

---

# SELECTION OF OUTPUT MODE FOR BAYESIAN OPTIMIZATION IN NOISY PROBLEMS

---

A PREPRINT

 **Chenxi Li\***

Department of Information Systems and Management Engineering  
Southern University of Science and Technology  
Shenzhen, China  
12333226@mail.sustech.edu.cn

September 2, 2024

## ABSTRACT

This paper investigates the application of Bayesian optimization for selecting final output results in noisy optimization problems. In noise-free scenarios, the best observation can be directly utilized as the final optimization result. However, in the presence of noise, each observation is affected, making it inappropriate to directly use the best observation. Considering that surrogate models in Bayesian optimization can provide predicted means for any point in the search space, and that these predictions tend to be more reliable than actual noisy observations, we propose a result output method based on the predicted values of the surrogate models. For instance, when aiming to find the minimum of an objective function, we can output the point with the lowest predicted mean among all observed points, or the point with the lowest predicted mean in the entire search space. These methods may be more suitable than using direct observations, especially in high-noise environments. However, the choice of output method is influenced not only by the noise level but also by factors such as the response surface characteristics and the dimensionality of the search space. In this paper, we conduct numerical experiments using the BBOB function set to evaluate the advantages and disadvantages of different output methods for Bayesian optimization in varying noise environments, and provide informed recommendations for selecting output methods.

**Keywords** Bayesian optimization · Gaussian process · blackbox optimization · noisy problems · BBOB

## 1 Introduction

Bayesian optimization is recognized as an advanced framework for black-box optimization, distinguished by its efficiency in approximating the global optimum with a limited number of objective function evaluations. This efficiency has led to its widespread adoption across various fields, including drug research and development, as well as cost-effective optimization tasks such as tuning neural network parameters and other objective function evaluations [Shahriari et al., 2016]. The algorithm operates as a model-based sequential optimization method. At each iteration, Bayesian optimization devises a strategy that balances exploration and exploitation based on the current information to select the next evaluation point. Specifically, Bayesian optimization consists of two main components: the surrogate model and the acquisition function. The surrogate model, typically a Gaussian process [Rasmussen and Williams, 2005], simulates the objective function based on current observation points. The acquisition function then leverages this model to determine the location of the next observation point, thereby optimizing outcomes more efficiently with fewer evaluations.

Bayesian optimization demonstrates its efficiency in many scenarios, especially in low-dimensional, noise-free, and expensive black-box optimization problems. When the objective function is noisy, Bayesian optimization faces

---

\*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

a critical challenge: the observations corresponding to the observation points are no longer accurate. This issue can impact the effectiveness of Bayesian optimization in two ways: the acquisition function and the output results. The most commonly used acquisition function is the Expected Improvement (EI) [Jones et al., 1998], which is based on the expectation of improvement over the current optimal value. By calculating the expected improvement of a specific point relative to the current optimal value, the point with the highest expected improvement is selected as the next observation point. In cases of high noise levels, the current optimal value may be significantly disturbed, affecting the point selection strategy of EI and leading to a tendency to over-explore. Similarly, the noise level also impacts the final output results, as the minimum observation in this case is not accurate. An intuitive solution to these problems is to replace the observed optimum with the optimum of the predicted mean derived by the surrogate model. Here, the predicted mean is the surrogate models extrapolation of the objective function based on the available information. In noisy situations, these predictions are often more reliable than direct observations. The main purpose of this research is to explore the circumstances under which we should trust predicted values more than observed values. To the best of our knowledge, this topic has not been discussed in detail in current literature, so this paper attempts to address it through numerical experiments. To ensure the experimental conclusions are representative, this study uses the advanced test benchmark environment "BBOB," which includes 24 noise-free test functions, modified to a noisy form for the numerical experiments. The details of BBOB functions can be found in [Hansen et al., 2010].

The structure of this article is segmented as follows: section 2 provides a detailed introduction to Bayesian optimization, including the surrogate model and acquisition function. Section 3 introduces the BBOB benchmark functions and experimental settings. In Section 4, we first discuss the use of observations and predictions in a noise-free environment, highlighting the differences in optimization effectiveness as the result output. Then, the experiments for noisy problems are discussed in detail. Section 5 is a summary of the paper.

## 2 Bayesian optimization

Assume  $f(x)$  is an unknown target black-box function, Consider the optimization problem:

$$x^* = \arg \max_{x \in \mathcal{X}} f(x) \quad (1)$$

where  $\mathcal{X}$  is the design space, assumed to be compact. The objective function  $f(x)$  has no explicit expression; for any  $x_0 \in \mathcal{X}$  (observation point), the corresponding objective function value  $f(x_0)$  can be calculated (observation value). Bayesian optimization frameworks commonly employ Gaussian processes (see Section 2.1) as surrogate models to construct the posterior distribution of the objective function, providing a high degree of flexibility. With Gaussian process modeling, the model can yield the predicted mean and uncertainty for any point in the search space. Leveraging these predicted values and uncertainties, we can design acquisition functions (see Section 2.2) to balance exploration and exploitation, thereby selecting the next observation point. The specific algorithm of Bayesian optimization is outlined in **Algorithm 1**.

Note that **Algorithm 1** provides the standard Bayesian optimization algorithm under the assumption of a noise-free condition. In the presence of noise, outputting the point corresponding to the minimum observed value as the result may not be appropriate. Two alternative predictive output methods may be considered: outputting the point with the smallest predicted mean among the observation points or outputting the point with the smallest predicted mean throughout the entire design space as predicted by the surrogate model. The following experimental segment will focus on examining and contrasting the optimization efficacy of these output methods.

---

### Algorithm 1: Bayesian optimization

---

**Input:** number of initial point  $n_0$ , number of max iteration  $n$

- 1 Randomly get initial data  $D_{1:n_0}$  and update GP model;
- 2 **for**  $t = 1, 2, \dots, n$  **do**
- 3     Find  $x_t$  by optimizing the acquisition function over the GP  $x_t = \arg \max_x u(x|D_{1:n_0+n})$ ;
- 4     Sample the objective function:  $y_t = f(x_t) + \epsilon_t$ ;
- 5     Augment the data  $D_{1:n_0+n} = D_{1:n_0+n-1}, (x_t, y_t)$  and update GP model;
- 6 **end**

**Result:**  $(x_{t^*}, y_{t^*}) \in D_{1:n_0+n}, t^* = \arg \max y_{1:n_0+n}$

---

## 2.1 Gaussian process

Gaussian process is a nonparametric model that is fully characterized by its prior mean function  $m(x) : \mathcal{X} \rightarrow \mathbb{R}$  and positive definite kernel functions (Can also be thought of as a covariance function)  $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ . For any finite set of points within the search space  $x_{1:n}$ , define  $f_i := f(x_i)$  as the corresponding value of the objective function at  $x_i$ ,  $y_i := f(x_i) + \epsilon_i$  is the corresponding noisy observation at  $x_i$ . In the Gaussian process model, we assume that  $\mathbf{f} := f_{1:n}$  obeys a joint Gaussian distribution, and that  $\mathbf{y} := y_{1:n}$  obeys a normal distribution given  $\mathbf{f}$ , which is expressed as follows:

$$\mathbf{f} \mid \mathcal{X} \sim \mathcal{N}(\mathbf{m}, \mathbf{K}) \quad (2)$$

$$\mathbf{y} \mid \mathbf{f}, \sigma^2 \sim \mathcal{N}(\mathbf{f}, \sigma^2 \mathbf{I}) \quad (3)$$

