Collaborative and Distributed Bayesian Optimization via Consensus

Xubo Yue, Yang Liu, Albert S. Berahas, Blake N. Johnson, and Raed Al Kontar

***Abstract*— Optimal design is a critical yet challenging task within many applications. This challenge arises from the need for extensive trial and error, often done through simulations or running field experiments. Fortunately, sequential optimal design, also referred to as Bayesian optimization when using surrogates with a Bayesian flavor, has played a key role in accelerating the design process through efficient sequential sampling strategies. However, a key opportunity exists nowadays. The increased connectivity of edge devices sets forth a new collaborative paradigm for Bayesian optimization. A paradigm whereby different clients collaboratively borrow strength from each other by effectively distributing their experimentation efforts to improve and fast-track their optimal design process. To this end, we bring the notion of consensus to Bayesian optimization, where clients agree (i.e., reach a consensus) on their next-to-sample designs. Our approach provides a generic and flexible framework that can incorporate different collaboration mechanisms. In lieu of this, we propose transitional collaborative mechanisms where clients initially rely more on each other to maneuver through the early stages with scant data, then, at the late stages, focus on their own objectives to get client-specific solutions. Theoretically, we show the sub-linear growth in regret for our proposed framework. Empirically, through simulated datasets and a real-world collaborative sensor design experiment, we show that our framework can effectively accelerate and improve the optimal design process and benefit all participants.**

***Note to Practitioners*—The proposed algorithm allows multiple clients to collaboratively distribute their trial-and-error efforts to fast-track and improve the optimal design process. In the algorithm, each client performs a test locally and then shares the results with an orchestrator. Using the information from all clients, the orchestrator then finds the best new experiment that each client should undertake and sends those back for the next round of experiments. Through this process, all clients can leverage each other's strengths and optimize their designs with far fewer experiments than each client operating in isolation. This is confirmed through many simulation examples, along with a real-life sensor design experiment where multiple collaborating agents seqeuntially coordinate their experimentation efforts. The goal is to rapidly discover the biosensor design and measurement format parameters that find the maximum amount of captured target analyte.**

***Index Terms*— Optimal design, collaborative Bayesian optimization, consensus, sensor design.**

Received 5 November 2024; accepted 7 January 2025. Date of publication 13 January 2025; date of current version 11 April 2025. This article was recommended for publication by Associate Editor X. Zhang and Editor X. Xie upon evaluation of the reviewers' comments. This work was supported in part by NSF CAREER under Award 2144147 and Award CBET-2144310. *(Corresponding author: Raed Al Kontar.)*

Xubo Yue is with the Department of Mechanical and Industrial Engineering, Northeastern University, Boston, MA 02115 USA.

Yang Liu and Blake N. Johnson are with the Grado Department of Industrial and Systems Engineering, Virginia Tech, Blacksburg, VA 24061 USA.

Albert S. Berahas and Raed Al Kontar are with the Department of Industrial & Operations Engineering, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: alkontar@umich.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TASE.2025.3529349, provided by the authors.

Digital Object Identifier 10.1109/TASE.2025.3529349

I. Introduction

THE success of many real-world applications critically depends on trial and error. Often the goal is to manipulate a set of variables, called *designs*, to achieve a desired response. For instance, material scientists perform time-consuming and expensive experiments to determine optimal compositions [1] (i.e., the portion of all ingredients) that produce a material with desired properties. Similarly, additive manufacturers must calibrate many design parameters, including laser power, beam diameter, and hatching pattern through trial & error [2] so that their product matches its intended shape. Those studies, whether through experimentation or simulations like finite element models, consume resources (e.g., time and budget) that can significantly limit progress.

Fortunately, sequential design has played a key role in accelerating the design process [3], [4]. In the engineering and statistical literature, sequential design can be categorized into two types [5]: (i) *sequential design for exploration*, that aims to explore a response surface, (ii) *sequential design for optimization*, that aims to find designs that optimize a target response. Our work focuses on sequential design for optimization, known as sequential optimal design. This field has also been coined more recently as Bayesian optimization (BO) when using surrogates with a Bayesian flavor (e.g., Gaussian processes - $\mathcal{GP}$s). Hereon, we will use the term Bayesian optimization since, without loss of generality, we exploit $\mathcal{GP}$s in the process.

Rather than running exhaustive brute-force experiments over a dense grid, BO employs a sequential strategy to conduct experiments and observe new samples. It first fits a surrogate that estimates the design-response relationship from existing data. Afterward, a utility, also known as acquisition, function [6] is defined to hint at the benefits of sampling/experimenting at new design points. Based on this utility metric, the next-to-sample design point is selected. The procedure is sequentially iterated over several rounds till budget constraints or exit conditions are met. Needless to say, sequential optimal design/BO has been extensively studied [7], [8] and has found success in a wide variety of disciplines across physics, chemistry, mechanical engineering [9], [10], amongst many others.

In lieu of the aforementioned successes, this work aims to bring BO to a collaborative paradigm. The main question we

ask is: **How can multiple clients collaborate to improve and fast-track their design processes**? With today's advances in computation and communication power at edge devices [11], it has become more plausible for potentially dispersed clients to share information, distribute trial and error efforts, and fast-track the design process so that all participants gain benefit. Here, collaborating clients can be scientists, robots, multiple finite element simulations, etc.

However, to enable collaboration, key challenges exist. The first challenge is how to distribute the sequential optimization process. While BO has been extensively studied in the past decades, few literature exists on collaborative BO; see Sec. III for related work. The second challenge is heterogeneity. Despite trying to optimize similar processes, clients may operate under different external factors and conditions. As such, retaining client-specific optimal solutions is of importance. The third challenge is privacy. To encourage participation, a collaborative process should refrain from sharing client-specific outcomes.

In an effort to address the opportunity and stated challenges above, we summarize our contributions below:

- **Bayesian optimization via Consensus:** We bring the notion of "consensus" to BO where clients perform experiments locally and agree (i.e., reach a consensus) on their individualized next-to-sample designs. Our approach provides a generic and flexible framework that can incorporate different collaboration mechanisms. It hinges upon a consensus matrix that evolves with iterations and flexibly determines the next-to-sample designs across all clients.
- **Regret minimization:** We show that, under some mild conditions, with a high probability, the cumulative regret for each client has a sublinear growth rate. This implies that our collaborative algorithm can bring clients within regions of optimal designs.
- **The power of collaboration for optimal sensor design:** We perform an experiment for accelerated optimization of sensor design through collaborative finite element analysis (FEA) workflows. The experiment showcases the ability of collaboration to significantly outperform its non-collaborative counterpart.

II. Motivation

Our work is motivated by optimal sensor design. Biosensors are critical bioanalytical technologies that enable the selective detection of target analytes and have broad applications ranging from medical diagnostics, bioprocess monitoring [12], health monitoring [13], food and water safety, and environmental monitoring [14]. The ability to obtain timely and accurate data enables more precise control over processes, improved patient outcomes, and enhanced safety.

A biosensor is defined as a device that is based on an integrated biorecognition element and transducer [15]. There are two main categories of biosensors: (1) device-based biosensors, and (2) methods-based biosensors (e.g., nanobiosensors) [16]. Device-based biosensors are often integrated with microfluidics and exhibit form factors that can be physically integrated with and removed from processes (e.g., thin-film, dip-stick) and preserve the characteristics of the sample to be analyzed. Alternatively, methods-based biosensors based on solutions and suspensions of functionalized particles provide detection by mixing with a sample, and thus, are relatively destructive with respect to the sample. Thus, device-based biosensors are commonly used in a "flow-and-measure" or "dip-and-measure" format. In contrast, methods-based biosensors, such as nanobiosensors, are typically used in a "mix-and-measure" format.

While biosensors have been created for sensitive and selective detection of many target analytes ranging from small molecules, proteins, nucleic acids, cells, biomarkers, and pathogens, there remains a need to further optimize biosensor design (e.g., form factor, functionalization) and performance (e.g., sensitivity, measurement confidence, and speed) to meet the constraints and requirements of industrial and commercial applications [17]. Current thrust areas in biosensor design and performance optimization can be categorized as driven by experimentation, simulation, data analytics, or combinations thereof [18], [19]. For example, high-throughput experimentation, sensor arrays, FEA, and data-driven biosensing have been leveraged to improve the understanding, design, and performance of biosensors, particularly device-based biosensors whose design, utility, and performance are often linked to the characteristics of experimental measurement formats, such as microfluidic channel design, flow field parameters (e.g., flow rate), and other parameters of the measurement format (e.g., sample injection time). However, it remains a challenge to optimize biosensor performance, particularly in maximizing the amount of target analyte detected, given the high-dimensionality of the design space associated with biosensor design, functionalization (e.g., concentration of immobilized biorecognition elements), and measurement format parameters. In particular, there remains a need to establish closed-loop self-driving workflows for engineering high-performance biosensors, such as by optimizing the biosensor design, functionalization protocols, and measurement formats that synergize with experimentation, simulation, and machine learning. Given the current limitation, our central hypothesis is:

Hypothesis: Collaboration can accelerate the pace of optimal sensor design and yield optimal design with minimal resource expenditures.

Fig. 1 provides a microcosm of our collaborative solution, which we have tested in Sec. VI. Our closed-loop experiment features multiple collaborating agents (FEA simulations) that perform biosensor design. Then they will coordinate to decide on their next simulations. Our goal is to rapidly discover the biosensor design and measurement format parameters that find the maximum amount of captured target analyte.

III. Related Work

A. Historical Context

Optimal design has a rich history, initially focusing on fixed-sample designs, where the best option was chosen from a set. After World War II, adaptive designs emerged to improve

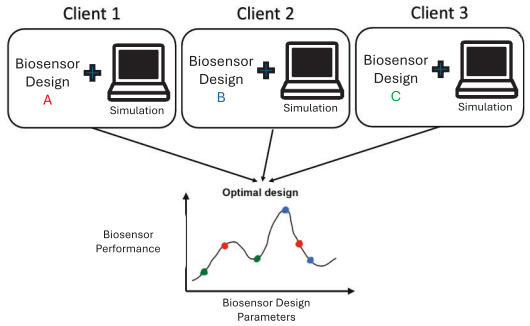

Fig. 1. Illustration of closed-loop biosensor and measurement format design optimization driven by consensus BO-driven simulation.

experimental efficiency. Early milestones include Friedman's sequential designs for military applications [20] and Box's adaptive experiments at Imperial Chemical Industries [21] aimed at enhancing chemical yields. These methods demonstrated that updating designs based on outcomes could greatly improve performance, laying the groundwork for modern BO. Later, Robbins and Bellman [22], [23] formalized sequential decision-making in uncertain environments, followed by Chernoff's [24] work on dynamic adjustments. Seminal contributions by Sacks and Ylvisaker [25] further advanced BO by proposing design selection through utility optimization with surrogate models, ushering in ongoing developments in BO today.

