Optimization strategies of

Bayesian global optimization

Platform – Bgolearn | MLMD

Bin CAO

Guangzhou Municipal Key Laboratory of Materials Informatics
The Hong Kong University of Science and Technology (GZ)

Jan. 6-th, 2024

https://github.com/Bin-Cao (binjacobcao@gmail.com)

Self introduction

曹斌 (Bin CAO)

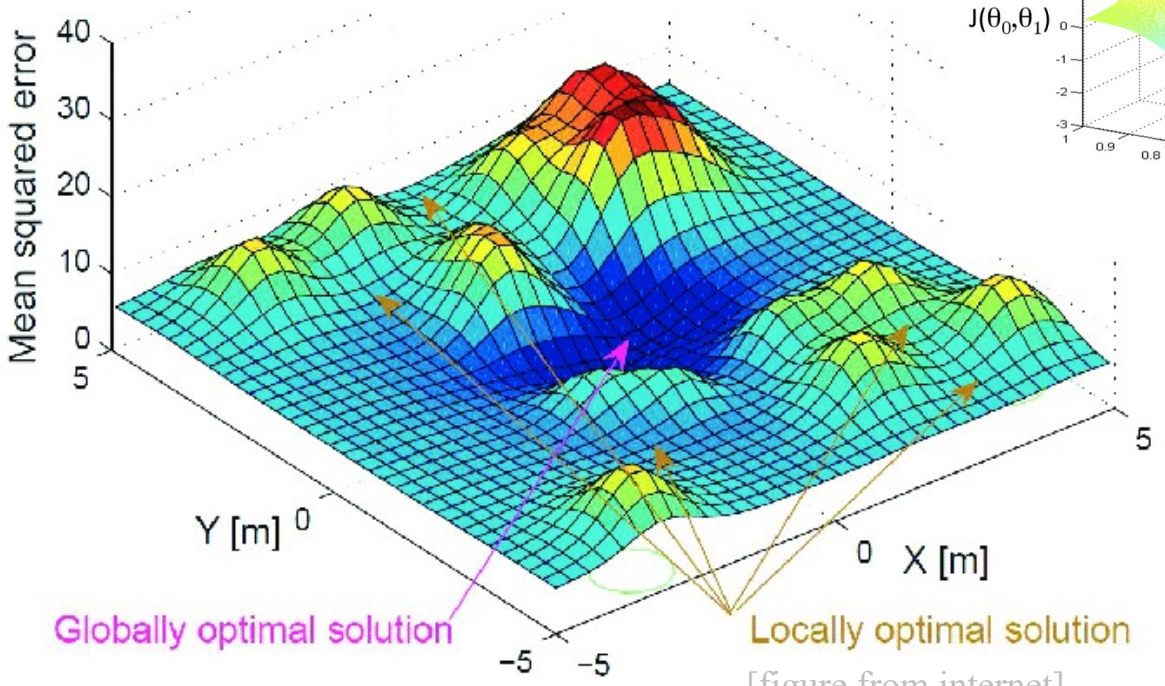
- 👨‍🎓 2023.9-Present : Hong Kong University of Science and Technology(GZ） / **PHD student** / Supervisor : Prof. Zhang Tongyi
- 👨‍🎓 2020.9-2023.6 : Shanghai University / **Mphil** / Supervisor : Prof. Zhang Tongyi
- 👨‍🎓 2016.9-2020.6 : Beijing University of Chemical Technology / **Bachelor**

Member of Chinese Crystallographic Society (CCrS) ;

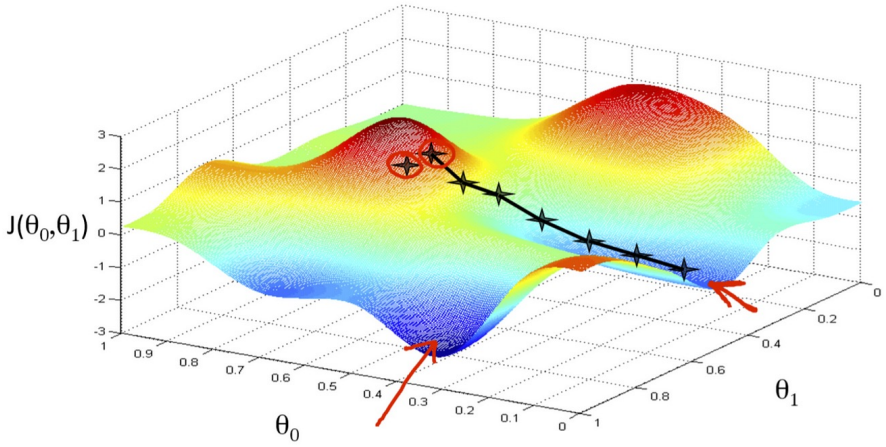
working on ML for crystal structure analysis, using X-ray spectral analysis technology and machine learning techniques. ML-based X-ray technology (XRD、XAFS、XPS) https://orcid.org/0000-0001-7273-6779

Non-convex Optimization

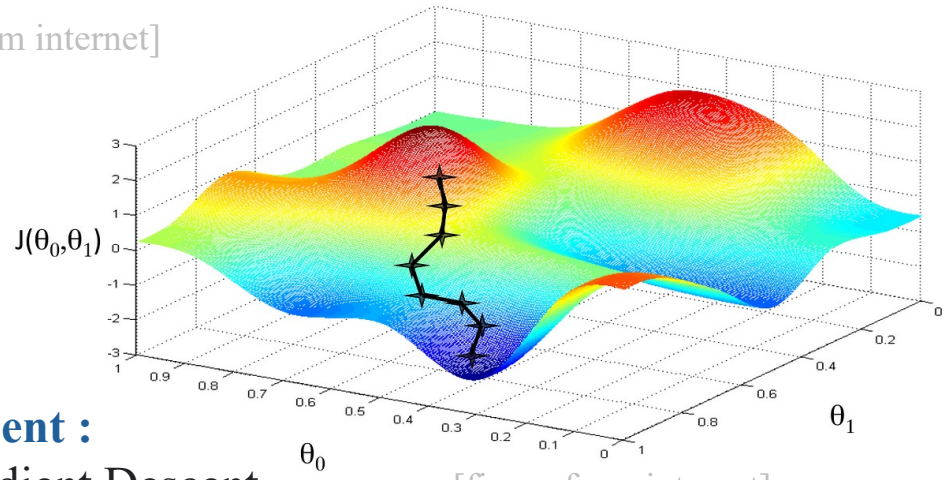
A non-convex optimization problem is characterized by having a non-convex objective function or non-convex constraints.



[figure from internet]



[figure from internet]



[figure from internet]

The function is expressed analytically

Gradient descent :

Stochastic Gradient Descent

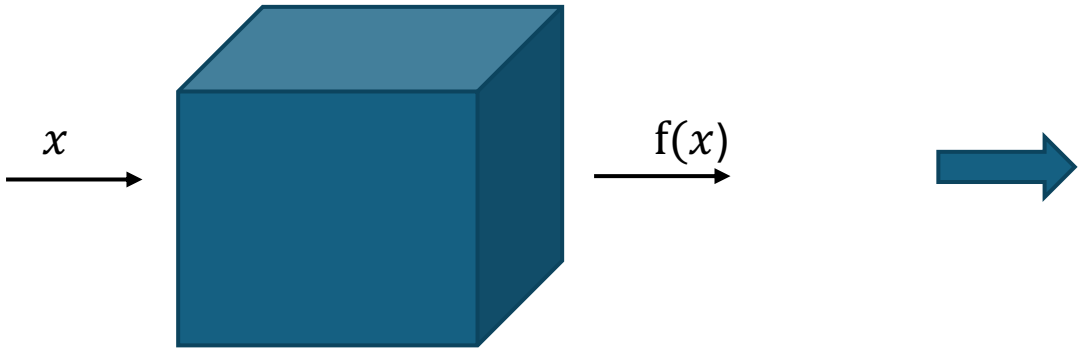
Steepest Gradient Descent

Newton Gradient Descent …

Non-convex Optimization

Goal : solve argmin_x f(x)

Where f(x) is a black box and is a time-consuming-to-evaluate function



Heuristic optimization algorithm:
particle swarm algorithm
simulated annealing algorithm
genetic algorithm
…

Bayesian optimization:
Expected Improvement algorithm
Upper confidence bound
Predictive Entropy Search
Knowledge Gradient
…

Diffacults :

