Collaborative Bayesian Optimization via Wasserstein Barycenters

Donglin Zhan, Haoting Zhang, Rhonda Righter, Zeyu Zheng, and James Anderson

***Abstract*—Motivated by the growing need for black-box optimization and data privacy, we introduce a collaborative Bayesian optimization (BO) framework that addresses both of these challenges. In this framework agents work collaboratively to optimize a function they only have oracle access to. In order to mitigate against communication and privacy constraints, agents are not allowed to share their data but can share their Gaussian process (GP) surrogate models. To enable collaboration under these constraints, we construct a central model to approximate the objective function by leveraging the concept of Wasserstein barycenters of GPs. This central model integrates the shared models without accessing the underlying data. A key aspect of our approach is a collaborative acquisition function that balances exploration and exploitation, allowing for the optimization of decision variables collaboratively in each iteration. We prove that our proposed algorithm is asymptotically consistent and that its implementation via Monte Carlo methods is numerically accurate. Through numerical experiments, we demonstrate that our approach outperforms other baseline collaborative frameworks and is competitive with centralized approaches that do not consider data privacy.**

I. Introduction

In numerous engineering applications, including materials design [1], control engineering [2], robotics [3] and machine learning parameter tuning [4], one must solve optimization problems where either the objective function is costly to evaluate, or, is unknown and only oracle access is available. A widely used method in this setting is Bayesian optimization (BO), which employs a "surrogate model" to approximate the unknown objective function [5]. One of the most widely-used surrogates for BO is the Gaussian process (GP) model [6], which assumes that the objective function is a realization of a GP. In conjunction with the surrogate model, an "acquisition function" uses predictions from the surrogate model to determine the next sample point. The BO algorithm is implemented by iteratively collecting observations, updating the model, and optimizing the acquisition function to select the decision variable to collect another observation.

In various control and machine learning scenarios, multiple agents collaborate to optimize an unknown function. For instance, agents may need to collaboratively fine-tune controller parameters in distributed control architectures, such as multi-agent robotic systems [7] or decentralized energy management systems [8]. Another example would be fine-tuning the hyper-parameters of several different neural network architectures in a parallel computing environment [9]. To facilitate this collaboration, observations at evaluated points are shared to update a joint surrogate model. The acquisition function is designed to select multiple points for parallel evaluation by multiple agents in the next iteration. In many cases, data privacy might be a concern, in which case agents are not allowed to share the data directly when updating the surrogate model. An illustrative example is found in autonomous vehicle platooning, where each vehicle locally optimizes its control parameters based on sensor data that cannot be directly shared due to privacy and security regulations [10]. Similarly, in distributed power grids, individual units optimize local control parameters without exchanging sensitive consumption or operational data directly, adhering strictly to privacy and security constraints [11], [12].

Data privacy concerns have been discussed extensively in the literature on federated learning (FL), c.f., [13], [14]. In FL, instead of sharing the data, each agent learns a model from its own dataset and shares the resulting model. The central model is constructed from the models shared by the agents, aiding in approximating the unknown function and facilitating downstream applications, including optimization. Existing federated learning algorithms primarily rely on parametric models, such as neural networks and kernel regression models [15], [16]. For example, FedAvg is a foundational algorithm that directly averages local updates to achieve a global model [13], and FedProx addresses agent heterogeneity by adding a proximal term to stabilize the learning process across diverse data distributions [14]. In these cases, agents share the learned parameters representing their surrogate models, and the central model is constructed by adaptively averaging these parameters using various methodologies. In contrast, the GP models used in BO algorithms are nonparametric, which poses a challenge for existing methods to construct a central model from the surrogate models shared by agents. In this work, we ask the question: *Is it possible to collaboratively learn in a Bayesian Optimization setting without sharing data?*

Contribution. We propose a framework for collaborative Bayesian Optimization (BO) with data privacy considerations. In this framework, agents approximate the objective function using GP models. Instead of sharing data, the agents share these GP models with a (central) server. The server then constructs a central GP model by leveraging the concept of the Wasserstein barycenter. Specifically, we treat GPs as probability measures and represent the central model as the probability measure that minimizes the squared Wasserstein

Donglin Zhan and James Anderson are with the Department of Electrical Engineering at Columbia University, New York, USA. Emails: {donglin.zhan,james.anderson}@columbia.edu. Haoting Zhang, Rhonda Righter, and Zeyu Zheng are with the Department of Industrial Engineering and Operation Research at the University of California at Berkeley, Berkeley, USA. Emails: {haoting_zhang,rrighter,zyzheng}@berkeley.edu.

distance to the local GP models. This central model integrates the local models without directly sharing data. Furthermore, the central model remains a GP and provides explicit uncertainty quantification. We propose a collaborative acquisition function to select decision variables for agents to collect observations in parallel. The proposed acquisition function not only addresses the exploration-exploitation trade-off during the optimization procedure but also leverages the central model while maintaining differences through consideration of the local models. Our main contributions are as follows:

- A collaborative BO framework with data privacy considerations, where the central model is constructed as the Wasserstein barycenter of GPs and remains a GP. This GP structure allows for explicit uncertainty quantification and facilitates the exploitation-exploration trade-off during the BO implementation.
- A collaborative acquisition function that selects decision variables for agents to collect observations in parallel, focusing on the collaborative knowledge gradient (Co-KG) function and proving the consistency of the framework based on Co-KG. We use a Monte Carlo method to approximate Co-KG, proving the consistency of the approximation.
- Our experimental results show that Co-KG achieves the best performance compared to other collaborative acquisition functions. We provide practical suggestions for selecting the hyperparameters for Co-KG implementation. Results indicate that our approach not only outperforms existing baseline approaches but also achieves comparable performance to BO algorithms where agents can directly share data to update the GP model without data privacy concerns.