B. Recent Development

Recent advances in BO include but are not limited to: (i) **Alternative surrogate models:** instead of using $\mathcal{GP}$ surrogates, [26], [27] suggest using Deep $\mathcal{GP}$s as an alternative. This credits to many desirable advantages of Deep $\mathcal{GP}$s, such as flexibility to non-stationarity and robustness in handling abrupt regime changes in the training data. Besides deep $\mathcal{GP}$s, Bayesian neural networks [28], [29] have also been widely adopted as surrogates. (ii) **Multi-objective BO:** where the goal is to simultaneously optimize multiple, often competing, responses [30], [31]. Here surrogates that simultaneously learn the multiple outputs, such as the Multi-output $\mathcal{GP}$ [32], are often exploited and developed. (iii) **Multi-fidelity BO:** when data is collected across multiple fidelities, it becomes crucial to choose the fidelity to use when running an experiment. Along this line, [33], [34] have developed various sequential strategies to tackle this challenge. (iv) **New utility functions:** developing new utility functions remains one of the most active areas in optimal design. Along this line, recent literature has investigated look-ahead utilities that chose a design based on its utility over a rolling future horizon [35], [36].

C. Collaborative BO

Though BO has caught major attention over the past century, to our knowledge, little literature exists on collaborative BO. Perhaps the two closest fields are batch sequential design (or batch BO) and federated BO. In batch BO, multiple designs are chosen from a common surrogate and utility. These designs are then distributed across multiple compute nodes to be evaluated in parallel [37], [38], [39]. Unfortunately, batch BO is designed for parallel computing, where it focuses on optimizing a single objective using a single surrogate learned from a centralized dataset. Therefore, it cannot handle cases where data comes from diverse and potentially heterogeneous sources, nor does it preserve privacy. All data needs to be agglomerated in one place to learn the surrogate and optimize the utility. In a similar fashion, federated BO [40], [41] tries to distribute experiments decided from a single objective. However, they do so while preserving privacy. For instance, [40] share function realizations of the posterior belief using random Fourier features to learn the common surrogate. A key limitation in the current federated BO (FedBO) literature is its restriction to Thompson Sampling, overlooking the diverse family of utilities that define unique search heuristics to balance exploration and exploitation. Different strategies yield varying performance across tasks; there is no single utility that fits all, but one needs to pick the right utility for their own tasks.

IV. The Collaborative BO Framework

A. Setting the Stage

We start by describing our problem setting and introducing notation. Assume there are $K \geq 2$ clients, and each client has a budget of T experiments across T iterations. Denote by $t \in \{0, 1 \ldots, T-1\}$ the iteration index. Clients can communicate with each other either via a central orchestrator or direct communication (See Sec. IV-B).

Each client $k \in [K] \coloneqq \{1, \ldots K\}$ has an initial dataset $\mathcal{D}_k^{(0)} = \{\boldsymbol{X}_k^{(0)}, \boldsymbol{y}_k^{(0)}\}$ with $N_k^{(0)}$ observations, where $\boldsymbol{X}_k^{(0)} = \{\boldsymbol{x}_{k,1}, \ldots, \boldsymbol{x}_{k,N_k^{(0)}}\}$ is a $D \times N_k^{(0)}$ design matrix that contains the initials designs $\boldsymbol{x}_{k,\cdot} \in \mathbb{R}^D$ and $\boldsymbol{y}_k^{(0)} = (y_{k,1}, \ldots, y_{k,N_k^{(0)}})^\intercal$ is an $N_k^{(0)} \times 1$ vector that contains the corresponding observed responses. The goal of each client is to find a set of client-specific optimal designs

$$\boldsymbol{x}_k^* = \underset{\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^D}{\operatorname{argmax}} f_k(\boldsymbol{x}),$$

where $\mathcal{X}$ is a subset in $\mathbb{R}^D$, and $f_k : \mathbb{R}^D \to \mathbb{R}$ is the true unknown, and continuous, design function each client aims to optimize. Clearly, using first or second-order optimization algorithms is not feasible since f_k is a black-box. To observe $f_k(\cdot)$ at a new design point $\boldsymbol{x}_k^{\text{new}}$ one needs to run an experiment and observe $y_k(\boldsymbol{x}_k^{\text{new}})$, that is possibly a noisy representation of $f_k(\boldsymbol{x}_k^{\text{new}})$; $y_k = f_k + \epsilon_k$, where ϵ_k is an additive noise. Therefore, the goal is to carefully decide on the next-to-sample design $\boldsymbol{x}_k^{\text{new}}$ so that an optimal design is reached with the fewest experiments possible.

To do so, at any time t, BO resorts to a utility function U, $U(y_k(\boldsymbol{x}); \mathcal{D}_k^{(t)}) : \mathbb{R}^D \to \mathbb{R}$, that quantifies the benefits gained if one were to conduct an experiment at a new D-dimensional design, $\boldsymbol{x}_k^{(t)\text{new}}$ [30], [42].

In a non-collaborative environment, each client k chooses the next-to-sample design by maximizing their own utility. However, since the utility is dependent on the response y_k, one cannot calculate the utility except at previously observed

designs. Here BO resorts to a surrogate $\hat{y}_k$ that estimates $\hat{y}_k(\boldsymbol{x})$ for any $\boldsymbol{x} \in \mathbb{R}^D$ using the dataset $\mathcal{D}_k^{(t)}$. Such surrogates are often probabilistic (e.g., $\mathcal{GP}$ [43]) and are capable of providing a predictive distribution $\mathbb{P}_{\hat{y}_k(\boldsymbol{x})|\mathcal{D}_k^{(t)}}$ over $\hat{y}_k(\boldsymbol{x})$. Equipped by the surrogate, client k, now chooses the next-to-sample design by maximizing the expected utility

$$\boldsymbol{x}_k^{(t)\text{new}} = \operatorname*{argmax}_{\boldsymbol{x}} \mathbb{E}_{\mathbb{P}_{\hat{y}_k|\mathcal{D}_k^{(t)}}}\Big[U(\hat{y}_k(\boldsymbol{x}); \mathcal{D}_k^{(t)})\Big]. \tag{1}$$

Hereon, for notation brevity, we write the expected utility in (1) as $\mathbb{E}_{\hat{y}_k|\mathcal{D}_k^{(t)}}\big[U(\hat{y}_k(\boldsymbol{x}))\big]$. It is also worth noting that expectation in (1) with respect to $\hat{y}_k$ is sometimes replaced with $\hat{f}_k$ where $\hat{y}_k = \hat{f}_k + \epsilon_k$, depending on the type of utility function. See utility functions in Sec. IV-D for some examples.

B. Collaborative BO via Consensus

Now in a collaborative framework, given an arbitrary t, we aim to allow clients to collaboratively decide on $\{\boldsymbol{x}_k^{(t)\text{new}}\}_{k=1}^K$. A natural idea is to maximize utility across all clients. This translates to the following problem:

$$\begin{aligned}\max_{\{\boldsymbol{x}_k\}_{k=1}^K} \mathbb{E}_k\Big[\mathbb{E}_{\hat{y}_k|\mathcal{D}_k^{(t)}}\big[U(\hat{y}_k(\boldsymbol{x}_k))\big]\Big] \\ = \max_{\{\boldsymbol{x}_k\}_{k=1}^K} \sum_{k=1}^K p_k\Big[\mathbb{E}_{\hat{y}_k|\mathcal{D}_k^{(t)}}\big[U(\hat{y}_k(\boldsymbol{x}_k))\big]\Big],\end{aligned} \tag{2}$$

where p_k is some weight coefficient for client k with $\sum_{k=1}^K p_k = 1$. Without loss of generality, hereon we assume $p_k = \frac{1}{K}$. As shown, the key difference of (1) from (2) is taking the expectation $\mathbb{E}_k$ over all participants. However, in itself, (2) does not allow entities to borrow strength from each other as it can be fully decoupled across the K entities.

In order to enable collaboration and distribute experiments across the K clients, we bring the notion of "consensus" [44], [45] to BO. In the context of BO, consensus allows clients to agree (reach a consensus) on their individualized next-to-sample designs through a consensus matrix. Specifically, we modify (2) to

$$\begin{aligned}\boldsymbol{x}_k^{(t)} = [\boldsymbol{x}_{\mathcal{C}}^{(t)}]_k = \operatorname*{argmax}_{\boldsymbol{x}_k}\Big[\mathbb{E}_{\hat{y}_k|\mathcal{D}_k^{(t)}}\big[U(\hat{y}_k(\boldsymbol{x}_k))\big]\Big] \\ \text{and} \quad \boldsymbol{x}_k^{(t)\text{new}} = \Big[(\boldsymbol{W}^{(t)} \otimes \boldsymbol{I}_D)\boldsymbol{x}_{\mathcal{C}}^{(t)}\Big]_k,\end{aligned} \tag{3}$$

where $[\cdot]_k$ represents the k^{th} block of a vector, $\boldsymbol{x}_{\mathcal{C}}^{(t)} = [\boldsymbol{x}_1^{(t)\top}, \boldsymbol{x}_2^{(t)\top}, \cdots, \boldsymbol{x}_K^{(t)\top}]^\top$ is concatenation of the designs across all clients, $\boldsymbol{W}^{(t)}$ is a consensus matrix of size $K \times K$,

$$\boldsymbol{W}^{(t)} = \begin{bmatrix} w_{11}^{(t)} & w_{12}^{(t)} & \cdots & w_{1K}^{(t)} \\ w_{21}^{(t)} & w_{22}^{(t)} & \cdots & w_{2K}^{(t)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{K1}^{(t)} & w_{K2}^{(t)} & \cdots & w_{KK}^{(t)} \end{bmatrix},$$

$\boldsymbol{I}_D$ is a $D \times D$ identity matrix, and $\otimes$ denotes the Kronecker product operation, i.e., $\boldsymbol{W}^{(t)} \otimes \boldsymbol{I}_D$ results in a matrix of size $DK \times DK$. The matrix $\boldsymbol{W}^{(t)}$ is a symmetric, doubly stochastic matrix (i.e., $\sum_k w_{kj}^{(t)} = \sum_j w_{kj}^{(t)} = 1$ for $j, k \in [K]$) with non-negative elements.