Note here we treat the objective function  $f(x)$  as a random variable, and equation (2) represents the prior distribution of the variable  $f(x)$ . where  $\mathbf{m}$  represents the prior mean function, and  $K_{i,j} := k(x_i, x_j)$  is the covariance matrix. After obtaining the observation data  $\mathcal{D}_n = (x_i, y_i)_{i=1}^n$ ,  $f(x)$  given the observed data  $\mathcal{D}_n$ , for the new value to be predicted  $f(x^*)$ , we have the following joint distribution:

$$\begin{bmatrix} \mathbf{y} \\ f(x^*) \end{bmatrix} \sim \mathcal{N} \left( \mathbf{m}, \begin{bmatrix} \mathbf{K} + \sigma^2 \mathbf{I} & \mathbf{k}(x^*) \\ \mathbf{k}(x^*)^T & k(x^*, x^*) \end{bmatrix} \right)$$

Using the conditional probability distribution formula of normal distribution, it is easy to deduce the mean and variance functions of  $f(x^*)$  as follows:

$$\mathbf{m}_n(x^*) = \mathbf{m}(x^*) + \mathbf{k}(x^*)^T (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} (\mathbf{y} - \mathbf{m}(\mathbf{x})) \quad (4)$$

$$\sigma_n^2(x^*) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}(\mathbf{x})^T (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}(\mathbf{x}) \quad (5)$$

The  $\mathbf{k}(x^*)$  in the formula is the vector of covariance between  $x^*$  and  $\mathbf{x}_{1:n}$ . By employing the Gaussian process model, we can obtain the predicted mean and predicted variance for each point in the search space, which provides an important basis for the design of the acquisition function.

## 2.2 Expected improvement

BO has a variety of acquisition function options, including Expected improvement [Jones et al., 1998], GP upper confidence bound [Srinivas et al., 2009], Thompson sampling [Thompson, 1933], and Entropy search methods [Hennig and Schuler, 2012]. This paper concentrates on the most commonly used method EI. The main idea of EI is to select the point with the greatest improvement expectation compared to the current optimal value as the next observation point. Specifically, we first define the improvement function  $\mathbf{I}$  (improvement function):

$$I(\mathbf{x}, v, \theta) := (v - f_n^*) \mathbb{I}(v > f_n^*) \quad (6)$$

where  $v \sim \mathcal{N}(m_n(\mathbf{x}), \sigma_n^2(\mathbf{x}))$ ,  $\mathbb{I}$  is indicator function,  $\theta$  is the hyperparameter set  $f_n^*$  is the current optimal value,  $m_n(\mathbf{x}), \sigma_n(\mathbf{x})$  are respectively the values returned by the Gaussian process model for the input value  $\mathbf{x}$ . Predicted mean and predicted variance. Take expectation for  $v$  to get the acquisition function EI:

$$u(\mathbf{x}; D_{1:n}) := \mathbb{E}[I(\mathbf{x}, v, \theta)] = (m_n(\mathbf{x}) - f_n^*) \Phi \left( \frac{m_n(\mathbf{x}) - f_n^*}{\sigma_n(\mathbf{x})} \right) + \sigma_n(\mathbf{x}) \phi \left( \frac{m_n(\mathbf{x}) - f_n^*}{\sigma_n(\mathbf{x})} \right) \quad (7)$$

Where  $\Phi$  and  $\phi$  are the cumulative distribution (CDF) and probability density distribution (PDF) of the standard normal distribution respectively.

When there is noise in the objective function, the current optimal value  $f_n^*$  is inaccurate. In situations of high noise level, the value of  $f_n^*$  may be more extreme, causing EI to mistakenly believe that the most points have minimal improvement expectations, making EI overly biased towards exploration rather than exploitation. One of the most intuitive solutions is to use the minimum predicted mean  $m_n^*$  of the surrogate model among the observed points to replace  $f_n^*$ . This actually improves EI from optimizing the optimal observation value with noise to optimizing the optimal predicted value of the surrogate model, thereby improving the algorithm's robustness to noise. We record the improved EI as EIM, and its expression is as follows:

$$u'(\mathbf{x}; D_{1:n}) := (m_n(\mathbf{x}) - m_n^*)\Phi\left(\frac{m_n(\mathbf{x}) - m_n^*}{\sigma_n(\mathbf{x})}\right) + \sigma_n(\mathbf{x})\phi\left(\frac{m_n(\mathbf{x}) - m_n^*}{\sigma_n(\mathbf{x})}\right) \quad (8)$$

In fact, there are many other variants of EI designed to deal with noisy problems, such as Augmented expected improvement(AEI)[Huang et al., 2006], the reinterpolation procedure(RI)[Forrester et al., 2006], Expected quantile improvement(EQI)[Picheny et al., 2012] and so on. The behavior of these methods has been detailed discussed in [Picheny et al., 2013]. The experiments in this article will not involve too many variations of EI, since there is no direct relationship between the output method of the final result and the point selection method of the acquisition function.

In subsequent experiments, we will use EI for the output method based on observed values; and use EIM for the output method based on predicted mean.

### 3 Test functions and experimental settings

#### 3.1 BBOB benchmark functions

The test functions in this article are selected from the 24 test functions of the BBOB noiseless test function group, with added noise to serve as our test benchmark problems. These functions are designated as F1-F24. In the original BBOB test's configurations, these functions are categorized into five types: separable functions (F1-F5), functions with low or moderate conditioning (F6-F9), functions with high conditioning and unimodal (F10-F14), multi-modal functions with adequate global structure (F15-F19), and multi-modal functions with weak global structure (F20-F24). Every function within the BBOB test function group is capable of acquiring novel test functions through alterations like shifting or rotating. In the subsequent experiments of this article, we fixed these 24 noise-free functions in advance. The optimal values, standard deviations, and simple descriptions of these functions are shown in Table 1. More detailed information about them can be found in [Hansen et al., 2010].

