RAPID DESIGN OF METAMATERIALS VIA MULTI-TARGET BAYESIAN OPTIMIZATION

BY YANG YANG 1, CHUNLIN JI * 2 AND KE DENG * 3

1 Department of Mathematical Sciences & Center for Statistical Science, Tsinghua University, yyang15@mails.tsinghua.edu.cn

2 Shenzhen Kuang-Chi Institute of Advanced Technology, chunlin.ji@kuang-chi.com

3 Center for Statistical Science & Department of Industry Engineering, Tsinghua University, kdeng@tsinghua.edu.cn

Composed of a large number of sub-wavelength unit cells with designable geometries, metamaterials have been widely studied to achieve extraordinary advantageous and unusual optical properties. However, ordinary computer simulator requires a time-consuming fine-tuning to find a proper design of metamaterial for a specific optical property, making the design stage a critical bottleneck in large scale applications of metamaterials. This paper investigates the metamaterial design under the framework of computer experiments, with emphasis on dealing with the challenge of designing numerous unit cells with functional responses simultaneously, which is not common in traditional computer experiments. We formulate the multiple related design targets as a multi-target design problem. Leveraging on the similarity between different designs, we propose an efficient Bayesian optimization strategy with a parsimonious surrogate model and an integrated acquisition function to design multiple unit cells with very few function evaluations. A wide range of simulations confirm the effectiveness and superiority of the proposed approach compared to the naive strategies where the multiple unit cells are dealt with separately or sequentially. Such a rapid design strategy has the potential to greatly promote large scale applications of metamaterials in practice.

1. Introduction. A metamaterial is an arrangement of artificial structural elements, engineered to achieve advantageous and unusual optical properties that are not found in naturally occurring materials [Pendry, Schurig and Smith (2006); Shalaev (2007); Padilla, Basov and Smith (2006)]. The advantage of metamaterials over their conventional counterparts comes from their designability, including the artificial elements’ shape, geometry, size, orientation and arrangement, which gives them an excellent flexibility in manipulating electromagnetic waves via blocking, absorbing, enhancing, or bending waves, to achieve benefits that go beyond the capability of conventional materials. For example, some optical metamaterial can produce a negative refractive index [Valentine et al. (2008)] and a cloak of carefully designed metamaterial can protect an object on ground from being detected by microwave (a.k.a. invisible cloak) [Liu et al. (2009)]. No natural materials achieve similar properties. Due to their superior optical properties, metamaterials demonstrate great potentials in various attractive applications in biosensor [Sanders and Manz (2000)], antennas [Li et al. (2010)] and absorber [Tao et al. (2008)] that popularly used in public safety, sensor detection and remote aerospace applications [Singh and Marwaha (2015)]. Metamaterials are generally composed of numerous sub-wavelength structural elements with designable geometries, also called unit cells. Figure 1 illustrates the macrostructure and the microstructure of a typical metamaterial, which is essentially a 3-dimensional array of unit cells. Each unit cell is a small piece of metal surrounded by insulating materials. In practice, people have the freedom to choose

*Corresponding author.

Keywords and phrases: Design of metamaterials, multi-target design, response surface learning, design of computer experiments, Bayesian optimization

1
different metals, surrounding materials and geometric shapes of the metal to create unit cells with different optical properties. For the metamaterial shown in Figure 1, the unit cells are designed to be \( I \)-shape copper sheets surrounded by resin. The geometric shape of each unit cell is determined by four parameters \( l, a, w_1 \) and \( w_2 \) as shown in the figure (the unit is mm). Let \( \mathbf{x} = (l, a, w_1, w_2) \) be the vector of geometric parameters of a unit cell. By specifying \( \mathbf{x} \) to be different values, we get unit cells of different designs, which achieve different responses (e.g., relative refractive index) for electromagnetic wave of a specific frequency. The surrounding materials insulate different unit cells to make sure that these unit cells work independently with little interference with each other. Carefully designing the geometry of different unit cells, we can fine-tune the field of optical response within the metamaterial to meet a specific need in practice. For example, if a field of negative refractive index is implemented, we would come up with an invisible cloak that can protect an object from being detected by microwave [Liu et al. (2009)].

\[ \text{Fig 1. A schematic diagram showing the relationship between the macrostructure and the microstructure of a metamaterial prototype.} \]

The design of metamaterial generally consists of two major processes [Cui, Smith and Liu (2010)]: (1) a system-level design stage to determine the field of optical response within the metamaterial based on the requirements of a practical problem, and (2) a cell-level design stage to find a proper geometric setting for each unit cell to fit the specific response target for the unit cell given by the system-level design. It is the key art of metamaterial research to find and implement proper designs for both stages efficiently for a given practical problem. In this paper, we assume that the system-level design is given and focus on the cell-level design only. Considering that there are typically tens of thousands of unit cells in the metamaterial of interest, many of which are unique in terms of optical property, the cell-level design becomes very challenging: we need to find a large number of distinct designs of unit cells to satisfy the complex and diverse needs for various unit cells requested by the system-level design.

Regards to the design response of unit cells, let \( \mathcal{F} = [f_L, f_U] \) be the frequency band in which the metamaterial is designed to work (referred to as Working Frequency Band or WFB), \( K \) be the number of distinct unit cells determined by the system-level design, and \( Z_k = \{z_k(f)\}_{f \in \mathcal{F}} \) be the specific electromagnetic response function for the \( k \)-th unit cell as requested by the system-level design. Our goal in the cell-level design stage is to find a series of designs \( \{X_k^1, \cdots, X_k^K\} \), where design \( X_k^k \) leads to the desired electromagnetic response function \( Z_k \) within the WFB \( \mathcal{F} \). To be concrete, let \( \mathbf{r}(\mathbf{x}) = \{r_f(\mathbf{x})\}_{f \in \mathcal{F}} \) be the electromagnetic response function of a unit cell with design \( \mathbf{x} \), and \( \Omega \) be the space of all possible designs of a unit cell with dimension \( d \), which is a compact subset of \( \mathbb{R}^d \). The research goal can be formally stated as below: for each \( k \in \{1, \cdots, K\} \), search \( \Omega \) to find a design \( \mathbf{x}_k \) satisfying \( D(\mathbf{r}(\mathbf{x}_k), Z_k) \leq \varepsilon_k \) and let \( X_k^* = \mathbf{x}_k \), where \( D(\cdot, \cdot) \) is a quantity measuring the deviation of \( \mathbf{r}(\mathbf{x}_k) \) from \( Z_k \) and \( \varepsilon_k \).
is the maximum admissible error level. Hereinafter, we refer to $Z_k$ as a target response and $X_k^*$ as an admissible design of $Z_k$ (with respect to deviation measurement $D$ and error level $\varepsilon_k$), respectively. Please note that there could be more than one or no admissible design in $\Omega$ for a target response $Z_k$ under a given error level $\varepsilon_k$. If the number of admissible designs is larger than 1, we just need to catch one of them; if the number of admissible designs is less than 1, however, we would have to expand the design space $\Omega$ to include more possible designs, or increase $\varepsilon_k$ if we are allowed to do so at the price of a downgraded quality of the metamaterial. From now on, we simplify the problem by assuming that there exists at least one admissible design in $\Omega$ for each target response $Z_k$. For this simplified scenario, the statistical nature of the cell-level design can be summarized as follow: finding $K$ design points $\{X_1^*, \cdots, X_K^*\}$ in the design space $\Omega$ to fit $K$ distinct target responses $\{Z_1, \cdots, Z_K\}$. Throughout this paper, we refer to a design problem of this type as a Multi-Target Design (MTD) problem.

To the best of our knowledge, the MTD problem is still an open problem that has never been formally discussed in the statistical literature, possibly because there was no strong appeal of finding multiple design points along one design path in the design space to fit distinct design targets before the boom of the metamaterial research in recent years. The MTD problem in metamaterial design faces a few critical challenges. First, the number of targets $K$ is typically very large, e.g., it is quite often that $K$ is at the order of $10^3$ in many real applications. Second, we need to consider the performance of a design over the whole WFB $\mathcal{F}$, leading to a functional response that is often challenging to deal with. Third, the design space is often high-dimensional and has a large volume, making it computationally expensive to search.

A relative of the MTD problem is the traditional Material Design (MD) problem, whose goal is to find the optimal design, structure or composition of a material to gain excellent properties. Unlike the MTD problem we study here, there is only one target in a MD problem which is finding the optimal one. An efficient approach to resolve the MD problem is to use the Bayesian Optimization (BO) [Mockus (1975); Jones, Schonlau and Welch (1998)]. As a sequential design strategy for global optimization of black-box functions that does not require derivatives, BO takes the Bayesian strategy to treat the unknown objective function as a random function and place a prior over it to capture beliefs about the behaviour of the function; after gathering the function evaluations, the prior is updated to form the posterior distribution over the objective function, based on which an acquisition function is constructed to determine the next query point. Numerous literatures have applied BO to solve the MD problems in various kinds of materials [Seko et al. (2015); Frazier and Wang (2016); Kiyohara et al. (2016); Ju et al. (2017); Packwood (2017); Packwood and Hitosugi (2017)]. For example, Seko et al. (2015) use BO to design the compounds of materials with the desirable physical property, the low lattice thermal conductivity (LTC). Ueno et al. (2016) improve BO and propose the COMBO algorithm to determine the atomic structure of a crystalline interface with the optimal rigid body translation. Recent work in biomaterials utilizes BO to choose the silk processing parameters, including fiber degumming, dissolving, and concentration to produce the silkworm silk with high viscosity spinning dopes [Yao et al. (2020)].

The MTD problem is also closely related to the Multi-Task Optimization (MTO) problem in machine learning/optimization, whose goal is to find the global optimum of each of the $K$ tasks [Gupta, Ong and Feng (2016); Da et al. (2016); Zheng et al. (2019)]. In fact, the two problems essentially share the same mathematical nature, although their research backgrounds are very different. The key difference between the two problems, however, lies on the fact that $K$ is often as large as a few thousand in the MTD problem considered in this study, while takes a much smaller value, typically 2 or 3, in the MTO problem in the literature. The earliest effort to resolve the MTO problem traced back to the MTBO algorithm by Swersky,
Snoek and Adams (2013), a BO approach which uses the multiple output Gaussian processes [Boyle and Frean (2004)] to jointly model the $K$ outputs of the $K$ tasks, and sequentially finds the next most promising task and the design point of that task to evaluate. Considering a MTO problem with $n$ points and $K$ tasks, MTBO has a complexity of $O(n^2K^2)$ in memory and $O(n^3K^3)$ in time, making it infeasible to extend to a large number of tasks. Later on, people tried to solve the MTO problem by utilizing evolutionary algorithms (EAs) [Gupta, Ong and Feng (2016); Da et al. (2016); Zheng et al. (2019); Bali et al. (2020)]. For example, Gupta, Ong and Feng (2016) develop the MFEA algorithm, which creates a cross-domain optimization platform that allows a single population of individuals from one task to influence the evolution of the population from other tasks concurrently via EAs; Bali et al. (2020) extend MFEA and propose MFEA-II, which seeks to minimize the negative interactions between tasks based on a data-driven online learning approach. While if the number of tasks increases rapidly or some tasks have negative inter-task interactions, the number of function evaluations or experiments consumed by those multi-task EAs will get explosive, impeding the overall performance.

A potential distant relative of the MTD problem in the statistical literature is the Multi-Objective Optimization (MOO) problem [Hwang and Masud (1979); Marler and Arora (2004)], whose goal is to find one feasible design point $x^*$ in the design space $\Omega$ that can minimize $K$ objective functions simultaneously under some constraints, i.e., $x^* = \arg\min_{x \in \Omega} f(x)$, restricted by $g_j(x) \leq a_j$ for $1 \leq j \leq m$, where $f = (f_1(x), \ldots, f_K(x))$ are the objective functions of interest, $m$ is the number of inequality constraints, and $g_j$s are the constraint functions. Usually there does not exist a single solution that can optimize all objective functions, so one should make trade-off to search for such solutions that can improve in at least one objective function without sacrificing the performance in other objective functions [Deb et al. (2002); Konak, Coit and Smith (2006)]. In the literature, many optimization algorithms have been developed to resolve the MOO problems efficiently, including VEGA [Schaffer (1987)], MOGA [Murata and Ishibuchi (1995)], NSGA-II [Deb et al. (2002)] and so on. But, clearly the MOO problem has a different goal from the MTD problem.

In this paper, we study the challenging MTD problem under the computer-experiment framework, and propose an efficient design strategy for the cell-level design of metamaterials via BO. Since the number of targets in the MTD problem is much more than its related problems we mentioned above, directly using traditional BO or MTBO is not practical. We extend BO and propose to summarize all key ingredients and requirements of the cell-level design into two proper objective variables shared by the $K$ targets and build a surrogate model to approximate the complex function that links the design variables and the objective variables by Gaussian Process (GP) [Santner, Williams and Notz (2003); Kleijnen (2009)]. We further develop the target-specific acquisition functions and integrate them into a summarized acquisition function to find the next potential design point. We find that the challenges in the cell-level design of metamaterials can be effectively resolved in a principled way via BO. A wide range of simulations confirm the effectiveness of the proposed approaches.

The later part of this paper is organized as follows. In Section 2, we discuss how to define proper objective variables that can effectively summarize all key ingredients and requirements of the cell-level design, and demonstrate how to build surrogate models to approximate the complex response function that links the design variables and the objective variables. Section 3 introduces a synergic strategy to resolve the MTD problem under the BO framework based on the established surrogate models, with the convergence of the proposed strategy discussed. A few alternative approaches are introduced in Section 4. Performance of the proposed method is evaluated and compared via simulations in Section 5. Finally, we conclude the study with discussions in Section 6.
2. Surrogate models for cell-level design of metamaterials. A critical challenge in metamaterial design lies on the fact that the concrete form of the response function \( r(\mathbf{x}) \) is typically unknown, leaving \( D(r(\mathbf{x}), \mathbf{Z}_k) \) an unknown function as well. To learn these unknown functions, we need to do physical or simulation-based experiments on a set of carefully selected design points in the design space \( \Omega \). Collecting the corresponding response data at these design points and fitting them via proper statistical methods, we can approximate \( r(\mathbf{x}) \) and/or \( D(r(\mathbf{x}), \mathbf{Z}_k) \) with surrogate models which can guide the cell-level design of metamaterials. In this section, we will discuss how to establish such a surrogate model efficiently.

2.1. Generating response data by computer simulation. Considering that it is timely and economically inefficient to implement unit cells with design \( \mathbf{x} \) physically, people take advantage of modern simulators to simulate the electromagnetic response of unit cells with different designs [Chen et al. (2006); Bilotti, Toscano and Vegni (2007)]. The simulator provides a full-wave simulation to produce the frequency responses of the unit cell with a specific design. But generally, this full-wave simulation is also time consuming, which may take tens of minutes for one unit cell. Digesting a given design \( \mathbf{x} \) and the WFB \( \mathcal{F} \), the simulator discretizes \( \mathcal{F} \) into \( M \) distinct frequency points denoted as \( \mathbb{F}_M \) and outputs the numerically approximated responses at these discretized frequency points, i.e., \( \mathbf{\bar{r}}(\mathbf{x}) = \{\mathbf{\bar{r}}_f(\mathbf{x})\}_{f \in \mathbb{F}_M} \), where \( M \) is a tuning parameter controlling the resolution of discretization and the computational cost. Here, we use the notation \( \mathbf{\bar{r}}_f(\mathbf{x}) \) to distinguish the numerically approximated response obtained by simulation from \( r_f(\mathbf{x}) \), the true physical response to frequency \( f \) of design \( \mathbf{x} \). In general, there exists a small deviation between \( \mathbf{\bar{r}}_f(\mathbf{x}) \) and \( r_f(\mathbf{x}) \), whose magnitude depends on many factors and is negatively correlated with the computation cost we invest in the simulation task. In practice, the deviation of \( \mathbf{\bar{r}}_f(\mathbf{x}) \) from \( r_f(\mathbf{x}) \) can be calibrated with the help of data obtained from physical experiments [Kennedy and O’Hagan (2001); Higdon et al. (2004); Guo et al. (2017a)]. In this paper, however, we simply ignore the difference between \( \mathbf{\bar{r}}_f(\mathbf{x}) \) and \( r_f(\mathbf{x}) \) and use \( \mathbf{\bar{r}}_f(\mathbf{x}) \) as the surrogate of \( r_f(\mathbf{x}) \) for all \( \mathbf{x} \in \Omega \) and \( f \in \mathbb{F}_M \). Thus, for a group of design points \( \mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n \) on which simulation experiments have been implemented, the observed data are \( \mathbf{R}_\mathcal{X} = \{r_f(\mathbf{x}) : \mathbf{x} \in \mathcal{X}, f \in \mathbb{F}_M\} \).