The new objective (3) has several interesting features. For the sake of compactness, unless necessary, we drop the superscript t in the subsequent discussion. First, the formulation presented in (3) is indeed reminiscent of recent optimization approaches coined as consensus optimization. To see this, notice that a doubly stochastic matrix $\boldsymbol{W}$ has the property that $(\boldsymbol{W} \otimes \boldsymbol{I}_D)\boldsymbol{x}_{\mathcal{C}} = \boldsymbol{x}_{\mathcal{C}}$ if and only if $\boldsymbol{x}_k = \boldsymbol{x}_j$ for all $k, j \in [K]$ [46]. As such, if we enforce $(\boldsymbol{W} \otimes \boldsymbol{I}_D)\boldsymbol{x}_{\mathcal{C}} = \boldsymbol{x}_{\mathcal{C}}$ as a constraint, we can solve

$$\begin{aligned}\max_{\boldsymbol{x}_k} \frac{1}{K}\sum_{k=1}^K\big[\mathbb{E}_{\hat{y}_k|\mathcal{D}_k}\big[U(\hat{y}_k(\boldsymbol{x}_k))\big]\big] \\ \text{subject to} \quad \boldsymbol{x}_k = \boldsymbol{x}_j, \quad \forall k, j \in [K].\end{aligned} \tag{4}$$

The equality constraint (often referred to as the consensus constraint) is imposed to enforce that local copies at every client are equal. That said, our formulation and setting have several distinguishing features that differentiate it from consensus optimization. The differences pertain to the goals of the two approaches and are related to the consensus matrix $\boldsymbol{W}$. In consensus optimization, the goal is for all clients to eventually agree on a common decision variable. However, in our collaborative BO paradigm, we do not want all clients to make the same decisions. Rather, we want clients to borrow strength from each other while at the same time allowing for personalized (per client) solutions. Even when clients are homogeneous, enforcing the constraint will significantly reduce exploration as everyone runs an experiment at the same location. Therefore, we do not explicitly enforce the consensus constraint. Instead, we allow $\boldsymbol{W}$ to play an aggregation role that decides on the next-to-sample designs $\boldsymbol{x}_k^{(t)\text{new}}$ given the current utility maximizers $\boldsymbol{x}_{\mathcal{C}}^{(t)}$. More importantly, $\boldsymbol{W}^{(t)}$ is time-varying and, in the limit, converges to the identity matrix (see Sec. IV-C for more details and examples of the consensus matrices used). This allows clients to borrow strength in the initial stages of the optimization, yet, eventually, make personalized decisions.

Second, the consensus matrix $\boldsymbol{W}$ is doubly stochastic. By the Birkhoff–von Neumann theorem [47], any doubly stochastic matrix can be expressed as a convex combination of multiple permutation matrices. Mathematically, there exists L non-negative scalars $\{\eta_l\}_{l=1}^L$ such that $\sum_{l=1}^L \eta_l = 1$ and $\boldsymbol{W} = \sum_{l=1}^L \eta_l \boldsymbol{P}_l$. For example,

$$\begin{aligned}\boldsymbol{W} = \begin{bmatrix} 0.2 & 0.3 & 0.5 \\ 0.6 & 0.2 & 0.2 \\ 0.2 & 0.5 & 0.3 \end{bmatrix} = 0.2\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} + 0.2\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ + 0.1\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} + 0.5\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},\end{aligned}$$

where each permutation matrix $\boldsymbol{P}_l$ can be viewed as an allocator that assigns one design solution to one client. In this example, we can see that

$$(\boldsymbol{P}_1 \otimes \boldsymbol{I}_D)\boldsymbol{x}_{\mathcal{C}} = \left(\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \otimes \boldsymbol{I}_D\right)\begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_1 \end{bmatrix}.$$

This implies that the first allocator assigns the solution from client 1 to client 2, and so forth. This is an interesting phenomenon that is related to batch BO, where a batch of candidate design points is selected, and then multiple experiments run in parallel, each corresponding to one candidate solution. Now, instead of sticking to one allocation strategy, the consensus approach can be viewed as a weighted average of all possible assignments, where the designed consensus matrix dictates the weights, hence the impact of clients on each other.

Third, over the course of the optimization, the consensus step $(\boldsymbol{W} \otimes \boldsymbol{I}_D)\boldsymbol{x}_{\mathcal{C}}$ naturally yields K solutions to all clients (i.e., one for each) so that clients can explore and exploit the solution space independently and in a distributed manner. For example, suppose $D = 1$ and consider a consensus matrix $\boldsymbol{W} = \begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix}$ and two local solutions $\boldsymbol{x}_1 = 5, \boldsymbol{x}_2 = 7$. The consensus step will yield $\boldsymbol{W}\boldsymbol{x}_{\mathcal{C}} = (5.6, 6.4)^\intercal$ such that client 1 will take solution 5.6, and client 2 will take solution 6.4.

Fourth, from the previous example, it can be seen that the consensus matrix $\boldsymbol{W}$ controls how much one client will affect the design choices for other clients. As a result, the matrix $\boldsymbol{W}$ adds a layer of flexibility in optimizing (3) and allows for heterogeneous clients. More specifically, in Sec. IV-C, we will show two approaches to design $\boldsymbol{W}^{(t)}$ at each iteration t such that $\boldsymbol{W}^{(t)} \to \boldsymbol{I}$. The intuition is as follows. In the early stages, client k may not have enough observations to obtain a high-quality surrogate model and therefore needs to borrow information from other clients. In the late stages, as client k accumulates sufficient data and can construct a high-quality local surrogate model, it will focus more on its local design problem to find client-specific optimal design points.

Finally, the consensus constraint provides a naturally distributed approach to solve (3) sequentially. To see this, recall that each client can run T experiments across T iterations and the initial dataset is $\mathcal{D}_k^{(0)}$. Then a natural algorithm would take the following form: At iteration $t \in \{0, \ldots, T-1\}$, given $\boldsymbol{x}_{\mathcal{C}}^{(t)}$, all clients calculate $\boldsymbol{x}_k^{(t)\text{new}} = [(\boldsymbol{W}^{(t)} \otimes \boldsymbol{I}_D)\boldsymbol{x}_{\mathcal{C}}^{(t)}]_k$. Now, each client k conducts an experiment at $\boldsymbol{x}_k^{(t)\text{new}}$, and augments its dataset with the new observation. Each client then fits a surrogate (e.g., $\mathcal{GP}$) using the current dataset $\mathcal{D}_k^{(t+1)}$ and accordingly constructs the utility function $U(\hat{y}_k(\boldsymbol{x}_k))$. Then the clients maximize their local utility $\max_{\boldsymbol{x}_k} \mathbb{E}_{\hat{y}_k|\mathcal{D}_k^{(t+1)}}\left[U(\hat{y}_k(\boldsymbol{x}_k))\right]$ to obtain a new candidate solution $\boldsymbol{x}_k^{(t+1)}$. Finally, all clients send their solutions to each other or a central orchestrator, and the process repeats.

Algorithm 1 summarizes our Collaborative Bayesian Optimization via Consensus (CBOC) framework. Fig. 2 presents a flowchart illustrating the collaborative BO. A detailed complexity analysis can be found in Appendix 1 (see the Supplementary Material). Overall, compared to standard BO, the computational complexity of CBOC remains nearly identical. Notably, our collaborative framework enjoys some nice theoretical properties (See Sec. IV-E).

It is worthwhile noting that the collaborative process can be done in a centralized or decentralized manner where clients share $\boldsymbol{x}_k^{(t)}$ either with a central orchestrator or with each other directly.

Algorithm 1 CBOC: Collaborative BO via Consensus

Data: T iterations, K number of clients, initial data $\{\mathcal{D}_k^{(0)}\}_{k=1}^K$, initial consensus matrix $\boldsymbol{W}^{(0)}$, initial designs to share $\{\boldsymbol{x}_k^{(0)}\}_{k=1}^K$

for $t = 0, 1, \cdots T-1$ **do**

  for $k = 1, \cdots, K$ **do**

    Consensus: client k computes the consensus point $\boldsymbol{x}_k^{(t)\text{new}} = [(\boldsymbol{W}^{(t)} \otimes \boldsymbol{I}_D)\boldsymbol{x}_{\mathcal{C}}^{(t)}]_k$

    Experiment: client k conducts experiment using $\boldsymbol{x}_k^{(t)\text{new}}$ and observes $y_k(\boldsymbol{x}_k^{(t)\text{new}})$

    Data Augmentation: client k augments dataset by $\mathcal{D}_k^{(t+1)} = \mathcal{D}_k^{(t)} \cup \{\boldsymbol{x}_k^{(t)\text{new}}, y_k(\boldsymbol{x}_k^{(t)\text{new}})\}$

    Surrogate Modeling: client k updates its surrogate model

    Optimization: client k finds their local utility maximizer $\boldsymbol{x}_k^{(t+1)} = \text{argmax}_{\boldsymbol{x}_k} \mathbb{E}_{\hat{y}_k|\mathcal{D}_k^{(t+1)}}\left[U(\hat{y}_k(\boldsymbol{x}_k))\right]$ and shares this candidate sample

  end

  Update $\boldsymbol{W}^{(t)}$ **(Sec. IV-C))**

end

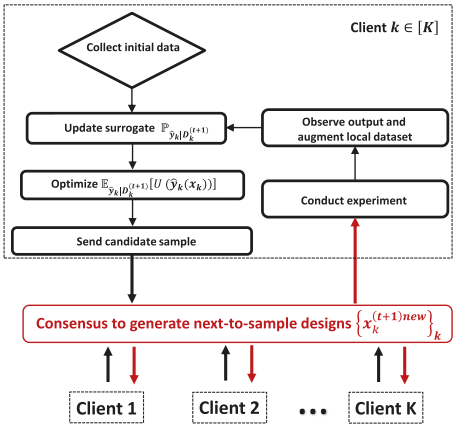

Fig. 2. Flowchart of CBOC.

C. Designing the Consensus Matrix

From Algorithm 1, it can be seen that one of the key components in (3) is the consensus matrix $\boldsymbol{W}$. In essence, as we have discussed earlier, the consensus matrix $\boldsymbol{W}$ controls how much one client will affect the design choices for other clients. Therefore, one needs to carefully design the consensus matrix.

In this section, we propose two approaches for designing the consensus matrix. The first approach assumes one does not have any prior information on different clients and uniformly adjusts all entries in $\boldsymbol{W}^{(t)}$ at each iteration t. The second approach carefully modifies the weights for each client based on the "leader", where the "leader" is defined as the client that has observed the best improvement (e.g., the most

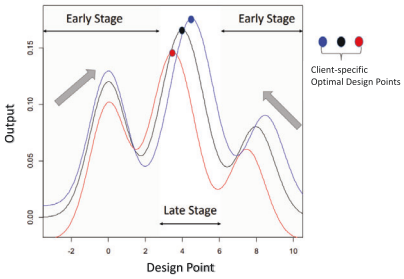

Fig. 3. Illustration of the collaborative process. Colors represent client-specific black-box functions.

significant improvement in the utility). Below, we will detail both approaches.

1) Uniform Transitional Matrix: We propose to assign equal weights to all clients at iteration 0 and then gradually decay off-diagonal elements

$$\boldsymbol{W}^{(0)} = \begin{bmatrix} \frac{1}{K} & \frac{1}{K} & \cdots & \frac{1}{K} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{1}{K} & \frac{1}{K} & \cdots & \frac{1}{K} \end{bmatrix}.$$

For simplicity, we assume all clients are connected. In practice, if clients i and j are not connected, we can set $w_{ij} = w_{ji} = 0$ and reweigh the other components of the $\boldsymbol{W}$ matrix to ensure it is doubly stochastic.