2Dfunctions	optimal value	std	comment
F1	79.48	12.57	sphere function,unimodal,presumably the most easy continuous domain search problem
F2	66.95	10044455.67	Globally quadratic and ill-conditioned(about $10^6$ ) function with smooth local irregularities.Conditioning is about $10^6$
F3	77.66	418.57	Highly multimodal function with a comparatively regular structure for the placement of the optima.
F4	77.66	171.86	Highly multimodal function with a structured but highly asymmetric placement of the optima.
F5	66.71	29.02	Purely linear function,solution is on the domain boundary
F6	65.87	237396.11	Unimodal,highly asymmetric function,
F7	92.94	431.55	unimodal, non-separable, conditioning is about 100.The function consists of many plateaus of different sizes.
F8	98.62	46480.79	(Rosenbrock function) in larger dimensions the function has a local optimum with an attraction volume of about 25%
F9	65.61	21715.05	rotated version of the previously defined f8.
F10	59.13	9634378.45	rotated version of the previously defined f2.
F11	76.27	21241083.28	A single direction in search space is a 1000 times more sensitive than all others.Conditioning is about $10^6$
F12	56.61	9607260659	conditioning is about $10^6$ , rotated, unimodal
F13	68.42	449.99	Resembles f12 with a non-differentiable bottom of valley
F14	77.31	41.04	The sensitivities of the $z_i$ -variables become more and more different when approaching the optimum
F15	70.03	521.74	Prototypical highly multimodal function which has originally a very regular and symmetric structure for the placement of the optima.
F16	71.35	79.07	Highly rugged and moderately repetitive landscape, where the global optimum is not unique.
F17	69.83	19.00	A highly multimodal function where frequency and amplitude of the modulation vary.Conditioning is low
F18	119.54	308.17	Moderately ill-conditioned counterpart to f17
F19	71.69	74.72	Resembling the Rosenbrock function in a highly multimodal way.
F20	71.29	45818.02	The most prominent 2D minima are located comparatively close to the corners of the unpenalized search area.
F21	124.08	12.21	The function consists of 101 optima with position and height being unrelated and randomly chosen.
F22	51.57	24.05	The function consists of 21 optima with position and height being unrelated and randomly chosen.Conditioning is about 1000
F23	85.39	20.12	Highly rugged and highly repetitive function with more than 10D global optima.
F24	93.30	18.18	Highly multimodal function with two funnels.

Table 1: BBOB functions

#### 3.2 Experimenteal settings

- **Dimension and budget:** In the experiment, the dimensions of the function are segmented into 2 and 4 dimensions. For 2-dimensional problems, the total budget for a single experiment is 100 observation points under noise-free conditions, and escalates to 300 points in noisy conditions. For four-dimensional problems, the total budget is 200 under no-noise conditions and increases to 400 under noisy conditions.

- **Initialization:** Following the result of [Bossek et al., 2020], we set the initial number of points for Gaussian process modeling to 10% of the total budget, and use Latin hypercube sampling to randomly select points in the search area.
- **loss rate** Denote  $y_a$  as the true value of the result obtained by the optimization algorithm,  $y_{opt}$  as the true optimal value. We define the loss rate:

$$loss = \frac{y_a - y_{opt}}{y_{opt}} \times 100\% \quad (9)$$

as the optimization performance evaluation index of the algorithm. Note that the minimum values of the 24 functions we tested are all positive and do not approach 0, so the loss rate is well defined. Similarly, we can also define the relative loss rate between the two algorithms:

$$relloss = \frac{y_1 - y_2}{y_2} \times 100\% \quad (10)$$

where  $y_1$  is the real value of the result obtained by optimization algorithm 1 and  $y_2$  is the real value of the result obtained by optimization algorithm 2. If the relative loss rate is negative, it means that the optimal result of Algorithm 1 is better than Algorithm 2.

- **Randomness processing:** To ensure the generality of the results, we will use different random seeds to repeat each set of experiments 30 times and then average the results. The randomness in the Bayesian optimization process is mainly reflected in the initial point position and the internal optimization of the acquisition function.
- **Noise settings:** The noise added to the test functions in the experiments of this article is Gaussian white noise, with a mean value of 0. The noise level is expressed as a proportion of the function’s standard deviation(5% for small noise, 20% for moderate, 50% for extremely noisy).

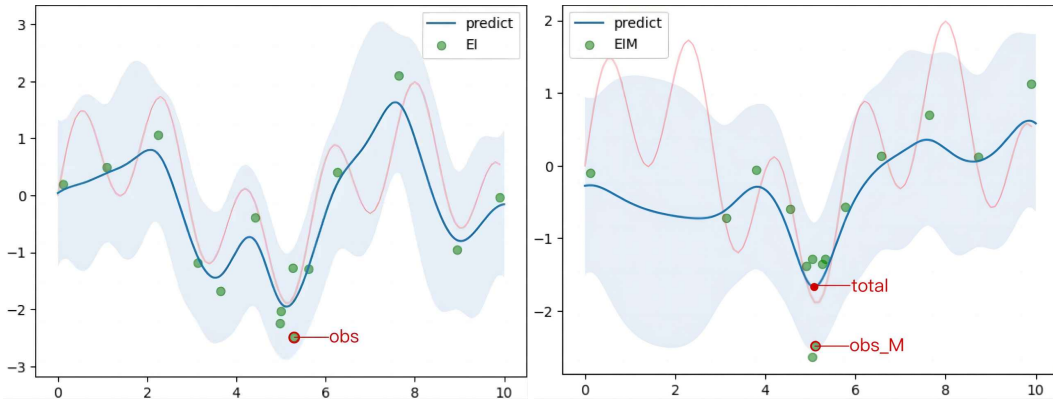
- **Acquisition function and output results:** As mentioned earlier, We will discuss three output methods:

1. Directly output the minimum observation value(Abbreviated as **obs**).

2. Output observation point with the minimum predicted mean.(Abbreviated as **obs\_M**).

3. Output the point with the minimum predicted mean over the total design space.(Abbreviated as **total\_M**).

The choice of output method will align with the selection of the acquisition function. If the output method based on observed values (**obs**) is employed, then the Expected Improvement (EI) acquisition function will be used. Conversely, if the output method based on predicted values (**obs\_M** and **total\_M**) is employed, then we use EIM. The main reason for this is to ensure logical consistency. If we use **obs** as the output method, it indicates that we are more inclined to trust the observed values over the predicted values. Therefore, when selecting the acquisition function, we should correspondingly choose EI, which is optimized based on the observed values, and vice versa.



(a) Using EI for output method based on observation (b) Using EIM for output method based on prediction

Figure 1: The above two figures illustrate the results of Bayesian optimization for a one-dimensional noisy function. The pale red curve represents the true function, while the blue curve depicts the predicted mean function of the Gaussian Process (GP). The green dots indicate the observation points selected by the acquisition function. The acquisition function used in the left figure is Expected Improvement (EI), which corresponds to the output method '**obs**', where the point with the smallest observation value is directly outputted. The acquisition function for the right figure is Expected Improvement of the Mean (EIM), showing two outputs based on the predicted values. The method '**total\_M**' corresponds to the minimum value of the predicted mean function (blue curve), and '**obs\_M**' is the point with the smallest predicted mean (blue curve) among all observed points (green dots).