2.2. Functional versus summarized responses. Since the simulator outputs a whole vector of \( \mathbf{\bar{r}}(\mathbf{x}) \) at once for a given design \( \mathbf{x} \), it is natural to regard \( \mathbf{\bar{r}}(\mathbf{x}) \) as a vector-value function of \( \mathbf{x} \), and \( r(\mathbf{x}) \) as a functional response of \( \mathbf{x} \) accordingly. In the literature, many efforts have been given to learn \( r(\mathbf{x}) \) as a functional response of \( \mathbf{x} \) via equivalent physical model and rational regression [Majumdar et al. (2014); Guo et al. (2017b)]. But, it is in general challenging and computationally expensive to model the functional response of the entire frequency band.

In various practical cases, however, the WFB of interest is only a small proportion of the entire frequency band. In these cases, the response has small fluctuations in the WFB, and thus we can turn to utilize certain statistics to measure the discrepancy between the designed response and its corresponding target response. Due to such an insight, instead of directly working on the challenging functional response \( r(\mathbf{x}) \), in this study we work on simple responses summarized from \( r(\mathbf{x}) \) as below: specifying the deviation measurement \( D(\cdot, \cdot) \) by the \( L_2 \)-norm, we have

\[
(2.1) \quad d_k(\mathbf{x}) := D(r(\mathbf{x}), \mathbf{Z}_k) = \frac{1}{f_U - f_L} \int_{\mathcal{F}} |r_f(\mathbf{x}) - z_k(f)|^2 df
\]

as the summarized response of design \( \mathbf{x} \) with respect to target \( \mathbf{Z}_k \). Clearly, the vector of summarized responses \( \{d_k(\mathbf{x})\}_{k=1}^K \) encodes all useful information about design \( \mathbf{x} \) for the MTD problem of interest.
2.3. The target-free responses. In practice, the target response function \( Z_k \) often degenerates to a constant \( z_k \) in the WFB \( F \), i.e., \( z_k(f) = z_k \) for \( \forall f \in F \). In this case, we notice that for \( V(x) \in \Omega \),

\[
d_k(x) = \frac{1}{f_U - f_L} \int_F |r_f(x) - z_k|^2 df = (E(x) - z_k)^2 + V(x),
\]

where

\[
E(x) = \frac{1}{f_U - f_L} \int_F r_f(x) df \quad \text{and} \quad V(x) = \frac{1}{f_U - f_L} \int_F (r_f(x) - E(x))^2 df
\]

stand for the mean and variance of the response function \( r_f(x) \), respectively. In other words, the vector of summarized responses \( \{d_k(x)\}_{k=1}^K \) can be determined by two target-free responses \( E(x) \) and \( V(x) \), suggesting that we can search for qualifying designs based on \( E(x) \) and \( V(x) \) without knowing the concrete form of the complicated functional response \( r(x) \).

This fact demonstrates great advantages in practice, both conceptually and computationally. First, the summarized simple responses \( E(x) \) and \( V(x) \) enjoy a much simpler mathematical structure than the functional response \( r(x) \), and can be handled by a simpler model at a much cheaper price. Second, because \( E(x) \) and \( V(x) \) both come from local smoothing of \( r(x) \), they in general enjoy a better continuity with respect to \( x \) than \( r(x) \), resulting in an improved efficiency of statistical inference. Last and the most important, by organizing \( d_k(x) \) as a function of \( E(x), V(x) \) and \( z_k \), we identify \( E(x) \) and \( V(x) \) as the common building blocks for all response vectors \( \{d_k(x)\}_{k=1}^K \), leading to a dramatic reduction on the dimensionality of the problem. Moreover, considering that \( E(x) \) and \( V(x) \) reflect information about \( r(x) \) in orthogonal dimensions, it is reasonable to model them separately. All these facts greatly simplify the problem, and make it possible to build an effective surrogate model for the cell-level design of metamaterials and achieve efficient synergic design via Bayesian optimization, which we will introduce in details later.

In practice, the summarized responses \( E(x) \) and \( V(x) \) can be approximated by

\[
\tilde{E}(x) = \frac{1}{M} \sum_{j=1}^{M} \tilde{r}_{f_j}(x) \quad \text{and} \quad \tilde{V}(x) = \frac{1}{M} \sum_{j=1}^{M} (\tilde{r}_{f_j}(x) - \tilde{E}(x))^2.
\]

With a reasonably large \( M \) and a properly specified simulator, the difference between \( (\tilde{E}(x), \tilde{V}(x)) \) and \( (E(x), V(x)) \) can be safely ignored. Therefore, in this study, we will treat

\[
e(x) = \tilde{E}(x) \quad \text{and} \quad v(x) = \log \tilde{V}(x)
\]

as the working responses from simulation experiments and establish efficient design strategies for the cell-level design of metamaterials based on them. Note that we take \( v(x) = \log \tilde{V}(x) \), instead of \( \tilde{V}(x) \) itself as a working response, because \( v(x) \) takes values in the whole real line and makes it easier for statistical modeling and inference. From now on, we utilize

\[
d_k(x) = (e(x) - z_k)^2 + \exp(v(x)),
\]

as the working summarized response function.

2.4. Building surrogate models by Gaussian process. Previously, we have established \( e(x) \) and \( v(x) \) as the working responses. In this subsection, we discuss how to build surrogate models for them via Gaussian Process (GP). Because the two response functions \( e(x) \) and \( v(x) \) can be proceeded separately in an exactly same way, without loss of generality, here we use \( y(x) \) to stand for either \( e(x) \) or \( v(x) \) and describe the steps of building the surrogate model with \( y(x) \).
Gaussian Process is a powerful tool for non-parametric fitting of complex functions and has been widely used in computer experiments [Santner, Williams and Notz (2003); Kleijnen (2009)]. For any \( n \)-dimensional design vector \( X = (x_1, \cdots, x_n) \) from the design space \( \Omega \), let \( y(X) = (y(x_1), \cdots, y(x_n))^T \) be the corresponding response vector, which is abbreviated to \( y \) in case of no ambiguity. The GP model assumes that \( y \) follows a \( n \)-dimensional Gaussian distribution as follows:

\[
y(X) \sim N(\mu, \sigma^2, \Phi, X),
\]

where \( \mu \) is the \( n \)-dimensional column vector whose elements all equal to 1, and \( \Sigma \) is a \( n \times n \) covariance matrix whose \((i, j)\) element satisfies

\[
\Sigma_{ij} = \text{Cov}(y(x_i), y(x_j)) = \sigma^2 \mathcal{K}_\phi(x_i, x_j),
\]

where \( \mathcal{K}_\phi(\cdot, \cdot) \) is a pre-given correlation structure or kernel with \( \phi = (\phi_1, \ldots, \phi_d)^T \) as parameters. A popular choice of \( \mathcal{K}_\phi(\cdot, \cdot) \) is the Gaussian kernel

\[
\mathcal{K}_\phi(x_i, x_j) = \exp \left( -\sum_{s=1}^{d} (x_{is} - x_{js})^2 \phi_s \right).
\]

Evidently, the statistical properties of GP are completely determined by model parameters \( \theta = (\mu, \sigma^2, \phi) \) when the functional form of \( \mathcal{K}_\phi(\cdot, \cdot) \) is given. Hereinafter, we refer to the Gaussian process with parameters \( \theta \) as \( \text{GP}_\theta \).

In practice, the unknown parameters \( \mu, \sigma^2 \) and \( \phi \) can be estimated based on the maximum likelihood principle. To be concrete, given observed response \( y \) at \( n \) design points in \( X \), the log-likelihood function of \( y \) is

\[
\ell(\mu, \sigma^2, \phi) = \log f(y | \mu, \sigma^2, \phi, X)
\]

\[
= -\frac{1}{2\sigma^2}(y - \mu \cdot 1_n)^T R_\phi^{-1}(y - \mu \cdot 1_n) - \frac{n}{2} \ln(\sigma^2) - \frac{1}{2} \ln |R_\phi|,
\]

where \( R_\phi = (\mathcal{K}_\phi(x_i, x_j))_{1 \leq i, j \leq n} \) is the correlation matrix of \( y \). It is easy to check that for a fixed \( \phi \) (and thus, \( R_\phi \)), the log-likelihood \( \ell(\mu, \sigma^2, \phi) \) takes its maximum value at

\[
\hat{\mu} = \frac{1_n^T R_\phi^{-1} y}{1_n^T R_\phi^{-1} 1_n},
\]

\[
\hat{\sigma}^2 = \frac{1}{n} (y - \hat{\mu} \cdot 1_n)^T R_\phi^{-1} (y - \hat{\mu} \cdot 1_n).
\]

Further define \( \hat{\phi} = \arg \max_{\phi \in D_\phi} \ell(\mu, \sigma^2, \phi) \), where \( D_\phi = [\phi^l, \phi^u] \times [\phi^l, \phi^u] \times \cdots \times [\phi^l, \phi^u] \) is a hyper-rectangle in \( \mathbb{R}_+^d \). So \( \phi^l = (\phi^l_1, \ldots, \phi^l_d) \) and \( \phi^u = (\phi^u_1, \ldots, \phi^u_d) \) are the lower bound and upper bound of \( \phi \), respectively. Let \( \hat{\mu} = \mu_\phi \) and \( \hat{\sigma}^2 = \sigma^2_\phi \), we have

\[
(\hat{\mu}, \hat{\sigma}^2, \hat{\phi}) = \arg \max_{\phi \in D_\phi} \ell(\mu, \sigma^2, \phi),
\]

as the Maximum Likelihood Estimation (MLE) of \( \theta = (\mu, \sigma^2, \phi) \). The maximization involved can be achieved by standard optimization algorithms such as Genetic Algorithm [Roustant, Ginsbourger and Deville (2012)], Nelder-Mead [Olsson and Nelson (1975)] and so on.

With the estimated parameters \( \hat{\theta} \), the response value \( y(x) \) of an untested design point \( x \) can be predicted based on \( \text{GP}_\theta \) in the light of the observed responses \( y \). Let \( y(x|X) \) be the conditional random variable \( y(x) \) given \( y \) under \( \text{GP}_\theta \). Previous studies [Jones, Schonlau and Welch (1998); Roustant, Ginsbourger and Deville (2012)] have shown that by using the Bayes rule,

\[
y(x|X) \sim N \left( \mu_\phi(x), \sigma^2_\phi(x) \right),
\]
where
\[
\mu_{\theta}(x) = \hat{\mu} + \rho_x^T R_{\phi}^{-1} (y - \hat{\mu} \cdot 1_n),
\]
\[
\sigma^2_{e}(x) = \sigma^2 (1 - \rho_x^T R_{\phi}^{-1} \rho_x),
\]
with \( \rho_x = (K_{\phi}(x,x_1), \ldots, K_{\phi}(x,x_n))^T \) stands for the correlation vector between \( y(x) \) and \( y \). Practically, the distribution of \( y(x|\mathcal{X}) \) provides us a lot of information about \( y(x) \) based on \( y \) before a physical or simulation experiment is implemented for an untested design point \( x \), and thus can serve as an ideal surrogate model for the true response \( y(x) \). With the increase of \( n \), the number of implemented design points along the process, \( y(x|\mathcal{X}) \) would approximate \( y(x) \) with better accuracy and less uncertainty.

From the above analysis, we can obtain the surrogate models of \( e(x) \) and \( v(x) \) denoted by \( e(x|\mathcal{X}) \) and \( v(x|\mathcal{X}) \) respectively by specifying \( y \) as \( e \) or \( v \). In this study, we use notations \( \hat{e}(x) \) and \( \hat{s}^2_e(x) \) to refer to respectively the mean and variance of \( e(x|\mathcal{X}) \), and \( \hat{v}(x) \) and \( \hat{s}^2_v(x) \) to refer to the mean and variance of \( v(x|\mathcal{X}) \). Thus, we have
\[
e(x|\mathcal{X}) \sim N(\hat{e}(x), \hat{s}^2_e(x)) \quad \text{and} \quad v(x|\mathcal{X}) \sim N(\hat{v}(x), \hat{s}^2_v(x)).
\]

Based on \( e(x|\mathcal{X}) \) and \( v(x|\mathcal{X}) \), the following surrogate model can be established for \( d_k(x) \):
\[
d_k(x|\mathcal{X}) = (e(x|\mathcal{X}) - z_k)^2 + \exp(v(x|\mathcal{X})),
\]
which is a function of two independent random variables \( e(x|\mathcal{X}) \) and \( v(x|\mathcal{X}) \). And thus, it is easy to see that
\[
d_k(x|\mathcal{X}) \sim \chi^2 + \text{LN},
\]
i.e., \( d_k(x|\mathcal{X}) \) has the same distribution as the summation of a generalized Chi-Square random variable and an independent log-normal random variable. We denote the distribution of \( d_k(x|\mathcal{X}) \) as \( E_{k,\mathcal{X}} \).

Please note that because \( e(x|\mathcal{X}) \) and \( v(x|\mathcal{X}) \) degenerate to point masses at \( e(x) \) and \( v(x) \) respectively for all \( x \in \mathcal{X} \), we also have \( d_k(x|\mathcal{X}) \) degenerates to a point mass at \( d_k(x) \) for all \( x \in \mathcal{X} \).

3. Bayesian optimization with synergic design. Bayesian Optimization (BO) is a sequential design framework that has been widely used in computer design to improve design efficiency [Jones, Schonlau and Welch (1998); Shahriari et al. (2016)]. Based on a surrogate model established from data collected from the implemented experiments to approximate the response function, BO chooses design points in the next step by optimizing an acquisition function which quantifies the expected reward of the design points to be selected. Once the experiments at the chosen design points are implemented, the surrogate model is updated based on the new data, thus we complete the loop. There are four key steps in a typical BO procedure: initializing the system with pilot experiments, building/updating the surrogate model, designing the acquisition function, and optimizing the acquisition function to find new design points. The previous section has established surrogate models based on Gaussian process. In this section, we will discuss how to implement the other three key steps.

3.1. Initialization via pilot experiments. BO depends on a surrogate model to guide the learning process. To obtain an effective surrogate model at the beginning, pilot experiments are needed to explore the huge design space and initialize the system. In this study, we use the Latin hypercube design (LHD) [Tang (1993)] for this purpose due to its space-filling property and straightforward implementation. Using LHD can sample points which spread evenly over each input space [Santner, Williams and Notz (2003)]. Let us consider the construction of a
$d$-dimensional LHD with $n$ points taking values in the hypercube $[0, 1]^d$. Let $A = (a_{ij})$ be a $n \times d$ matrix, where each column is formed by independently generating a permutation on $\{1, \ldots, n\}$. Then a $n \times d$ LHD matrix $B = (x_{ij})$ can be generated by

$$x_{ij} = \frac{a_{ij} - u_{ij}}{n}, \text{for } i = 1, \ldots, n, j = 1, \ldots, d,$$

where the $u_{ij}$s are $U[0, 1]$ random variables [Xiong, Qian and Wu (2013)]. More sophisticated design strategies with the space-filling property, e.g., the variates of LHD [Owen (1992); Ye (1998)] and the uniform designs [Fang et al. (2000); Sun, Wang and Xu (2019)] can also be used to initialize the pilot experiments.