The analytical expression of the function is **not known**
The computation of f(x) is associated with **significant time cost**

Target of Optimization

The optimization algorithms applied to black-box problems are collectively geared towards solving the challenge of effective sampling

$x_1 \rightarrow \mathrm{f}(x_1)$
$x_2 \rightarrow \mathrm{f}(x_2)$
...
$x_n \rightarrow \mathrm{f}(x_n)$
(n is **a finite number**)

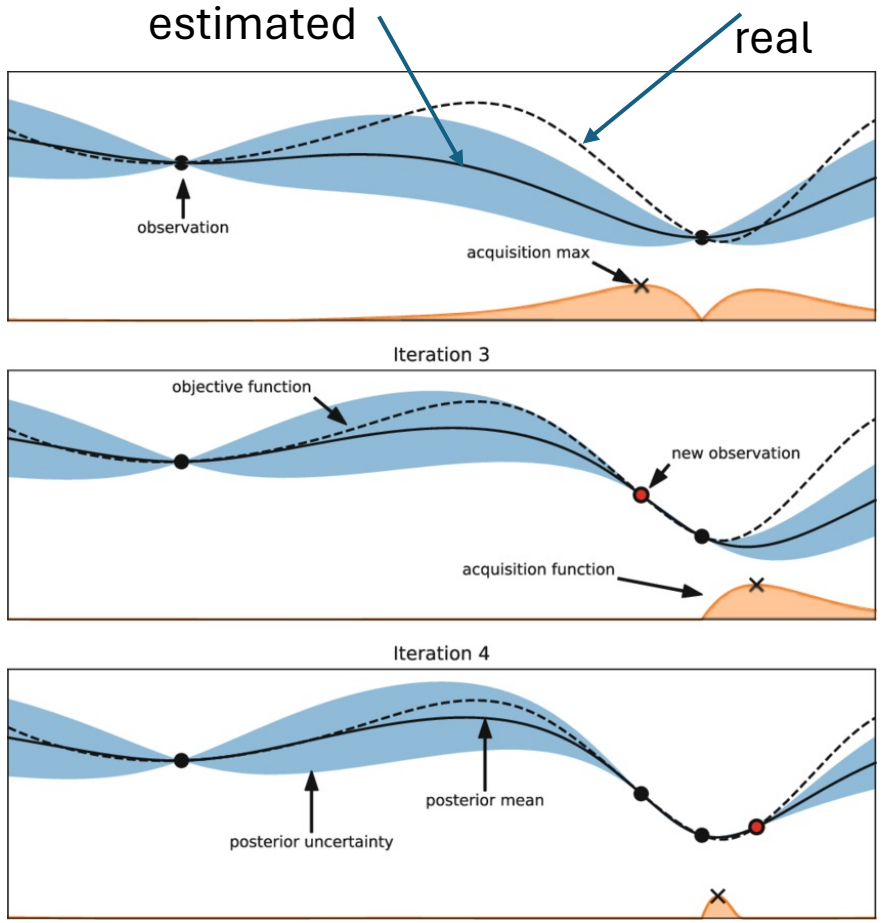
$x_* \rightarrow \mathrm{f}(x_*)$

Many optimization algorithms leverage historical information to guide the selection of the next sampling point, determining which infill data,

$\boldsymbol{x_{n+1} \rightarrow f(x_{n+1})}$ is most informative in the pursuit of deriving the optimal solution x_*.

The differences

All optimization algorithms incorporate a fundamental concept of an optimization (response) surface



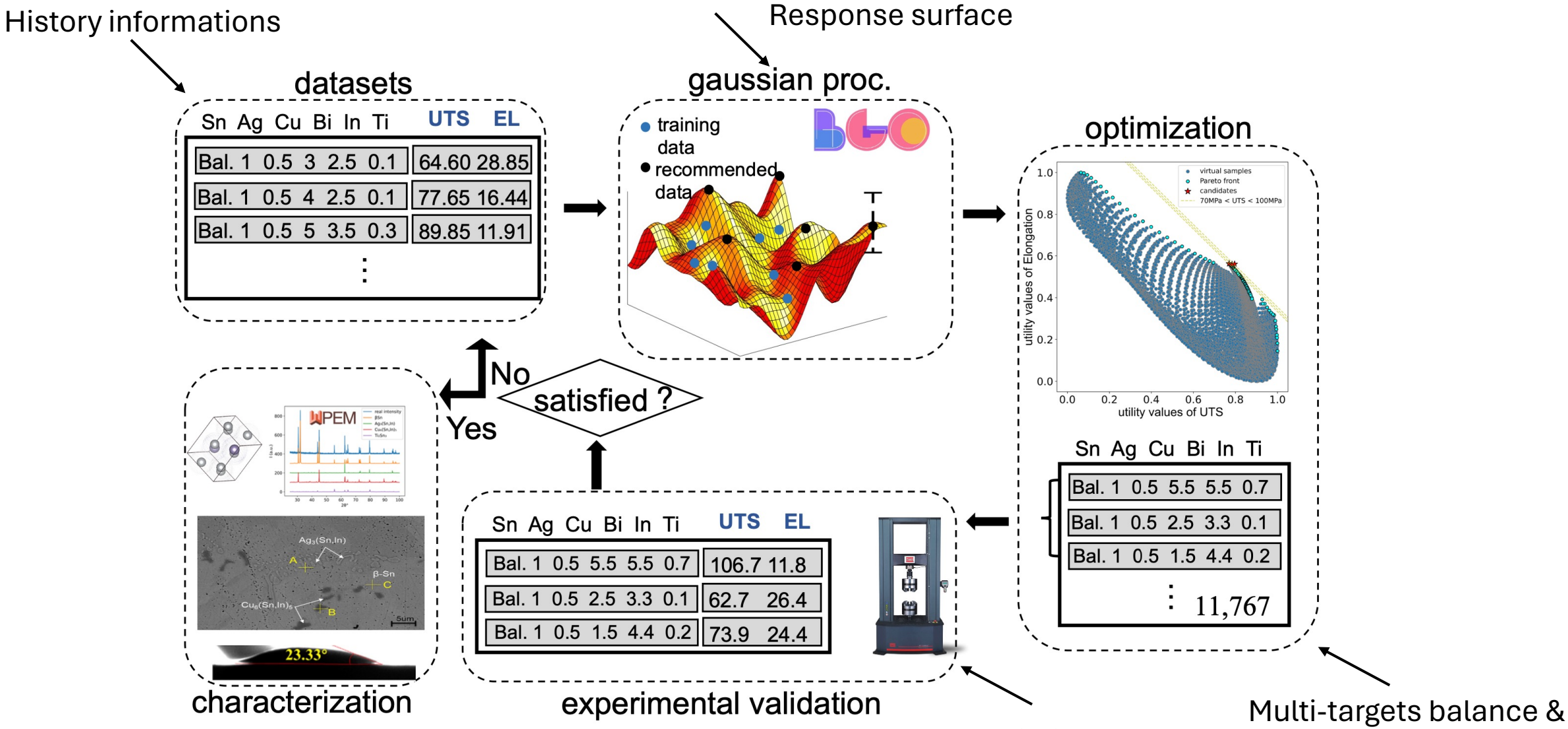
[figure from internet]

The optimization surface **undergoes changes** with the introduction of infill data.

This response surface can be **implicit**, as seen in heuristic optimization, or explicit, as **observed** in Bayesian optimization.

A notable distinction lies in Bayesian optimization's approach, as **it does not unconditionally trust the derived response surface**. Instead, it carefully navigates the **exploration-exploitation trade-off**, balancing the need for exploring new regions with exploiting the current knowledge to achieve optimal results.