## 4 Experimental results and analysis.

### 4.1 Noise-free experimental results

We first focus on the experimental results for the noiseless problem. In a noise-free scenario, the methods **obs** and **obs\_M** are exactly the same since the observations at the observation points are all accurate. Therefore, we only need to compare the **obs** and **total\_M** methods to analyze whether the predictive output method can yield better points as the final result. Naturally, it is reasonable to consider **obs** as the most appropriate output method under noise-free conditions. The main purpose of our noise-free experiments is to investigate the specific gap in optimization efficiency between these two output methods. Table 2 details the specific differences between these two output methods:

2D	obs	total_M	total_M vs obs	4D	obs	total_M	total_M vs obs
F1	0.00%	0.00%	0.00%	F1	0.00%	0.00%	0.00%
F2	5.71%	59.70%	50.71%	F2	6589.66%	7861.45%	585.31%
F3	10.22%	27.81%	16.80%	F3	26.95%	56.09%	23.06%
F4	5.01%	18.19%	12.62%	F4	29.90%	63.32%	26.56%
F5	0.00%	0.00%	0.00%	F5	0.00%	0.00%	0.00%
F6	1.52%	3.45%	1.89%	F6	31.23%	105.48%	58.40%
F7	0.08%	0.18%	0.10%	F7	0.05%	0.09%	0.05%
F8	0.07%	1.75%	1.68%	F8	4.33%	5.16%	0.84%
F9	0.12%	3.66%	3.54%	F9	5.76%	11.17%	5.25%
F10	15.30%	1282.36%	1133.40%	F10	454.52%	4275.66%	1253.11%
F11	8.50%	719.52%	656.81%	F11	22.84%	2774.84%	2255.78%
F12	1645.98%	2907234.96%	1209546.08%	F12	526280.42%	27772809.81%	11673.63%
F13	0.96%	1.78%	0.81%	F13	27.39%	12.98%	-11.03%
F14	0.01%	0.02%	0.02%	F14	0.01%	0.03%	0.01%
F15	5.26%	23.30%	17.33%	F15	22.63%	64.67%	34.49%
F16	0.74%	26.55%	25.71%	F16	3.20%	14.22%	10.55%
F17	0.52%	4.43%	3.89%	F17	0.91%	1.20%	0.30%
F18	0.76%	11.46%	10.62%	F18	2.40%	3.50%	1.08%
F19	0.25%	10.95%	10.68%	F19	2.90%	14.37%	11.13%
F20	1.64%	5.11%	3.41%	F20	2.78%	6.00%	3.13%
F21	0.07%	0.14%	0.07%	F21	0.70%	0.84%	0.13%
F22	0.46%	0.75%	0.29%	F22	3.62%	3.87%	0.24%
F23	4.48%	50.64%	44.24%	F23	3.10%	24.36%	20.63%
F24	5.11%	27.33%	21.20%	F24	20.30%	49.25%	24.22%

Table 2: Noise-free results

The first two columns in the table show the loss rate of the output methods **obs**, **obs\_M** relative to the global optimal value of each test function. The third column is the relative loss rates of these two output methods (If positive, **obs** is better than **total\_M**). For two-dimensional problems, the predictive method **total\_M** is significantly less efficient in optimization than **obs** when applied to the functions F2, F10, F11, and F12. A review of Table 1 reveals a common feature of these functions: their conditioning levels are significantly high, exceeding  $10^6$ . The standard deviation of each of these functions is very large, resulting in high noise levels. This high noise level can easily mask the original structural characteristics of the objective function, making it difficult to build a suitable model for the Gaussian process to predict. By examining Table 2, it can be seen that the loss rate of **obs** is likewise relatively high on these four functions compared to the others, which further suggests that the surrogate model struggles due to high conditioning levels. Therefore, it can be concluded that for noiseless problems, low accuracy in surrogate model modeling can greatly increase the loss rate of predictive output methods, in which case observational output method **obs** is more plausible. For simpler functions, the results in the third column show that the loss ratio of **total\_M** relative to **obs** is

mostly above 10%. For more complex functions such as F10, F12, etc., the gap between the optimization results will increase further. Therefore, for noise-free problems, we should indeed directly use the observation-based Bayesian optimization result output method (The loss rate of **total\_M** relative to **obs** in all test functions is greater than 0). For 4D problems, the conclusion remains same, however, after increasing the dimension, we find that the relative loss rate of these two output methods begins to decrease on many functions. This will be discussed later in the article.

## 4.2 Experimental results for the 2D noise problem

In the previous section, we briefly compared the optimization results of the observative and predictive outputs on 24 noise-free functions. In this section, we will select some functions and add noise to them as new optimization problems. To ensure the consistency of the conclusions, we will use EI as the acquisition function for the observable output method (**obs**), and EIM for the predictive output method (**obs\_M**, **total\_M**). It was noted that in the noisy problems, **obs** is no longer exactly the same as **obs\_M**, as the observed value will no longer be accurate. (See Section 3.2 for details on noise settings). We will use the Instant Regret as the vertical axis of the figures, and the number of iterations of BO as the horizontal axis to generate curves for analysis. For example, in each iteration of BO, **obs** selects a point as the output result. We subtract the optimal value of the function from the value of the real function corresponding to the selected point to get the instant regret value of **obs** at the corresponding round. **obs** and **obs\_M** can do the same for similar curves. For comparison, we add a curve corresponding to a completely randomized output method, which randomly selects a point in the search space as the output for each round of iteration.