3.2. The acquisition function based on expected improvement. A unique challenge in the cell-level design of metamaterials is the MTD problem, in which we are requested to find a series of designs to fit $K$ different design targets within one design path (with $K$ in the order of $10^3$-$10^5$). With so many targets to fit simultaneously, great conceptual and computational challenges have been posted for Bayesian optimization.

To make the problem traceable, we start with the simple situation where only one target is considered. Let $\mathcal{X} = \{x_1, \ldots, x_n\}$ be the set of $n$ implemented designs and $\xi(\mathcal{X}) = \{\xi(x) : x \in \mathcal{X}\}$ be the corresponding responses, where $\xi(x) = (e(x), v(x))$ is the sufficient response vector of design $x$. For the $k$-th design target, we define

$$d_{k,n}^e = \min_{x \in \mathcal{X}} d_k(x) \quad \text{and} \quad x_{k,n}^* = \arg\min_{x \in \mathcal{X}} d_k(x)$$

to be the smallest distance and optimal design with respect to $z_k$ in design set $\mathcal{X}$, respectively. Evidently, we have

$$d_{k,n}^e = d_k(x_{k,n}^*) = (e_{k,n}^* - z_k)^2 + \exp(v_{k,n}^*),$$

where $e_{k,n}^* = e(x_{k,n}^*)$ and $v_{k,n}^* = v(x_{k,n}^*)$. For $x \not\in \mathcal{X}$, let $d_k(x|\mathcal{X})$ be its estimated divergence from target $k$ based on the surrogate models of $e(x)$ and $v(x)$. We define the target-specific expected improvement (tEI) on target $k$ of design $x$ given $\mathcal{X}$ as

$$I_k(x|\mathcal{X}) = \mathbb{E} \left[ \max \left\{ d_{k,n}^e - d_k(x|\mathcal{X}), 0 \right\} \right],$$

where the expectation is with respect to random variable $d_k(x|\mathcal{X})$. Evidently, $I_k(x|\mathcal{X})$ quantifies the expected improvement on target $k$ of a hypothetical implementation of design $x$ predicted by the surrogate model based on $\xi(\mathcal{X})$. In this paper, we use $I_k(x|\mathcal{X})$ defined in (3.2) as the basic building block for the acquisition function.

Now, let’s discuss the general case where $K$ distinct design targets are involved. Given the set of implemented designs $\mathcal{X}$, we define

$$\mathcal{L}_p = \{ k : d_{k,n}^e > \epsilon_k \} \quad \text{and} \quad \mathcal{Z}_p = \{ z_k : k \in \mathcal{L}_p \}$$

to be the pending index set and the set of pending design targets that are not fulfilled yet, respectively. For $x \not\in \mathcal{X}$, we define the summarized expected improvement (sEI) of $x$ with respect to target $k$ given $\mathcal{X}$ as

$$I(x|\mathcal{X}) = \frac{1}{|\mathcal{L}_p|} \sum_{k \in \mathcal{L}_p} I_k(x|\mathcal{X}).$$

Evidently, $I(x|\mathcal{X})$ integrates the overall expected improvement on all pending targets of a hypothetical implementation of design $x$ predicted by the surrogate model based on $\xi(\mathcal{X})$. In this paper, we use $I(x|\mathcal{X})$ as the acquisition function to search for the next design point. Acquisition function $I(x|\mathcal{X})$ has two unique features: first, it takes only the pending targets into
account, leaving all fulfilled targets behind; second, it aggregates expected improvements on all pending targets with equal weights. These features make sure that the searching algorithm always focuses on the pending targets and finds the new design point in a global instead of local fashion.

In the literature, the acquisition function based on expected improvement has been widely used in Bayesian optimization for classic design problems [Jones, Schonlau and Welch (1998); Jones (2001)]. There are also other criteria to design acquisition functions, including the probability of improvement (PI) [Jones (2001)], the upper confidence bound (UCB) [Srinivas et al. (2010)], the entropy search (ES) [Hennig and Schuler (2012)] and the knowledge gradient (KG) [Scott, Frazier and Powell (2011)]. Here, we choose the expected improvement criterion because of its straightforward interpretation and easy computation.

3.3. An upper bound of the acquisition function. Because \( d_k(x|X) \sim F_{F_kX} \), the distribution of the summation of a generalized \( \chi^2 \) random variable and an independent log-normal random variable, we have

\[
I_k(x|X) = \mathbb{E} \left[ \max \{ d^\ast_{k,n} - d_k(x|X), 0 \} \right]
= \int \left( d^\ast_{k,n} - t \right) \mathbb{I}(t \leq d^\ast_{k,n}) dF_{F_kX}(t),
\]

\[
(3.4)
= d^\ast_{k,n} \cdot \int \mathbb{I}(t \leq d^\ast_{k,n}) dF_{F_kX}(t) - \int t \cdot \mathbb{I}(t \leq d^\ast_{k,n}) dF_{F_kX}(t).
\]

However, due to the complicated structure of distribution \( F_{F_kX} \), no analytic form of \( I_k(x|X) \) is available based on (3.4). Although it is possible to use Monte Carlo methods to approximate \( I_k(x|X) \), the Monte Carlo approximation is computationally inefficient and makes the downstream optimization of \( I(x|X) \) as difficult as its components \( I_k(x|X) \)'s which have no analytic forms. These facts post critical computational challenges in this study.

To avoid these computational challenges, in this subsection we propose to replace the acquisition function \( I(x|X) \) by a slightly loose upper bound denoted by \( \tilde{I}(x|X) \), which enjoys much easier computation. To be concrete, we define

\[
\delta_{d,k}(x) = d^\ast_{k,n} - d_k(x|X),
\]

\[
\delta_{e,k}(x) = (e^\ast_k - z_k) - (e(x|X) - z_k),
\]

\[
\delta_{e,k}(x) = \exp(v^\ast_k) - \exp(v(x|X)).
\]

Evidently, we have

\[
\delta_{d,k}(x) = d^\ast_{k,n} - d_k(x|X) = \left( e^\ast_k - z_k \right)^2 + \exp(v^\ast_k) - \left( (e(x|X) - z_k)^2 + \exp(v(x|X)) \right)
\]

\[
= \left( e^\ast_k - z_k \right)^2 - (e(x|X) - z_k)^2 + \exp(v^\ast_k) - \exp(v(x|X))
\]

\[
= \delta_{e,k}(x) + \delta_{\nu,k}(x).
\]

Considering that

\[
\max \{ \delta_{d,k}(x), 0 \} \leq \max \{ \delta_{e,k}(x), 0 \} + \max \{ \delta_{\nu,k}(x), 0 \},
\]

we have

\[
\tilde{I}_k(x|X) = \mathbb{E} \left[ \max \{ \delta_{d,k}(x), 0 \} \right]
\leq \mathbb{E} \left[ \max \{ \delta_{e,k}(x), 0 \} \right] + \mathbb{E} \left[ \max \{ \delta_{\nu,k}(x), 0 \} \right]
\]

\[
= \tilde{I}_{e,k}(x|X) + \tilde{I}_{\nu,k}(x|X) = \tilde{I}_k(x|X),
\]
where
\[
\tilde{I}_{c,k}(x|X) = \mathbb{E} \left[ \max \{ \delta_{c,k}(x), 0 \} \right], \quad \tilde{I}_{v,k}(x|X) = \mathbb{E} \left[ \max \{ \delta_{v,k}(x), 0 \} \right],
\]
and \( \tilde{I}_k(x|X) \) is an upper bound of \( I_k(x|X) \). Compared to \( I_k(x|X) \), \( \tilde{I}_k(x|X) \) enjoys much easier computation as it separates the two independent random variables \( e(x|X) \) and \( v(x|X) \) into two additive terms, namely \( \tilde{I}_{c,k}(x|X) \) and \( \tilde{I}_{v,k}(x|X) \), whose concrete form is given by the following proposition (with detailed calculations provided in Appendix A).

**Proposition 1.**
\[
\tilde{I}_{c,k}(x|X) = \left( \hat{\delta}_{c,k}(x) - \hat{s}_c^2(x) \right) \left[ \Phi(h_{c,k}(x)) - \Phi(l_{c,k}(x)) \right] + s_c^2(x) \left[ h_{c,k}(x) \phi(l_{c,k}(x)) - l_{c,k}(x) \phi(h_{c,k}(x)) \right],
\]
\[
\tilde{I}_{v,k}(x|X) = \exp(v_{k,n}^*) \Phi(m_{v,k}(x)) - g_v(x) \Phi(f_{v,k}(x)),
\]
where
\[
\hat{\delta}_{c,k}(x) = (e_{k,n}^* - \hat{z}_k)^2 - (\hat{e}(x) - \hat{z}_k)^2,
\]
\[
h_{c,k}(x) = \frac{\hat{z}_k - \hat{e}(x) + |\hat{z}_k - e_{k,n}^*|}{\hat{s}_c(x)},
\]
\[
l_{c,k}(x) = \frac{\hat{z}_k - \hat{e}(x) - |\hat{z}_k - e_{k,n}^*|}{\hat{s}_c(x)},
\]
\[
m_{v,k}(x) = \frac{v_{k,n}^* - \hat{v}(x)}{\hat{s}_v(x)},
\]
\[
f_{v,k}(x) = m_{v,k}(x) - \hat{s}_v(x),
\]
\[
g_v(x) = \exp \left\{ \hat{v}(x) + \frac{\hat{s}_v^2(x)}{2} \right\}.
\]

An interesting special case happens when the WFB \( \mathcal{F} \) shrinks to a frequency point. In such a special case, term \( V(x) \) in \((2.2)\) vanishes, and thus
\[
I_k(x|X) = \tilde{I}_k(x|X) = \tilde{I}_{c,k}(x|X),
\]
i.e., \( \tilde{I}_k \) becomes a sharp upper bound of \( I_k \). In this paper, we define
\[
\tilde{I}(x|X) = \frac{1}{|L_p|} \sum_{k \in L_p} \tilde{I}_k(x|X)
\]
to be the upper bound of \( I(x|X) \), and use it as the working acquisition function (i.e., the surrogate of \( \tilde{I}(x|X) \)) in practice.

3.4. **Determining the next design point.** According to the standard procedure of Bayesian optimization, we determine the new design point \( x_{n+1} \) by maximizing the acquisition function \( \tilde{I}(x|X) \). Considering the challenges in computing \( \tilde{I}(x|X) \), however, we propose to maximize \( \tilde{I}(x|X) \), the upper bound of \( I(x|X) \) with an analytic form, instead:
\[
x_{n+1} = \arg \max_{x \in \Omega} \tilde{I}(x|X).
\]

Considering that \( \tilde{I}(x|X) \) is a continuous function of \( x \) given \( X \), we use the *rgenoud* package in R [Mebane and Sekhon (2011)] to resolve the above optimization problem. The *rgenoud* package provides a powerful global optimizer by combining Evolutionary Algorithm (EA)
with a derivative-based (i.e., quasi-Newton) method. The primary advantage of using EA in \textit{rgenoud} is to avoid being trapped in local optimal solutions encountered in a fast derivative-based method. The combination of EA and quasi-Newton makes the algorithm converge to the global optimum faster. Thus, \textit{rgenoud} can solve a wide variety of difficult optimization problems and is particularly suitable for our case. To run the package, we need the derivative of \(\tilde{I}(x|\lambda)\) as input, which we will provide the details below. Considering that

\[
\nabla \tilde{I}(x|\lambda) = \frac{1}{|L_p|} \sum_{k \in L_p} \nabla \tilde{I}_{e,k}(x|\lambda) + \frac{1}{|L_p|} \sum_{k \in L_p} \nabla \tilde{I}_{v,k}(x|\lambda),
\]

we only need to focus on \(\nabla \tilde{I}_{e,k}(x|\lambda)\) and \(\nabla \tilde{I}_{v,k}(x|\lambda)\). With \(\tilde{I}_{e,k}(x|\lambda)\) and \(\tilde{I}_{v,k}(x|\lambda)\) given in closed forms, it is easy to calculate \(\nabla \tilde{I}_{e,k}(x|\lambda)\) and \(\nabla \tilde{I}_{v,k}(x|\lambda)\). The proposition below provides the details.

**Proposition 2.**

\[
\nabla \tilde{I}_{e,k}(x|\lambda) = \left( h_{e,k}(x)l_{e,k}(x) + \hat{\delta}_{e,k}(x) - \hat{s}_{e,k}(x) \right) [\phi(h_{e,k}(x)) \nabla h_{e,k}(x) - \phi(l_{e,k}(x)) \nabla l_{e,k}(x)] \\
\quad + \left( \nabla \hat{\delta}_{e,k}(x) - 2 \hat{s}_{e,k}(x) \nabla \hat{s}_{e,k}(x) \right) [\Phi(h_{e,k}(x)) - \Phi(l_{e,k}(x))] \\
\quad + 2 \hat{s}_{e,k}(x) \nabla \hat{s}_{e,k}(x) [h_{e,k}(x) \phi(l_{e,k}(x)) - l_{e,k}(x) \phi(h_{e,k}(x))]
\]

(3.9)

\[
\nabla \tilde{I}_{v,k}(x|\lambda) = \exp(v_{\phi,k}(x)) \phi(m_{v,k}(x)) \nabla m_{v,k}(x) - \Phi(f_{v,k}(x)) \nabla g_v(x) - g_v(x) \phi(f_{v,k}(x)) \nabla f_{v,k}(x),
\]

(3.10)

where

\[
\nabla h_{e,k}(x) = - \frac{\nabla \hat{e}(x) + h_{e,k}(x) \nabla \hat{s}_{e,k}(x)}{\hat{s}_{e,k}(x)},
\]

\[
\nabla l_{e,k}(x) = - \frac{\nabla \hat{e}(x) + l_{e,k}(x) \nabla \hat{s}_{e,k}(x)}{\hat{s}_{e,k}(x)},
\]

\[
\nabla \hat{\delta}_{e,k}(x) = -2(\hat{e}(x) - z_k) \nabla \hat{e}(x),
\]

\[
\nabla m_{v,k}(x) = - \frac{\nabla \hat{v}(x) + m_{v,k}(x) \nabla \hat{s}_{v,k}(x)}{\hat{s}_{v,k}(x)},
\]

\[
\nabla g_v(x) = g_v(x) (\nabla \hat{v}(x) + \hat{s}_{v,k}(x) \nabla \hat{s}_{v,k}(x)),
\]

\[
\nabla f_{v,k}(x) = \nabla m_{v,k}(x) - \nabla \hat{s}_{v,k}(x).
\]

### 3.5. The algorithm.

Now, we summarize the key steps of the BO procedure discussed previously into a formal algorithm as shown in Algorithm 1. Taking the target response vector \(Z = \{z_1, \ldots, z_K\}\) and the error level vector \(\epsilon = \{\epsilon_1, \ldots, \epsilon_K\}\) as inputs, the algorithm implements the four key steps of BO and outputs the best design point \(X^*_p\) for each design target \(z_k\) discovered along the design path. We set \(L_p = \{1, \ldots, K, Z_p = Z\) in the initialization. Considering that multiple design targets involved are integrated and optimized simultaneously in this algorithm to resolve the MTD problem, we refer to such a BO procedure as a synergic design on multiple design targets as \(BO_{syn}\).
Algorithm 1 The $BO_{\text{syn}}$ algorithm for the cell-level design of metamaterials

Input: $Z$ and $e$

Initialization. Select a set of pilot design points $X = \{x_1, \ldots, x_n\}$ by LHD, run simulation experiments at all selected pilot design points to get the corresponding responses $e(X) = \{e(x)\}_{x \in X}$ and $v(X) = \{v(x)\}_{x \in X}$, and set $D_e = \{(x, e(x))\}_{x \in X}$, $D_v = \{(x, v(x))\}_{x \in X}$;

while $Z_\pi \neq \emptyset$ do

Surrogate model for $e(x)$. Calculate the MLE $\hat{\theta}$, for response $e(x)$ based on data $D_s$ as in (2.7), and establish the surrogate model $e(x|X)$ as in (2.8);

Surrogate model for $v(x)$. Calculate the MLE $\hat{\theta}$, for response $v(x)$ based on data $D_s$ as in (2.7), and establish the surrogate model $v(x|X)$ as in (2.8);

Working acquisition function. Calculate the acquisition function $\tilde{I}(x|X)$ as in (3.7);

Next query. Find the next design point $x_{n+1} = \arg \max_{x \in \Omega} \tilde{I}(x|X)$, and run simulation experiment at $x_{n+1}$ to get corresponding responses $e(x_{n+1})$ and $v(x_{n+1})$;

Status updating. Update $X = X \cup \{x_{n+1}\}$, $D_e = D_e \cup \{(x_{n+1}, e(x_{n+1}))\}$, and $D_v = D_v \cup \{(x_{n+1}, v(x_{n+1}))\}$;

for $k$ in $L_p$ do

Calculate $d^k_{x_\pi} = \min_{x \in X} d_k(x)$;

if $d^k_{x_\pi} \leq e_k$ then

Set $X^\pi = \arg \min_{x \in X} d_k(x)$ and update $L_p = L_p \setminus \{k\}$ and $Z_p = Z_p \setminus \{z_k\}$;

end if

end for

end while

Return $\{X_1, \ldots, X_k\}$.

