to select which confounder to measure
- Case study brought by material scientists