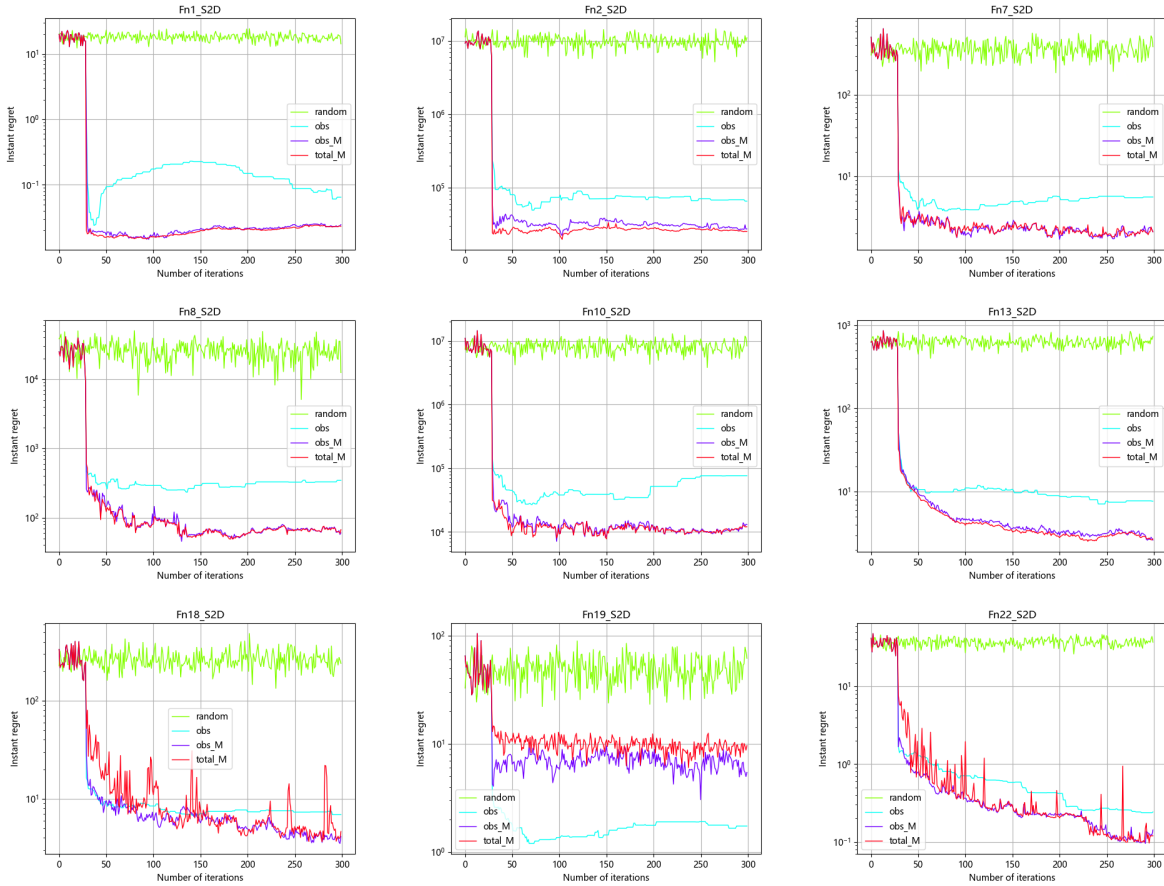


Figure 2: 2D optimization results (small noise)

Since the random seeds used in the initialization are identical, all experiments start with the exact same 20 initial points. Therefore, the curves corresponding to each output method for the first 20 rounds of iterations are exactly overlapped. After that, the Instant Regret curves for different output methods begin to show differences. We can arbitrarily take vertical intercepts of the curves to compare the advantages and disadvantages of the four output methods in a particular round. For example, we can take a vertical intercept of the curve at round 200 on the horizontal

axis and compare the average instant regret value of the four curves at round 200. The closer this value is to 0, the closer the output is to the true optimum. Since the budget of an optimization problem is often not fixed, we are more interested in the overall position and trend of these curves, rather than only comparing the average Instant Regret value after 300 rounds when the budget is exhausted.

Let's start with two predictive output methods. From Figure2, there is little difference between **obs\_M** and **total\_M** in the final optimization results, but the output of **total\_M** shows great instability in some functions. For example, for the function Fn18, the output of **total\_M** in turns 280-290 fluctuates greatly. Considering that our budget is actually arbitrary, if the budget happens to be between 280-290, the final output returned by **total\_M** will be very poor. Compared with **total\_M**, the output value of **obs\_M** is more stable. Therefore, for two-dimensional problems with small noise, we recommend using **obs\_M** instead of **total\_M** if you prefer a predictive output method.

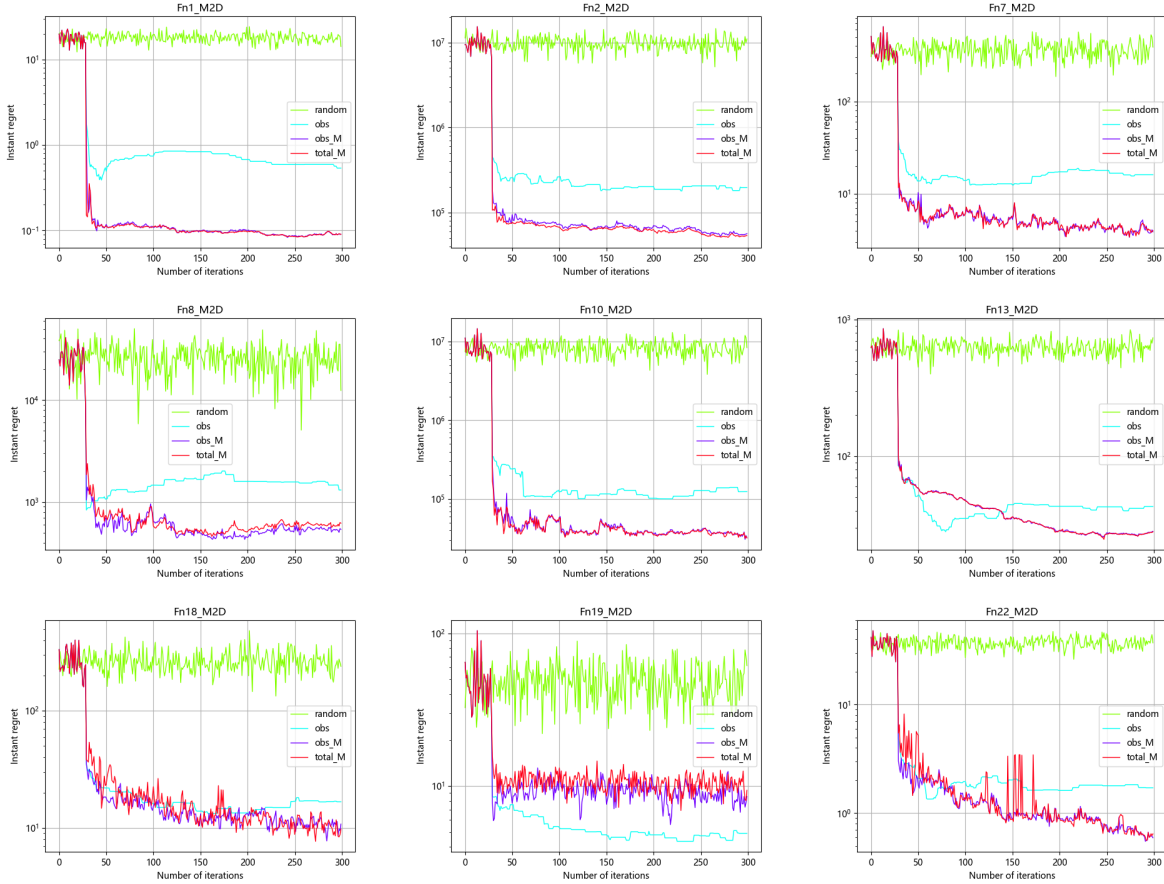


Figure 3: 2D optimization results(Medium noise)

As for the observational output method **obs**, we can see from Figure2 that the final output result of **obs** is much worse than **obs\_M** on most of the test functions, and its curve will always be above the curve of **obs\_M** after fewer rounds of iterations. This means that for any arbitrary given budget, the final output of **obs** is worse than **obs\_M**. However, in terms of optimization of the test function Fn19, we find that **obs** has a clear advantage over other output methods. We further observe the distribution characteristics of the output points for the three output methods on the function. We find that most of the points output by **obs\_M** and **total\_M** are concentrated in an area where the function's value is generally high. Even if a small number of points visit the area where the true optimum is located in some experiments, they do not linger in that area for long. In contrast, when **obs** accesses the region with a low function value, where the true optimal value is located, its subsequent output points tend to stay in this region. This indicates that the GP model has made a significant misjudgment in predicting the value of function Fn19, and the effectiveness of the **obs** point selection method is not influenced by the model's quality but is only related to the noise level. The performance of **obs\_M** and **total\_M**, however, is more dependent on the model's predictive accuracy. For Fn19, its small standard deviation results in a relatively low absolute noise value. However, its highly multimodal



characteristics severely impact the modeling accuracy of the GP, leading to better output results from **obs** compared to the other two output methods.