A key advantage of the $BO_{\text{syn}}$ algorithm lies on the fact that it integrates appeals of all pending targets into one acquisition function and determines the next query based on it, achieving a global, instead of local optimization in choosing the design path. This feature plays a critical role in resolving the MTD problem. Below, we demonstrate the insight via a toy example. For simplicity, we choose a one-dimensional continuous function $y(x) = \sin x$ with $x \in [0, 2\pi]$ as the true response function and set two target responses which are $z_1 = 1$ and $z_2 = -1$. We are supposed to find two designs $x_1^*$ and $x_2^*$ in the design space $\Omega = [0, 2\pi]$ satisfying $|y(x_1^*) - z_1| \leq \varepsilon_1$ and $|y(x_2^*) - z_2| \leq \varepsilon_2$ with $\varepsilon_1 = \varepsilon_2 = 10^{-4}$. Note that because the concept of WFB degenerates to a single point in this toy example, we have $I = \tilde{I}$ and $I_k = \tilde{I}_k$ for $k \in \{1, 2\}$. Figure 2 provides a graphical illustration of working process of the $BO_{\text{syn}}$ algorithm, where the 4 panels (a)-(d) stand for the 4 iterations of the algorithm. In each panel, two figures are presented: the figure on the top illustrates the surrogate model versus the true response function with the investigated design path highlighted; the one at the bottom compares the sEI function $I(x)$ to the two tEI functions $I_1(x)$ and $I_2(x)$. From the figure, we can see that the $BO_{\text{syn}}$ algorithm works effectively under the guidance of the sEI function $I(x)$, with a global view in choosing the next query.
3.6. Convergence of the BO$_{syn}$ algorithm. In this subsection, we will prove that the solutions reported by the BO$_{syn}$ algorithm converge to the global optimums under mild conditions. To be concrete, recall that $X^*_k = \arg \min_{x \in \Omega} d_k(x)$ is the optimal solution in the design space $\Omega$ to design target $z_k$, $X^*_{k,n} = \arg \min_{x \in X} d_k(x)$ is the solution reported by BO$_{syn}$ after $n$ design points summarized by $\hat{X}$, and define the loss function of the BO$_{syn}$ solutions with respect to the optimal ones as

$$L_n(BO_{syn}) = \sum_{k=1}^{K} [d_k(x^*_{k,n}) - d_k(X^*_k)].$$

Clearly, the desired result can be proved by showing that $L_n(BO_{syn})$ converges to zero with $n$ goes to infinity.
Based on the definition, \( L_n(BO_{syn}) \) is composed of three elements, the summarized response function \( d_k(\cdot) \), a group of \( K \) optimal design points \( \{X^*_k\}_{k=1}^K \), and a sequence of \( n \) design points summarized by \( X \). But, thinking a bit deeper, we will find that \( L_n(BO_{syn}) \) is essentially determined by the two working responses \( e(\cdot) \) and \( v(\cdot) \), as all the three elements \( d_k(\cdot) \), \( \{X^*_k\}_{k=1}^K \), \( K \) and \( X \) are deterministic with the initialization of the pilot design points fixed. In this paper, we model both \( e(\cdot) \) and \( v(\cdot) \) by GP\( _\theta \), where parameter \( \theta = (\mu, \sigma^2, \phi) \) with \( \mu, \sigma^2 \) and \( \phi \) determining the mean, variance and correlation structure (or, kernel) of the GP, respectively. Let \( \mathcal{K}_\phi(\cdot, \cdot) \) be the GP kernel with parameter \( \phi \in D_\phi \), a hyper-rectangle in \( \mathbb{R}_d^d \) determined by the pre-given lower bound \( \phi' = (\phi'_1, \ldots, \phi'_d) \) and the upper bound \( \phi'' = (\phi''_1, \ldots, \phi''_d) \). For the compact set \( \Omega \subset \mathbb{R}_d^d \) and each specific \( \phi \in D_\phi \), a reproducing kernel Hilbert space (RKHS) \( \mathcal{H}_\phi(\Omega) \) can be derived from kernel \( \mathcal{K}_\phi(\cdot, \cdot) \), which contains all nicely behaved functions \( \mathcal{H}_e(\Omega) \) with an inner product \( \langle \cdot, \cdot \rangle \) satisfying the reproducing property: \( \langle y, \mathcal{K}(x, \cdot) \rangle = y(x), \forall y \in \mathcal{H}_\phi(\Omega) \). The RKHS norm is defined as \( ||y|| = \sqrt{\langle y, y \rangle} \). Define \( \mathcal{H}_e(\Omega) = \mathcal{H}_\phi(\Omega) \), which satisfies \( \mathcal{H}_e(\Omega) \subseteq \mathcal{H}_\phi(\Omega) \) for \( \forall \phi \in D_\phi \). And also the Gaussian kernel we use in Section 2.4 has nicely isotropic and regular properties based on Bull (2011). We have the following theorem that guarantees the convergence of the \( BO_{syn} \) algorithm for the MTD problem under mild conditions.

**Theorem 1.** If \( e(\cdot), v(\cdot) \in \mathcal{H}_\phi(\Omega) \), \( ||e|| \leq R_e \) and \( ||v|| \leq R_v \), where \( R_e > 0 \) and \( R_v > 0 \), then for any set of pilot design points in initialization, we always have

\[
\lim_{n \to \infty} L_n(BO_{syn}) = 0.
\]

For the general BO task seeking to model an unknown objective function \( y(x) \) defined on the design space \( \Omega \) with GP and finding its minimum \( \min_{x \in \Omega} y(x) \), Vazquez and Bect (2010) have shown that when the parameters of GP prior are known and fixed, the BO approach with the expected improvement acquisition function converges to the minimum of any \( y \in \mathcal{H}_\phi(\Omega) \). Bull (2011) extends the result and provides the convergence rate of the expected improvement strategy for both fixed parameters and estimated parameters of GP priors under some mild conditions. Our proof for Theorem 1 is built on the top of these two previous works. The details are provided in Appendix B.

### 4. Alternative Approaches.

**4.1. Bayesian optimization with independent or sequential design.** Alternatively, we can also decompose a MTD problem into \( K \) simple sub-problems, each for one target response \( z_k \), and deal with them one by one separately in parallel or sequentially by classic design methods focusing on one design target only. In this section, we briefly discuss these alternative approaches, whose efficiency is in general lower than the \( BO_{syn} \) approach established previously.

A straightforward alternative solution is to deal with the \( K \) design targets of interest one by one separately. Algorithm 2 summarizes the Bayesian optimization procedure with this independent design strategy, which is referred to as \( BO_{ind} \). Different from the \( BO_{syn} \) algorithm, which evolves by considering all pending targets simultaneously with the surrogate model \( d_k(x|X) \) in (2.10) of \( d_k(x) \) constructed by two independent GP models for \( e(x) \) and \( v(x) \) based on all implemented design points \( X \), the \( BO_{ind} \) deals with the \( K \) design targets in parallel, with one GP model based on the target-specific implemented design points \( X_k \) for each \( d_k(x) \) as its surrogate model directly denoted by \( \hat{d}_k(x|X_k) \). In such a setting, we can simply use a new acquisition function \( I_{d,k}(x|X_k) \) based on \( \hat{d}_k(x|X_k) \), and find the next query by maximizing it. The definition and detailed calculations of \( I_{d,k}(x|X_k) \) are provided in Appendix C.
The BO\textsubscript{ind} algorithm enjoys a simple structure and can be implemented in a parallel fashion easily, but is statistically inefficient as the information in experiments implemented for one design target cannot be utilized by other design targets, and computationally expensive since we need to build \( K \) distinct surrogate models, each for one design target.

Another possible solution is to deal with the \( K \) design targets sequentially so that the information of implemented experiments in early stages can be accumulated for design targets coming later. Algorithm 3 summarizes the Bayesian optimization procedure with this sequential design strategy, which is referred to as \( BO\textsubscript{seq} \). Different from the \( BO\textsubscript{ind} \) algorithm, which starts over for each design target, the \( BO\textsubscript{seq} \) algorithm uses all implemented experiments as initialization for a new target.

### Algorithm 2

**The \( BO\textsubscript{ind} \) algorithm for the cell-level design of metamaterials**

**Input:** \( Z \) and \( \varepsilon \)

**for** \( (k = 1; k \leq K; k++) \) **do**

**Initialization.** Select a set of pilot design points \( X_k = \{x_1, \ldots, x_n\} \) by LHD, run simulation experiments at all selected pilot design points to get the functional responses \( r(X_k) = [r(x)]_{x \in X_k} \) and calculate \( d_1(x) \) as in (2.1) for all \( x \in X_k \). Set \( D_k = \{ (x, d_1(x)) \}_{x \in X_k} \), \( d'_{k,n} = \min_{x \in X_k} d_1(x) \) and \( X'_k = \arg \min_{x \in X_k} d_1(x) \);

**while** \( d'_{k,n} > \varepsilon_k \) **do**

**Surrogate model for** \( d_1(x) \). Calculate the MLE \( \hat{\theta}_d \) for target-specific response \( d_1(x) \) based on data \( D_k \) as in \( 2.7 \), and establish the surrogate model \( \tilde{d}_1(x|X_k) \) for \( d_k(x) \) as in \( 2.8 \);

**Acquisition function.** Calculate the acquisition function \( I_{d,k}(x|X_k) \) as discussed in Appendix C.

**Next query.** Find the next design point \( x_{k,n+1} = \arg \max_{x \in \Omega} I_{d,k}(x|X_k) \) and run simulation experiment at \( x_{k,n+1} \) to get corresponding response \( d_k(x_{k,n+1}) \);

**Status updating.** Update \( X_k = X_k \cup \{x_{k,n+1}\} \) and \( D_k = D_k \cup \{(x_{k,n+1}, d_k(x_{k,n+1}))\} \);

**end if**

**end while**

**end for**

**Return** \( (X'_1, \ldots, X'_K) \).

### Algorithm 3

**The \( BO\textsubscript{seq} \) algorithm for the cell-level design of metamaterials**

**Input:** \( Z \) and \( \varepsilon \)

**Pilot experiments.** Select a set of pilot design points \( X = \{x_1, \ldots, x_n\} \) by LHD, run simulation experiments at all selected pilot design points to get the functional responses \( r(X) = [r(x)]_{x \in X} \);

**Target sorting.** Sort the \( K \) design targets \( z_1, \ldots, z_K \) by a certain order (e.g., increasing), and without loss of generality still denote the sorted targets as \( z_1, \ldots, z_K \);

**for** \( (k = 1; k \leq K; k++) \) **do**

**Initialization.** Calculate \( d_1(x) \) as in (2.1) for all \( x \in X \), and set \( D = \{(x, d_1(x))\}_{x \in X} \), \( d'_{k,n} = \min_{x \in X} d_1(x) \), and \( X'_k = \arg \min_{x \in X} d_1(x) \);

**while** \( d'_{k,n} > \varepsilon_k \) **do**

**Surrogate model for** \( d_1(x) \). Calculate the MLE \( \hat{\theta}_d \) for target-specific response \( d_1(x) \) based on data \( D \) as in \( 2.7 \), and establish the surrogate model \( \tilde{d}_1(x|X) \) for \( d_k(x) \) as in \( 2.8 \);

**Acquisition function.** Calculate the acquisition function \( I_{d,d}(x|X) \) as discussed in Appendix C.

**Next query.** Find the next design point \( x_{k,n+1} = \arg \max_{x \in \Omega} I_{d,d}(x|X) \), and run simulation experiment at \( x_{k,n+1} \) to get corresponding response \( d_k(x_{k,n+1}) \);

**Status updating.** Update \( X = X \cup \{x_{k,n+1}\} \), \( D = D \cup \{(x_{k,n+1}, d_k(x_{k,n+1}))\} \);

**end if**

**end while**

**end for**

**Return** \( (X'_1, \ldots, X'_K) \).
4.2. Bayesian optimization with batches of points. For each pending target in the \(BO_{ind}\) and \(BO_{seq}\) algorithm, only one design point is sampled in each iteration by maximizing the acquisition function \(I_{d,k}(x|X_k)\) or \(I_{d,k}(x|\mathcal{X})\), resulting too many sequential sampling iterations. In some real applications, computer experiments can be implemented in parallel clusters so that multiple design points can be evaluated simultaneously. We can take advantage of the parallel computation and implement multiple design points in each iteration. Ginsbourger, Riche and Carraro (2008) proposed a multi-point optimization strategy, called multi-points expected improvement (q-EI), to query \(q\) design points in each iteration. The acquisition function \(q\)-EI has an analytic form when \(q = 2\). Let \(I_{d,k}(x, \mathbf{x}'|\mathbf{X}_k)\) denote the 2-EI acquisition function involving two design variables \((x, \mathbf{x}')\). Replacing \(I_{d,k}(x|\mathcal{X})\) with \(I_{d,k}(x, \mathbf{x}'|\mathbf{X}_k)\) in \(BO_{ind}\) and \(I_{d,k}(x|\mathcal{X})\) with \(I_{d,k}(x, \mathbf{x}'|\mathcal{X})\) in \(BO_{seq}\), we obtain batched algorithms of the \(BO_{ind}\) and \(BO_{seq}\), referred as to \(BO_{ind-batch}\) and \(BO_{seq-batch}\), respectively. The calculations of \(I_{d,k}(x, \mathbf{x}'|\mathbf{X}_k)\) and \(I_{d,k}(x, \mathbf{x}'|\mathcal{X})\) are detailed in Appendix C. Algorithm 4 and Algorithm 5 demonstrate the pseudocodes of \(BO_{ind-batch}\) and \(BO_{seq-batch}\).

4.3. Bayesian optimization with joint-modeling design. We then consider an intermediate strategy of \(BO_{syn}\) by replacing the summarized acquisition function \(\hat{I}(x|\mathcal{X})\) in Algorithm 1 with the target-specific acquisition function \(\hat{I}_k(x|\mathcal{X})\) and optimize \(\hat{I}_k(x|\mathcal{X})\) for each pending target. The joint modeling of \(e(x)\) and \(v(x)\) is kept, and we refer to the intermediate algorithm as \(BO_{jm}\) and detail it in Algorithm 6. The \(BO_{jm}\) algorithm can assess how much the summarized acquisition function \(\hat{I}(x|\mathcal{X})\) in \(BO_{syn}\) contributes to the gain in performance.