A. Related Literature

BO has been widely applied to solve black-box optimization problems, including materials/engineering design [1] and parameter tuning [4]. The BO approach involves a statistical surrogate model, typically a GP, learned from observations at evaluated points, and an acquisition function, constructed from the surrogate, for deciding the next evaluation point. The algorithm iteratively collects observations, updates the model, and optimizes the acquisition function, with common acquisition functions including expected improvement [17], knowledge gradient [18] and upper confidence bound [19]. Despite its popularity, BO approaches have been largely restricted to moderate-dimensional problems due to the computational complexities brought by Gaussian processes (GPs). Thus, large streams of work have focused on high-dimensional BO, where the approximation of the covariance matrix is employed or an additive/sparse structure of the GP is imposed; see [20], [21]. There have also been many recent contributions to multi-objective/task BO [22], [23], budgeted BO [24], and multi-fidelity BO [25]. Additionally, BO within a collaborative framework has also been explored. Work in [26] casts BO in a FL setting, where agents are randomly selected at each iteration, and the central GP model is approximated by a parametric model. [27] considers optimizing a weighted mean of acquisition functions based on multiple GPs to facilitate collaboration. [28] enhances the efficiency of agents with constrained communication graphs via distributed Thompson sampling.

II. Problem Statement & Main Procedure

We consider the black-box optimization problem:

$$\max_{x\in\mathcal{X}} \; f(x),$$

where $x \in \mathbb{R}^d$ is the decision variable, $\mathcal{X}$ is the continuous feasible set, and $f : \mathcal{X} \mapsto \mathbb{R}$ is a black-box objective function. Each evaluation of $f(x)$ is expensive, so the exploitation-exploration trade-off is a concern. In addition, each evaluation includes observation noise, i.e., for the t-th sample we observe $y_t = f(x_t)+\epsilon_t$. and we assume a Gaussian noise model where $\epsilon_t \overset{i.i.d.}{\sim} \mathcal{N}\left(0, \sigma_\epsilon^2\right)$.

We use a Gaussian process (GP) to model the objective function. That is, the objective function $f(x) \sim \mathcal{GP}\left(0, K\left(x, x'\right)\right)$, where $K\left(x, x'\right) \doteq \mathrm{Cov}\left(f\left(x\right), f\left(x'\right)\right)$ is a pre-specified kernel function that quantifies the similarity of the surrogate model f evaluated at different decion variables x and x'. A common selection is the radial basis function (RBF) kernel

$$K_{\mathrm{RBF}}\left(x, x'\right) = \exp\left\{-\frac{\|x - x'\|^2}{\sigma^2}\right\}, \tag{1}$$

where $\|\cdot\|$ denotes the Euclidean norm of a vector and σ^2 is a user-specified hyperparameter. The kernel function $K\left(x, x'\right)$ is associated with an operator $\Phi_K : L^2(\mathcal{X}) \to L^2(\mathcal{X})$ with

$$[\Phi_K \phi](x) = \int_{\mathcal{X}} K(x, s)\phi(s)\,\mathrm{d}s \quad \forall \phi \in L^2(\mathcal{X}),$$

where $L^2(\mathcal{X})$ is the space of functions that are square-integrable over the domain $\mathcal{X}$. We refer to [29], [30] for detailed discussions of the kernel operator.

The GP model enjoys explicit uncertainty quantification associated with the function approximation. Conditional on the historical dataset $\tilde{\mathcal{S}} = \{(x_1, y_1), \ldots, (x_t, y_t)\}$, the objective function $f(x)$ remains a GP model:

$$f(x) \mid \tilde{\mathcal{S}} \sim \mathcal{GP}\left(\tilde{\mu}\left(x\right), \tilde{K}\left(x, x'\right)\right), \tag{2}$$

where

$$\tilde{\mu}\left(x\right) = \boldsymbol{K}_t\left(x\right)^\top \left(\tilde{\boldsymbol{K}}_t + \sigma_\epsilon^2 \boldsymbol{I}_t\right)^{-1} \tilde{\boldsymbol{y}}_t, \tag{3}$$

$$\tilde{K}\left(x, x'\right) = K\left(x, x'\right) - \boldsymbol{K}_t\left(x\right)^\top \left(\tilde{\boldsymbol{K}}_t + \sigma_\epsilon^2 \boldsymbol{I}_t\right)^{-1} \boldsymbol{K}_t\left(x'\right).$$

Here $\tilde{\boldsymbol{y}}_t = \left(y_1, \ldots, y_t\right)^\top \in \mathbb{R}^t$ denotes the aggregated observation vector; $\tilde{\boldsymbol{K}}_t \in \mathbb{R}^{t\times t}$ is the kernel matrix with (τ, τ')-th entry $K\left(x_\tau, x_{\tau'}\right)$; and $\boldsymbol{K}_t\left(x\right) = \left(K\left(x, x_1\right), K\left(x, x_2\right), \ldots, K\left(x, x_t\right)\right)^\top \in \mathbb{R}^t$ is the vector of kernel function values between x and $\{x_\tau\}_{\tau=1}^t$. With the explicit inference of $f(x)$ as in (2), an acquisition function is constructed to account for both exploitation and exploration.

A popular choice of acquisition function is the knowledge gradient (KG) [18], which is defined by

$$\alpha_{\mathrm{KG}}(x) = \mathbb{E}_{\tilde{\mathcal{S}}}\left[\max_{x' \in \mathcal{X}} \mathbb{E}\left[f(x') \mid \tilde{\mathcal{S}}_x^*\right]\right], \tag{4}$$

where $\tilde{\mathcal{S}}$ represents the historical observations, and $\tilde{\mathcal{S}}_x^* = \tilde{\mathcal{S}} \cup \{(x, y(x))\}$ is the updated dataset if an additional observation $y(x)$ is decided to be collected at the point x. In this manner, the posterior mean $\mathbb{E}\left[f(x') \mid \tilde{\mathcal{S}}_x^*\right]$ serves as an approximation of the objective function with a future observation $(x, y(x))$ taken into consideration, and the acquisition function (4) is maximized to select the next decision variable.