Therefore, when the model’s prediction accuracy is poor, we indeed need to consider whether it might be better to use **obs**. However, we believe that this discussion is not very meaningful in essence. For an optimization problem, if we can anticipate that the model’s predictive performance will be poor, our primary focus should be on improving the model itself rather than debating which output method to choose. For most problems with small noise, when the model is capable of making reasonable predictions, we still recommend to use **obs\_M**.

Figure3 illustrates the optimization results for medium noise levels. It can be seen that with increased noise, the gap between the observational output method **obs** and the predictive output method widens further for most of the test functions. Overall, the conclusions for the choice of output methods are generally consistent with those under small noise. That is, for most problems with small noise, the use of the **obs\_M** output method is the best choice. The use of **obs** only needs to be considered if one is convinced that BO’s surrogate has low predict accuracy; however, in such cases, replacing the model may be the better choice. If we increase the noise level further (see Appendix B for details), the above conclusions still hold true, and when noise level is too high, the advantage of **obs** in Fn19 even ceases to exist.

### 4.3 Experimental results for the 4D noise problem

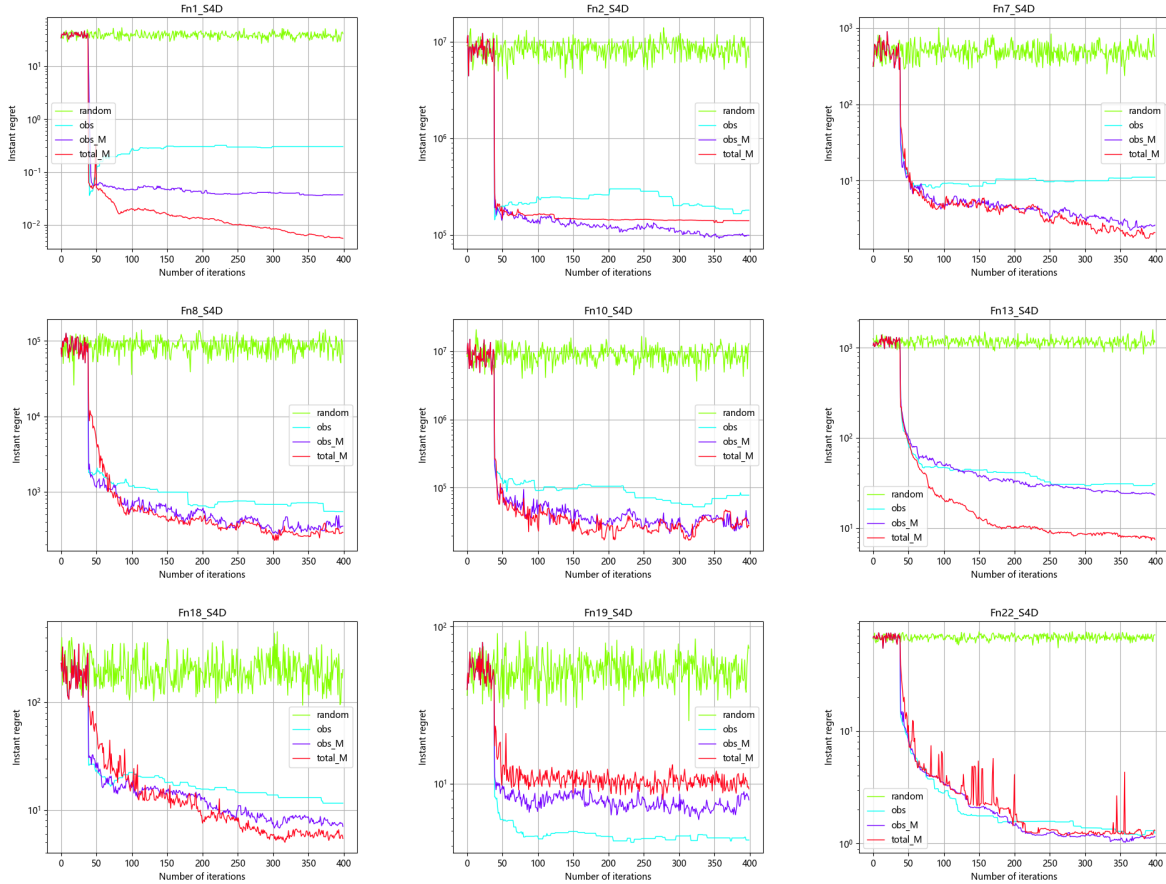


Figure 4: 4D optimization results (small noise)

In this section we increase the dimensionality of the optimisation problem from 2 to 4, thereby expanding the search space geometrically. Figure 4 shows the optimization results for a low noise level problem. We can clearly see that the output method **total\_M** starts to become more competitive. For example, on the function Fn1, for which, on function Fn1, which is the simplest and least difficult to optimize, **total\_M** yields significantly better results

than the other output methods. The same is true for function Fn13. There is also a small advantage over **obs\_M** on functions Fn7 and Fn18.

This is mainly due to the fact that **obs\_M** can only select among the observed points. Despite increasing the maximum number of iterations to 400 for the four-dimensional problem, 400 points are still too few for a four-dimensional search space. As discussed earlier, both EI and EIM acquisition functions must balance exploitation and exploration tasks. This reduces the proportion of the 400 points that are focused on exploitation. In contrast, **total\_M** can focus entirely on exploitation without being limited to observed points, making it more likely to find local optimal points.

For higher-dimensional problems, the vast search space makes finding the global optimal solution extremely difficult, so finding a local optimal solution may be a better choice. However, it is worth noting that increasing the dimensionality does not solve the instability problem of **total\_M**. For example, on Fn22 **total\_M** still exhibits large fluctuations, indicating that it may still produce poor optimization results in some experiments. While **total\_M** is competitive in higher-dimensional optimization problems, it carries the risk of instability. Therefore, its use should be approached with caution. In contrast, **obs\_M** remains relatively stable, outperforming **obs** on all test problems except Fn19 and performing comparably to or better than **total\_M** on most problems. Overall, **obs\_M** continues to be the most reasonable choice for four-dimensional problems with low noise levels.