Baysian global Optimization



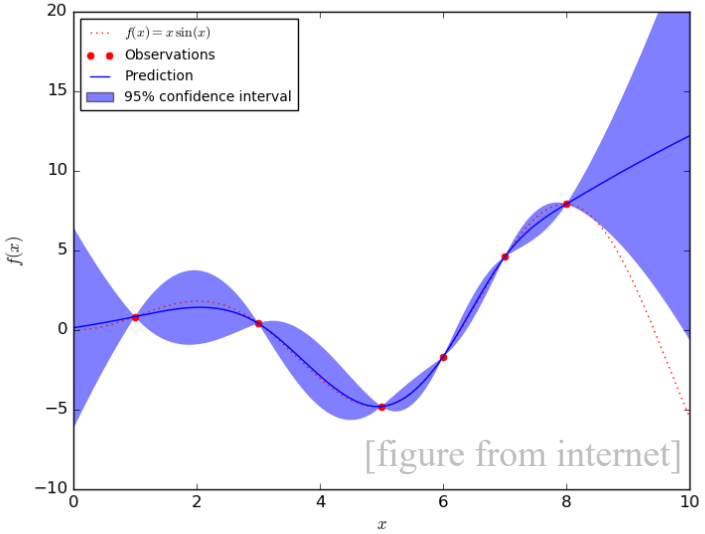
Cao, Bin and Su, Tianhao … and Zhang, Tong-Yi.Available at SSRN: https://ssrn.com/abstract=4686075 or http://dx.doi.org/10.2139/ssrn.4686075

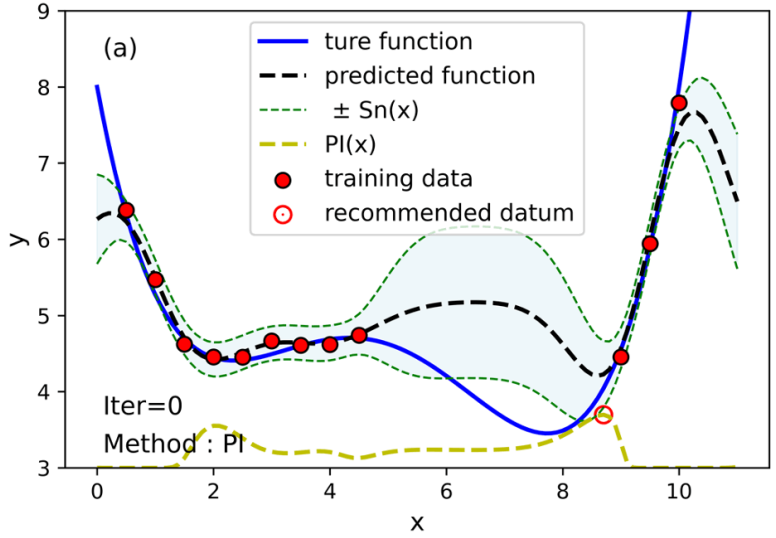
Baysian global Optimization : Response surface

The first and most important step is to **establish the response surface** with collect data pairs

$$x_1 \rightarrow f(x_1)$$
$$x_2 \rightarrow f(x_2)$$
$$\dots$$
$$x_n \rightarrow f(x_n)$$

A ML model





Iterations	x_i	$y_i(x_i)$
1-th	8.7000	4.0418

The reliability is insufficient

Bgolearn provided **three modules** for establishing the surface :

1. Gaussian Process-Based
2. Boosting Sampling-Based (SVM, RF, MLP, etc.)
3. Multi-model-Based

Baysian global Optimization : Virtual space

In BGO, the virtual space refers to the area constituted by all unevaluated data at the current step. In a discrete scenario, we are discussing situations where the data points are **distinct**.

In Mathematic :

$x_1 \rightarrow f(x_1)$
$x_2 \rightarrow f(x_2)$
...
$x_n \rightarrow f(x_n)$

Data be recorded

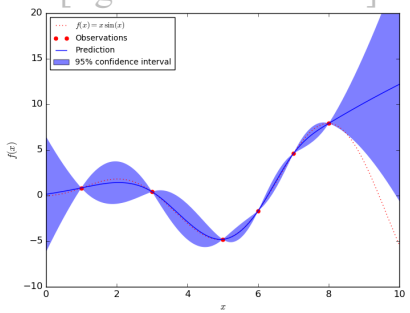
V : $x_i(i \notin (1,2,\dots n))$

The data within the virtual space, denoted as V, must encompass the optimal solution

$x_1 \rightarrow f(x_1)$
$x_2 \rightarrow f(x_2)$
...
$x_n \rightarrow f(x_n)$

→ ML mdoel →

[figure from internet]



Is x_i with the minimum value of $g(x_i)$ considered the best one? In BGO, the model is not entirely convinced, highlighting the exploration-exploitation tradeoff

All data within the virtual space are assessed by the response surface, **taking into account associated uncertainties**

V : $x_i(i \notin (1,2,\dots n))$

G : $g(x_i)(i \notin (1,2,\dots n))$

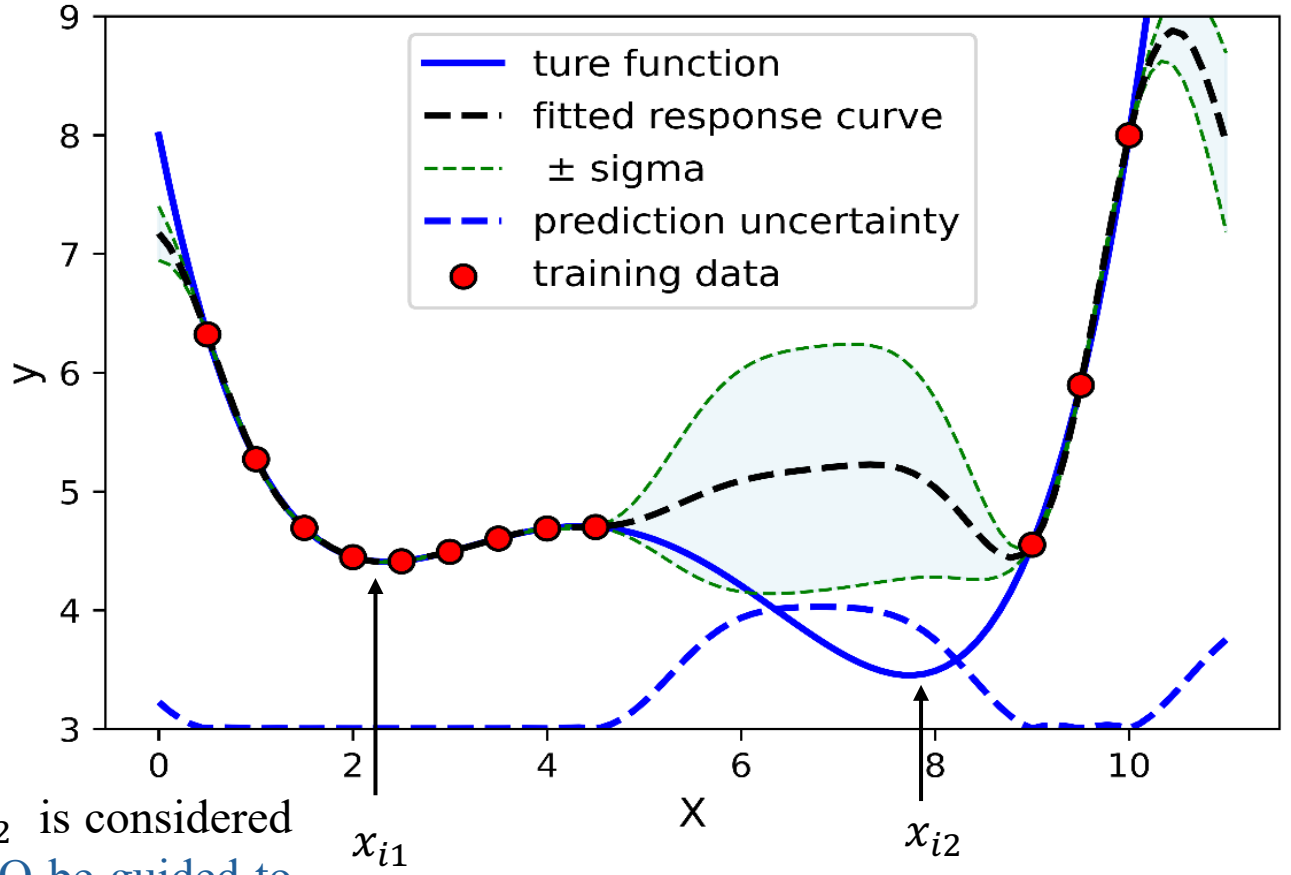
Baysian global Optimization : Ranking

Through the computation of the response surface, we can obtain all the evaluated function values and **their associated uncertainties** for each data point in V

V : $x_i\left(i \notin (1,2,\dots n)\right)$

G : $g(x_i)\left(i \notin (1,2,\dots n)\right)$
S : $\mathrm{std}(x_i)\left(i \notin (1,2,\dots n)\right)$

We use 'g' to denote the evaluated **means** and 'std' to represent the evaluated **standard variance.**



While x_{i1} exhibits the lowest $g(x_{i1})$, however x_{i2} is considered optimal from a global perspective. How can BGO be guided to recommend solutions closer to x_{i2}?