As the optimization progresses, we gradually increase diagonal elements w_{kk} to 1 and decrease the off-diagonal elements w_{kj} for $k \neq j$ to 0. For example, if we adjust weights linearly, then

$$\boldsymbol{W}^{(t+1)} = \boldsymbol{W}^{(t)} + \begin{bmatrix} \frac{K-1}{TK} & -\frac{1}{TK} & \cdots & -\frac{1}{TK} \\ \vdots & \vdots & \vdots & \vdots \\ -\frac{1}{TK} & -\frac{1}{TK} & \cdots & \frac{K-1}{TK} \end{bmatrix}. \quad (5)$$

Ultimately, $\boldsymbol{W}^{(t)}$ will converge to the identity matrix $\boldsymbol{I}$. The intuition behind this design is illustrated in Fig. 3. In this figure, each colored line represents a black-box design function for a client. In the early stages, client k does not have enough observations to obtain a high-quality utility and therefore needs to borrow more information from other clients. As client k has more data in the late stages, it will focus more on its own design problem to find client-specific optimal designs.

Note that $\boldsymbol{W}^{(t)} \to \boldsymbol{I}$, is mandatory if clients have some heterogeneity. To see this, assume all clients have recovered their optimal design $\boldsymbol{x}_k^*$, then if $\boldsymbol{W} \neq \boldsymbol{I}$, the consensus step will always move the experiment away from $\boldsymbol{x}_k^*$.

2) Leader-Driven Matrix: Another approach is to adjust the weights of $\boldsymbol{W}$ dynamically based on the "leader", i.e., the client that observed the best improvement. As a result, all other clients will follow the leader and explore the region that potentially contains the best solutions. Initially, we create two matrices $\boldsymbol{W}_1^{(0)}$ and $\boldsymbol{W}_2^{(0)}$, and assign equal weights to all entries of both matrices. Matrix $\boldsymbol{W}_1$ will be used as a baseline, and is updated using (5). Matrix $\boldsymbol{W}_2$ will be modified based on $\boldsymbol{W}_1$ and by incorporating the leader information. The purpose of using $\boldsymbol{W}_1$ is to ensure that $\boldsymbol{W}_2$ still converges to $\boldsymbol{I}$ so that every single client will eventually focus on their own design objective.

At iteration t, each client k shares a pair $(\boldsymbol{x}_k^{(t)}, S_k^{(t)})$, where S_k is a reward that quantifies the gained benefit at client k. As a simple example, we could set $S_k^{(t)} = \max \mathbb{E}_{\hat{y}_k|\mathcal{D}_k^{(t)}}\left[U(\hat{y}_k(\boldsymbol{x}_k))\right]$. We then sort $S_k^{(t)}$ and find $k^{*(t)} = \text{argmax}_k\{S_1^{(t)}, S_2^{(t)}, \ldots, S_K^{(t)}\}$, and treat client $k^{*(t)}$ as a leader at the current iteration. In essence, client $k^{*(t)}$ can hint to others that the neighborhood of $\boldsymbol{x}_{k^{*(t)}}^{(t)}$ is an area of potential improvement. To do so, during the consensus step, for the non-lead clients $k \neq k^{*(t)}$, we decrease their weights by $\frac{1}{TK}$ and increase their off-diagonal weights with the leader $k^{*(t)}$ by $\frac{K-1}{TK}$. To this end, we adjust blocks of $\boldsymbol{W}_2^{(t)}$ as follows:

$$\begin{aligned}
[\boldsymbol{W}_2^{(t)}]_{\neq k^{*(t)}, k^{*(t)}} &= [\boldsymbol{W}_1^{(t)}]_{\neq k^{*(t)}, k^{*(t)}} + \frac{K-1}{TK}, \\
[\boldsymbol{W}_2^{(t)}]_{k^{*(t)}, \neq k^{*(t)}} &= [\boldsymbol{W}_1^{(t)}]_{k^{*(t)}, \neq k^{*(t)}} + \frac{K-1}{TK}, \\
[\boldsymbol{W}_2^{(t)}]_{\neq k^{*(t)}, \neq k^{*(t)}} &= [\boldsymbol{W}_1^{(t)}]_{\neq k^{*(t)}, \neq k^{*(t)}} - \frac{1}{TK}, \\
[\boldsymbol{W}_2^{(t)}]_{k^{*(t)}, k^{*(t)}} &= [\boldsymbol{W}_1^{(t)}]_{k^{*(t)}, k^{*(t)}} - \frac{(K-1)^2}{TK},
\end{aligned}$$

where $[\boldsymbol{W}_2^{(t)}]_{\neq i,j}$ represents all elements that are in the j^{th} column but not in the i^{th} row of $\boldsymbol{W}_2^{(t)}$. Interestingly, the weight $[\boldsymbol{W}_2^{(t)}]_{k^{*(t)}, k^{*(t)}}$ shrinks to maintain double stochasticity. Such an assignment allows the leader $k^{*(t)}$ to explore new regions.

Here we also suggest two heuristics: (i) To avoid the situation where the same client is selected in succession (this will cause the same client to keep exploring rather than exploiting), we propose to select the second largest index from $\{S_1^{(t+1)}, S_2^{(t+1)}, \ldots, S_K^{(t+1)}\}$ if $k^{*(t)} = k^{*(t+1)}$. (ii) In the extreme case where $[\boldsymbol{W}_2^{(t)}]_{k^{*(t)}, k^{*(t)}}$ is negative, set $[\boldsymbol{W}_2^{(t)}]_{k^{*(t)}, k^{*(t)}}$ to zero and reweigh other components accordingly.

To see an example, consider a scenario where $D = 1$, $K = 3$ and $T = 10$. We initialize both $\boldsymbol{W}_1^{(0)}$ and $\boldsymbol{W}_2^{(0)}$ as

$$\boldsymbol{W}_1^{(0)} = \boldsymbol{W}_2^{(0)} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}.$$

Suppose each client received $(\boldsymbol{x}_1^{(0)}, 1)$, $(\boldsymbol{x}_2^{(0)}, 5)$, $(\boldsymbol{x}_3^{(0)}, 4)$. Then we adjust $\boldsymbol{W}_2^{(0)}$ as

$$\boldsymbol{W}_2^{(0)} = \begin{bmatrix} \frac{1}{3} - \frac{1}{30} & \frac{1}{3} + \frac{2}{30} & \frac{1}{3} - \frac{1}{30} \\ \frac{1}{3} + \frac{2}{30} & \frac{1}{3} - \frac{4}{30} & \frac{1}{3} + \frac{2}{30} \\ \frac{1}{3} - \frac{1}{30} & \frac{1}{3} + \frac{2}{30} & \frac{1}{3} - \frac{1}{30} \end{bmatrix}.$$

In this example, client 2 observed that sampling at $\boldsymbol{x}_2^{(0)}$ yields the highest benefits, and therefore we put more weight on client 2. As a result, clients 1 and 3 will explore towards $\boldsymbol{x}_2^{(0)}$. On the other hand, we decrease $[\boldsymbol{W}_2^{(0)}]_{2,2}$ to make sure that client 2 does not over-explore the region that contains $\boldsymbol{x}_2^{(0)}$.

D. Contextualization Under a Specific Surrogate and Utility

Now, given the generic framework presented in IV-B and IV-C, we will contextualize Algorithm 1 and provide a concrete example. The iteration index t is dropped for simplicity unless necessary.

1) Gaussian Process Surrogate: At each iteration, we place a $\mathcal{GP}$ prior on the surrogate $\hat{f}_k$. We model the additive noise term ϵ_k as independent and identically distributed (*i.i.d.*) noise that follows a normal distribution with zero mean and v_k^2 variance. Now, given a new input location $\boldsymbol{x}^{\text{test}}$, the posterior predictive distribution of $\hat{f}_k(\boldsymbol{x}^{\text{test}}) \sim \mathbb{P}_{\hat{f}_k(\boldsymbol{x}^{\text{test}})|\mathcal{D}_k^{(t)}} := \mathcal{N}(\mu_k(\boldsymbol{x}^{\text{test}}; \mathcal{D}_k), \sigma_k^2(\boldsymbol{x}^{\text{test}}; \mathcal{D}_k))$, where

$$\begin{aligned}\mu_k(\boldsymbol{x}^{\text{test}}; \mathcal{D}_k) &= \boldsymbol{K}(\boldsymbol{x}^{\text{test}}, \boldsymbol{X}_k)\big(\boldsymbol{K}(\boldsymbol{X}_k, \boldsymbol{X}_k) + v_k^2\boldsymbol{I}\big)^{-1}\boldsymbol{y}_k,\\ \sigma_k^2(\boldsymbol{x}^{\text{test}}; \mathcal{D}_k) &= \boldsymbol{K}(\boldsymbol{x}^{\text{test}}, \boldsymbol{x}^{\text{test}})\\ &\quad - \boldsymbol{K}(\boldsymbol{x}^{\text{test}}, \boldsymbol{X}_k)(\boldsymbol{K}(\boldsymbol{X}_k, \boldsymbol{X}_k)\\ &\quad + v_k^2\boldsymbol{I}\big)^{-1}\boldsymbol{K}(\boldsymbol{X}_k, \boldsymbol{x}^{\text{test}}),\end{aligned} \tag{6}$$

and $\boldsymbol{K}(\cdot,\cdot) : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ is a covariance matrix whose entries are determined by some kernel function $\mathcal{K}(\cdot,\cdot)$. Similarly, $\mathbb{P}_{\hat{y}_k|\mathcal{D}_k^{(t)}}$ is derived by simply adding v_k^2 to $\sigma_k^2(\boldsymbol{x}^{\text{test}})$.

2) Utility Function: Given the $\mathcal{GP}$ surrogate, one can build a utility function that measures the benefits of conducting an experiment using a set of new design points. One common example is the expected improvement (EI) utility expressed as [48]

$$\begin{aligned}&\mathbb{E}_{\hat{f}_k|\mathcal{D}_k^{(t)}}\big[U(\hat{f}_k(\boldsymbol{x}))\big]\\ &= \mathbb{E}_{\hat{f}_k|\mathcal{D}_k^{(t)}}\Big[(\hat{f}_k(\boldsymbol{x}) - y_k^{*(t)})^+\Big]\\ &= \mathtt{EI}_k^{(t)}(\boldsymbol{x}) = \sigma_k^{(t)}(\boldsymbol{x}; \mathcal{D}_k^{(t)})\phi(z_k^{(t)}(\boldsymbol{x}))\\ &\quad + (\mu_k^{(t)}(\boldsymbol{x}; \mathcal{D}_k^{(t)}) - y_k^{*(t)})\Phi(z_k^{(t)}(\boldsymbol{x})),\end{aligned}$$

where $a^+ = \max(a, 0)$, $y_k^{*(t)} = \max \boldsymbol{y}_k^{(t)}$ is the current best response, $\phi(\cdot)$ (or $\Phi(\cdot)$) is a probability density function (PDF) (or cumulative distribution function (CDF)) of a standard normal random variable, and $z_k^{(t)}(\boldsymbol{x}) = \frac{\mu_k^{(t)}(\boldsymbol{x};\mathcal{D}_k^{(t)}) - y_k^{*(t)}}{\sigma_k^{(t)}(\boldsymbol{x};\mathcal{D}_k^{(t)})}$. Here note that the expectation is taken with respect to $\hat{f}_k$ rather than $\hat{y}_k$. Another example is the knowledge gradient (KG) [37] defined as $\mathbb{E}_{\hat{y}_k|\mathcal{D}_k^{(t)}}\big[U(\hat{y}_k(\boldsymbol{x}))\big] = \mathtt{KG}_k^{(t)}(\boldsymbol{x}) = \mathbb{E}_{\hat{y}_k|\mathcal{D}_k^{(t)}}\Big[\mu_k^{*(t)}(\boldsymbol{x}; \mathcal{D}_k^{(t)} \cup \{\boldsymbol{x}, \hat{y}_k(\boldsymbol{x})\})\Big] - \max_{\boldsymbol{x}'}(\mu_k^{(t)}(\boldsymbol{x}'; \mathcal{D}_k^{(t)}))$, where $\mu_k^{*(t)}(\boldsymbol{x}; \mathcal{D}_k^{(t)} \cup \{\boldsymbol{x}, \hat{y}_k(\boldsymbol{x})\})$ is the maximum of the updated posterior mean of the $\mathcal{GP}$ surrogate by taking one more sample at location $(\boldsymbol{x}, \hat{y}_k(\boldsymbol{x}))$. KG can be interpreted as finding the new sampling location $\boldsymbol{x}$ that potentially increases the maximum updated posterior mean. Hereon, in the remainder of this paper, we drop $\mathcal{D}_k$ in μ_k, σ_k for the sake of compactness.