In this work, we specifically consider a collaborative framework, where there are N agents independently, locally, and and in parallel sampling decision variables to maximize an identical black-box objective function $f(x)$. At each iteration, these agents communicate through a server. The server collects information (which will be specified later) from the agents and then decides which decision variables to sample for them. We denote the t-th data pair collected by the n-th agent as $(x_{n;t}, y_{n;t})$. In the setting where data privacy is not an issue, all such pairs $(x_{n;t}, y_{n;t})$ are sent to the server. The server then constructs a central GP model from these samples. This is precisely the parallel BO or batch BO [9], [31] algorithm, where the server selects a batch of N decision variables $x_{n;t+1}$'s at each iteration, and there is no distinction between the agents.

In contrast, our work assumes data privacy *is* a concern. In other words, the data $(x_{n;t}, y_{n;t})$ collected by each agent, is not allowed to be sent to the server. Instead, each local model, updated by the data collected by each agent, is shared. Specifically, we denote the historical dataset the n-th agent has collected as $\tilde{\mathcal{S}}_n$ for $i = 1, \ldots, N$. We eliminate the dependency of the size of each dataset for notational simplicity. We denote the posterior of the black box function as $\tilde{f}_n \doteq f(x) \mid \tilde{\mathcal{S}}_n$, which is characterized by the associated posterior mean function $\tilde{\mu}_n(x)$ and the posterior kernel function $\tilde{K}_n(x, x')$ as in (3). We implement the collaborative framework by sending the posterior GPs from the agents to the server. Thus, in each iteration, the server receives a set of GP models $\left\{\tilde{f}_1, \ldots, \tilde{f}_N\right\}$ from the agents. To attain a central model, we use the Wasserstein barycenter [32] of the GPs. Informally, he Wasserstein barycenter can be thought of as a weighted average of probability measures. The Wasserstein barycenter is robust to outliers and captures the central tendency of multiple distributions by respecting the underlying geometry of the data. It effectively combines multimodal distributions and aligns them before averaging, preserving distributional characteristics. Specifically, note that a GP is equivalent to a probability measure defined on $\mathcal{X}$. The central model, represented by the Wasserstein barycenter, is defined as the probability measure that minimizes the squared Wasserstein distance to local GP models:

$$f^c = \inf_{f' \in \mathcal{P}(\mathcal{X})} \sum_{n=1}^{N} \left[W_2\left(f', \tilde{f}_n\right)\right]^2. \tag{5}$$

Here, $\mathcal{P}(\mathcal{X})$ denotes the set of all probability measures on $\mathcal{X}$, and $W_2(\cdot, \cdot)$ denotes the 2-Wasserstein distance:

$$W_2(\mu, \nu) \doteq \left(\inf_{\gamma \in \Gamma[\mu,\nu]} \int_{(x,x') \in \mathcal{X} \times \mathcal{X}} \|x - x'\|^2 \, \mathrm{d}\gamma(x, x')\right)^{\frac{1}{2}},$$

where μ and ν are probability measures defined on $\mathcal{X}$, and $\Gamma[\mu, \nu]$ denotes the set of probability measures defined on $\mathcal{X} \times \mathcal{X}$, with marginal distributions μ and ν; see [33]. The central model f^c defined by a Wasserstein barycenter of multiple GP models is itself a GP.

Proposition 1 ([34]). *Let $\left\{\tilde{f}_n\right\}_{i=1}^N$ be a set of GPs with $\tilde{f}_n \sim \mathcal{GP}\left(\tilde{\mu}_n(x), \tilde{K}_n(x, x')\right)$. There exists a unique barycenter $f^c \sim \mathcal{GP}(\mu^c(x), K^c(x, x'))$ defined as in (5). If f^c is non-degenerate, the associated mean function $\mu^c(x)$ and the kernel function $K^c(x, x')$ satisfy that*

$$\mu^c(x) = \frac{1}{N} \sum_{n=1}^{N} \tilde{\mu}_n(x),$$

and

$$\sum_{n=1}^{N} \left(\Phi_{K^c}^{\frac{1}{2}} \Phi_{\tilde{K}_n} \Phi_{K^c}^{\frac{1}{2}}\right)^{\frac{1}{2}} = N \Phi_{K^c},$$

where Φ_K denotes the operator that is associated with the kernel function $K(x, x')$.

Proposition 1 defines the Wasserstein barycenter of GPs with the help of kernel operators. In practice, we compute the barycenter by discretizing GPs to multivariate normal distributions [33]; we elaborate on this in Section III.

We now introduce a general acquisition function for our framework, given by:

$$\alpha(\mathbf{x}) = \underbrace{\alpha^c(\mathbf{x})}_{\text{central}} + \beta_t \sum_{n=1}^{N} \underbrace{\alpha_n(x_n)}_{\text{local}}, \tag{6}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ represents the joint vector of decision variables for the agents. The function $\alpha^c(\mathbf{x})$ denotes the acquisition function based on the central model, which selects N decision variables jointly, while $\alpha_n(x_n)$ refers to the acquisition function for the n-th local model, focusing specifically on x_n. This approach facilitates a collaborative acquisition strategy that leverages the central model while maintaining differentiation through consideration of the local models. The hyperparameter β_t is used to balance this trade-off in the t-th iteration, indicating the extent of collaboration. When $\beta_t \to 0$, the acquisition function approaches a parallel acquisition function focusing on the central model, eliminating the effects of the differences between local models. When $\beta_t \to \infty$, each decision variable x_n is selected to maximize its acquisition function based on the local model, i.e., there is no collaboration between agents.

Specifically, we consider an increasing sequence of β_t and let $\beta_t \to \infty$. In early iterations, there is insufficient data, so the central model collects information from local

models to help construct a more accurate approximation of the objective function, which facilitates the optimization procedure [35]. On the other hand, constructing a central GP model requires approximation (see details in the next section), which introduces additional bias and uncertainty. As the iterations progress and more data is collected, each local model becomes more reliable on its own, reducing the reliance on the central model. Therefore, we attach more weight to each local GP with increasing β_t. We also discuss the effects of this hyper-parameter through numerical experiments.