Exploration and Exploitation

How China discovered new oil fields ?

[figure from internet]

In regions with existing oil fields (A sites), there is a heightened likelihood of discovering new oil fields. Conversely, provinces lacking oil fields (B sites) exhibit the potential for discovering larger reserves of oil.

Prioritizing attention on A sites may diminish the probability of finding larger reserves. Meanwhile, focusing efforts on B sites may seem futile, but there have a probability of discovering larger reserves. The challenge lies in the strategic choice between these options.

风浪越大，鱼越贵

[figure from internet]

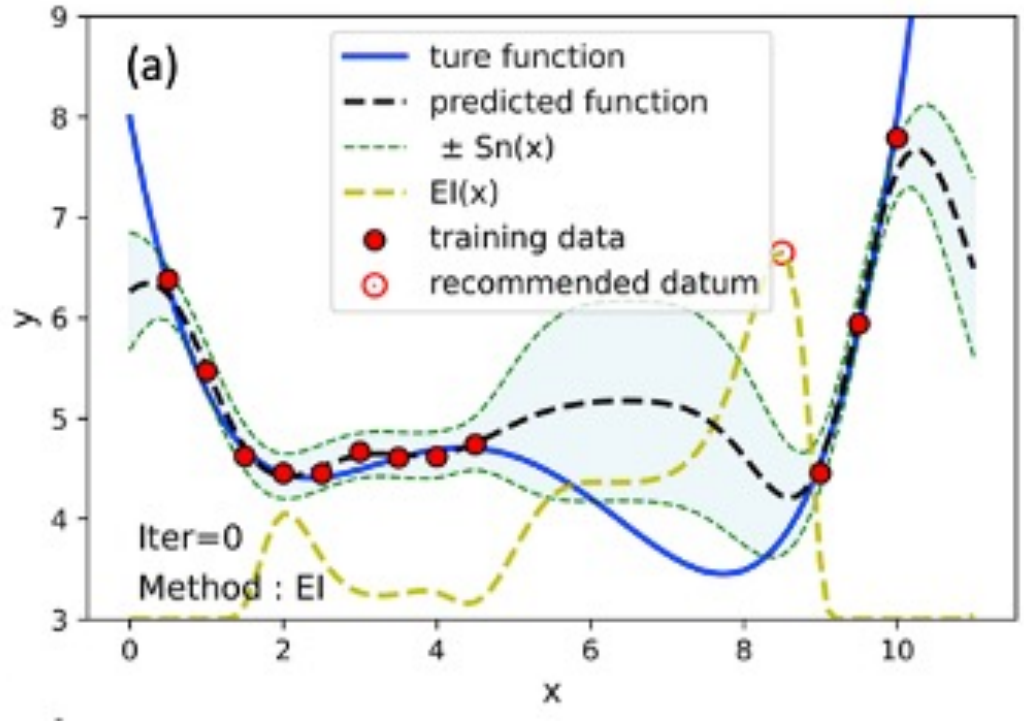
Exploration and Exploitation

The utility function is incorporated into BGO, exemplified by the Expectation Maximization algorithm

$$I_n(\boldsymbol{x}) = max\left(f_{min}^{(n)} - g(\boldsymbol{x}), 0\right)$$

$$EI_n(\boldsymbol{x}) = E_{\hat{y}_n(\boldsymbol{x})}\left(max\left(f_{min}^{(n)} - \hat{g}_n(\boldsymbol{x}), 0\right)\right)$$

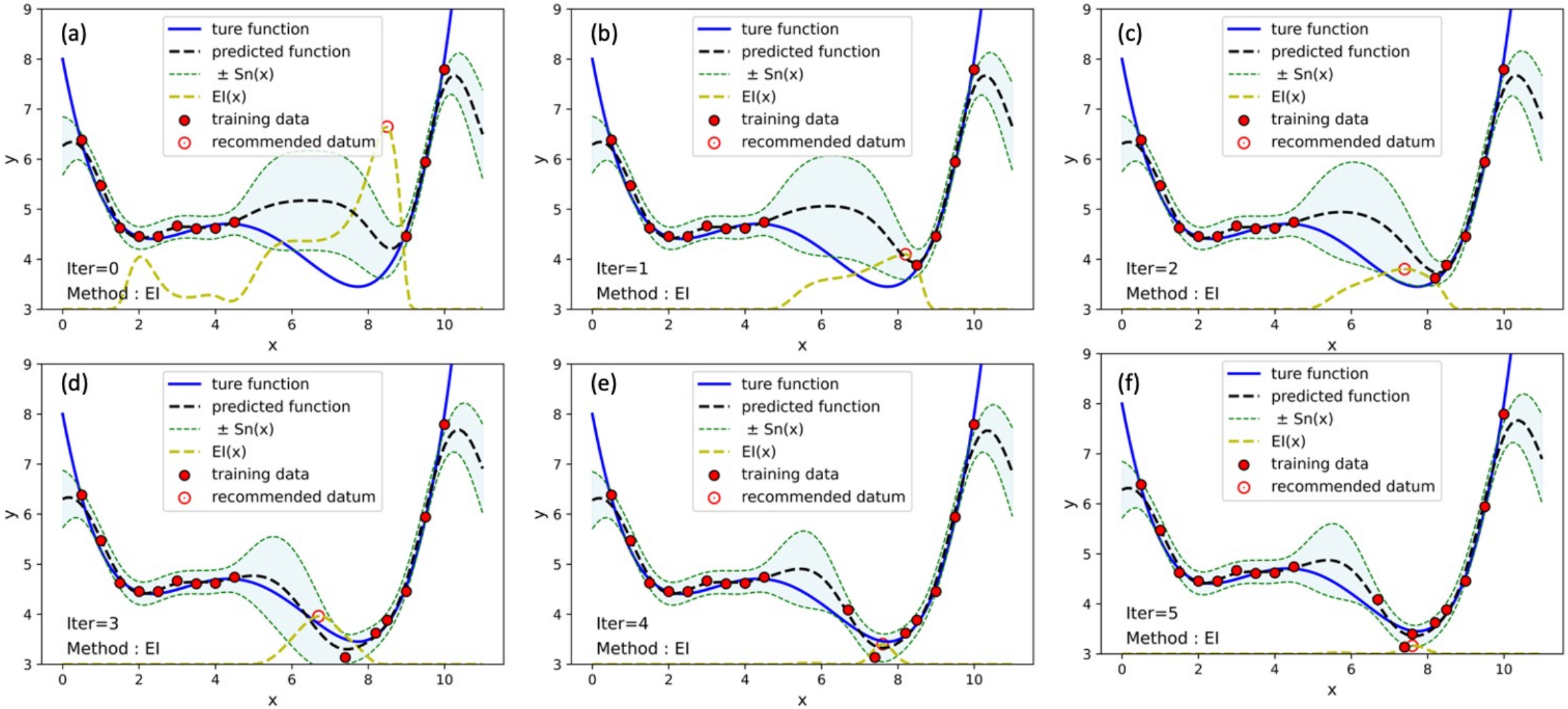
$$EI_n(\boldsymbol{x}) = s_n\left[\left(\frac{f_{min}^{(n)} - g_n}{s_n}\right)\Phi\left(\frac{f_{min}^{(n)} - g_n}{s_n}\right) + \phi\left(\frac{f_{min}^{(n)} - g_n}{s_n}\right)\right]$$