E. Theoretical Analysis

Despite its immense success, BO theory is still in its infancy due to many fundamental challenges. First and foremost, the black-box nature of f_k renders theory hard to derive due to the lack of known structure. Second, we still have a limited understanding of the properties, such as Lipschitz continuity, concavity, or smoothness, of many commonly employed utility functions. For example, even the EI utility in general is not Lipschitz continuous or concave. Third, despite recent advances in laying the theoretical foundations for understanding the generalization error bounds of $\mathcal{GP}$s [49], [50], bridging the gap between these bounds and errors incurred in the utility function remains an open problem and a rather challenging one.

To circumvent these open problems while providing a theoretical proof of concept, we derive some theoretical insights that are confined to the EI utility in conjunction with a smooth $\mathcal{GP}$ kernel and a homogeneity assumption. While we believe that our results extend beyond these settings, we leave this analysis as an enticing challenge for future research.

We focus on regret defined as $r_k^{(t)} = f_k(\boldsymbol{x}_k^*) - f_k(\boldsymbol{x}_k^{(t)\text{new}})$ for client k at iteration t. Intuitively, regret measures the gap between the design function evaluated at the optimal solution $\boldsymbol{x}_k^*$ and the one evaluated at the consensus solution $\boldsymbol{x}_k^{(t)\text{new}}$, at iteration t. By definition, $r_k^{(t)} = 0$ if CBOC recovers the global optimal solution. Our theoretical guarantee studies cumulative regret, more specifically, $R_{k,T} = \sum_{t=1}^T r_k^{(t)}$.

By definition, $\mathtt{EI}_k^{(t)}(\boldsymbol{x}) = \mathbb{E}_{\hat{f}_k|\mathcal{D}_k^{(t)}}\Big[(\hat{f}_k(\boldsymbol{x}) - y_k^{*(t)})^+\Big] \geq 0, \forall \boldsymbol{x} \in \mathbb{R}^D$. Therefore, we define a small positive constant κ such that when $\mathtt{EI}_k^{(t)}(\boldsymbol{x}) < \kappa$, client k stops its algorithm at iteration t. This stopping criterion is only used for theoretical development. In practice, we will run our algorithm for T iterations or until all budgets are exhausted.

Below, we present our main Theorem and the sketch of our proof. Detailed information and all supporting Lemmas and required assumptions can be found in Appendices 1-2 (see the Supplementary Material).

Theorem 1: (Homogeneous Clients) Suppose $f_1 = f_2 = \cdots = f_K$, and suppose a squared exponential kernel function $\mathcal{K}_k(\boldsymbol{x}, \boldsymbol{x}') = u_k^2 \exp\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\ell_k^2}\right)$ is used for the $\mathcal{GP}$ surrogate, where u_k is the variance scale parameter and ℓ_k is the length parameter, and each client uses the EI utility. Without loss of generality, assume $\mathcal{K}_k(\boldsymbol{x}, \boldsymbol{x}') \leq 1$ and the initial sample size for each client is 2. Under some assumptions (Appendix 2) (see the Supplementary Material), given any doubly stochastic $\boldsymbol{W}^{(t)}$ with non-negative elements, for $\epsilon > 0, \delta_1 \in (0, T)$, with probability at least $(1-\frac{\delta_1}{T})^T$, the cumulative regret after $T > 1$ iterations is

$$\begin{aligned}R_{k,T} = \sum_{t=1}^T r_k^{(t)} &\leq \sqrt{\frac{6T\big[(\log T)^3 + 1 + C\big](\log T)^{D+1}}{\log(1 + v_k^{-2})}}\\ &\quad + \sqrt{\frac{2T(\log T)^{D+4}}{\log(1 + v_k^{-2})}} + \sum_{t=1}^T \mathcal{O}\left(\frac{1}{(\log(2+t))^{0.5+\epsilon}}\right)\\ &\sim \mathcal{O}(\sqrt{T \times (\log T)^{D+4}}),\end{aligned}$$

where $C = \log[\frac{1}{2\pi\kappa^2}]$.

Theorem 1 shows that the cumulative regret of Algorithm 1 has a sublinear growth rate in terms of the number of iterations. This implies that as the algorithm proceeds with more iterations, the consensus solution for client k will be close to the optimal solution $\boldsymbol{x}_1^* = \cdots = \boldsymbol{x}_k^*$.

Theorem 1 relies on the assumption of homogeneity. When heterogeneity is present, where at least one pair of design functions has some differences, providing convergence results is challenging. To our knowledge, the theory here is vacant. Even if one were to provide a regularity assumption on the heterogeneity, say f_k's are uniformly bounded and have optimal designs restricted within a small ball, it remains an open

problem to understand how this regularity will propagate to the utility where we are able to understand the structural similarities and differences across $\{\texttt{EI}_k\}_{k=1}^K$. Indeed, this stems from the aforementioned second and third challenges in traditional BO. That said, Theorem 1 may serve as a proof of concept, showcasing that our approach will drive clients to a region that contains optimal designs. Afterward, each client will focus on their own objective to obtain a client-specific optimal design (recall Fig. 3). As shown in Sec. IV-C and Sections V-VI, our empirical results highlight the advantageous properties of our approach under heterogeneous settings.

1) Proof Sketch: We hereby provide a proof sketch for Theorem 1. We first decompose the regret $r_k^{(t)}$ as

$$r_k^{(t)} = \Big\{ \underbrace{f_k(\boldsymbol{x}_k^*) - y_k^{*(t)}}_{\text{A}} + \underbrace{y_k^{*(t)} - f_k(\boldsymbol{x}_k^{(t)\text{new}})}_{\text{B}} \Big\}.$$

We let $\text{A} = f_k(\boldsymbol{x}_k^*) - y_k^{*(t)}$ and $\text{B} = y_k^{*(t)} - f_k(\boldsymbol{x}_k^{(t)\text{new}})$. Next, we aim to bound terms A and B.

- Term A defines the difference between the optimal value of the design function and the current best-observed output value. Here, we first use a concentration inequality from [51] to bound the difference between the mean function $\mu_k^{(t)}(\cdot)$ of the $\mathcal{GP}$ surrogate and the truth $f_k(\cdot)$. We then show that, with a high probability, term A is upper bounded by the EI value evaluated at the optimal design $\boldsymbol{x}_k^*$ plus a scaled predictive standard deviation term. Mathematically, with probability $1-\delta$, $\delta \in (0,1)$,

$$f_k(\boldsymbol{x}_k^*) - y_k^{*(t)} \leq \texttt{EI}_k^{(t)}(\boldsymbol{x}_k^*) + \sqrt{\beta_k^{(t)}}\sigma_k^{(t)}(\boldsymbol{x}_k^*),$$

  where $\{\beta_k^{(t)}\}_t$ is a non-decreasing sequence such that $\beta_k^{(t)} \sim \mathcal{O}((\log \frac{t}{\delta})^3)$.
- Term B defines the difference between the current best-observed output value and the underlying true function $f_k(\cdot)$ evaluated at the consensus solution. To proceed, we further expand B as

$$\begin{aligned}\text{B} &= y_k^{*(t)} - f_k(\boldsymbol{x}_k^{(t)\text{new}}) \\ &= y_k^{*(t)} - \mu_k^{(t)}(\boldsymbol{x}_k^{(t)\text{new}}) + \mu_k^{(t)}(\boldsymbol{x}_k^{(t)\text{new}}) - f_k(\boldsymbol{x}_k^{(t)\text{new}}).\end{aligned}$$

  We then bound B using a similar strategy in bounding term A and obtain

$$\begin{aligned}\text{B} \leq \sigma_k^{(t)}(\boldsymbol{x}_k^{(t)\text{new}})\Big(&\tau(-z_k^{(t)}(\boldsymbol{x}_k^{(t)\text{new}})) \\ &- \tau(z_k^{(t)}(\boldsymbol{x}_k^{(t)\text{new}})) + \sqrt{\beta_k^{(t)}}\Big),\end{aligned}$$

  where $\tau(z_k^{(t)}(\boldsymbol{x})) \coloneqq z_k^{(t)}(\boldsymbol{x})\Phi(z_k^{(t)}(\boldsymbol{x})) + \phi(z_k^{(t)}(\boldsymbol{x}))$.

Our next goal is to show that the summation of $r_k^{(t)}$ over T iterations (i.e., the cumulative regret) is bounded. Here, note that there are two key components that appear in A and B: $\sigma_k^{(t)}(\cdot)$ and $\tau(\cdot)$. First, we analyze the behavior of the cumulative predictive variance $\sum_{t=1}^T \sigma_k^{2(t)}(\cdot)$. We show that, with a squared exponential kernel, $\sum_{t=1}^T \sigma_k^{2(t)}(\boldsymbol{x}_k^{(t)\text{new}}) \leq \frac{2}{\log(1+v_k^{-2})}\mathcal{O}((\log T)^{D+1})$. Second, we show that $\tau(-z_k^{(t)}(\boldsymbol{x}_k^{(t)\text{new}})) \leq 1 + \sqrt{C}$, where $C = \log[\frac{1}{2\pi\kappa^2}]$. After some algebraic manipulation, we obtain the upper bound stated in Theorem 1. □

V. Simulation Studies

In this section, we validate CBOC on several simulation datasets. We consider a range of simulation functions from the Virtual Library of Simulation Experiments [52] as the underlying black-box design functions for the clients. Our goal is to showcase the benefit of collaboration in finding client-specific optimal designs.