Furthermore, this collaborative acquisition function is flexible in terms of the selections of $\alpha^c(\mathbf{x})$ and $\alpha_n(x)$ in different scenarios. For example, $\alpha^c(\mathbf{x})$ and $\alpha_n(x)$ can be parallel knowledge gradient (q-KG) [9] and knowledge gradient (KG), parallel expected improvement [36] and expected improvement, and parallel predictive entropy search [37] and entropy search respectively. We compare different selections of collaborative acquisition functions in Section IV-A.

We focus on the KG function due to its proven success in practical implementations. We propose an acquisition function named *collaborative knowledge gradient* (Co-KG):

$$\begin{aligned}\alpha_{\text{Co-KG}}(\mathbf{x}) \doteq & \mathbb{E}_{\tilde{\mathcal{S}}}\left\{\max_{x'\in\mathcal{X}}\mathbb{E}\left[f^c(x') \mid \tilde{\mathcal{S}}^*_{\mathbf{x}}\right]\right. \\ & \left.+\beta_t\left\{\sum_{n=1}^{N}\max_{x'\in\mathcal{X}}\mathbb{E}\left[f(x') \mid \tilde{\mathcal{S}}^*_{x_n}\right]\right\}\right\}.\end{aligned} \tag{7}$$

Here $\tilde{\mathcal{S}} = \cup_{n=1}^{N}\tilde{\mathcal{S}}_n$ represents the historical observations, $\tilde{\mathcal{S}}^*_{\mathbf{x}} = \cup_{n=1}^{N}\tilde{\mathcal{S}}^*_{x_n}$ represents the updated dataset if additional observations are collected as $\tilde{\mathcal{S}}^*_{x_n} = \tilde{\mathcal{S}}_n \cup \{(x_n, y(x_n))\}$, consistent with the definition in (10). In other words, $\alpha^c(\mathbf{x})$ is selected to be q-KG defined on the central GP model $\alpha_{q\text{-KG}}(\mathbf{x}) = \mathbb{E}_{\tilde{\mathcal{S}}}\left[\max_{x'\in\mathcal{X}}\mathbb{E}\left[f^c(x') \mid \tilde{\mathcal{S}}^*_{\mathbf{x}}\right]\right]$, which selects N decision variables in parallel based on the central GP model f^c. Additionally, $\alpha_n(x)$ is a regular KG function for each local GP, as defined in (10). Although (7) involves the set of observations, the construction of the function is facilitated by the central and local GP models, rather than directly utilizing the data, i.e., the data privacy is preserved.

With the acquisition function defined in (7), our framework for collaborative BO (formalized in Algorithm 1) proceeds as follows: First, each agent independently and randomly collect observations and construct GP models in a warm-up stage (line 2). Then, the procedure begins to iterate. At each iteration, the server first collects the GP models from the agents (line 4) to construct a central GP model (line 5). Next, the server maximizes the acquisition function (7) to select the decision variables $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ for each agent (line 6). Consequently, each agent collects observations with the decision variable selected by the server, updates the local GP model, and the procedure moves to the next iteration. This procedure is repeated for T iterations. When the last iteration terminates, each agent submits the optimizer $\hat{x}^*_n$ and the corresponding optimal value μ^*_n of $\max_{x\in\mathcal{X}}\tilde{\mu}_n(x)$ to the server (line 10). The server then determines $\hat{x}^*$ as the decision variable that maximizes across all agents' optimal values (line 11). Implementation details are given in the next section. Here we focus on the general procedure and provide consistency results. We make the following assumptions.

Assumption 1.

1) *The feasible set $\mathcal{X}$ is a compact set.*
2) *Given the kernel function $K(x, x')$, there exists a constant $\tau > 0$ and a continuous function $\rho : \mathbb{R}^d \mapsto \mathbb{R}_+$ such that $K(x, x') = \tau^2\rho(x - x')$. Moreover, ρ satisfies:*
   a) *$\rho(|\delta|) = \rho(\delta)$, where $|\cdot|$ is interpreted component-wise;*
   b) *$\rho(\delta)$ is decreasing in δ component-wise for $\delta \geqslant \mathbf{0}$;*
   c) *$\rho(\mathbf{0}) = 1, \rho(\delta) \to 0$ as $\|\delta\| \to \infty$*
   d) *there exist some $0 < C < \infty$ and $\varepsilon, u > 0$ such that*

$$1 - \rho(\delta) \leqslant \frac{C}{|\log(\|\delta\|)|^{1+\varepsilon}},$$

   for all δ such that $\|\delta\| < u$.

The second condition in the assumptions above is a standard requirement for general kernel functions, such as the RBF kernel function in (1). For other kernel functions that satisfy this condition, we refer to [38].

Theorem 2. *Under Assumption 1, the collaborative BO with Co-KG summarized in Algorithm 1 is consistent. That is,*

$$\lim_{T\to\infty} f(\hat{x}^*) \overset{a.s.}{=} \max_{x\in\mathcal{X}} f(x).$$

Specifically, as the iterations $T \to \infty$, the posterior variance of each local GP model shrinks to zero everywhere, and the posterior mean function converges to the true objective function. Consequently, the limit of function value at the selected decision variable approaches the ground-truth optimal value. The proof is deferred to the supplements [1] in the online version.

Algorithm 1 Collaborative BO with Co-KG.

1: **Input:** The prior kernel function of local GP models;
2: Warm-up stage: Each agent collects observations and updates GP models as in (3);
3: **for** $t = 1, 2, \ldots, T$ **do**
4: The server collects local GP models from each agent;
5: The server constructs the central GP as in (5);
6: The server selects $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ by maximizing the Co-KG function (7);
7: Each agent collects a new observation at x_n;
8: Each agent updates the posterior mean $\tilde{\mu}_n(x)$ and the posterior kernel function $\tilde{K}_n(x, x')$;
9: **end for**
10: Each agent reports $\hat{x}^*_n = \arg\max_{x\in\mathcal{X}}\tilde{\mu}_n(x)$ and $\mu^*_n = \tilde{\mu}_n(\hat{x}^*_n)$ to the server;
11: The server outputs $\hat{x}^* = \arg\max_{x^*_1, x^*_2, \ldots, x^*_N}\mu^*_n$.