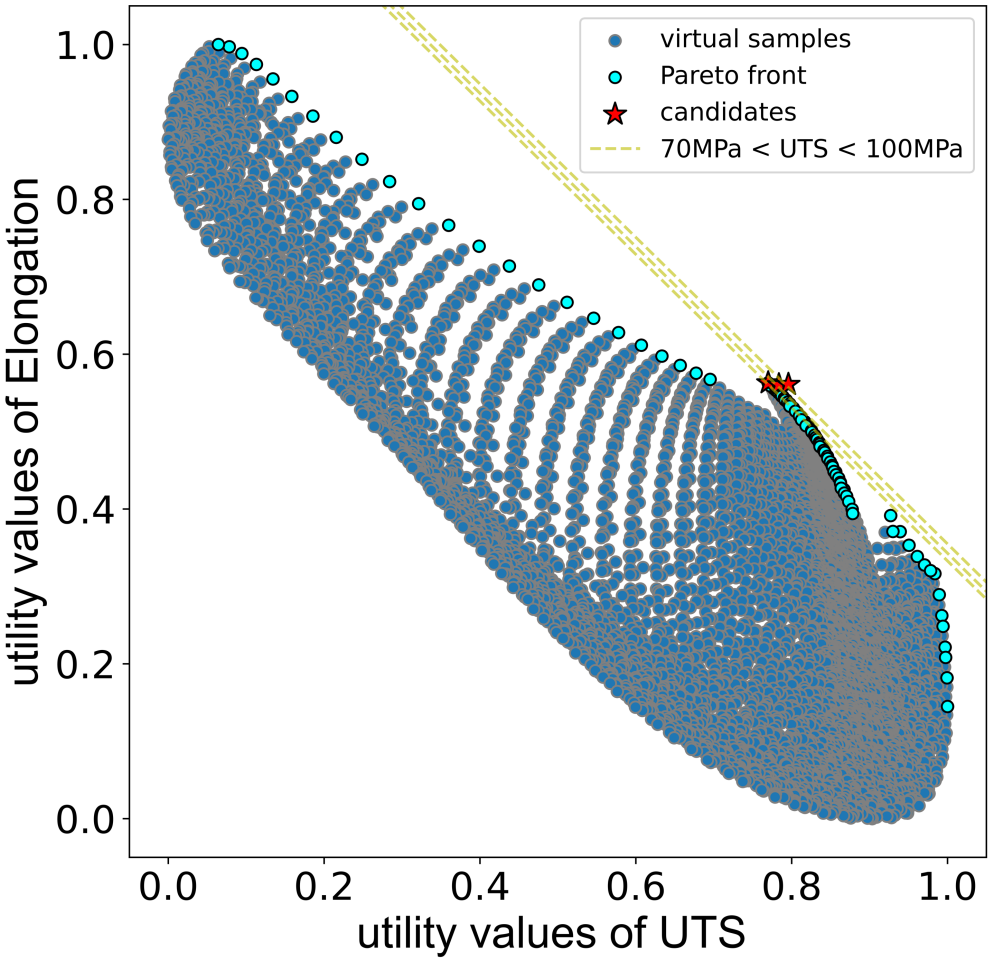
Recommendations are made **by considering both the evaluated values and their associated uncertainties**. The recommended data must then undergo measurement to acquire the real function value (experiments or simulations)

Exploration and Exploitation

The response surface undergoes updates with each iteration



Multi-targets selection :



In the context of multi-tasking, a straightforward approach involves evaluating each data point in the virtual space twice. This yields the 'score' for each data, and a **Pareto space** is then formed.

Selecting a data point from the Pareto front involves various methods:

1.Diagonal search

2.Distance search

3.Equal value search

4.Crowding distance search

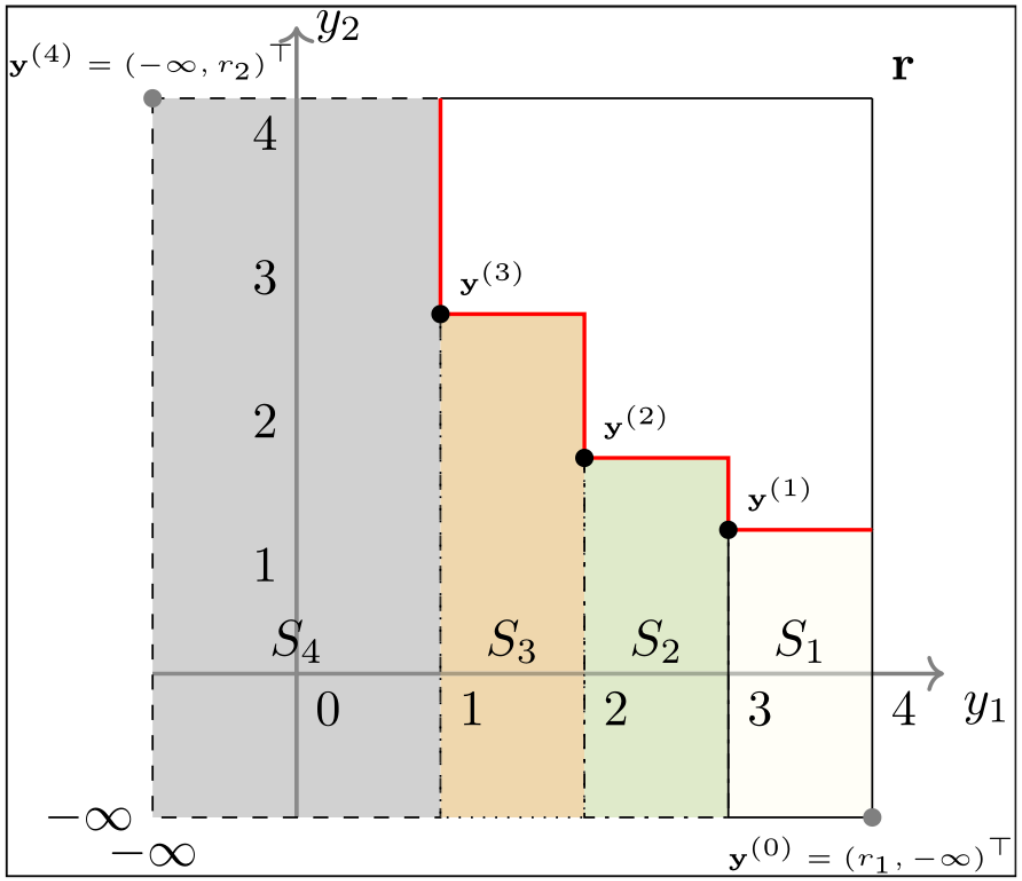
5. …

Each method **recommends points located on the front**, contributing to the multi-task optimization process.

Multi-targets BGO :

In multi-target BGO, the evaluation of infill data focuses on enhancing the influence **across all targets**.

Hypervolume Improvement



[figure from https://doi.org/10.1016/j.swevo.2018.10.007]

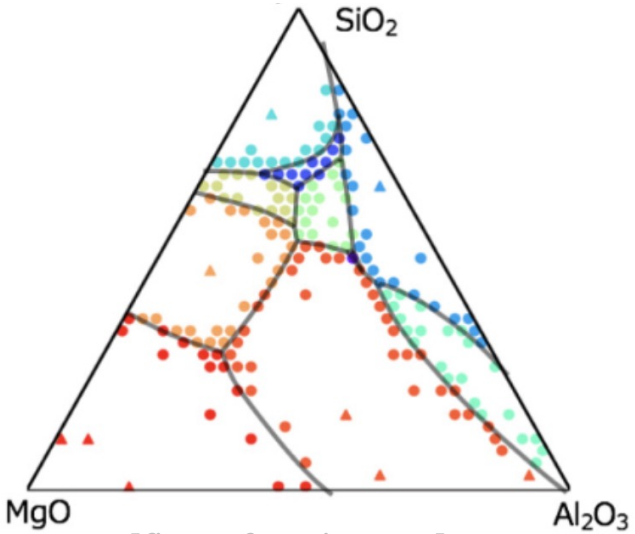
Algorithm 1 MOBGO algorithm.