We set the number of iterations for each testing function to $T = 20D$. At iteration 0, $\boldsymbol{W}^{(0)}$ is initialized as a uniform matrix where each entry has a weight equal to $\frac{1}{K}$. We use the uniform transitional approach (CBOC-U) or the leader-driven approach (CBOC-L) to adjust $\boldsymbol{W}^{(t)}$ at every iteration t. The initial dataset for each client $\{\mathcal{D}_k^{(0)}\}_{k=1}^K$ contains $5D$ randomly chosen design points. The performance of each client is evaluated using a Gap metric G_k [35] defined as

$$G_k = \frac{|y_k^{*(0)} - y_k^{*(T)}|}{|y_k^{*(0)} - y_k^*|},$$

where $y_k^{*(0)}$ (or $y_k^{*(T)}$) is the best observed response at iteration 0 (or T), and $y_k^* = f_k(\boldsymbol{x}_k^*)$ is the true optimal response. A larger G_k implies better performance. If the optimal solution is recovered (i.e., $y_k^{*(T)} = y_k^*$), then $G_k = 1$. We compare CBOC using the EI utility with the following benchmark models: (1) Individual: each client k conducts BO without collaboration; (2) FedBO: the state-of-the-art federated BO algorithm that builds upon Thompson sampling [41]. For both benchmarks, we collect $5D$ initial samples and conduct experiments for $20D$ iterations; (3) CentralBO: the centralized BO setup involves K clients, with each client starting with $5D$ initial points and running for $20D$ iterations. During each iteration, all clients send their local data to a central server, where a global surrogate model is fitted. However, the expected utility is evaluated only on the local data at each client. Our code is available at this GitHub link.

A. Testing Function 1: Levy-D

Levy-D is an D-dimensional function in the form of

$$\begin{aligned}f(\boldsymbol{x}) = &\sin^2(\pi\omega_1) \\ &+ \sum_{d=1}^{D-1}(\omega_d - 1)^2[1 + 10\sin^2(\pi\omega_d + 1)] \\ &+ (\omega_D - 1)^2[1 + \sin^2(2\pi\omega_D)],\end{aligned}$$

where $\omega_d = 1 + \frac{{}^d x - 1}{4}, \forall d \in [D]$ and $\boldsymbol{x} = ({}^1x, \ldots, {}^d x \ldots, {}^D x)^\intercal \in [-10, 10]^D$. We first consider the homogeneous scenario where $f_1(\boldsymbol{x}) = \cdots = f_K(\boldsymbol{x}) = f(\boldsymbol{x})$. We focus on maximizing the design function $-f_k(\boldsymbol{x})$ (i.e., minimizing $f_k(\boldsymbol{x})$), $\forall k \in [K]$. In Table I, we report the average Gap across $K = 5$ clients over 30 independent runs, defined as follows:

$$\bar{G} = \frac{1}{30}\sum_{i=1}^{30}\frac{1}{K}\sum_{k=1}^{K} G_k^{(i)},$$

where $G_k^{(i)}$ is the Gap metric for client k at the i-th run.

TABLE I

THE AVERAGE GAP ACROSS $K = 5$ CLIENTS OVER 30 INDEPENDENT RUNS UNDER A HOMOGENEOUS SETTING. WE REPORT STANDARD DEVIATIONS OVER 30 RUNS IN BRACKETS

Functions	CBOC-L	Individual	FedBO	CentralBO
Levy-2	**0.993(±0.002)**	0.931(±0.004)	0.990(±0.002)	0.982(±0.001)
Levy-4	**0.987(±0.003)**	0.926(±0.007)	0.951(±0.006)	**0.989(±0.003)**
Levy-8	**0.969(±0.010)**	0.925(±0.013)	0.938(±0.009)	**0.973(±0.002)**

TABLE II

THE AVERAGE GAP ACROSS $K = 10$ CLIENTS OVER 30 INDEPENDENT RUNS UNDER A HETEROGENEOUS SETTING

Functions	CBOC-L	Individual	FedBO	CentralBO
Levy-2	**0.990(±0.001)**	0.942(±0.005)	0.958(±0.003)	0.982(±0.001)
Levy-4	**0.984(±0.002)**	0.933(±0.012)	0.940(±0.009)	**0.989(±0.003)**
Levy-8	**0.949(±0.008)**	0.917(±0.008)	0.903(±0.011)	**0.973(±0.002)**

Second, we consider the heterogeneous scenario where each client has a different underlying truth. To do so, for each client k, we transform the Levy function $f(\boldsymbol{x})$ to

$$f_k(\boldsymbol{x}) = a_1 f(\boldsymbol{x} + \text{vec}(a_3)) + a_2\,,$$

where $a_1 \sim \texttt{Uniform}(0.5, 1)$, $a_2, a_3 \sim \mathcal{N}(0, 1)$, and $\text{vec}(a_3)$ is a D-dimensional vector whose elements are all equal to a_3. This transformation will shift and re-scale the original function, creating heterogeneous functional forms. The homogeneous scenario can be viewed as a special case when $a_1 = 1, a_2 = a_3 = 0$. We set $K = 10$. Other settings remain unchanged. Results are shown in Table II.

From Tables I-II, we can derive two key insights. First, collaborative methods yield superior performance than non-collaborative competitors through higher average Gap metrics. This evidences the importance of collaboration in improving the optimal design process. Second, CBOC outperforms all benchmarks. This credits to CBOC's ability to address heterogeneity through a flexible consensus framework that allows clients to collaboratively explore and exploit the design space and eventually obtain client-specific solutions.

B. An Illustrative Example

To visualize the performance of our method, we provide an illustration using the Levy-2 function, $K = 2$, and the same heterogeneity structure as in Sec. V-A. Specifically, the two design functions are set to $f_1(\boldsymbol{x}) = f(\boldsymbol{x} + \text{vec}(1)) + 1$, $f_2(\boldsymbol{x}) = 2f(\boldsymbol{x} + \text{vec}(2)) + 2$. Here, each client starts with five two-dimensional designs, and then both CBOC-L and Individual are run for $T = 40$. Fig. 4 demonstrates the landscape of the original Levy-2 function and shows contour plots of f_1, f_2. The selected design points for each client are marked as red for CBOC-L and green for Individual, and the iteration indices are labeled next to those points. We did not label all points for better visualization.

From Fig. 4, we can see that through collaboration CBOC-L samples more frequently near optimal regions, as evident by the larger number of red points close to the optimal. This allows CBOC-L to reach the region that contains the optimal design (white-most region) for both clients in 25 iterations. On the other hand, Individual required around 38 iterations for client 1, while for client 2, even at $T = 40$, the optimal region was still not explored yet. This again highlights

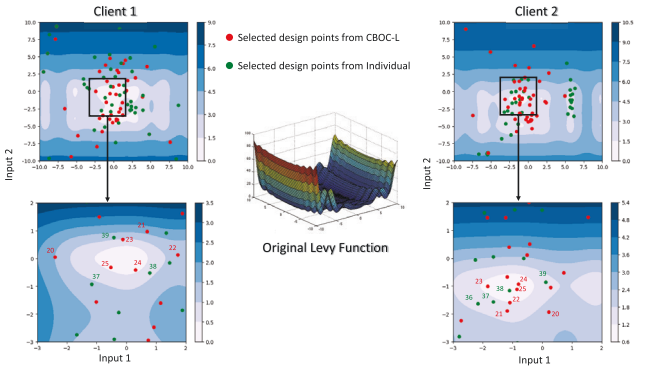

Fig. 4. Contour plot of f_1, f_2 and selected design points. The x-axis shows the first design input (i.e., ${}^1x$), and the y-axis shows the second design input (i.e., ${}^2x$).

TABLE III

THE AVERAGE GAP ACROSS K CLIENTS OVER 30 INDEPENDENT RUNS UNDER A HETEROGENEOUS SETTING

Functions	CBOC-L	CBOC-U	Individual	FedBO
$K = 5$	**0.475(±0.053)**	0.462(±0.052)	0.350(±0.055)	0.370(±0.047)
$K = 10$	**0.516(±0.049)**	0.501(±0.029)	0.364(±0.035)	0.422(±0.040)
$K = 15$	**0.577(±0.033)**	0.553(±0.036)	0.356(±0.022)	0.496(±0.053)
$K = 20$	**0.592(±0.036)**	0.572(±0.015)	0.335(±0.028)	0.535(±0.051)

Functions	CentralBO
$K = 5$	0.346(±0.061)
$K = 10$	0.350(±0.039)
$K = 15$	0.373(±0.042)
$K = 20$	0.347(±0.033)

the benefit of collaboration. *More importantly, Fig. 4 highlights the ability of our method to operate under heterogeneity as both clients were able to reach their distinct optimal design neighborhood, and they do so much faster than operating in isolation.*

C. Testing Function 2: Shekel-10

Shekel-10 is a four-dimensional ($D = 4$) with 10 local minima. It has the functional form:

$$f(\boldsymbol{x}) = -\sum_{i=1}^{10}\left(\sum_{d=1}^{4}({}^dx - F_{di})^2 + \xi_i\right)^{-1}.$$

We defer the specification of F_{di} and ξ_i to Appendix 4 (see the Supplementary Material).

Similar to the heterogeneous scenario in Sec. V-A, for each client k, we transform the Shekel-10 function $f(\boldsymbol{x})$ to $f_k(\boldsymbol{x}) = a_1 f(\boldsymbol{x} + \text{vec}(a_3)) + a_2$, where $a_1 \sim \texttt{Uniform}(0.5, 1)$ and $a_2 \sim \mathcal{N}(0, 2)$, $a_3 \sim \mathcal{N}(0, 1)$. We test all benchmarks using $K = 5, 10, 15$ and 20 clients.

Table III shows that CBOC outperforms both the non-collaborative method and state-of-the-art FedBO. Interestingly, we observe that the performance of CBOC improves as more clients participate in the collaboration process. For example, the average Gap for CBOC when $K = 5$ is 0.475. As we increase K to 20, the average Gap becomes 0.592. Fig. 5 plots the evolution of the average Gap with respect to iterations. This result further demonstrates the benefits of collaboration.

D. Other Testing Functions

Finally, we test our methods on three other functions: Branin, Ackley-D, and Harmann-6. We use $K = 10$.

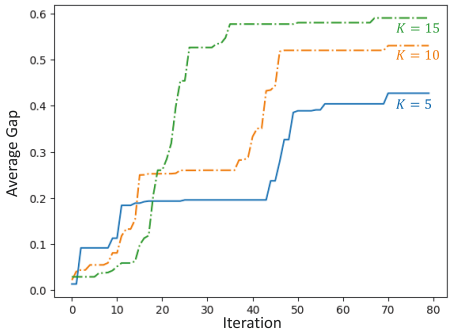

Fig. 5. The evolution of the average Gap with respect to iterations and collaborators.

TABLE IV
THE AVERAGE GAP ACROSS $K = 10$ CLIENTS OVER 30 INDEPENDENT RUNS UNDER A HETEROGENEOUS SETTING

Functions	CBOC-L	CBOC-U	Individual	FedBO
Branin	**0.992(±0.000)**	0.990(±0.001)	0.975(±0.001)	0.986(±0.001)
Ackley-5	**0.656(±0.057)**	0.641(±0.053)	0.501(±0.039)	0.632(±0.041)
Hartmann-6	**0.968(±0.005)**	0.959(±0.002)	0.941(±0.001)	0.955(±0.003)

Functions	CentralBO
Branin	0.967(±0.002)
Ackley-5	0.513(±0.004)
Hartmann-6	0.940(±0.005)

Experimental details and function specifications are deferred to the appendix (see the Supplementary Material).