[1] The online version of the manuscript is in https://github.com/jd-anderson/Collab_Bayesian_Opt/

III. Implementation

Here we describe the process of calculating the Wasserstein barycenter of a Gaussian process and the optimization of Co-KG based on discretization, which is a standard procedure in BO literature. We discretize the feasible set $\mathcal{X}$ (the domain of GP) to $\mathcal{X}_D = \left\{x^{(1)}, x^{(2)}, \ldots, x^{(D)}\right\}$, where $|\mathcal{X}_D| = D$. The agents send both $\tilde{\boldsymbol{\mu}}_n = \left(\tilde{\mu}_n\left(x^{(1)}\right), \tilde{\mu}_n\left(x^{(2)}\right), \ldots, \tilde{\mu}_n\left(x^{(D)}\right)\right)^\top \in \mathbb{R}^D$ and

$$\tilde{\boldsymbol{K}}_n = \begin{pmatrix} \tilde{K}_n\left(x^{(1)}, x^{(1)}\right) & \ldots & \tilde{K}_n\left(x^{(1)}, x^{(D)}\right) \\ \vdots & \ddots & \vdots \\ \tilde{K}_n\left(x^{(D)}, x^{(1)}\right) & \ldots & \tilde{K}_n\left(x^{(D)}, x^{(D)}\right) \end{pmatrix} \in \mathbb{R}^{D\times D}$$

to the server, where $\tilde{\mu}_n(x)$ and $\tilde{K}_n(x, x')$ denote the posterior mean and kernel functions associated with the n-th local model, updated as in (3). We note that, in the t-th iteration, the posterior mean and kernel functions are constructed using more than t observations since agents have independently collected observations during the warm-stage as in Algorithm 1. Then, we attain the mean vector of the (discretized) central model as $\boldsymbol{\mu}^c = \frac{1}{N}\sum_{n=1}^N \tilde{\boldsymbol{\mu}}_n$ and the kernel matrix $\boldsymbol{K}^c$ by solving the equation

$$\sum_{n=1}^N \left((\boldsymbol{K}^c)^{\frac{1}{2}} \tilde{\boldsymbol{K}}_n (\boldsymbol{K}^c)^{\frac{1}{2}}\right)^{\frac{1}{2}} = N\boldsymbol{K}^c. \tag{8}$$

The equation (8) can be efficiently solved by numerical methods. When the size of discretization $D \to \infty$, $\mathcal{N}(\boldsymbol{\mu}^c, \boldsymbol{K}^c)$ approximates f^c defined in (5) arbitrarily well [34].

This discretization also helps maintain data privacy. Specifically, given the full knowledge of (i) the posterior mean and covariance functions, and (ii) all the decision variables that have been selected, the data can be inferred as in (3). In our implementation, since the posterior mean and covariance functions that the agents send are discretized, this inference cannot be implemented. Additionally, as in Algorithm 1, there is a warm-up stage where the selected decision variables by each agent are not revealed to the server. These two steps main the data privacy requirement, i.e., the server is not able to reveal the exact values of data from the posterior mean and covariance functions sent by the agents.

We now describe the optimization of the acquisition function Co-KG defined in (7). Since both q-KG and KG involve taking expectations, and do not admit an explicit solution, we use a Monte Carlo (MC) based approximation. Specifically, we express the posterior mean function of the central GP model $\mathbb{E}\left[f^c(x') \mid \tilde{\mathcal{S}}^*_{\mathbf{x}}\right]$ as

$$\mu^c(x') + \boldsymbol{K}^c(\mathbf{x}, x')^\top \left(\boldsymbol{\Sigma}(\mathbf{x})\right)^{-1}\left(y^*(\mathbf{x}) - \boldsymbol{\mu}^c(\mathbf{x})\right),$$

where $\boldsymbol{K}^c(\mathbf{x}, x') = \left(K^c(x_1, x'), \ldots, K^c(x_N, x')\right)^\top \in \mathbb{R}^N$,

$$\boldsymbol{\Sigma}(\mathbf{x}) = \begin{pmatrix} K^c(x_1, x_1) & \ldots & K^c(x_1, x_N) \\ \vdots & \ddots & \vdots \\ K^c(x_N, x_1) & \ldots & K^c(x_N, x_N) \end{pmatrix} + \sigma_\epsilon^2 \boldsymbol{I}_N \in \mathbb{R}^{N\times N},$$

and $\boldsymbol{\mu}^c(\mathbf{x}) = \left(\mu^c(x_1), \mu^c(x_2), \ldots, \mu^c(x_N)\right)^\top \in \mathbb{R}^N$. Note that $y^*(\mathbf{x}) - \boldsymbol{\mu}^c(\mathbf{x}) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}(\mathbf{x}))$. We have

$$\mathbb{E}\left[f^c(x') \mid \tilde{\mathcal{S}}^*_{\mathbf{x}}\right] = \mu^c(x') + \boldsymbol{\sigma}^c(\mathbf{x}, x')\,\boldsymbol{\xi},$$

where $\boldsymbol{\sigma}^c(\mathbf{x}, x') = \boldsymbol{K}^c(\mathbf{x}, x')^\top \left(\boldsymbol{D}(\mathbf{x})^\top\right)^{-1}$, $\boldsymbol{D}(\mathbf{x})$ is the Cholesky factor of the matrix $\boldsymbol{\Sigma}(\mathbf{x})$, and $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_N)$. In this manner, when a sample of $\boldsymbol{\xi}$ and the decision variables $\mathbf{x}$ are fixed, $\mathbb{E}\left[f^c(x') \mid \tilde{\mathcal{S}}^*_{\mathbf{x}}\right]$ is a deterministic function of x' so can be optimized over $x' \in \mathcal{X}$. With a similar argument and $\xi \sim \mathcal{N}(0, 1)$, the posterior mean function associated with the n-th local GP model can be expressed as

$$\mathbb{E}\left[f(x') \mid \tilde{\mathcal{S}}^*_{x_n}\right] = \tilde{\mu}_n(x') + \tilde{\sigma}_n(x_n, x')\,\xi,$$

where $\tilde{\sigma}_n(x_n, x') = \tilde{K}_n(x_n, x') / \sqrt{\tilde{K}_n(x_n, x_n) + \sigma_\epsilon^2}$.