Input: Objective functions $\mathbf{y}$, initialization size μ, termination criterion T_c
Output: Pareto-front approximation $\mathcal{P}$

1 Initialize μ points $(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(\mu)})$;
2 Evaluate the initial set of μ points: $(\mathbf{y}^{(1)} = \mathbf{y}(\mathbf{x}^{(1)}), \ldots, \mathbf{y}^{(\mu)} = \mathbf{y}(\mathbf{x}^{(\mu)}))$;
3 Store $(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(\mu)})$ and $(\mathbf{y}^{(1)} = \mathbf{y}(\mathbf{x}^{(1)}), \ldots, \mathbf{y}^{(\mu)} = \mathbf{y}(\mathbf{x}^{(\mu)}))$ in D: $D = ((\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \ldots, (\mathbf{x}^{(\mu)}, \mathbf{y}^{(\mu)}))$;
4 Compute the non-dominated subset of D and store it in $\mathcal{P}$;
5 $g = \mu$;
6 **while** $g <= T_c$ **do**
7 Train Kriging models $M_1, \cdots, M_d$ based on D ;
8 Use an optimizer (opt) to find the promising point $\mathbf{x}^*$ based on surrogate models M, with the infill criterion C;
9 Update D: $D = D \cup (\mathbf{x}^*, \mathbf{y}(\mathbf{x}^*))$;
10 Update $\mathcal{P}$ as the non-dominated subset of D;
11 $g = g + 1$;
12 Return $\mathcal{P}$

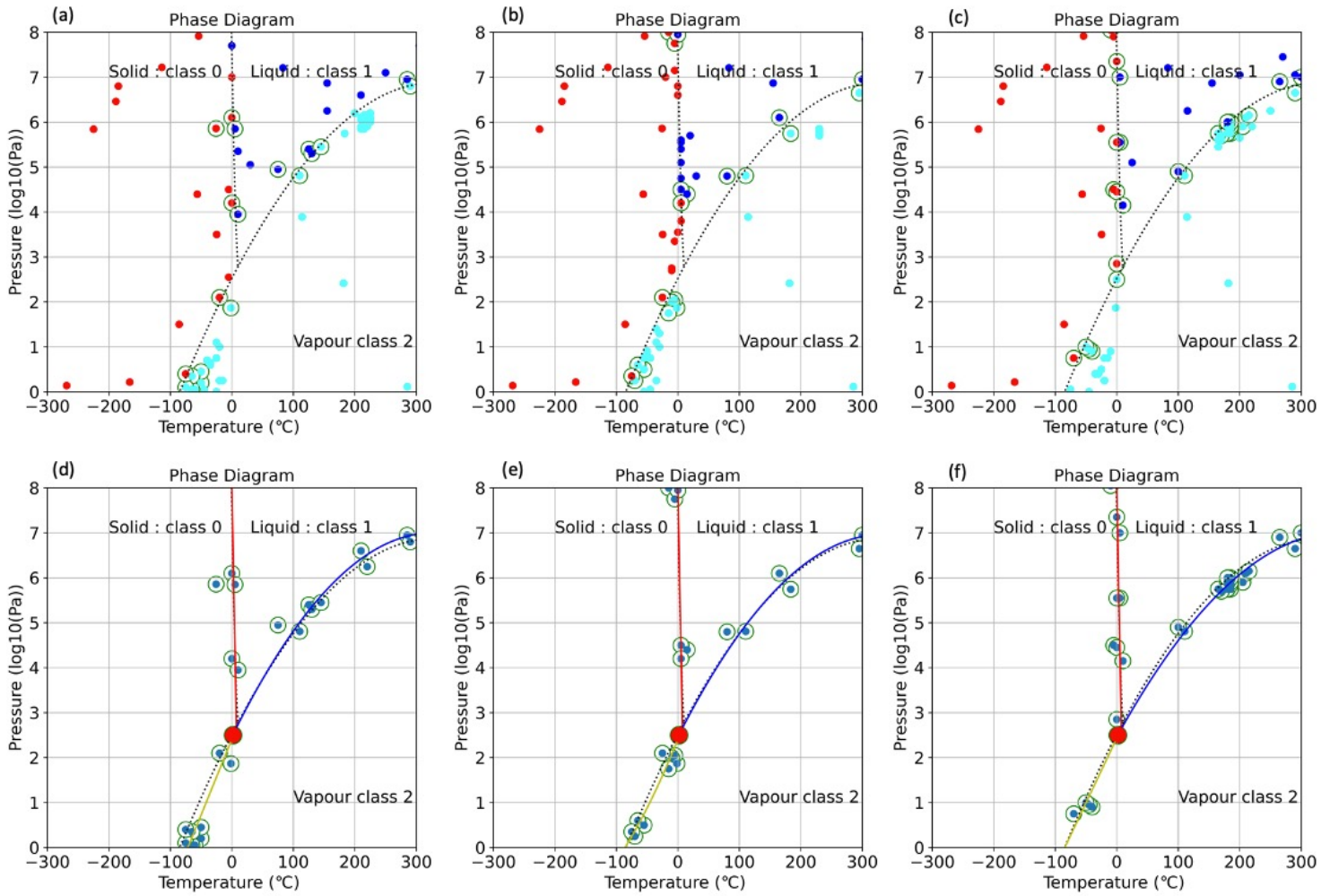
BGO classification

For **classification** problems, BGO proves to be an excellent optimization method in identifying boundaries between different categories



[figure from internet]

Find the phase boundaris



Bgolearn is a computational approach that enables material composition-oriented design and performance-oriented optimization. By leveraging existing experimental data, Bgolearn searches for the optimal material composition design within a specified composition space in order to maximize or minimize the desired performance metric.

Only takes three steps

step 1 ：

```
pip install Bgolearn
```

step 2 ：

```
# import your dataset (Samples have been characterized)
# 导入研究的数据集(已经表征过的样本)
data = pd.read_csv('data.csv')
x = data.iloc[:,:-1]
y = data.iloc[:,-1]
# virtual samples which have same feature dimension with x
# 设计的虚拟样本, 与x具有相同的维度
vs = pd.read_csv('VS.csv')
```

step 3 ：

```
# instantiate class
# 实例化类
Bgolearn Bgolearn = BGOS.Bgolearn()
# Pass parameters to the function
# 传入参数
Mymodel = Bgolearn.fit(data_matrix = x,
Measured_response = y, virtual_samples = vs)
# derive the result by EI
# 通过EI导出结果
Mymodel.EI()
```

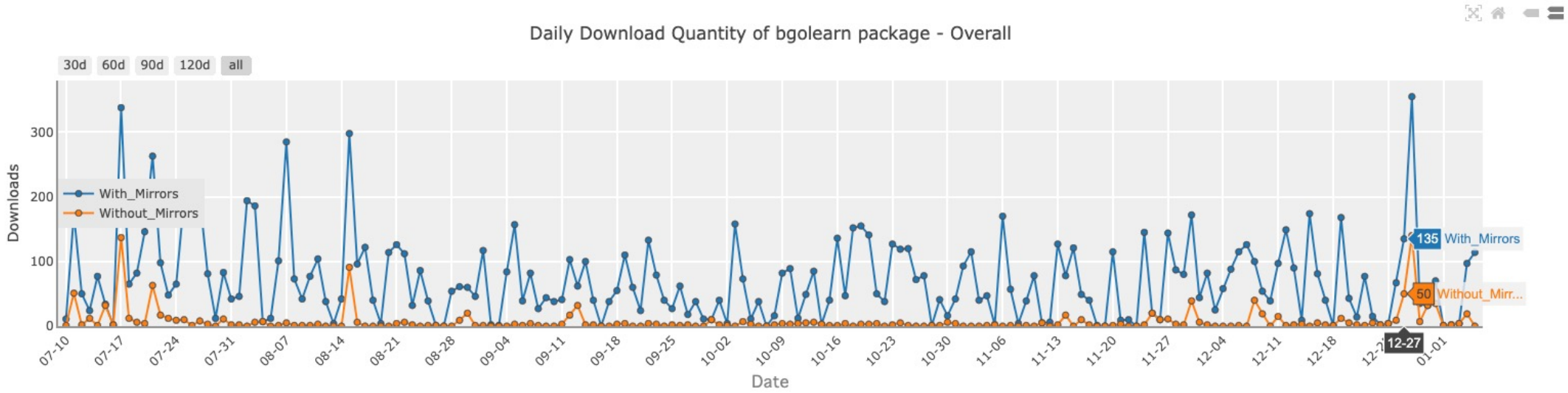
for regression

- 1.Expected Improvement algorith (期望提升函数)
- 2.Expected improvement with “plugin” (有 “plugin”的期望提升函数)
- 3.Augmented Expected Improvement (增广期望提升函数)
- 4.Expected Quantile Improvement (期望分位提升函数)
- 5.Reinterpolation Expected Improvement (重插值期望提升函数)
- 6.Upper confidence bound (高斯上确界函数)
- 7.Probability of Improvement (概率提升函数)
- 8.Predictive Entropy Search (预测熵搜索函数)
- 9.Knowledge Gradient (知识梯度函数)

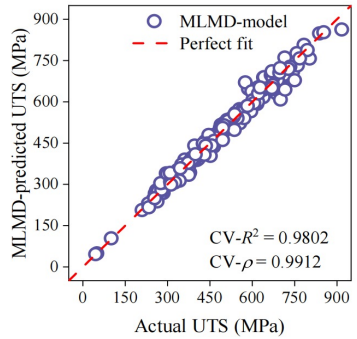
for classification

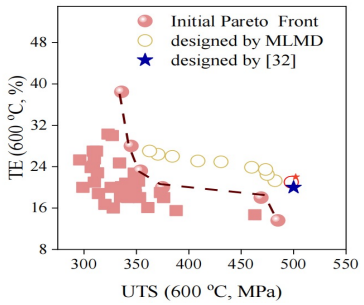
- 1.Least Confidence (欠信度函数)
- 2.Margin Sampling (边界函数)
- 3.Entropy-based approach (熵索函数)