**Algorithm 4** The \(BO_{ind-batch}\) algorithm for the cell-level design of metamaterials

**Input:** \(\mathcal{X}\) and \(e\)

for \((k = 1; k \leq K; k++)\) do

**Initialization.** Select a set of pilot design points \(X_k = \{x_1, \ldots, x_n\}\) by LHD, run simulation experiments at all selected pilot design points to get the functional responses \(r(X_k) = [r(x)|x \in X_k]\) and calculate \(d_l(x)\) as in (2.1) for all \(x \in X_k\). Set \(D_k = [(x, d_l(x))]_{x \in X_k}\), \(d_{l,n}^* = \min_{x \in X_k} d_l(x)\) and \(X_k^* = \arg\min_{x \in X_k} d_l(x)\);

while \(d_{l,n}^* > e_k\) do

**Surrogate model for** \(d_l(x)\). Calculate the MLE \(\hat{\theta}_d\) for target-specific response \(d_l(x)\) based on data \(D_k\) as in (2.7), and establish the surrogate model \(\hat{d}_l(x|\mathcal{X})\) for \(d_l(x)\) as in (2.8);

**Acquisition function.** Calculate the acquisition function \(I_{d,k}(x, \mathbf{x}'|\mathcal{X})\) as discussed in Appendix C.

**Next query.** Find the next design point \((x_{l,n+1}, x_{l,n+2}) = \arg\max_{x \in \mathcal{D}_k} I_{d,k}(x, \mathbf{x}'|\mathcal{X})\) and run simulation experiment at \(x_{l,n+1}\) and \(x_{l,n+2}\) to get corresponding responses \(d_l(x_{l,n+1})\) and \(d_l(x_{l,n+2})\);

**Status updating.** Update \(X_k = X_k \cup \{x_{l,n+1}, x_{l,n+2}\}\) and \(\mathcal{D}_k = \mathcal{D}_k \cup [(x_{l,n+1}, d_l(x_{l,n+1}))] \cup [(x_{l,n+2}, d_l(x_{l,n+2}))];\)

if \(\min(d_l(x_{l,n+1}), d_l(x_{l,n+2})) < d_{l,n}^*\) then

Update \(d_{l,n}^* = \min(d_l(x_{l,n+1}), d_l(x_{l,n+2}))\). If \(d_l(x_{l,n+1}) < d_l(x_{l,n+2})\), set \(X_k^* = x_{l,n+1};\) otherwise, set \(X_k^* = x_{l,n+2};\)

end if

end while

end for

Return \(X_k, \ldots, X_K\).
Algorithm 5 The BOseq-batch algorithm for the cell-level design of metamaterials

Input: $\mathcal{Z}$ and $\varepsilon$

Pilot experiments. Select a set of pilot design points $X = \{x_1, \ldots, x_p\}$ by LHD, run simulation experiments at all selected pilot design points to get the functional responses $r(\cdot) = (r(\cdot))_{x \in X}$;

Target sorting. Sort the $K$ design targets $z_1, \ldots, z_K$ by a certain order (e.g., increasing), and without loss of generality still denote the sorted targets as $z_1, \ldots, z_K$;

for $k = 1:k \leq K,k++$ do

Initialization. Calculate $d_k(x)$ as in (2.1) for all $x \in X$, and set $\mathcal{D} = \{(x, d_k(x))\}_{x \in X}$; $d_{\varepsilon_k} = \min_{x \in X} d_k(x)$ and $X_k = \{x \in X : d_{\varepsilon_k} \leq d_k(x)\};$

while $d_{\varepsilon_k} > \varepsilon_k$ do

Surrogate model for $d_k(x)$. Calculate the MLE $\hat{\theta}_d$ for target-specific response $d_k(x)$ based on data $\mathcal{D}$ as in (2.7), and establish the surrogate model $\hat{d}_k(x|\mathcal{X})$ for $d_k(x)$ as in (2.8);

Acquisition function. Calculate the acquisition function $I_d(|x, \mathcal{X}|)$ as discussed in Appendix C.

Next query. Find the next design point $(x_{k,n_1}, x_{k,n_2}) = \arg\max_{x \in \mathcal{X}} I_d(|x, \mathcal{X}|)$ and run simulation experiment at $x_{k,n_1}$ and $x_{k,n_2}$ to get corresponding responses $d_k(x_{k,n_1})$ and $d_k(x_{k,n_2})$;

Status updating. Update $\mathcal{D} = \mathcal{D} \cup \{(x_{k,n_1}, d_k(x_{k,n_1}))\} \cup \{(x_{k,n_2}, d_k(x_{k,n_2}))\}$;

if $\min\{d_k(x_{k,n_1}), d_k(x_{k,n_2})\} < d_{\varepsilon_k}$ then

Update $d_{\varepsilon_k} = \min\{d_k(x_{k,n_1}), d_k(x_{k,n_2})\}$. If $d_k(x_{k,n_1}) < d_k(x_{k,n_2})$, set $X_k^* = x_{k,n_1}$; otherwise, set $X_k^* = x_{k,n_2}$;

end if

end while

end for

Return $(X_1^*, \ldots, X_K^*)$.

Algorithm 6 The BOjm algorithm for the cell-level design of metamaterials

Input: $\mathcal{Z}$ and $\varepsilon$

Initialization. Select a set of pilot design points $X = \{x_1, \ldots, x_p\}$ by LHD, run simulation experiments at all selected pilot design points to get the corresponding responses $e(\cdot) = (e(\cdot))_{x \in X}$ and $v(\cdot) = (v(\cdot))_{x \in X}$, and set $\mathcal{D}_e = \{(x, e(x))\}_{x \in X}$, $\mathcal{D}_v = \{(x, v(x))\}_{x \in X}$;

while $\mathcal{Z}_p \neq \emptyset$ do

Surrogate model for $e(x)$. Calculate the MLE $\hat{\theta}$, for response $e(x)$ based on data $\mathcal{D}_e$ as in (2.7), and establish the surrogate model $\hat{e}(x|\mathcal{X})$ as in (2.8);

Surrogate model for $v(x)$. Calculate the MLE $\hat{\theta}$, for response $v(x)$ based on data $\mathcal{D}_v$ as in (2.7), and establish the surrogate model $\hat{v}(x|\mathcal{X})$ as in (2.8);

for $k \in L_p$ do

Working acquisition function. Calculate the acquisition function $\tilde{I}_d(x|\mathcal{X})$;

Next query. Find the next design point $x_{k,n+1} = \arg\max_{x \in \mathcal{X}} \tilde{I}_d(x|\mathcal{X})$, and run simulation experiment at $x_{k,n+1}$ to get corresponding responses $e(x_{k,n+1})$, $v(x_{k,n+1})$ and $d_k(x_{k,n+1})$;

Status updating. Update $\mathcal{D}_e = \mathcal{D}_e \cup \{(x_{k,n+1}, e(x_{k,n+1}))\}$, $\mathcal{D}_v = \mathcal{D}_v \cup \{(x_{k,n+1}, v(x_{k,n+1}))\}$, and $\mathcal{D}_d = \mathcal{D}_d \cup \{(x_{k,n+1}, d_k(x_{k,n+1}))\}$;

if $d_k(x_{k,n+1}) \leq \varepsilon_k$ then

Set $X_k^* = x_{k,n+1}$, and update $L_p = L_p \setminus \{k\}$ and $\mathcal{Z}_p = \mathcal{Z}_p \setminus \{z_k\}$;

end if

end for

end while

Return $(X_1^*, \ldots, X_K^*)$.

4.4. Genetic algorithm. Solutions beyond the BO framework are available as well. For example, the evolutionary algorithms, which mimic the natural evolution of a population of individuals with distinct genetic codes. As one of the evolutionary algorithms, Genetic Algorithm (GA) has been widely used in metamaterial design [Li et al. (2012); Zhao et al. (2011)]. In GA, each design point $x = (x_1, \ldots, x_d)$ is associated with a chromosome $C_x$, which is typically a binary vector corresponding to the binary code of $x$: we transform each real
number \( x_j \) into its binary code \( C_j \) of length \( n_c \) and concatenate \( C_j \)'s to get the chromosome \( C_x = (C_1, \cdots, C_d) \). The code length \( n_c \) is usually determined by the decimal place accuracy and the range of \( x_j \). For example, if we want to code \( x_j \in [0,1.5] \) as a chromosome with a precision of 3 decimal places, the range \([0,1.5]\) with length 1.5 should be divided into at least \((1.5 - 0) \times 10^3\) equal-size intervals to represent and distinguish different values of \( x_j \). Since the inequality \( 2^{10} < (1.5 - 0) \times 10^3 < 2^{11} \) holds, we need to code a binary vector with length \( n_c \) to be at least 11 to meet the required precision. We also use the precision of 3 decimal places in rest of the paper. For each design target \( z_k \), we define \( f_k(x) = -d_k(x) \) as the fitness of a design point \( x \) and use the roulette wheel selection [Haupt and Haupt (2004)] (which assumes the selection probability is proportionate to the fitness) to select design points in the current generation to generate the next generation by random mating. Here, we follow the settings in [Li et al. (2012); Zhao et al. (2011)] to allow a single-point crossover operation with a crossover probability \( p_c = 0.8 \), a single-point mutation operation with a mutation probability \( p_m = 0.08 \) in each random mating, and set the population size of each generation \( n_p = 20 \). To get better performance, we also deal with the \( K \) design targets in a sequential fashion as in the \( BO_{seq} \) algorithm, in which information of implemented experiments in early stages is kept to guide the design of targets afterwards. We refer to such a GA-based sequential design strategy as \( GA_{seq} \), and summarize its key steps in Algorithm 7.

Algorithm 7 The \( GA_{seq} \) for the cell-level design of metamaterials

**Input:** \( Z, \varepsilon, n_c, p_c, \) and \( p_m \)

**Pilot experiments.** Randomly select a set of pilot design points \( X = \{x_1, \ldots, x_N\} \), run simulation experiments at all selected design points to get the functional responses \( r(x) = [r(x)]_{k \in K} \).

**Target sorting.** Sort the \( K \) design targets \( z_1, \cdots, z_K \) by a certain order (e.g., increasing), and without loss of generality still denote the sorted targets as \( z_1, \cdots, z_K \);

for \( (k = 1;k \leq K;k++) \) do

**Initialization.** Calculate \( d_k(x) \) as in (2.1) for all \( x \in X \), and set \( d_{k,\text{opt}} = \min_{x \in X} d_k(x) \), \( X_k^* = \arg \min_{x \in X} d_k(x) \);

while \( d_{k,\text{opt}} > \varepsilon_k \) do

Calculate the probability \( p_k = \frac{f_k(x)}{\sum_{k \in K} f_k(x)} \) for each \( x \in X \);

**Selection.** Select \( n_p \) individuals denoted as \( X_i \) from \( X \) with probabilities \( \{p_{k \in K} \} \);

**Crossover.** Generate the child individuals with crossover probability \( p_c \) on each pair of parent individuals from \( X_i \);

**Mutation.** Update \( X_i \) with the mutation probability \( p_m \);

**Status updating.** Update \( X = X \cup X_i \) and calculate \( X_k^* = \arg \min_{x \in X} d_k(x) \);

if \( d_k(x^*) < d_{k,\text{opt}} \) then

Update \( d_{k,\text{opt}} = d_k(x^*) \) and \( X_k^* = x^* \);

end if

end while

end for

Return: \( (X_1^*, \ldots, X_K^*) \).

4.5. MFEA and MFEA-II. Some multi-task EAs widely used in MTO problems can also be used to resolve MTO problems. For example, the MFEA [Gupta, Ong and Feng (2016)] and its extension MFEA-II [Bali et al. (2020)] are the mainstream algorithms in MTO. MFEA firstly generates a population of \( N \) individuals for each target and then iteratively updates the population. Let \( P^k \) denote the set of individuals of the \( k \)th target. For the \( i \)th individual, \( \tau_i \) represents the target index which the individual is associated with, and \( F_i = 1/r_{\tau_i} \) represents the individual’s fitness function, where \( r_{\tau_i} \) is the rank of the \( i \)th individual \( x_i \) on target \( \tau_i \), which is proportional to \( d_{\tau_i}(x_i) \). Two individuals from the same target index implement intra-target crossover operation, and two individuals from different targets implement inter-target
crossover operation with a pre-given scalar parameter labeled as the random mating probability \((rmp)\), \(rmp \in [0, 1]\). We set \(rmp = 0.3\) as in Gupta, Ong and Feng (2016). Algorithm 8 summarizes the pseudocode of MFEA.

Algorithm 8 The MFEA for the cell-level design of metamaterials

<table>
<thead>
<tr>
<th>Input: (Z) and (\varepsilon)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initialization</strong> Generate (N \cdot K) individuals as the initial population (P(0));</td>
</tr>
<tr>
<td><strong>for</strong> each individual (x_i) in (P(0)) <strong>do</strong></td>
</tr>
<tr>
<td>Assign the target index (\tau_i = \text{mod}(i, K) + 1) for (x_i);</td>
</tr>
<tr>
<td>Evaluate (x_i) for target (\tau_i) only;</td>
</tr>
<tr>
<td><strong>end for</strong></td>
</tr>
<tr>
<td>Set (t = 1) and (c = 0);</td>
</tr>
<tr>
<td><strong>while</strong> (c \leq K) <strong>do</strong></td>
</tr>
<tr>
<td>Sample (N \cdot K/2) individuals denoted as (P(t)) based on the fitness function;</td>
</tr>
<tr>
<td>Set (P_c(t) = \emptyset);</td>
</tr>
<tr>
<td><strong>while</strong> #(P_c(t) \leq NK) <strong>do</strong></td>
</tr>
<tr>
<td>Randomly sample two individuals ([m_i, m_j]) from (P(t)) without replacement;</td>
</tr>
<tr>
<td><strong>if</strong> (\tau_i = \tau_j) <strong>then</strong></td>
</tr>
<tr>
<td>Implement intra-task crossover operation between (m_i) and (m_j) and generate two new individuals ([m_a, m_b]). Assign ([m_a, m_b]) with target index (\tau_i);</td>
</tr>
<tr>
<td><strong>else if</strong> (\text{rand} \leq rmp) <strong>then</strong></td>
</tr>
<tr>
<td>Implement inter-task crossover operation between (m_i) and (m_j) and generate two new individuals ([m_a, m_b]). Assign ([m_a, m_b]) with target index (\tau_i) or (\tau_j);</td>
</tr>
<tr>
<td><strong>else</strong></td>
</tr>
<tr>
<td>Implement mutation operation on (m_i) to obtain new individual (m_a) and assign (m_a) with target index (\tau_i);</td>
</tr>
<tr>
<td>Implement mutation operation on (m_j) to obtain new individual (m_b) and assign (m_b) with target index (\tau_j);</td>
</tr>
<tr>
<td><strong>end if</strong></td>
</tr>
<tr>
<td>Evaluate ([m_a, m_b]) for their assigned target only;</td>
</tr>
<tr>
<td><strong>for</strong> (x) in ([m_a, m_b]) <strong>do</strong></td>
</tr>
<tr>
<td>Let (k \in [\tau_i, \tau_j]) be the target index of the individual (x);</td>
</tr>
<tr>
<td><strong>if</strong> (d_k(x) \leq \varepsilon_k) <strong>then</strong></td>
</tr>
<tr>
<td>Set (X^*_k = x) and (c = c + 1);</td>
</tr>
<tr>
<td><strong>end if</strong></td>
</tr>
<tr>
<td><strong>end for</strong></td>
</tr>
<tr>
<td>Set (P_c(t) = P_c(t) \cup [m_a, m_b]);</td>
</tr>
<tr>
<td><strong>end while</strong></td>
</tr>
<tr>
<td>Set (P(t + 1) = P_c(t), t = t + 1);</td>
</tr>
<tr>
<td><strong>end while</strong></td>
</tr>
<tr>
<td>Return: ((X^<em>_1, \ldots, X^</em>_K)).</td>
</tr>
</tbody>
</table>

In MFEA, the random mating probability \(rmp\) is a fixed parameter. However, when negative correlations exist between different targets, the computation efficiency of MFEA is sensitive to the choice of \(rmp\). Bali et al. (2020) improved MFEA and proposed MFEA-II by assuming that there exists an unknown random mating probability matrix \(RMP\) with size \(K \times K\) between \(K\) different targets. MFEA-II uses a probabilistic model to estimate and update \(RMP\) based on the data generated during the evolution process. The framework of MFEA-II is basically similar to Algorithm 8, and we only need to replace the fixed scalar \(rmp\) in MFEA with a matrix \(RMP\) estimated by maximum likelihood in MFEA-II. Thus there is no need to detail the pseudocode of MFEA-II.