Similar to our previous analysis, Table IV shows the superiority of our approach.

VI. CASE STUDY

A case study on consensus BO-driven closed-loop biosensor design optimization was performed (see Fig. 6). The closed-loop workflow was based on simulation (FEA) guided by a decentralized computing process for collaboration (i.e., consensus BO). Given the demand for device-based biosensors for industrial process analytical technology (e.g., for bioprocess monitoring and control) and health monitoring (e.g., via wearable sensors), the case study was focused on closed-loop optimization of a device-based biosensor. Specifically, our case study focused on optimizing a device-based biosensor with a milli-scale transducer used in a continuous flow format, which is consistent with several types of device-based biosensors, including electrochemical, mechanical, and electromechanical biosensors.

Biosensor performance, specifically, transient binding of the target analyte, was calculated using commercially available FEA software (COMSOL Multiphysics, COMSOL). The transient fractional surface coverage of the bound target analyte on the biosensor surface was calculated by numerical solution of a coupled convection-diffusion-reaction model in 2D using a time-dependent study. The convection-diffusion-reaction model was constructed by coupling laminar flow and transport of diluted species physics with a surface reaction. The computational domain consisted of a 6×1 mm^2 (length $\times$ width) rectangular fluidic channel that encompassed the domain $x \in [-3, 3]$ mm and $y \in [0, 1]$ mm. The biosensor surface (i.e., surface 1) on which the binding reaction between immobilized biorecognition elements and target analyte occurred encompassed the domain $x \in [0, 1]$ mm at $y = 0$. The computational domain contained five additional surfaces on which boundary and initial conditions were applied. The coupled partial differential equations associated with the convection-diffusion-reaction model were solved subject to the following boundary and initial conditions:

For laminar flow physics:

- Surface 2 (inlet): normal inflow velocity = u_{in}.
- Surface 3 (outlet): static pressure = 0 Pa.
- Surfaces 1 and 4-6 (top and bottom walls, including the sensing surface): no slip.

Here, no slip represents a traditional no slip boundary condition. The fluid properties were defined by water and obtained from the FEA software's material library. The initial velocity and pressure fields were zero.

For transport of diluted species physics:

- Surface 1: $J_{0,c} = -r_{\text{ads}} + r_{\text{des}}$.
- Surface 2: $c = c_0$.
- Surface 3: $J_{0,c} = \boldsymbol{n} \cdot D_i \nabla c_i$.
- Suraces 4-6: no flux.

In the above equations, $J_{0,c}$ is the mass flux, $r_{\text{ads}} = k_{\text{on}} c c_s$ is the adsorption (binding) rate of the target analyte, c is the concentration of the target analyte in solution, c_s is the concentration of immobilized biorecognition element, $r_{\text{des}} = k_{\text{off}} c_s^*$ is the desorption rate, c_s^* is the concentration of occupied sites, $c_0 = c_{\text{in}} \text{GP}(t)$, $c_{\text{in}} = 1$ nM is the concentration of the target analyte in the injected sample, GP(t) is a time-dependent Gaussian pulse function with integral normalization and integral value of unity, and no flux represents a traditional no flux boundary condition. The following values of transport properties and rate constants were used in the simulation: $D_i = 1 \times 10^{-11}$ m^2/s is the diffusivity of the target analyte in solution, the binding (forward) rate constant (k_{on}) $= 1 \times 10^6$ M^{-1}s^{-1}, and the unbinding (reverse) rate constant (k_{off}) $= 1 \times 10^{-3}$ s^{-1}. The initial concentration field was zero.

For surface reaction chemistry:

- Surface 1: reaction rate $= r_{\text{ads}} - r_{\text{des}}$.
- Surfaces 2-6: no reaction.

The following surface properties were used: ρ_s = density of surface sites and site occupancy number = 1. The surface diffusion of the bound target analyte was assumed to be zero. The initial concentration of bound target was zero.

The convection-diffusion-reaction model was discretized and solved subject to the aforementioned boundary and initial conditions using a physics-controlled adaptive mesh that contained 373004 (plus 351428 degrees of freedom) in the final mesh for a given combination of inputs $[x_1, x_2, x_3] = [u_{\text{in}}, \rho_s, \text{GP}_{\text{std}}]$, where GP_{std} is the standard deviation of the Gaussian pulse associated with the injected sample. The specific values of $u_{\text{in}}, \rho_s, \text{GP}_{\text{std}}$ examined were selected by traditional BO and the consensus BO model, which are referred to as "non-collaborative" and "collaborative" learning, respectively.

The simulation's output of interest (i.e., the quantity to be optimized) was the maximum fractional surface coverage of the bound target analyte (θ_{max}) during the transient binding response, which is a fundamental characteristic of biosensor

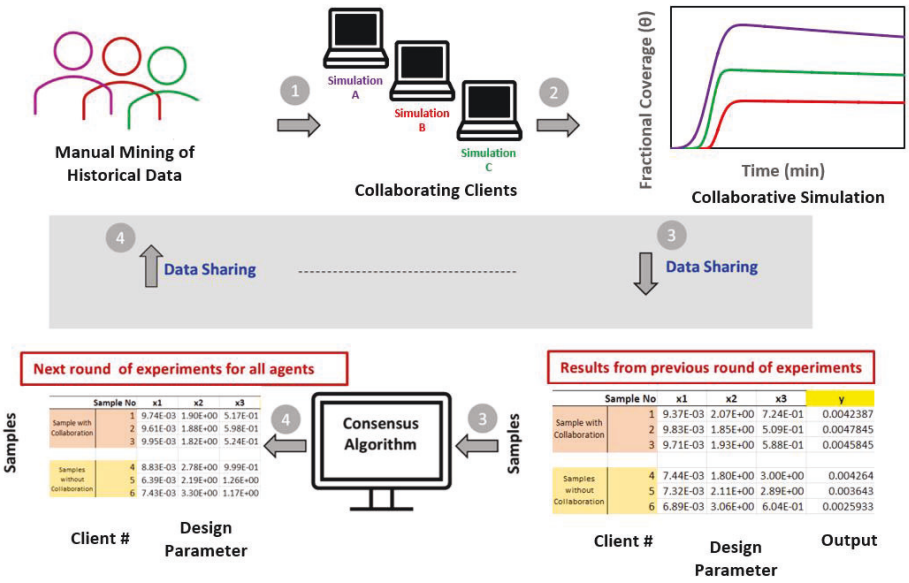

Fig. 6. A case study on consensus BO-driven closed-loop optimization for biosensor design was performed, focusing on optimizing a milli-scale transducer-based device biosensor in a continuous flow format. The case study involved two groups: a non-collaborative group and a collaborative group, each consisting of three clients. Selected design points from collaborative groups were shared to each other to run the CBOC algorithm.

function that is associated with the performance characteristics of sensitivity, detection limit, dynamic range, and speed.

The objective of this case study was to discover (i.e., learn) the biosensor design and measurement format parameters that maximize $\theta_{\max}$ (i.e., the maximum amount of captured target analyte). An optimal solution (i.e., design) was sought within the design space of $x_1 \in [1 \times 10^{-4}, 1 \times 10^{-2}]$ m/s, $x_2 \in [1.8 \times 10^{-8}, 3.3 \times 10^{-8}]$ mol/m^2, and $x_3 \in [0.5, 3]$ min. The range and limits of the design space were selected based on practical values used in previous biosensing studies [53], [54], [55], [56]. In summary, the objective of this case study was to learn the optimal combination of $u_{\text{in}}, \rho_s, \text{GP}_{\text{std}}$ that maximized the amount of captured target analyte and to compare the learning performance achieved by traditional BO performed by a group of independent clients that do not share information with that achieved by collaborative learning performed by a group of clients that share information and make decisions for next-to-test values using consensus BO.

The case study involved two groups: a non-collaborative group and a collaborative group, each consisting of three clients. The clients are distributed across diverse experimental settings to establish a heterogeneous environment. The non-collaborative group used conventional centralized BO algorithms, where each client solves its own problem and there is no communication among clients. In contrast, the collaborative group utilized the CBOC algorithm (Algorithm 1) to collaboratively find optimal solutions. For CBOC, we adopted the leader-driven matrix mentioned in Sec. IV-C. Fig. 7 shows the trend of biosensor performance, specifically maximum amount of captured target species, throughout the iterative closed-loop workflow. As shown in Fig. 7, the self-driving workflow driven by consensus BO was capable of learning an optimized biosensor design and measurement format parameter selection after ten rounds of experimentation as evidenced

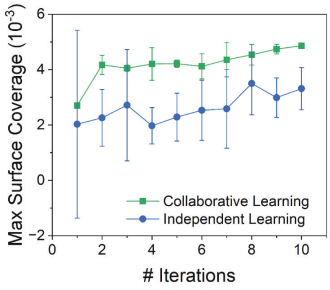

Fig. 7. Trend in the biosensor's maximum fractional surface coverage (i.e., the maximum amount of captured target analyte) vs. iteration number (i.e., sequential rounds of simulation) via BO without and with collaboration after ten iterations.

by the trend of $\theta_{\max}$. Additionally, the average biosensor performance was higher and exhibited lower variance in the collaborative group relative to the non-collaborative group, which highlights the value of leveraging collaboration in closed-loop high-throughput experimentation.

This case study serves as a proof-of-concept showcasing that collaboration in optimal design can reap benefits for the participating entities. We envision that the underlying principles and methodologies demonstrated can extend to different biosensor types and varied applications, though with some domain-specific considerations. Our collaborative framework supports various BO methods, allowing customization for different biosensors, such as electrochemical, optical, and mass-sensitive types, by adjusting calibration parameters to match unique response profiles. It can also be adapted for diverse applications, including environmental monitoring, healthcare diagnostics, and food safety - for instance, detecting pollutants, identifying biomarkers, or finding food contaminants. While additional calibration may be needed for each use case to address sample complexity and analyte specifics,

this flexible framework could significantly accelerate biosensor deployment across these fields.

VII. Discussion and Conclusion

In this paper, we propose a collaborative Bayesian optimization framework built upon a consensus mechanism. Our experiments on simulated data and a real-world sensor design case study show that collaboration through our proposed framework can accelerate and improve the optimal design process. Specifically, `CBOC` achieves up to a 28% (and 35%) improvement in the GAP metric on complex 4-dimensional functions, comparing to the federated BO (and individual non-collaborative BO). In real-world sensor discovery data, `CBOC` enhances the maximum captured target analyte by approximately 50% (see Fig. 7).

Collaborative BO is still in its infancy stage, and few prior studies exist along this line. Indeed, there are many avenues of potential improvement. These avenues include: (i) extending the consensus framework to multi-objective and multi-fidelity settings, (ii) developing a resource-aware approach that allows some clients to perform more or fewer experiments depending on their resources, (iii) understanding the theoretical conditions needed for the collaborative design to outperform its non-collaborative counterpart. If successful, this may lead to targeted algorithms that exploit these conditions to improve performance. We hope this research inspires future work along these avenues and beyond.