Therefore, to approximate the Co-KG function in (7), we first generate $\{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_M\}$, where $\boldsymbol{\xi}_m \overset{i.i.d.}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{I}_N)$. Then, we have the MC approximation

$$\hat{\alpha}_{\text{Co-KG}}(\mathbf{x}) \doteq \max_{\mathbf{z}} \frac{1}{M}\sum_{m=1}^M \left\{\left\{\mu^c(z_m) + \boldsymbol{\sigma}^c(\mathbf{x}, z_m)\,\boldsymbol{\xi}_m\right\} + \beta_t \sum_{n=1}^N \left\{\tilde{\mu}_n(z_{n,m}) + \tilde{\sigma}_n(x_n, z_{n,m})\,\xi_{m,n}\right\}\right\}. \tag{9}$$

Here $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ is the set of decision variables; $\mathbf{z} = (z_1, z_2, \ldots, z_M, z_{1,1}, \ldots, z_{N,M})$ are optimizers to maximize the sampled posterior mean functions; and $x_n, z_m, z_{n,m} \in \mathcal{X}_D$. In addition, $\xi_{m,n}$ denotes the n-th entry of $\boldsymbol{\xi}_m$. Since x_n, z_m and $z_{n,m}$ are restricted to be within $\mathcal{X}_D$, we have the exact values of quantities in (9) in terms of $\tilde{\boldsymbol{\mu}}_n$'s, $\tilde{\boldsymbol{K}}_n$'s, $\boldsymbol{\mu}^c$, and $\boldsymbol{K}^c$. Thus, we can optimize $\mathbf{x}$ and $\mathbf{z}$ altogether to maximize the approximated Co-KG function; see also [39].

Let $\alpha^* = \max_{\mathbf{x}} \alpha_{\text{Co-KG}}(\mathbf{x})$ and $\hat{\alpha}^* = \max_{\mathbf{x}} \hat{\alpha}_{\text{Co-KG}}(\mathbf{x})$. Also, we let $\hat{\mathbf{x}}^* \in \arg\max_{\mathbf{x}} \hat{\alpha}_{\text{Co-KG}}(\mathbf{x})$ and $\boldsymbol{\mathcal{X}}^*$ be the set of optimizers of $\max_{\mathbf{x}} \alpha_{\text{Co-KG}}(\mathbf{x})$. Next we provide the consistency of the maximization based on this MC approximation.

Theorem 3. *If the kernel function is continuously differentiable, we have*

$$\lim_{M\to\infty} \hat{\alpha}^* \overset{a.s.}{=} \alpha^* \quad \textit{and} \quad \lim_{M\to\infty} \operatorname{dist}(\hat{\mathbf{x}}^*, \boldsymbol{\mathcal{X}}^*) \overset{a.s.}{=} 0,$$

where M denotes the number of samples generated to construct the MC approximation. Furthermore, $\forall \delta > 0, \exists K < \infty, \beta_t > 0$ such that

$$\mathbb{P}\left(\operatorname{dist}(\hat{\mathbf{x}}^*_N, \mathcal{X}^*) > \delta\right) \leqslant K e^{-\beta_t M} \quad \forall M \geqslant 1.$$

Proof. We omit the details due to lack of space, however it follows from a simple adaptation of Theorem 3 from [39], which is itself grounded in the theoretical foundations of the sample average approximation method [40]. □

We note that the condition that the kernel function is continuously differentiable guarantees that the objective function (the sample path of the GP) is continuously differentiable as

well. As a result, the sample path of the central GP is also continuously differentiable, supported by the properties of the Wasserstein distance and the Fréchet mean [41], [42].

IV. Experiments

We first compare the performances of different collaborative acquisition functions in order to justify the use of the knowledge gradient function. Next, we explore the algorithm performance on the hyperparameter selection. Lastly, we compare our approach (Algorithm 1) with several baseline approaches, as well as a parallel BO algorithm without data privacy concerns.

Across all sets of experiments, we fix the number of agents at $N = 4$. We also normalize the feasible set to the box $\mathcal{X} = [0, 1]^2$ and discretize the feasible set using a 20×20 uniform meshgrid. Additional experiments on the effects of the number of agents and different discretization strategies are also included in the supplementary materials. Our experiments were conducted with Botorch [39] and Python 3.9 on a computer equipped with two AMD Ryzen Threadripper 3970X 32-Core Processors, 128 GB memory, and a Nvidia GeForce RTX A6000 GPU with 48GB of RAM.

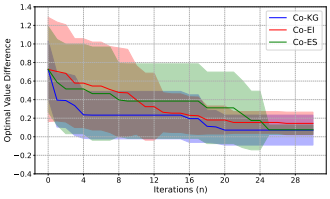

Fig. 1. Optimal value differences with different acquisition functions on the black-box objective function $f_1(x)$.

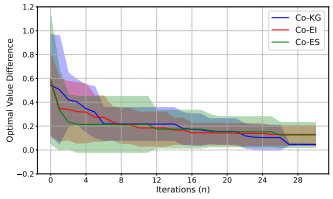

Fig. 2. Optimal value differences with different acquisition functions on the black-box objective function $f_2(x)$.

A. Collaborative Acquisition Function Comparison

We conduct experiments to compare different collaborative acquisition functions within the general form of (6). Specifically, we consider (i) the collaborative knowledge gradient (Co-KG) function as in (7); (ii) the collaborative expected improvement (Co-EI) function with $\alpha^c(\mathbf{x})$ selected as the parallel expected improvement function and $\alpha_n(x)$ selected as the expected improvement function [36]; and (iii) the collaborative entropy search (Co-ES) function with $\alpha^c(\mathbf{x})$ selected as the parallel expected improvement function and $\alpha_n(x)$ selected as the expected improvement function [37].