5. Simulation studies. In this section, we evaluate the performance of the proposed methods via simulations. Three groups of simulation studies are implemented for this purpose. The first group of simulation studies focus on MTD problems with a simple real-value
response function \( y(x) \) defined on a bounded region of \( \mathbb{R}^d \). The second group of simulation studies consider a more general case where functional instead of real-value response \( r(x) = \{r_f(x)\}_{f \in F} \) is involved. With the functional form of \( y(x) \) and \( r(x) \) explicitly known in simulation, we have the full control to evaluate and compare the proposed methods. The third group of simulation studies consider the general metamaterial design problem for a given WFB \( [f_L, f_U] \) with \( f_L < f_U \), where the functional response \( r(x) \) is not precisely known and relies on computer simulation to achieve function evaluation, the same setting we encounter in metamaterial design in practice.

Given the \( K \) target responses, the total number of function evaluations is defined by \( N_f \). We evaluate the performance of a design strategy by the average number of function evaluations for reaching a target response defined by \( N_{\text{ave}} = N_f / K \). Evidently, a smaller \( N_{\text{ave}} \) indicates an algorithm of better performance. In practical applications, we expect \( N_{\text{ave}} \) to be as small as possible. The nine methods of interest, \( GA_{\text{seq}}, MFEA, MFEA-II, BO_{\text{ind}}, BO_{\text{seq}}, BO_{\text{ind-batch}}, BO_{\text{seq-batch}}, BO_{\text{pm}} \) and \( BO_{\text{syn}} \) are compared in the following simulation studies.

### 5.1. The MTD problem with simple responses

In this simulation study, we focus on the MTD problem with a simple real-value response function \( y(x) \) defined on a bounded region of \( \mathbb{R}^d \). We specify the underlying response function \( y(x) \) with a 2-dimensional Branin function or a 6-dimensional Rastrigin function respectively as demonstrated in Table 1, and allow the number of targets \( K \) to be 10, 100, or 1,000, leading to \( 2 \times 3 = 6 \) distinct simulation settings. For all 6 settings, we assume that the design targets \( \{z_1, \cdots, z_K\} \) sit on the equal-size grid of the value space of the response function \( y(x) \), and set the error level \( \varepsilon_k = 10^{-4} \) for all \( k \) to guarantee that adjacent targets can be fitted with enough precision.

<table>
<thead>
<tr>
<th>Response function Configuration</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Branin ((d = 2)) ( y(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^3 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10, ) ( x_1 \in [-5, 10], x_2 \in [0, 15]. )</td>
<td></td>
</tr>
<tr>
<td>Rastrigin ((d = 6)) ( y(x) = 10d + \sum_{j=1}^{d} \left[x_j^2 - 10\cos(2\pi x_j)\right], ) ( x_i \in [-5, 5], ) for all ( j = 1, \cdots, d. )</td>
<td></td>
</tr>
</tbody>
</table>

Table 2 summarizes the mean of the average number of function evaluations and the standard deviations (in the brackets) of 10 independent experiments used by different methods to resolve the design problems. From Table 2, we can see that the six BO algorithms, \( BO_{\text{ind}}, BO_{\text{seq}}, BO_{\text{ind-batch}}, BO_{\text{seq-batch}}, BO_{\text{pm}} \) and \( BO_{\text{syn}} \) uniformly outperform the three evolutionary algorithms, \( GA_{\text{seq}}, MFEA \) and \( MFEA-II \), with around 90% reduction of simulation cost. The efficiency of the BO algorithms are more prominent especially when the dimension \( d \) increases. MFEA and MFEA-II need to generate and evaluate new individuals proportional to \( K \) for each generation without dynamically updating the pending target set \( Z_p \); therefore, the efficiency of MFEA and MFEA-II decreases when increasing \( K \). Interestingly, we find that as \( K \) increases, the average number of function evaluations of \( GA_{\text{seq}} \) decreases a lot. The reason for this phenomenon is that the distance between two adjacent targets becomes smaller for a larger \( K \), which makes it easier for \( GA_{\text{seq}} \) to find the targets. Among the six BO algorithms, \( BO_{\text{syn}} \) performs the best with a 30-70% reduction of simulation cost; \( BO_{\text{pm}} \) is less efficient than \( BO_{\text{syn}} \), but outperforms all other alternative approaches; \( BO_{\text{seq}} \) and \( BO_{\text{seq-batch}} \) are more...
efficient than $BO_{ind}$ and $BO_{ind\text{-batch}}$; using batches in BO (i.e., $BO_{ind\text{-batch}}$ or $BO_{seq\text{-batch}}$) evaluates multiple design points in parallel which may reduce the number of iterations but does not significantly improve the average number of function evaluations.

### Table 2

*Average number of function evaluations used by different methods for simple responses.*

<table>
<thead>
<tr>
<th>Problem settings</th>
<th>EAs</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response function</td>
<td>K (G_{A_{seq}})</td>
<td>(MFEA)</td>
</tr>
<tr>
<td>Branin ((d = 2))</td>
<td>10</td>
<td>198 (\pm) 218</td>
</tr>
<tr>
<td>100</td>
<td>72.1 (\pm) 14.4</td>
<td>502 (\pm) 142</td>
</tr>
<tr>
<td>1000</td>
<td>16.5 (\pm) 5.99</td>
<td>996 (\pm) 145</td>
</tr>
<tr>
<td>Rastrigin ((d = 6))</td>
<td>10</td>
<td>582 (\pm) 419</td>
</tr>
<tr>
<td>100</td>
<td>403 (\pm) 106</td>
<td>548 (\pm) 227</td>
</tr>
<tr>
<td>1000</td>
<td>172 (\pm) 10.9</td>
<td>1263 (\pm) 93</td>
</tr>
</tbody>
</table>

5.2. The MTD problem with functional responses. In this simulation study, we evaluate the proposed methods for the more general cases where functional responses are involved. We modify the simple test function \(y(x)\) in Table 1 by a functional test function \(r(x) = \{r_f(x)\}_{f \in F}\), where

\[
r_f(x) = \frac{||x||}{A} \cdot \beta(f) + y(x),
\]

with \(\beta(f) = \sin(\pi f), A = 10^3, f \in F = [0, 2]\), and the region of \(x\) is the same as in Table 1. We still use the 6 distinct simulation settings and the design targets defined in Section 5.1. By using (2.3), we can calculate \(E(x)\) and \(V(x)\). In Figure 3, we illustrate the landscape of the responses \(E(x)\) and \(V(x)\) of the two test functions on 2,000 design points uniformly sampled from the design space. According to Figure 3, we notice that the variances of some design targets cannot approach zero, so it is not possible to reach an admissible error level of \(10^{-4}\) for all design targets. We revise the admissible error level to \(\varepsilon_k = 2 \times 10^{-4}\) for all \(k\).

We implement 10 independent experiments and summarize the mean of the average number of function evaluations and the standard deviations (in the brackets) in Table 3. Compared with Table 2, we find that the proposed methods have similar results over the functional responses demonstrated in Table 3. The BO algorithms still have the competitive performances, and \(BO_{syn}\) performs the best.
Figs 3. The scatter plot of $E(x)$ and $V(x)$ defined in (2.3) for the functional responses of (a) Branin and (b) Rastrigin in the design space.

<table>
<thead>
<tr>
<th>Problem settings</th>
<th>EAs</th>
<th>Methods</th>
<th>BO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response function</td>
<td>$K$</td>
<td>$GA_{seq}$</td>
<td>MFEA</td>
</tr>
<tr>
<td>Branin ($d = 2$)</td>
<td>10</td>
<td>193 (184)</td>
<td>293 (159)</td>
</tr>
<tr>
<td>100</td>
<td>73.6 (17.9)</td>
<td>491 (139)</td>
<td>480 (140)</td>
</tr>
<tr>
<td>1000</td>
<td>15.4 (6.21)</td>
<td>987 (201)</td>
<td>814 (186)</td>
</tr>
<tr>
<td>Rastrigin ($d = 6$)</td>
<td>10</td>
<td>592 (397)</td>
<td>471 (262)</td>
</tr>
<tr>
<td>100</td>
<td>412 (125)</td>
<td>545 (238)</td>
<td>511 (202)</td>
</tr>
<tr>
<td>1000</td>
<td>169 (11.9)</td>
<td>1305 (183)</td>
<td>1281 (164)</td>
</tr>
</tbody>
</table>

5.3. Design of metamaterials. In this subsection, we demonstrate how to apply the proposed methods to the practice of metamaterial design. Here, we focus on I-shape unit cells as shown in Figure 1, whose capability of forming metamaterials with good properties has been proved by previous studies [Liu et al. (2009); Liu and Ji (2014)]. An I-shape unit contains 4 geometric parameters $l, a, w_1$ and $w_2$, whose ranges are (1.4, 9), (1.4, 8), (0.2, 0.8) and (0.2, 0.8), respectively (the unit is mm). Our primary goal is to design metamaterials with a specific electric-permittivity pattern for electromagnetic wave nearby 13GHz, a task similar to the study in [Liu and Ji (2014)] aiming to design metamaterials that can steer the wave beam of 13GHz to a desired direction. Here, we consider two WFBs nearby 13GHz, the narrow one $F_1 = [12.99, 13.01]$GHz and the wider one $F_2 = [12.9, 13.1]$GHz.
Figure 4 and Figure 5 illustrate the landscape of the responses \( \tilde{E}(x) \) and \( \tilde{V}(x) \) for \( \mathcal{F}_1 \) and \( \mathcal{F}_2 \), respectively. The figures are obtained via numerical interpolation based on a larger scale computer simulation on 5,000 design points uniformly sampled from the design space. From the figures we can see that \( \tilde{E}(x) \) and \( \tilde{V}(x) \) are smooth functions in the design space, confirming that it is reasonable to build surrogate models of them by Gaussian process. Please note that the 5,000 simulations are implemented only for visualization and exploratory study of the response surfaces, and will not be used in the EAs and BO algorithms for design of metamaterials. Figure 6 shows the scatter plot of \( \tilde{E}(x) \) versus \( \tilde{V}(x) \) for \( \mathcal{F}_1 \) and \( \mathcal{F}_2 \), suggesting that the \( I \)-shape unit cells under study can achieve small \( \tilde{V}(x) \) for \( \tilde{E}(x) \) close to \(-600, -200 \) and \( 100 \) within WFBs \( \mathcal{F}_1 \) and \( \mathcal{F}_2 \). Based on the figure, we pick up 1,000 target responses (i.e., permittivity) to fit as follow: we first specify the range of target responses to be \( R = [-590, -500] \cup [-250, -150] \cup [40, 110] \) (i.e., the region of \( \tilde{E}(x) \) whose \( \tilde{V}(x) \) is possible to be smaller than 0.1 according to Figure 6(a)), and then select \( Z = \{z_1, \ldots, z_{1000}\} \) from this range with an equal-size grid. The smallest gap between two adjacent design targets is 0.26 in such a setting, which can separate the design targets reasonably well.

![Figure 4](image1.png)

![Figure 5](image2.png)

**Fig 4.** The landscape of \( \tilde{E}(x) \) and \( \tilde{V}(x) \) defined in (2.4) for \( \mathcal{F}_1 = [12.99, 13.01] \) GHz based on the numerical simulations of the metamaterials on 5,000 design points uniformly sampled from the design space. Subfigures (a) and (b) demonstrate the landscape of \( \tilde{E}(x) \) and \( \tilde{V}(x) \) as functions of \( a \) and \( l \) when \( w_1 = w_2 = 0.5 \); and, subfigures (c) and (d) demonstrate the landscape of \( \tilde{E}(x) \) and \( \tilde{V}(x) \) as functions of \( w_1 \) and \( w_2 \) when \( a = l = 5 \).
Fig 5. The landscape of $\tilde{E}(\mathbf{x})$ and $\tilde{V}(\mathbf{x})$ defined in (2.4) for $\mathcal{F}_2 = [12.9, 13.1]GHz$ based on the numerical simulations of the metamaterials on 5,000 design points uniformly sampled from the design space. Subfigures (a) and (b) demonstrate the landscape of $\tilde{E}(\mathbf{x})$ and $\tilde{V}(\mathbf{x})$ as functions of $a$ and $l$ when $w_1 = w_2 = 0.5$; and, subfigures (c) and (d) demonstrate the landscape of $\tilde{E}(\mathbf{x})$ and $\tilde{V}(\mathbf{x})$ as functions of $w_1$ and $w_2$ when $a = l = 5$.

Fig 6. The scatter plot of $\tilde{E}(\mathbf{x})$ and $\tilde{V}(\mathbf{x})$ defined in (2.4) for (a) $\mathcal{F}_1 = [12.99, 13.01]GHz$ and (b) $\mathcal{F}_2 = [12.9, 13.1]GHz$ based on the numerical simulations of the metamaterials on 5,000 design points uniformly sampled from the design space.
Then we apply the nine methods to find proper designs of I-shape unit cells for the 1,000 targets under the two WFBs $\mathcal{F}_1$ and $\mathcal{F}_2$. The error level $\varepsilon$ is specified to 0.1 and 1 for $\mathcal{F}_1$ and $\mathcal{F}_2$, respectively. We implement 5 independent experiments and report the mean of the average number of function evaluations and the standard deviations (in the brackets). The results are summarized in Table 4, suggesting that $BO_{syn}$ outperforms all competing methods significantly with the reduction on computational cost of at least 35%.

<table>
<thead>
<tr>
<th>Problem settings</th>
<th>Methods</th>
<th>EAs</th>
<th>MFEA</th>
<th>MFEA-II</th>
<th>BOind</th>
<th>BOind-batch</th>
<th>BOseq</th>
<th>BOseq-batch</th>
<th>BOjm</th>
<th>BOsyn</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{F}_1$</td>
<td>1000</td>
<td>102.31</td>
<td>1377</td>
<td>1337</td>
<td>40.41</td>
<td>42.07</td>
<td>11.92</td>
<td>13.84</td>
<td>8.28</td>
<td>7.68</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(241.57)</td>
<td>(253.20)</td>
<td>(10.41)</td>
<td>(4.14)</td>
<td>(6.34)</td>
<td>(1.04)</td>
<td>(1.28)</td>
<td>(0.53)</td>
<td>(0.76)</td>
</tr>
<tr>
<td>$\mathcal{F}_2$</td>
<td>1000</td>
<td>106.51</td>
<td>1485</td>
<td>1425</td>
<td>56.22</td>
<td>58.81</td>
<td>16.74</td>
<td>17.86</td>
<td>11.97</td>
<td>10.29</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(15.81)</td>
<td>(221.43)</td>
<td>(203.17)</td>
<td>(4.98)</td>
<td>(3.39)</td>
<td>(1.57)</td>
<td>(1.39)</td>
<td>(0.83)</td>
<td>(0.92)</td>
</tr>
</tbody>
</table>

6. Conclusions. Motivated by the great appeal of metamaterials in industry and the critical bottleneck of metamaterial design, we propose a novel framework to achieve rapid design of metamaterials based on computer experiments and Bayesian optimization. We point out that the statistical nature of the cell-level design of metamaterials is a multi-target design problem, in which the designer needs to find $K$ distinct designs in the design space to fit the $K$ various design targets simultaneously. Although the multi-target design problem can be transformed to a series of simple single-target design problems by dealing with the $K$ design targets one by one separately, this trivial solution often suffers from low efficiency and huge waste of computation power.