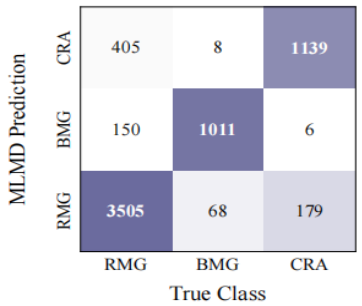
Preserved in various domestic and international mirrors, including Tsinghua University, Alibaba, Tencent, University of Science and Technology of China, etc., with over **5,000 downloads**

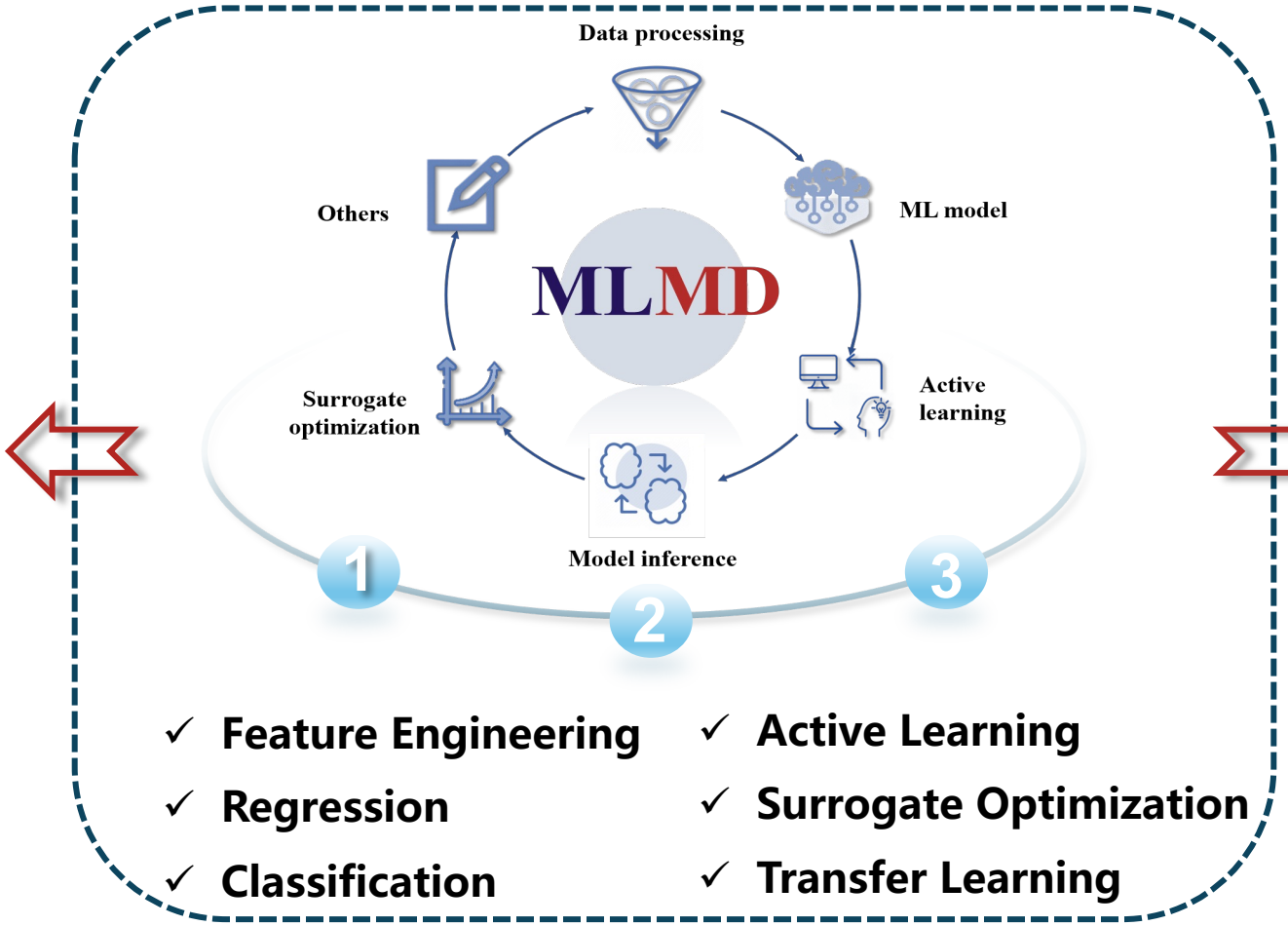


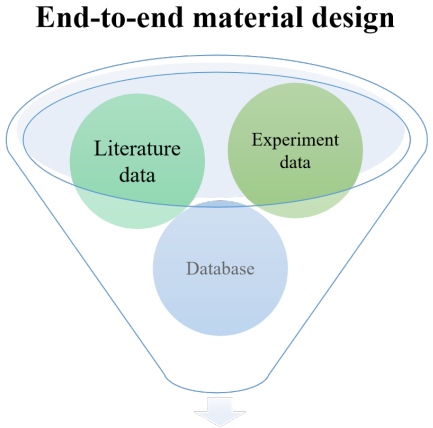
MLMD —Materials design platform









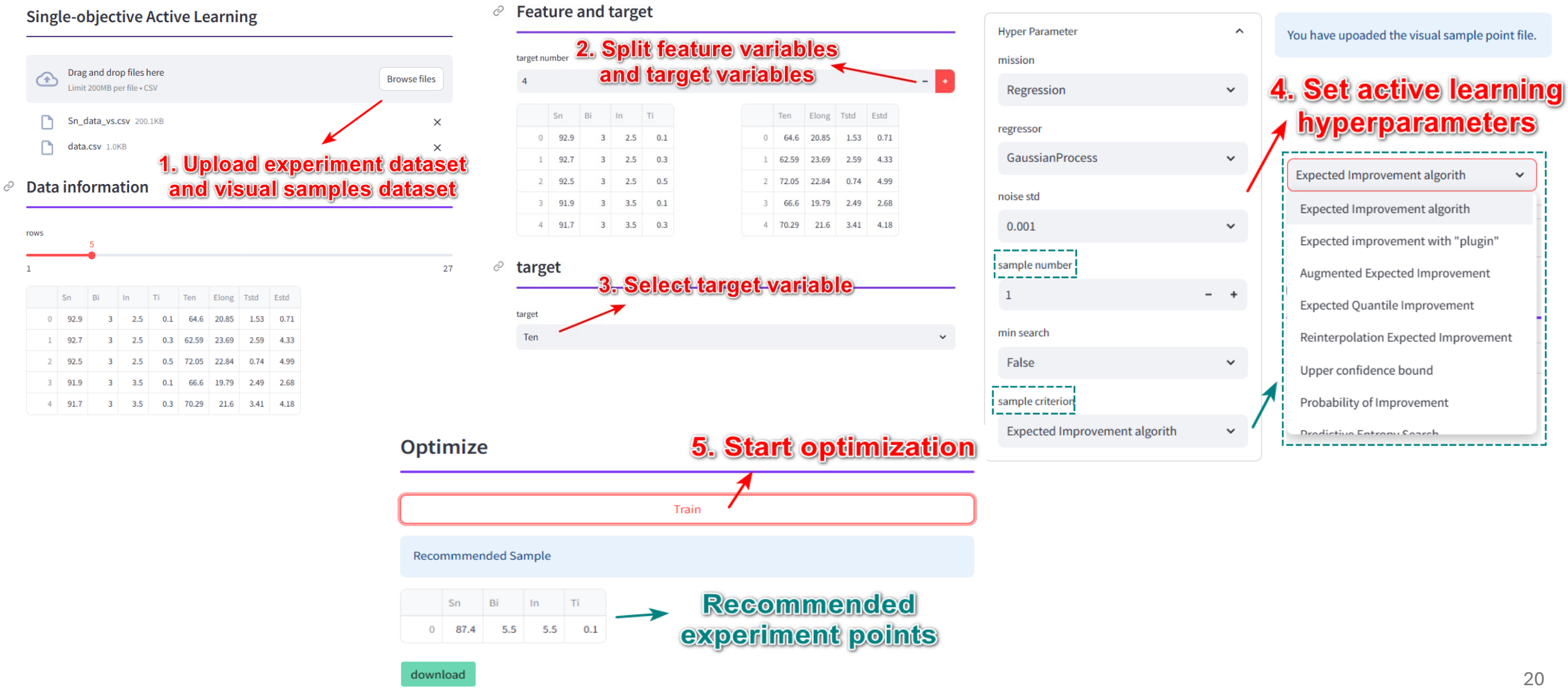


- ✓ Feature Engineering
- ✓ Regression
- ✓ Classification
- ✓ Active Learning
- ✓ Surrogate Optimization
- ✓ Transfer Learning

MLMD Code

MLMD (**M**achine **L**earning for **M**aterial **D**esign) aims at utilizing general and frontier AI algorithms to accelerate the end-to-end material design with programming-free.