Regarding the hyperparameter in the collaborative functions, we set $\beta_t = \log(2t+1)$. We compare the performance of these algorithms using the two functions: a function with quadratic terms and trigonometric terms

$$f_1(x) = x_1^2 + x_2^2 + \sin(2\pi x_1) + \cos(2\pi x_2),$$

and the Rosenbrock function

$$f_2(x) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2.$$

When collecting the observations, we add Gaussian noise as $y = f(x) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2 = 0.02)$. Regarding the variance of the noise, σ_ϵ^2, each agent estimates it from observations collected in the warm-up stage using maximum likelihood estimation, denoted by $\hat{\sigma}_n^2$. Then the variance is $\left(\sum_{n=1}^N \hat{\sigma}_n^2\right)/N$ fixed by the server. The experimental results are in Figure 1 and Figure 2, where we report the *optimal value difference* defined by

$$\arg\max_{x \in \mathcal{X}} f(x) - f(\hat{x}^*).$$

Here $\hat{x}^*$ is the maximizer selected by the server after the iterations end; see Algorithm 1. The experimental results presented are mean performances based on 10 repetitions, with standard deviations represented by a shadow around the mean-value line. Each repetition of the optimization procedure includes 30 iterations, and each agent has 5 observations associated with randomly selected decision variables before the start of the iterations. From the experimental results in Figure 1, we observe that Co-KG consistently outperforms Co-EI and Co-ES in terms of achieving lower optimal value differences across iterations. In contrast, in Figure 2, Co-KG does not perform as well as Co-EI or Co-ES in the initial iterations but surpasses them as the iterations increase. The reason is that the Co-KG function is overconfident in the set of experiments associated with $f_2(x)$, relying too heavily on the conditional mean function. When there is insufficient data, the conditional mean is not accurate enough in approximating the objective function. However, as the iterations increase and more data becomes available, Co-KG, which relies on the conditional mean, outperforms the other two approaches due to a more accurate approximation of the objective function. This advantage is also evident in the results shown in Figure 1. Since $f_1(x)$ is relatively simple, the conditional mean serves as a satisfactory approximation even when there is insufficient data at the beginning of the iterations, leading to Co-KG's preferable performance. Considering both experimental results, we focus on the acquisition function Co-KG in the following experiments.

B. Hyperparameter Analysis

We now explore the dependence of Co-KG on the hyperparameter β_t. We numerically evaluate how a time-varying hyperparameter β_t will affect the performance of Co-KG, where t denotes the iteration of the procedure. Specifically, we consider a decreasing sequence of hyperparameters $\beta_t = e^{-t/2}$ and an increasing sequence $\beta_t = \log(2t+1)$. We also include $\beta_t = 1$ for comparison. We follow the setting of section IV-A to further investigate the impact of hyperparameter.

The experimental results are included in Figure IV-B. We see that the performance of Co-KG does not significantly depend on the hyperparameter β_t in the initial iterations. In comparison, as the iterations progress, an increasing sequence of β_t outperforms the other two selections in both sets of experiments. The reason is two-fold: In early iterations, there is insufficient data, so the central model collects information from local models to help construct a more accurate approximation of the objective function, which helps the overall

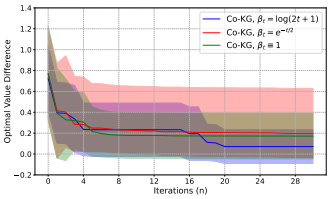

Fig. 3. Optimal value difference in iterations with different selections of β_t on $f_1(x)$.

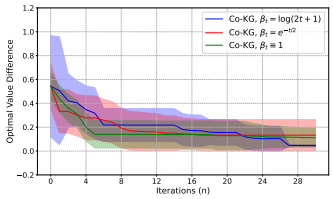

Fig. 4. Optimal value difference in iterations with different selections of β_t on $f_2(x)$.

optimization procedure [35]. On the other hand, constructing a central GP model requires approximation, which introduces additional bias and uncertainty. As the iteration increases and more data is collected, each local model becomes more reliable, reducing the reliance on the central model. Additionally, from the perspective of the surrogate models, β_t also addresses the exploration-exploitation trade-off. A higher β_t favors the exploration of local models to gather diverse information about the objective function. Increasing β_t as the iterations progress, to manage the exploration-exploitation trade-off, is also supported by classical BO literature [19].

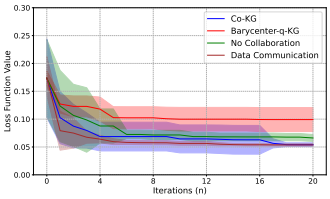

Fig. 5. Loss function values in iterations with compared BO approaches on Breast Cancer Dataset.

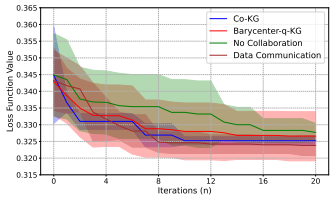

Fig. 6. Loss function values in iterations with compared BO approaches on California Housing Dataset.

C. Collaborative Framework Comparison

We compare our framework using Co-KG with several baseline approaches including (i) the BO approach with the parallel Knowledge Gradient function focusing on the central model constructed by the Wasserstein Barycenter (Barycenter-q-KG); (ii) the BO approach with the Knowledge Gradient function implemented by each agent without collaboration (No Collaboration); and (iii) the BO approach with the parallel Knowledge Gradient function with permission to Data Communication (Data Communication).

Regarding the black-box optimization problem, we consider the task of hyperparameter tuning for learning 3-layer neural networks from data. Specifically, the decision variable is the learning rate and the hidden layer node size of the neural networks. The unknown objective function is the loss function we would minimize. The neural networks are learned from two datasets. The first dataset is related to the breast cancer[2], where the neural network is learned to predict the breast cancer classification label given predictive attributes. The second dataset is describes California housing prices[3], where the neural network is learned to predict median value of houses in different districts given demographic attributes. We do not impose noise on the observations in the real dataset.