This paper proposes a synergic strategy to fit the $K$ design targets under the framework of Bayesian optimization. Measuring the deviation of the response curve $r(x)$ of a design $x$ from the desired response $z_k$ by their $L_2$ distance over the working frequency band $\mathcal{F} = [f_L, f_U]$ of the metamaterial of interest, namely $d_k(x) = \frac{1}{f_U - f_L} \int_{f_L}^{f_U} |r_f(x) - z_k|^2 df$, we use $d_k(x)$ as the target-specific response with respect to design target $k$. Observing that $d_k(x)$ can be further decomposed into two target-free components $E(x)$ and $V(x)$, which stand for the mean and variance of response curve $r(x)$ of design $x$, respectively, we propose to build surrogate models for $E(x)$ and $V(x)$ via Gaussian Process based on the data obtained, and use them as the building blocks to generate surrogate models for $d_k(x)s$. Such a strategy dramatically reduces the dimensionality of the problem and makes use of the information in observed data in an efficient way. We further calculate the working target-specific expected improvement of a potential query point $x$ on each target $z_k$, namely $\tilde{I}_k(x|X)$, and integrate them into a working summarized expected improvement $\tilde{I}(x|X)$ by summation. The next query point is determined by maximizing $\tilde{I}(x|X)$. Simulation studies confirm that the proposed method works effectively and leads to a significant reduction on the number of experiments compared to alternative approaches.

Based on the requirements of a practical problem, the system-level design of metamaterial provides the optical response of each unit cell. Then in the cell-level design stage, the proposed synergic strategy can find the target designs of unit cells efficiently. By arranging those unit cells to produce masks, we can obtain the metamaterial surfaces through photolithography. At present, the photolithography of metamaterials is a mature technology with the processing error well controlled. By using calibration methods, the responses produced...
by the computer simulations can be consistent with the responses produced by the physical experiments. So we can design various geometric shapes of the unit cells of metamaterials to constitute the functional device, which makes the proposed approach practically valuable.

APPENDIX

Appendix A. Detailed calculations of $\tilde I_{e,k}(x|X)$ and $\tilde I_{v,k}(x|X)$ in Proposition 1. Let $F_{e,X}^X$ and $F_{v,X}^X$ be the cumulative distribution of $e(x|X)$ and $v(x|X)$. Because

$$\delta_{d,k}(x) = d_{k,n}^e - d_k(x|X) = \left[ (e_k^* - z_k)^2 - (e(x|X) - z_k)^2 \right] + \left[ \exp(v_k^*) - \exp(v(x|X)) \right]$$

we have

$$\tilde I_{e,k}(x|X) = \mathbb{E}\left[ \max \{\delta_{e,k}(x), 0\} \right]$$

$$= \int_{A_e} \left[ (e_k^* - z_k)^2 - (t - z_k)^2 \right] \cdot dF_{e,X}^X(t)$$

(A.1)

$$= \left( e_k^* - z_k \right)^2 \int_{A_e} 1 \cdot dF_{e,X}^X(t) + 2z_k \int_{A_e} t \cdot dF_{e,X}^X(t) - \int_{A_e} t^2 \cdot dF_{e,X}^X(t),$$

where

$$A_e = \{e(x|X) : (e(x|X) - z_k)^2 \leq (e_{k,n}^* - z_k)^2, x \in \Omega\} = \{t : z_k - |e_{k,n}^* - z_k| \leq t \leq z_k + |e_{k,n}^* - z_k|\},$$

$$A_v = \{v(x|X) : \exp(v(x|X)) \leq \exp(v_k^*), x \in \Omega\} = \{t : t \leq v_k^*\}.$$

By rearranging the second part of (A.2), we have

$$\tilde I_{v,k}(x|X) = \mathbb{E}\left[ \max \{\delta_{v,k}(x), 0\} \right],$$

$$= \int_{A_v} \left[ \exp(v_k^*) - \exp(t) \right] \cdot dF_{v,X}^X(t)$$

(A.2)

$$= \exp(v_k^*) \int_{A_v} 1 \cdot dF_{v,X}^X(t) - \int_{A_v} \exp(t) \cdot dF_{v,X}^X(t),$$

where

$$w(x|X) = v(x|X) + \tilde s^2_v(x), \text{i.e., } w(x|X) \sim N(\tilde v(x) + \tilde s^2_v(x), \tilde s^2_v(x)), \text{ and } F_{w,X}^X is the cumulative distribution of w(x|X). \text{ So the calculations of } \tilde I_{e,k}(x|X) \text{ and } \tilde I_{v,k}(x|X) \text{ can be resolved by the integration of the zeroth, first and second moment of a normal random variable on a closed interval which we will study next.}$$

Considering a normal random variable $Y$ with mean $\mu$ and variance $\sigma^2$ whose density function is denoted by $f_Y(y)$, then $Y$ conditional on $l \leq Y \leq u$ with $-\infty \leq l \leq u \leq \infty$ is a truncated normal random variable [Johnson, Kotz and Balakrishnan (1994)]. Denoting the conditional random variable as $X = Y|l \leq Y \leq u$, its density function, $f_X(x)$, is given by

$$f_X(x) = \frac{1}{J} \phi \left( \frac{x - \mu}{\sigma} \right) = \frac{f_Y(x)}{J}, \text{ for } l \leq x \leq u,$$

where $J = \Phi \left( \frac{u - \mu}{\sigma} \right) - \Phi \left( \frac{l - \mu}{\sigma} \right)$.
The $n$-th moment of $Y$ on the integration interval $[l, u]$ is given by
\[
\mathbb{E}[Y^n \cdot \mathbb{I}(l \leq Y \leq u)] = \int_l^u y^n f_Y(y)dy = J \cdot \int_l^u x^n f_X(x)dx
\]
(A.3)

The result of (A.3) makes it possible to calculate the $n$-th moment of the normal random variable $Y$ on integration interval $[l, u]$ by giving the $n$-th moment of the truncated normal random variable $X$. Since the mean and variance of $X$ have been given by Johnson, Kotz and Balakrishnan (1994), we can obtain

(A.4) \[ \mathbb{E}[1 \cdot \mathbb{I}(l \leq Y \leq u)] = \Phi\left(\frac{u-a}{s}\right) - \Phi\left(\frac{l-a}{s}\right), \]
\[ \mathbb{E}[Y \cdot \mathbb{I}(l \leq Y \leq u)] = J \cdot \mathbb{E}(X) \]
(A.5)
\[ = a \left[ \Phi\left(\frac{u-a}{s}\right) - \Phi\left(\frac{l-a}{s}\right) \right] - s \left[ \phi\left(\frac{u-a}{s}\right) - \phi\left(\frac{l-a}{s}\right) \right], \]
\[ \mathbb{E}[Y^2 \cdot \mathbb{I}(l \leq Y \leq u)] = J \cdot \left[ \mathbb{E}(X) + \mathbb{E}^2(X) \right] \]
\[ = (s^2 + a^2) \left[ \Phi\left(\frac{u-a}{s}\right) - \Phi\left(\frac{l-a}{s}\right) \right] - (su + sa)\phi\left(\frac{u-a}{s}\right) \]
\[ + (sl + sa)\phi\left(\frac{l-a}{s}\right). \]
(A.6)

Let $(l, u, a, b) = (z_k - |e_{k,n}^* - z_k|, z_k + |e_{k,n}^* - z_k|, \hat{e}(x), \hat{e}_n(x))$ and $(l, u, a, b) = (-\infty, v_{k,n}^*, \hat{v}(x), \hat{v}_n(x))$, we can use (A.4)-(A.6) to obtain
\[
\tilde{I}_{e,k}(x|\lambda) = \left( \hat{\delta}_{e,k}(x) - \hat{\delta}_{e,k}(x) \right) \left[ \Phi(h_{e,k}(x)) - \Phi(l_{e,k}(x)) \right] + \hat{\delta}_{e,k}^2(x) \left[ \Phi(h_{e,k}(x)) - \Phi(l_{e,k}(x)) \right],
\]
\[ \tilde{I}_{v,k}(x|\lambda) = \exp(v_{k,n}^* - \hat{v}(x)) \Phi\left(m_{v,k}(x) - \hat{v}_n(x) \phi\left( f_{v,k}(x) \right) \right), \]
where
\[
\hat{\delta}_{e,k}(x) = (e_{k,n}^* - z_k)^2 - (\hat{e}(x) - z_k)^2, \\
h_{e,k}(x) = \frac{z_k - \hat{e}(x) + |z_k - e_{k,n}^*|}{\hat{\delta}_e(x)}, \\
l_{e,k}(x) = \frac{z_k - \hat{e}(x) - |z_k - e_{k,n}^*|}{\hat{\delta}_e(x)}, \\
m_{v,k}(x) = \frac{v_{k,n}^* - \hat{v}(x)}{\hat{\delta}_v(x)}, \\
f_{v,k}(x) = \frac{m_{v,k}(x) - \hat{v}_n(x)}{\hat{\delta}_v(x)}, \\
g_v(x) = \exp\left\{ \hat{v}(x) + \frac{\hat{\delta}^2_v(x)}{2} \right\}. \]

Appendix B. Proof of Theorem 1. Proof. Suppose $X$ contains the first $n$ design points chosen by the $BO_{syn}$ algorithm, then the next design point $x_{n+1}$ satisfies (3.8) for large $n$. 


(rather than being an pilot design point in initialization). Now, we prove the theorem by two steps:

1. **we prove**

   \[ \lim_{n \to \infty} \tilde{I}(x_{n+1}|X) = 0; \]  \hspace{1cm}

2. **we show that (A.7) is equivalent to**

   \[ \lim_{n \to \infty} L_n(BO_{syn}) = 0. \]

   **Step 1:** Suppose \( y(x) \in \mathcal{H}_K(\Omega), \|y\| \leq R_y \), where \( R_y > 0 \) is a constant. Based on a pre-given set of \( n \) design points \( X \subset \Omega \), fitting a GP model to obtain the posterior mean \( \hat{y}(x) \) and the posterior variance \( \hat{s}_y^2(x) \) of \( y(x) \). Given any sequence \( \{x_n\} \subset \Omega \), Bull (2011) has shown that the posterior variance \( \hat{s}_y^2(x) \) and the prediction error \( |y(x) - \hat{y}(x)| \) satisfy

   \[ \lim_{n \to \infty} \hat{s}_y^2(x_{n+1}) = 0 \quad \text{and} \quad \left| \frac{y(x_{n+1}) - \hat{y}(x_{n+1})}{\hat{s}_y(x_{n+1})} \right| \leq U_y, \]

   where \( U_y^2 = R_y^2 \prod_{j=1}^d (\phi_j^u/\phi_j^v) \).

   Specifying \( \{x_n\} \) as the sequence found by the \( BO_{syn} \) algorithm, we apply the above results to \( e(\cdot) \) and \( v(\cdot) \) and obtain

   \[ \lim_{n \to \infty} \hat{s}_e^2(x_{n+1}) = 0, \quad \lim_{n \to \infty} \hat{s}_v^2(x_{n+1}) = 0, \]

   \[ \left| \frac{e(x_{n+1}) - \hat{e}(x_{n+1})}{\hat{s}_e(x_{n+1})} \right| \leq U_e, \quad \left| \frac{v(x_{n+1}) - \hat{v}(x_{n+1})}{\hat{s}_v(x_{n+1})} \right| \leq U_v, \]

   where \( U_e^2 = R_e^2 \prod_{j=1}^d (\phi_j^e/\phi_j^v) \) and \( U_v^2 = R_v^2 \prod_{j=1}^d (\phi_j^u/\phi_j^v) \). (A.8) and (A.9) also imply that

   \[ \lim_{n \to \infty} \hat{e}(x_{n+1}) = e(x_{n+1}), \quad \lim_{n \to \infty} \hat{v}(x_{n+1}) = v(x_{n+1}). \]

   With \( e(\cdot) \) and \( v(\cdot) \) bounded, it is easy to prove that \( d_k(\cdot) \) is also bounded:

   \[ ||d_k(x)|| = (e(x) - z_k)^2 + \exp(v(x)) \leq (R_e + |z_k|)^2 + \exp(R_v) = R_{d_k}, \quad \forall x \in \Omega, k = 1, \ldots, K. \]

   Since the sequence \( \{d_k(x_{k,n})\} \) is monotone decreasing and bounded for each \( k \), it has a limit. The Cauchy sequence satisfies \( \lim_{n \to \infty} d_k(x_{k,n}) - d_k(x_{k,n+1}) = 0 \). By decomposing \( d_k(\cdot) \) into two parts including \( e \) and \( v \), we have

   \[ \lim_{n \to \infty} (z_k - e_{k,n}^*)^2 \quad \lim_{n \to \infty} \exp(v_{k,n}) - \exp(v_{k,n+1}) = 0, \]

   where \( e_{k,n}^* = e(x_{k,n}) \) and \( v_{k,n} = v(x_{k,n}) \).