#### 4.4 Other complementary experiments

We have added three sets of complementary experiments to Appendices. The experiments in Appendix A demonstrate the output of the three methods under more extreme conditions. In fact, even in experiments with moderate noise levels in 2D, we observed that for some functions, the quality of the points selected by EI and EIM is not even as good as that of completely random selection. This suggests that the GP model has been somewhat compromised on these functions. Under more extreme conditions, such as 2D high noise levels and 4D medium noise levels, the GP model may degrade further and even fail completely. Due to space constraints, we have included a discussion of the results from these experiments in the appendix.

The experiments in Appendix B were performed under conditions of moderate noise levels in two dimensions. Unlike Section 4.2, all the output methods in the experiments in Appendix B use EI, with no EIM used in the optimization process. The purpose of this experiment is to demonstrate that the gap between **obs** and **obs\_M** or **total\_M** is not primarily caused by differences in acquisition functions. In fact, in Appendix B, you can see that even when the same acquisition function is used, the results remain consistent.

It is also worth mentioning that the form of the noise can likewise influence the choice of output methods. Our experiments above only discuss homogeneous noise, and the conclusions may change when the noise becomes non-homogeneous. We briefly discuss this scenario in Appendix C and give the corresponding experimental results.

## 5 Conclusion

In the previous section we conducted numerical experiments on optimisation test problems without noise and with different noise levels and dimensions. The main conclusions are summarised as follows:

- For the noise-free problem, directly using the observation-based output method **obs** is the most reasonable choice.
- For the noisy problem, using the predictive-based output is superior to the observation-based output in most of the cases, even when the noise level is low(5% sd). Considering that the output results of **total\_M** are not stable, the use of **obs\_M** is the most reasonable choice.
- For more complex problems, when the BO's surrogate model has low prediction accuracy, the predictive-based output method should be used with caution, since the model's prediction of the objective function is less credible. In this case, the choice of observation-based output will often give better output results, but in practice it is meaningless. If we can be sure that the model is less predictive, then we should consider replacing the model rather than discussing what output method to use.
- When the optimisation problem is of high dimensionality, the use of **total\_M** can be considered, but it still carries the risk of unstable output results.

Taking all these points together, it seems that using **obs\_M** is a relatively reasonable choice in almost all situations. Firstly, when there is no noise or very little noise in the objective function, our experiments show that the output of **obs** is much better than that of **total\_M**. In fact, the outputs of **obs\_M** and **obs** are essentially the same in this case since both select from observed points, and in the absence of noise or with very little noise, the observed values and the predicted mean of the model are nearly identical.

After increasing the noise level, **obs\_M** and **total\_M** generally outperform **obs**, with **obs\_M** avoiding the instability problem that plagues **total\_M**. Excluding situations where the objective function is too complex, leading to poor model predictions (where discussing the output method becomes less meaningful), our experiments consistently show that using **obs\_M** as the output method is always a suitable choice. Therefore, we can conclude that using **obs\_M** as the output method is a reasonable choice for most optimization problems.

## References

- Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016. doi:10.1109/JPROC.2015.2494218.
- Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 11 2005. ISBN 9780262256834. doi:10.7551/mitpress/3206.001.0001. URL <https://doi.org/10.7551/mitpress/3206.001.0001>.
- Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *J. of Global Optimization*, 13(4):455–492, dec 1998. ISSN 0925-5001. doi:10.1023/A:1008306431147. URL <https://doi.org/10.1023/A:1008306431147>.
- Nikolaus Hansen, Anne Auger, Steffen Finck, and Raymond Ros. Real-parameter black-box optimization benchmarking bbo-2010: Experimental setup. 2010. URL <https://api.semanticscholar.org/CorpusID:260830254>.
- Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. *arXiv e-prints*, art. arXiv:0912.3995, December 2009. doi:10.48550/arXiv.0912.3995.
- William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933. ISSN 00063444. URL <http://www.jstor.org/stable/2332286>.
- Philipp Hennig and Christian J. Schuler. Entropy search for information-efficient global optimization. *J. Mach. Learn. Res.*, 13(null):1809–1837, jun 2012. ISSN 1532-4435.
- D. Huang, Theodore Allen, William Notz, and Ning Zheng. Global optimization of stochastic blackbox systems via sequential kriging meta-models. *Journal of Global Optimization*, 34:441–466, 03 2006. doi:10.1007/s10898-005-2454-3.
- Alexander I. J. Forrester, Andy J. Keane, and Neil W. Bressloff. Design and analysis of "noisy" computer experiments. *AIAA Journal*, 44(10):2331–2339, 2006. doi:10.2514/1.20068. URL <https://doi.org/10.2514/1.20068>.
- Victor Picheny, David Ginsbourger, Yann Richet, and Grégory Caplin. Quantile-based optimization of noisy computer experiments with tunable precision. *Technometrics*, 55, 03 2012. doi:10.1080/00401706.2012.707580.
- Victor Picheny, Tobias Wagner, and David Ginsbourger. A benchmark of kriging-based infill criteria for noisy optimization. *Structural and Multidisciplinary Optimization*, 48, 09 2013. doi:10.1007/s00158-013-0919-4.
- Jakob Bossek, Carola Doerr, and Pascal Kerschke. Initial Design Strategies and their Effects on Sequential Model-Based Optimization. *arXiv e-prints*, art. arXiv:2003.13826, March 2020. doi:10.48550/arXiv.2003.13826.

## Appendix

### A Non-homogeneous noise experimental results

#### A.1 2-dimensional experiments

The basic settings of the experiments are the same as those in 4.2, with the only difference being that the noise settings are no longer homogeneous variance, but are set as follows:

$$f_{GN}(f, \beta) = f \times \exp(\beta \mathcal{N}(0, 1)) \quad (11)$$

where  $\mathcal{N}(0, 1)$  represents random sampling from the standard normal distribution, and  $\beta$  is the parameter that controls the magnitude of the noise, here we set it to 0.1. Note that the value of noise here will be related to the true value of the function. Also similar to 4.2, we give the following image of the experimental results:

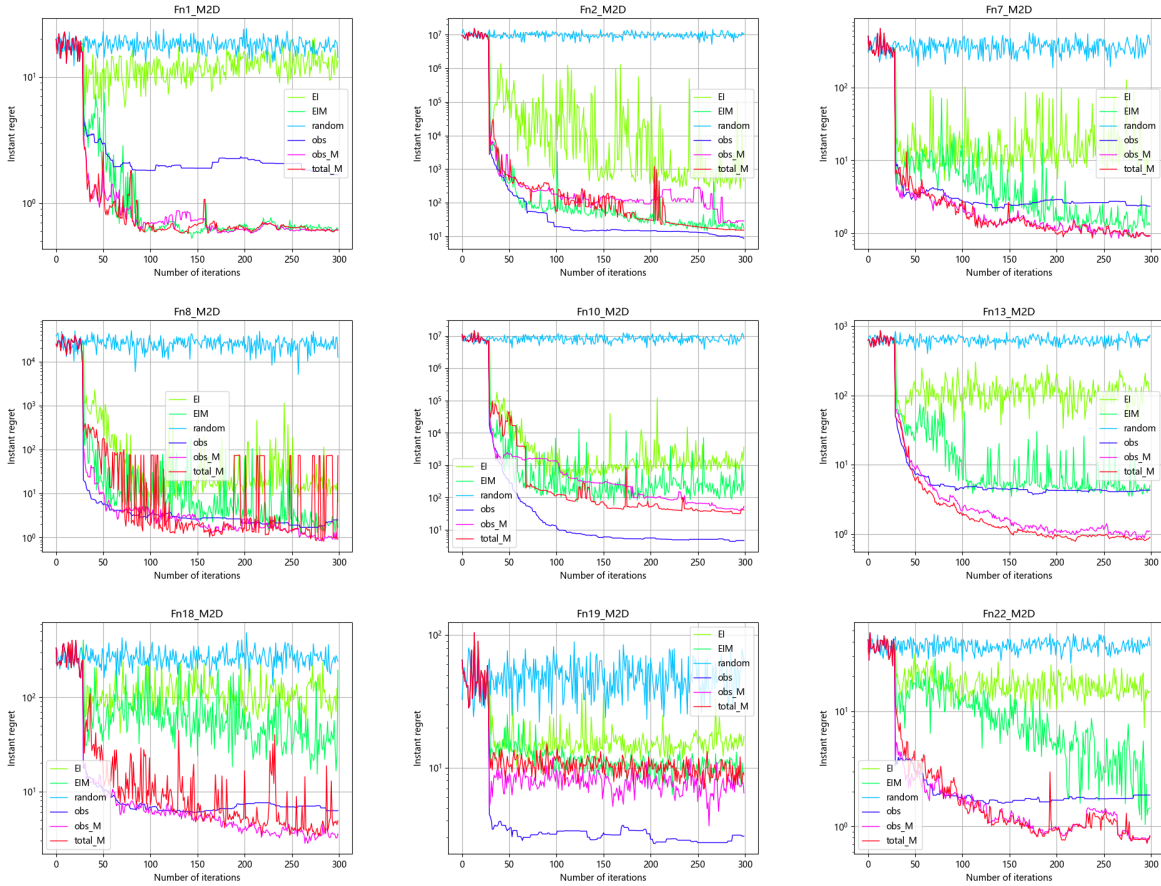


Figure 5: 2D optimization results(Non-homogeneous noise)

The results reflected in the figures are basically similar to those in 4.2, with the only difference being that for functions with large standard deviations (e.g. Fn2, Fn10), the observative output methods in non-homogeneous noise is much better than the predictive output method. This is because when the noise value is positively correlated with the true value of the function, the point with the smaller value of the function corresponds to the less noise, so if we observe that the value of a point is small, it is most likely because the function value corresponding to that point is also small, rather than due to extreme noise. In this case, the observant output method is more reliable than the predictive output methods, even if noise is affected(which is the main difference with the homogeneous variance conclusion).

## B Other homogeneous noise experimental results

### B.1 2-dimensional experiments

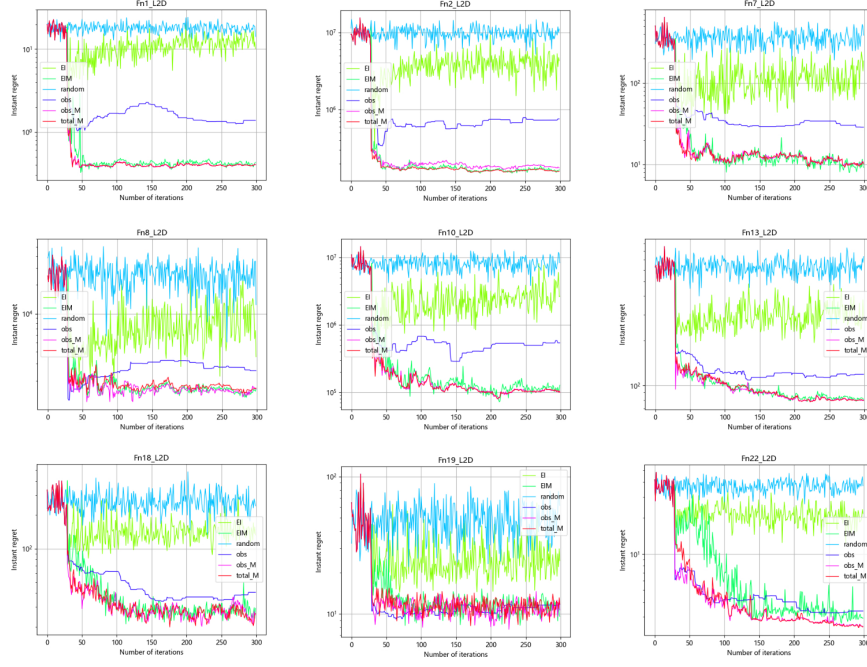


Figure 6: 2D optimization results(Large noise)

### B.2 4-dimensional experiments

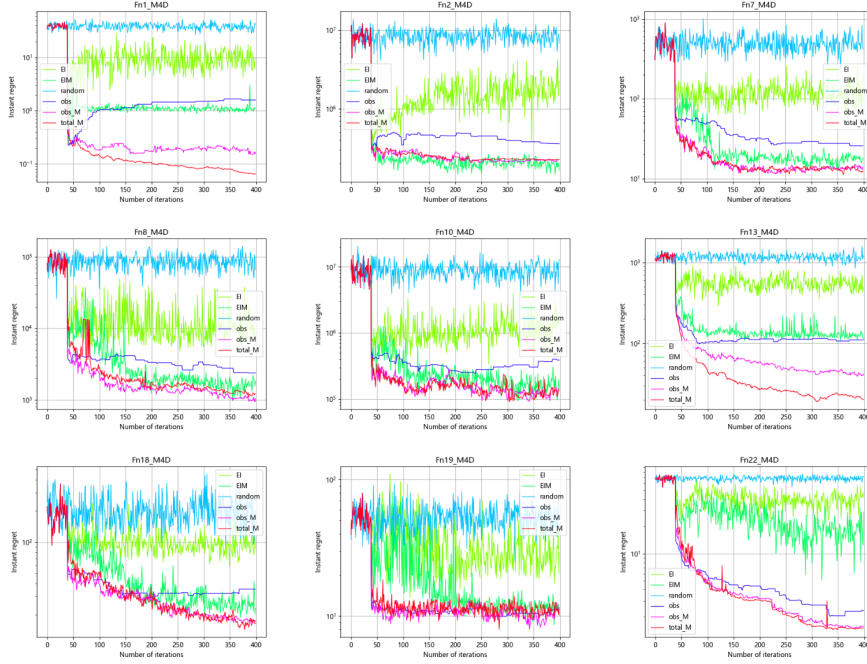


Figure 7: 4D optimization results(Medium noise)