References

[1] J. Zhang et al., "Rapid, autonomous high-throughput characterization of hydrogel rheological properties via automated sensing and physics-guided machine learning," *Appl. Mater. Today*, vol. 30, Feb. 2023, Art. no. 101720.

[2] C. Buchanan and L. Gardner, "Metal 3D printing in construction: A review of methods, research, applications, opportunities and challenges," *Eng. Struct.*, vol. 180, pp. 332–348, Feb. 2019.

[3] A. G. Kusne et al., "On-the-fly closed-loop materials discovery via Bayesian active learning," *Nature Commun.*, vol. 11, no. 1, p. 5966, 2020.

[4] C. Huang, Y. Ren, E. K. McGuinness, M. D. Losego, R. P. Lively, and V. R. Joseph, "Bayesian optimization of functional output in inverse problems," *Optim. Eng.*, vol. 22, no. 4, pp. 2553–2574, Dec. 2021.

[5] A. A. Ezzat, A. Pourhabib, and Y. Ding, "Sequential design for functional calibration of computer models," *Technometrics*, vol. 60, no. 3, pp. 286–296, Jul. 2018.

[6] R. B. Gramacy and H. K. H. Lee, "Adaptive design and analysis of supercomputer experiments," *Technometrics*, vol. 51, no. 2, pp. 130–145, May 2009.

[7] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, 2015.

[8] J. P. Kleijnen, *Design and Analysis of Simulation Experiments*. Cham, Switzerland: Springer, 2018.

[9] Y. Zhang, D. W. Apley, and W. Chen, "Bayesian optimization for materials design with mixed quantitative and qualitative variables," *Sci. Rep.*, vol. 10, no. 1, pp. 1–13, 2020.

[10] K. Wang and A. W. Dowling, "Bayesian optimization for chemical products and functional materials," *Current Opinion Chem. Eng.*, vol. 36, Jun. 2022, Art. no. 100728.

[11] R. Kontar et al., "The internet of federated things (IoFT)," *IEEE Access*, vol. 9, pp. 156071–156113, 2021.

[12] L. Rösner, F. Walter, C. Ude, G. John, and S. Beutel, "Sensors and techniques for on-line determination of cell viability in bioprocess monitoring," *Bioengineering*, vol. 9, no. 12, p. 762, Dec. 2022.

[13] L. Song, J. Chen, B. B. Xu, and Y. Huang, "Flexible plasmonic biosensors for healthcare monitoring: Progress and prospects," *ACS Nano*, vol. 15, no. 12, pp. 18822–18847, Dec. 2021.

[14] E. Cesewski and B. N. Johnson, "Electrochemical biosensors for pathogen detection," *Biosensors Bioelectron.*, vol. 159, Jul. 2020, Art. no. 112214.

[15] D. R. Thévenot, K. Toth, R. A. Durst, and G. S. Wilson, "Electrochemical biosensors: Recommended definitions and classification," *Anal. Lett.*, vol. 34, no. 5, pp. 635–659, Mar. 2001.

[16] B. N. Johnson and R. Mutharasan, "Biosensor-based microRNA detection: Techniques, design, performance, and challenges," *Analyst*, vol. 139, no. 7, p. 1576, 2014.

[17] A. C. Carpenter, I. T. Paulsen, and T. C. Williams, "Blueprints for biosensors: Design, limitations, and applications," *Genes*, vol. 9, no. 8, p. 375, Jul. 2018.

[18] J. Zhang et al., "Reduction of biosensor false responses and time delay using dynamic response and theory-guided machine learning," *ACS Sensors*, vol. 8, no. 11, pp. 4079–4090, Nov. 2023.

[19] J. Zhang et al., "Improving biosensor accuracy and speed using dynamic signal change and theory-guided deep learning," *Biosensors Bioelectron.*, vol. 246, Feb. 2024, Art. no. 115829.

[20] M. Friedman and L. J. Savage, "Planning experiments seeking maxima," in *Proc. Techn. Stat. Anal.*, 1947, pp. 365–372.

[21] G. Box and K. Wilson, "On the experimental designs for exploring response surfaces," *Ann. Math. Statist.*, vol. 13, pp. 1–45, Jan. 1951.

[22] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, Sep. 1951.

[23] R. Bellman, "A problem in the sequential design of experiments," *Sankhyā, Indian J. Statist.*, vol. 16, nos. 3–4, pp. 221–229, 1956.

[24] H. Chernoff, "Sequential design of experiments," *Ann. Math. Statist.*, vol. 30, no. 3, pp. 755–770, Sep. 1959.

[25] J. Sacks and D. Ylvisaker, "Statistical designs and integral approximation," in *Proc. 12th Bienn. Semiconductor Can. Math. Congr.*, R. Pyke, Ed., 1970, pp. 115–136.

[26] D. Ming, D. Williamson, and S. Guillas, "Deep Gaussian process emulation using stochastic imputation," *Technometrics*, vol. 65, no. 2, pp. 150–161, Apr. 2023.

[27] A. Sauer, R. B. Gramacy, and D. Higdon, "Active learning for deep Gaussian process surrogates," *Technometrics*, vol. 65, no. 1, pp. 4–18, Jan. 2023.

[28] J. Snoek et al., "Scalable Bayesian optimization using deep neural networks," in *Proc. Int. Conf. Mach. Learn.*, Feb. 2015, pp. 2171–2180.

[29] J. T. Springenberg, A. Klein, S. Falkner, and F. Hutter, "Bayesian optimization with robust Bayesian neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, Dec. 2016, pp. 1–9.

[30] B. Konomi, G. Karagiannis, A. Sarkar, X. Sun, and G. Lin, "Bayesian treed multivariate Gaussian process with adaptive design: Application to a carbon capture unit," *Technometrics*, vol. 56, no. 2, pp. 145–158, Apr. 2014.

[31] D. H. Svendsen, L. Martino, and G. Camps-Valls, "Active emulation of computer codes with Gaussian processes—Application to remote sensing," *Pattern Recognit.*, vol. 100, Apr. 2020, Art. no. 107103.

[32] R. Kontar, S. Zhou, C. Sankavaram, X. Du, and Y. Zhang, "Nonparametric modeling and prognosis of condition monitoring signals using multivariate Gaussian convolution processes," *Technometrics*, vol. 60, no. 4, pp. 484–496, Oct. 2018.

[33] X. He, R. Tuo, and C. F. J. Wu, "Optimization of multi-fidelity computer experiments via the EQIE criterion," *Technometrics*, vol. 59, no. 1, pp. 58–68, Jan. 2017.

[34] R. Stroh, J. Bect, S. Demeyer, N. Fischer, D. Marquis, and E. Vazquez, "Sequential design of multi-fidelity computer experiments: Maximizing the rate of stepwise uncertainty reduction," *Technometrics*, vol. 64, no. 2, pp. 199–209, Apr. 2022.

[35] S. Jiang, H. Chai, J. González, and R. Garnett, "Binoculars for efficient, nonmyopic sequential experimental design," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2019, pp. 4794–4803.

[36] X. Yue and R. A. Kontar, "Why non-myopic Bayesian optimization is promising and how far should we look-ahead? A study via rollout," in *Proc. Int. Conf. Artif. Intell. Statist.*, Nov. 2019, pp. 2808–2818.

[37] J. Wu and P. I. Frazier, "The parallel knowledge gradient method for batch Bayesian optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, Jan. 2016, pp. 1–9.

[38] W. Duan, B. E. Ankenman, S. M. Sanchez, and P. J. Sanchez, "Sliced full factorial-based Latin hypercube designs as a framework for a batch sequential design algorithm," *Technometrics*, vol. 59, no. 1, pp. 11–22, Jan. 2017.

[39] N. Hunt, "Batch Bayesian optimization," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, 2020.

[40] Z. Dai, K. H. Low, and P. Jaillet, "Federated Bayesian optimization via Thompson sampling," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2020, pp. 9687–9699.

[41] Z. Dai, B. K. H. Low, and P. Jaillet, "Differentially private federated Bayesian optimization with distributed exploration," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 9125–9139.

[42] I.-C. Lee, Y. Hong, S.-T. Tseng, and T. Dasgupta, "Sequential Bayesian design for accelerated life tests," *Technometrics*, vol. 60, no. 4, pp. 472–483, Oct. 2018.

[43] C. Currin, T. Mitchell, M. Morris, and D. Ylvisaker, "Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments," *J. Amer. Stat. Assoc.*, vol. 86, no. 416, p. 953, Dec. 1991.

[44] X. Shi, J. Cao, and W. Huang, "Distributed parametric consensus optimization with an application to model predictive consensus problem," *IEEE Trans. Cybern.*, vol. 48, no. 7, pp. 2024–2035, Jul. 2018.

[45] A. S. Berahas, R. Bollapragada, N. S. Keskar, and E. Wei, "Balancing communication and computation in distributed optimization," *IEEE Trans. Autom. Control*, vol. 64, no. 8, pp. 3141–3155, Aug. 2019.

[46] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[47] A. W. Marshall, I. Olkin, and B. C. Arnold, *Inequalities: Theory of Majorization and Its Applications* (Springer Series in Statistics). Midtown Manhattan, NY, USA: Springer, 1979.

[48] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. Global Optim.*, vol. 13, no. 4, p. 455, 1998.

[49] A. Lederer, J. Umlauft, and S. Hirche, "Uniform error bounds for Gaussian process regression with application to safe control," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2019, pp. 1–11.

[50] W. Wang, R. Tuo, and C. F. Jeff Wu, "On prediction properties of kriging: Uniform error bounds and robustness," *J. Amer. Stat. Assoc.*, vol. 115, no. 530, pp. 920–930, Apr. 2020.

[51] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," 2009, *arXiv:0912.3995*.

[52] S. Surjanovic and D. Bingham. *Virtual Library of Simulation Experiments: Test Functions and Datasets*. Accessed: Jan. 13, 2025. [Online]. Available: http://www.sfu.ca/~ssurjano

[53] M. Selmi, M. H. Gazzah, and H. Belmabrouk, "Optimization of microfluidic biosensor efficiency by means of fluid flow engineering," *Sci. Rep.*, vol. 7, no. 1, p. 5721, Jul. 2017.

[54] T. M. Squires, R. J. Messinger, and S. R. Manalis, "Making it stick: Convection, reaction and diffusion in surface-based biosensors," *Nature Biotechnol.*, vol. 26, no. 4, pp. 417–426, Apr. 2008.

[55] R. Baronas, F. Ivanauskas, J. Kulys, R. Baronas, F. Ivanauskas, and J. Kulys, "Biosensors acting in injection mode," in *Mathematical Modeling of Biosensors* (Springer Series in Statistics). Midtown Manhattan, NY, USA: Springer, 2021, pp. 183–205.

[56] B. N. Johnson and R. Mutharasan, "Sample preparation-free, real-time detection of microRNA in human serum using piezoelectric cantilever biosensors at attomole level," *Anal. Chem.*, vol. 84, no. 23, pp. 10426–10436, Dec. 2012.