   For any \( k \in \mathcal{L}_p^r \),

   \[ \tilde{I}_k(x|\mathcal{X}) = \tilde{I}_{e,k}(x|\mathcal{X}) + \tilde{I}_{v,k}(x|\mathcal{X}). \]

   Let

   \[ t_{e,k}(x) = \frac{z_k - e(x)}{\hat{s}_e(x)}; \quad \hat{t}_{e,k}(x) = \frac{z_k - \hat{e}(x)}{\hat{s}_e(x)}; \quad t_{e,k}^*(x) = \frac{z_k - e_{k,n}^*}{\hat{s}_e(x)}; \quad \hat{t}_{e,k}^*(x) = \frac{\hat{s}_e(x)}{\hat{s}_e(x)}; \quad u_{e,k}(x) = |\hat{t}_{e,k}(x)| - |t_{e,k}(x)|, \]

   \[ m_{e,k}(x) = |t_{e,k}^*(x)| - |\hat{t}_{e,k}^*(x)|, \quad u_{e,k}(x) = \frac{v_{k,n} - v(x)}{\hat{s}_v(x)}; \quad m_{v,k}(x) = \frac{v_{k,n} - \hat{v}(x)}{\hat{s}_v(x)}. \]
We observe that \( m_{e,k}(x) \leq |l_{e,k}^*(x)| \). Using (A.9), we have the following inequalities:

\[
|m_{e,k}(x) - u_{e,k}(x)| = |l_{e,k}(x) - |\hat{l}_{e,k}(x)|| \leq \frac{e(x) - \hat{e}(x)}{\hat{s}_e(x)} \leq U_e, \quad m_{e,k}(x) - u_{e,k}(x) = \left[ \frac{\nu(x) - \hat{\nu}(x)}{\hat{s}_\nu(x)} \right] \leq U_\nu.
\]

Further we have

(A.12) \( m_{e,k}(x) \in [u_{e,k}(x) - U_e, u_{e,k}(x) + U_e], \) \( m_{e,k}(x) \in [u_{e,k}(x) - U_e, u_{e,k}(x) + U_e]. \)

Denoting \( \tilde{I}_{e,k}(x|X) \) as a function of \( m_{e,k}(x) \), \( \tilde{I}_{e,k}(m_{e,k}(x)) = \tilde{I}_{e,k}(x|X) \), and \( F_{e,k}^{(1)}(m_{e,k}(x)) \) as the derivative of \( \tilde{I}_{e,k}(m_{e,k}(x)) \) with respect to \( m_{e,k}(x) \), then

\[
F_{e,k}^{(1)}(m_{e,k}(x)) = 2\hat{s}_e^2(x) \left[ \left| t_{e,k}^*(x) - m_{e,k}(x) \right| \left( \Phi(h_{e,k}(x)) - \Phi(l_{e,k}(x)) \right) \right] - 2\hat{s}_e^2(x) \left[ \phi(h_{e,k}(x)) - \phi(l_{e,k}(x)) \right],
\]

where \( h_{e,k}(x) = m_{e,k}(x), l_{e,k}(x) = m_{e,k}(x) - 2|t_{e,k}^*(x)| \) when \( \tilde{l}_{e,k}(x) \geq 0 \) and \( h_{e,k}(x) = 2|t_{e,k}^*(x)| - m_{e,k}(x), l_{e,k}(x) = -m_{e,k}(x) \) otherwise. For \( m_{e,k}(x) \leq |t_{e,k}^*(x)|, F_{e,k}^{(1)}(m_{e,k}(x)) \geq 0, \) and \( \tilde{I}_{e,k} \) is non-decreasing in \( m_{e,k}(x). \)

By the definition of the acquisition function and the BO synergic strategy, the next design point \( x_{n+1} \) will improve on the current prediction model. Combining (A.10) and (A.11), we have

\[
0 \leq \lim_{n \to \infty} \hat{s}_e^2(x_{n+1}) u_{e,k}^2(x_{n+1}) = \lim_{n \to \infty} \left[ |z_k - e_{k,n}^*| - |z_k - e(x_{n+1})| \right]^2 \leq \lim_{n \to \infty} \left[ |z_k - e_{k,n}^*| - |z_k - \hat{e}(x_{n+1})| + |e(x_{n+1}) - \hat{e}(x_{n+1})| \right] |z_k - e_{k,n}^*| + |z_k - e(x_{n+1})| = 0.
\]

Meanwhile notice that \( 0 \leq \Phi(\cdot) \leq 1, 0 \leq \phi(\cdot) \leq 0.5, \) and \( |\hat{s}_e(x)m_{e,k}(x)| \leq 2R_\epsilon < \infty, \) for any \( x \in \Omega. \) Using the fact \( \hat{s}_e(x_{n+1}) \) converges to 0, there exists a time \( N_1, \) for any \( n \geq N_1, \) such that \( u_{e,k}(x_{n+1}) + U_e \leq t_{e,k}^*(x_{n+1}). \) As \( \tilde{I}_{e,k} \) is the expectation of a non-negative quantity, \( \tilde{I}_{e,k} \geq 0. \)

Combining these bounds and the non-decreasing property of \( \tilde{I}_{e,k}, \) we obtain

\[
0 \leq \tilde{I}_{e,k}(m_{e,k}(x_{n+1})) \leq \tilde{I}_{e,k}(u_{e,k}(x_{n+1}) + U_e) \leq \hat{s}_e^2(x_{n+1}) \left[ \left( u_{e,k}(x_{n+1}) + U_e \right) \left( u_{e,k}(x_{n+1}) + U_e + 2m_{e,k}(x_{n+1}) \right) + 1 \right] + \hat{s}_e^2(x_{n+1}) \left[ \left( u_{e,k}(x_{n+1}) + U_e \right) \left( u_{e,k}(x_{n+1}) + U_e - 2t_{e,k}^*(x_{n+1}) \right) \right] = \hat{I}_{e,k}^u \to 0, \text{ as } n \to \infty.
\]

Denoting \( \tilde{I}_{v,k}(x|X) \) as a function of \( m_{v,k}(x) \), \( \tilde{I}_{v,k}(m_{v,k}(x)) = \tilde{I}_{v,k}(x|X) \), and let \( F_{v,k}(m_{v,k}(x)) = \tilde{I}_{v,k}(m_{v,k}(x))/\exp(v^*_n). \) Then the derivative of \( F_{v,k}(m_{v,k}(x)) \) with respect to \( m_{v,k}(x) \) is

\[
F_{v,k}^{(1)}(m_{v,k}(x)) = \frac{\partial F_{v,k}(m_{v,k}(x))}{\partial m_{v,k}(x)} = \tilde{s}_v(x) \exp \left\{ -\tilde{s}_v(x)m_{v,k}(x) + \frac{\tilde{s}_v^2(x)}{2} \right\} \Phi \left( m_{v,k}(x) - \tilde{s}_v(x) \right) \geq 0,
\]

so \( \tilde{I}_{v,k}(m_{v,k}(x)) \) is non-decreasing with respect to \( m_{v,k}(x). \) We have

\[
F_{v,k} \left( \frac{\tilde{s}_v(x)}{2} \right) = \Phi \left( \frac{\tilde{s}_v(x)}{2} \right) - \Phi \left( -\frac{\tilde{s}_v(x)}{2} \right),
\]

(A.14) \( F_{v,k}^{(1)}(m_{v,k}(x)) \leq \tilde{s}_v(x), \) when \( m_{v,k}(x) \geq \frac{\tilde{s}_v(x)}{2}. \)
Using the Taylor expansion of $\Phi(x)$, we have

$$\frac{1}{\sqrt{2\pi}}(x - \frac{1}{6}x^3) + \frac{1}{2} \leq \Phi(x) \leq \frac{1}{\sqrt{2\pi}}x + \frac{1}{2}, \text{ for } x \geq 0.$$  

Using (A.10), we have

$$\lim_{n \to \infty} \frac{\hat{s}_v(x_{n+1})u_{v,k}(x_{n+1})}{\hat{s}_v(x_{n+1})} = \lim_{n \to \infty} \frac{v^*_n}{v(x_{n+1})} - v(x_{n+1})$$

$$\leq \lim_{n \to \infty} \frac{v^*_n}{v^*_n} - v^*_n = 0.$$  

With (A.8), there exists a time $N_2$, such that $m_{v,k}(x_{n+1}) \geq \frac{\hat{s}_v(x_{n+1})}{2}$ for any $n \geq N_2$. Since $\tilde{I}_{v,k}$ is the expectation of a non-negative quantity, $\tilde{I}_{v,k} \geq 0$. Using (A.12), (A.14) and (A.15), we obtain

$$0 \leq \tilde{I}_{v,k}(m_{v,k}(x_{n+1})) \leq \exp(v^*_n) \left[ F_{v,k} \left( \frac{\hat{s}_v(x_{n+1})}{2} \right) + \hat{s}_v(x_{n+1}) \left( m_{v,k}(x_{n+1}) - \frac{\hat{s}_v(x_{n+1})}{2} \right) \right]$$

$$= \exp(v^*_n) \left[ \Phi \left( \frac{\hat{s}_v(x_{n+1})}{2} \right) - \Phi \left( -\frac{\hat{s}_v(x_{n+1})}{2} \right) + \hat{s}_v(x_{n+1}) \left( m_{v,k}(x_{n+1}) - \frac{\hat{s}_v(x_{n+1})}{2} \right) \right]$$

$$\leq \exp(v^*_n) \left[ \frac{\hat{s}_v(x_{n+1})}{\sqrt{2\pi}} + \hat{s}_v(x_{n+1})U_v + \hat{s}_v(x_{n+1})u_{v,k}(x_{n+1}) - \frac{s^2_v(x_{n+1})}{2} \right]$$

$$\leq \exp(v^*_n) \left[ \frac{\hat{s}_v(x_{n+1})}{\sqrt{2\pi}} + \hat{s}_v(x_{n+1})U_v + v^*_n - v^*_n - \frac{s^2_v(x_{n+1})}{2} \right] = \tilde{I}_{v,k} \to 0,$$

as $n \to \infty$.

Combining (A.13) and (A.16), we have

$$0 \leq \tilde{I}_k(x_{n+1} | \mathcal{X}) = \tilde{I}_{e,k}(x_{n+1} | \mathcal{X}) + \tilde{I}_{v,k}(x_{n+1} | \mathcal{X}) \leq \tilde{I}^u - \tilde{I}^u_{v,k} \to 0, \text{ as } n \to 0, \text{ for any } k \in \mathcal{L}_p.$$  

So we obtain

$$\lim_{n \to \infty} \tilde{I}(x_{n+1} | \mathcal{X}) = \lim_{n \to \infty} \frac{1}{|\mathcal{L}_p|} \sum_{k \in \mathcal{L}_p} \tilde{I}_k(x_{n+1} | \mathcal{X}) = 0.$$  

**Step 2:**

1. For any $k \in \mathcal{L}_p$, if $x^*_k \in \mathcal{X}$, obviously we have $\lim_{n \to \infty} d_k(x^*_k) = 0$ using the fact that $x^*_k = X^*_k$.

2. If there exists $k_0 \in \mathcal{L}_p$ such that $x^*_k \notin \mathcal{X}$, we will prove $\lim_{n \to \infty} d_{k_0}(x^*_k) = 0$ by contradiction. Since $x^*_k$ has not been explored by $BO_{syn}$, it should at least make an improvement on $\tilde{I}_{k_0}(x^*_k | \mathcal{X})$, thus we have $\tilde{I}_{k_0}(x^*_k | \mathcal{X}) > 0$. By the definition of $x_{n+1}$ in $BO_{syn}$,

$$\tilde{I}(x_{n+1} | \mathcal{X}) = \max_{x \in \Omega} \tilde{I}(x | \mathcal{X}) \geq \tilde{I}(x^*_{k_0} | \mathcal{X}) \geq \frac{1}{|\mathcal{L}_p|} \tilde{I}_{k_0}(x^*_{k_0} | \mathcal{X}) > 0,$$

and

$$\lim_{n \to \infty} \tilde{I}(x_{n+1} | \mathcal{X}) > 0$$

which contradicts (A.17). So we have $\lim_{n \to \infty} d_k(x^*_k) = 0$ for any $k \in \mathcal{L}_p$.

Let $\mathcal{L}_p = \{1, \ldots, K\}$, finally we obtain

$$\lim_{n \to \infty} L_n(BO_{syn}) = \sum_{k=1}^{K} \left( \lim_{n \to \infty} d_k(x^*_k) - d_k(x^*_k) \right) = 0.$$
Appendix C. Detailed calculations of $I_{d,k}(x|\mathcal{X})$ and $I_{d,k}(x,x'|\mathcal{X})$. For each target response $z_\ell$, we model $d_k(x)$ directly with one GP. Let $\tilde{d}_k(x|\mathcal{X})$ denote the conditional random variable of $d_k(x)$ given $\{d_k(x), x \in \mathcal{X}\}$ with mean $\mu_{d_k}(x)$ and variance $\hat{s}^2_{d_k}(x)$, and $F_{d_k,x}^X$ denote the cumulative distribution distribution of $d_k(x|\mathcal{X})$. Based on (2.8), $\tilde{d}_k(x|\mathcal{X}) \sim N(\mu_{d_k}(x), \hat{s}^2_{d_k}(x))$. Note that the surrogate model $\tilde{d}_k(x|\mathcal{X})$ is obtained by using one target-specific GP model for $d_k(x)$, which is different from the surrogate model $d_k(x|\mathcal{X})$ in (2.10) obtained by using two target-free GP models for $e(x)$ and $v(x)$ individually. Similar to the classic BO procedure [Mockus (1975); Jones, Schonlau and Welch (1998)], we define $\bar{I}_{d,k}(x|\mathcal{X})$, the acquisition function of $d_k(x)$ as follows

$$I_{d,k}(x|\mathcal{X}) = \mathbb{E}\left[\max\{d^\ast_{k,n} - \tilde{d}_k(x|\mathcal{X}), 0\}\right]$$

\[= \int_{\mathcal{A}_{d_k}} d^\ast_{k,n} \cdot dF_{d_k,x}^X(t) - \int_{\mathcal{A}_{d_k}} t \cdot dF_{d_k,x}^X(t)\]

\[= \left(d^\ast_{k,n} - \mu_{d_k}(x)\right) \Phi \left(\frac{d^\ast_{k,n} - \mu_{d_k}(x)}{\hat{s}_{d_k}(x)}\right) + \hat{s}_{d_k}(x) \Phi \left(\frac{d^\ast_{k,n} - \mu_{d_k}(x)}{\hat{s}_{d_k}(x)}\right),\]

where

$$\mathcal{A}_{d_k} = \{|\tilde{d}_k(x|\mathcal{X})| \leq d^\ast_{k,n}, x \in \Omega\} = \{|t| \leq d^\ast_{k,n}\}.$$ 

As for the 2-point acquisition function $I_{d,k}(x,x'|\mathcal{X})$, we directly use the results provided by Ginsbourger, Riche and Carraro (2008) and show its analytic form as follows:

$$I_{d,k}(x,x'|\mathcal{X}) = \mathbb{E}\left[\max\{\max\{d^\ast_{k,n} - \tilde{d}_k(x|\mathcal{X}), 0\}, \max\{d^\ast_{k,n} - \tilde{d}_k(x'|\mathcal{X}), 0\}\}\right]$$

\[= I_{d,k}(x|\mathcal{X}) + I_{d,k}(x'|\mathcal{X}) + B(x,x') + B(x',x),\]

where

$$B(x,x') = \left(\mu_{d_k}(x) - d^\ast_{k,n}\right) \delta(x,x') + \hat{s}_{d_k}(x) \epsilon(x,x'),$$

$$\epsilon(x,x') = \alpha_1 \phi \left(\frac{|\beta_1|}{1 + \alpha_1^2 \gamma_1^2}\right) \Phi \left(\frac{\gamma_1 + \alpha_1 \beta_1}{1 + \alpha_1^2 \gamma_1^2}\right) - \phi(\gamma_1) \Phi(\alpha_1 \gamma_1 + \beta_1),$$

$$\alpha_1 = \frac{\hat{s}_{d_k}(x) - \rho_{1,2} \hat{s}_{d_k}(x')}{\hat{s}_{d_k}(x')(1 - \rho_{1,2}^2)^{\frac{1}{2}}}, \quad \beta_1 = \frac{\mu_{d_k}(x) - \mu_{d_k}(x')}{{\hat{s}_{d_k}(x')}(1 - \rho_{1,2}^2)^\frac{1}{2}}, \quad \gamma_1 = \frac{d^\ast_{k,n} - \mu_{d_k}(x)}{\hat{s}_{d_k}(x)},$$

$\rho_{1,2}$ is the correlation between the random variable $\tilde{d}_k(x|\mathcal{X})$ and $\tilde{d}_k(x'|\mathcal{X})$, and $\Gamma$ stands for the cumulative distribution function of the bi-gaussian random variable $(\tilde{d}_k(x|\mathcal{X}), \tilde{d}_k(x'|\mathcal{X}) - \tilde{d}_k(x|\mathcal{X}))$.

In the $BO_{ind}$ and $BO_{ind-batch}$ algorithm, we can simply replace $X$ with $X_k$ and use the above results to obtain $I_{d,k}(x|\mathcal{X}_k)$ and $I_{d,k}(x,x'|\mathcal{X}_k)$.

Acknowledgement. We thank Mr. Yunshui Zhang of Department of Mathematical Sciences, Tsinghua University and Dr. Xiao Guo of Shenzhen Kuang-Chi Institute of Advanced Technology for insightful discussions in early stage of this study.

The second author was supported by the State Key Laboratory of Meta-RF Electromagnetic Modulation Technology and Guangdong Provincial Key Laboratory of Meta-RF Microwave.

The third author was supported in part by National Natural Science Foundation of China (Grant 11931001 & 11771242, DK PI) and Beijing Academy of Artificial Intelligence (Grant BAAI2019ZD0103, DK PI).
Disclosure statement. No potential conflict of interest was reported by the authors.

REFERENCES