Single-objective active learning in MLMD



Multi-objective active learning in MLMD

Multi-objective Active Learning

Drag and drop files here
Limit 200MB per file • CSV
Browse files

Sn_data_vs.csv 200.1KB
data-multi.csv 0.8KB

1. Upload experiment dataset and visual samples dataset

Data information

rows: 5 (1 – 27)

	Sn	Bi	In	Ti	Ten	Elong
0	92.9	3	2.5	0.1	64.6	20.85
1	92.7	3	2.5	0.3	62.59	23.69
2	92.5	3	2.5	0.5	72.05	22.84
3	91.9	3	3.5	0.1	66.6	19.79
4	91.7	3	3.5	0.3	70.29	21.6

Feature and target

target number: 2

2. The last two columns as target variables

	Sn	Bi	In	Ti
0	92.9	3	2.5	0.1
1	92.7	3	2.5	0.3
2	92.5	3	2.5	0.5
3	91.9	3	3.5	0.1
4	91.7	3	3.5	0.3

	Ten	Elong
0	64.6	20.85
1	62.59	23.69
2	72.05	22.84
3	66.6	19.79
4	70.29	21.6

target

target: Ten × Elong ×

Ten ref location: 0

Elong ref location: 0

Optimize

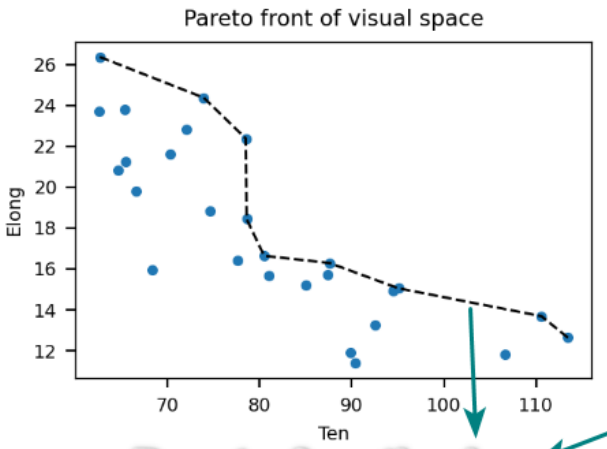
3. Start optimization

Opt

	Sn	Bi	In	Ti
0	93.1	2.4	2.9	0.1

Recommended experiment points

download



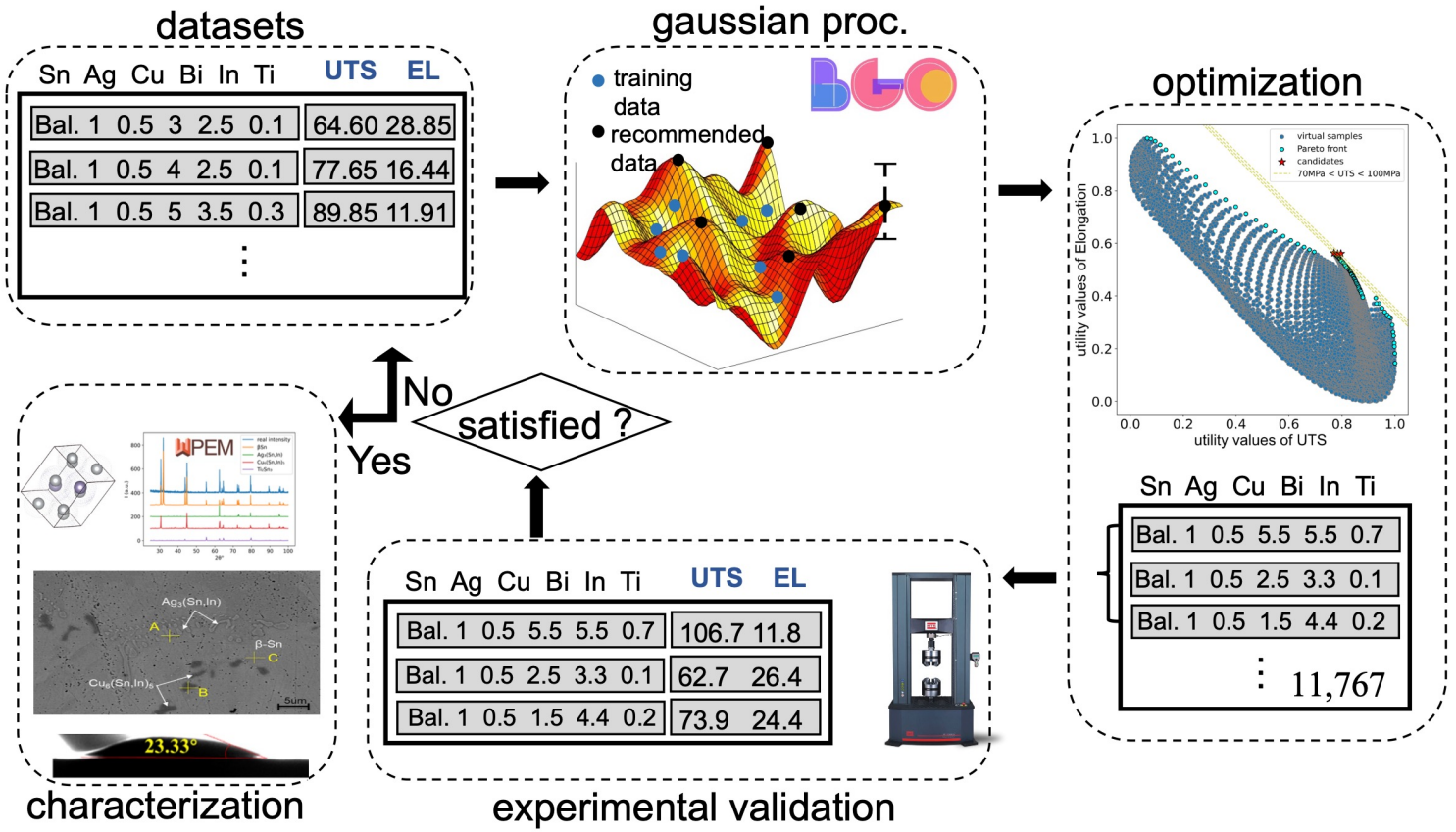
Pareto frontier in initial dataset

	Ten	Elong
0	113.44	12.64
1	110.56	13.69
2	95.07	15.05
3	87.58	16.29
4	80.49	16.64
5	78.59	18.44
6	78.49	22.38
7	73.94	24.37
8	62.67	26.36

download

Baysian global Optimization

Numerous scientific problems require careful consideration



Cao, Bin and Su, Tianhao … and Zhang, Tong-Yi.Available at SSRN: https://ssrn.com/abstract=4686075 or http://dx.doi.org/10.2139/ssrn.4686075

1. "What types of models and methodologies can be employed to estimate the mean and variance of candidates? (DeepNN, for instance, excels in high-dimensional spaces.)"

2. "How can one define a reasonable search space (virtual space)? (A space that is too large may introduce excessive randomness, while one that is too small might result in missing the optimal solutions.)"

3. "How does one strike a balance in the exploration-exploitation trade-off? (Especially, how should uncertainty be considered in multi-task optimization?)"

…

香港科技大学(广州)
THE HONG KONG
UNIVERSITY OF SCIENCE AND
TECHNOLOGY (GUANGZHOU)

Thank you!

If you use the PPT, please quote it as follows :

Bin CAO. (2024). Bgolearn: A Bayesian global optimization package. Retrieved from https://github.com/Bin-Cao/Bgolearn