We record the loss function values of training neural networks in Figure 5 and Figure 6. The results provide the following insights. First, the Data Communication approach achieves the best performance with the smallest loss function values on both datasets, since this approach has permission to share data and therefore exploits the data most effectively to construct the GP model. Second, our proposed Co-KG approach achieves performance comparable to the Data Communication approach and outperforms the other two compared approaches on both datasets. This indicates the effectiveness of 1) collaboration among agents (as seen in the comparison with No Collaboration) and 2) considering the differentiation between agents (as seen in the comparison with Barycenter-q-KG). Lastly, when the unknown objective function is relatively simple to optimize (as in the Breast Cancer Dataset), distributed methods (i.e., no collaboration) can already achieve acceptable performance. In these scenarios, inefficient collaboration (Barycenter-q-KG) might decrease the performance of BO approaches.

V. Conclusion

We consider a collaborative framework for Bayesian optimization (BO) with data privacy, where multiple agents collect data to optimize an identical black-box objective function, without sharing their data. In our framework, agents share the Gaussian process (GP) models constructed with their own data, and a server builds a central GP model from the shared local GP models using the concept of the Wasserstein barycenter. We propose a general acquisition function that takes both the central model and local models into consideration and selects decision variables for agents in each iteration. We specifically focus on the knowledge gradient algorithm and propose a collaborative knowledge gradient (Co-KG) function. We establish the consistency of the BO approach based on Co-KG. To approximate Co-KG, we employ a Monte Carlo method and prove the consistency of this approximation as well. Additionally, we conduct numerical experiments to demonstrate that Co-KG outperforms other collaborative acquisition functions within our framework and achieves superior performance compared to other collaborative frameworks. We also show that our framework with Co-KG can achieve performance comparable to approaches that do not have data privacy concerns.

VI. Acknowledgements

James Anderson acknowledges funding from NSF grants ECCS 2144634 and 2231350 and the Columbia Data Science Institute.

References

[1] P. I. Frazier and J. Wang, "Bayesian optimization for materials design," in *Information science for materials discovery and design*. Springer, 2016, pp. 45–75.

[2]https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic

[3]https://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.html

[2] S. F. Ghoreishi and M. Imani, "Bayesian optimization for efficient design of uncertain coupled multidisciplinary systems," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 3412–3418.

[3] R. Martinez-Cantin, N. De Freitas, E. Brochu, J. Castellanos, and A. Doucet, "A bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot," *Autonomous Robots*, vol. 27, pp. 93–103, 2009.

[4] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," *Advances in neural information processing systems*, vol. 25, 2012.

[5] P. I. Frazier, "Bayesian optimization," in *Recent advances in optimization and modeling of contemporary problems*. Informs, 2018, pp. 255–278.

[6] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2.

[7] C.-D. Liang, M.-F. Ge, J.-Z. Xu, Z.-W. Liu, and F. Liu, "Secure and privacy-preserving formation control for networked marine surface vehicles with sampled-data interactions," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 2, pp. 1307–1318, 2021.

[8] L. Chen, N. Li, L. Jiang, and S. H. Low, "Optimal demand response: Problem formulation and deterministic case," *Control and optimization methods for electric smart grids*, pp. 63–85, 2012.

[9] J. Wu and P. Frazier, "The parallel knowledge gradient method for batch Bayesian optimization," *Advances in neural information processing systems*, vol. 29, 2016.

[10] D. Pan, D. Ding, X. Ge, Q.-L. Han, and X.-M. Zhang, "Privacy-preserving platooning control of vehicular cyber–physical systems with saturated inputs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 4, pp. 2083–2097, 2022.

[11] J. Du, C. Jiang, E. Gelenbe, L. Xu, J. Li, and Y. Ren, "Distributed data privacy preservation in iot applications," *IEEE Wireless Communications*, vol. 25, no. 6, pp. 68–76, 2018.

[12] F. Zhou, J. Anderson, and S. H. Low, "Differential privacy of aggregated dc optimal power flow data," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 1307–1314.

[13] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[14] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[15] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[16] H. Yuan and T. Ma, "Federated accelerated stochastic gradient descent," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5332–5344, 2020.

[17] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[18] P. Frazier, W. Powell, and S. Dayanik, "The knowledge-gradient policy for correlated normal beliefs," *INFORMS journal on Computing*, vol. 21, no. 4, pp. 599–613, 2009.

[19] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," in *Proceedings of the 27th International Conference on Machine Learning*. Omnipress, 2010.

[20] Z. Wang, C. Gehring, P. Kohli, and S. Jegelka, "Batched large-scale Bayesian optimization in high-dimensional spaces," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2018, pp. 745–754.

[21] F. Jimenez and M. Katzfuss, "Scalable Bayesian optimization using vecchia approximations of Gaussian processes," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2023, pp. 1492–1512.

[22] S. Daulton, S. Cakmak, M. Balandat, M. A. Osborne, E. Zhou, and E. Bakshy, "Robust multi-objective Bayesian optimization under input noise," in *International Conference on Machine Learning*. PMLR, 2022, pp. 4831–4866.

[23] Z. J. Lin, R. Astudillo, P. Frazier, and E. Bakshy, "Preference exploration for efficient Bayesian optimization with multiple outcomes," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 4235–4258.

[24] R. Astudillo, D. Jiang, M. Balandat, E. Bakshy, and P. Frazier, "Multi-step budgeted Bayesian optimization with unknown evaluation costs," *Advances in Neural Information Processing Systems*, vol. 34, pp. 20 197–20 209, 2021.

[25] K. Kandasamy, G. Dasarathy, J. Schneider, and B. Póczos, "Multi-fidelity Bayesian optimisation with continuous approximations," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1799–1808.

[26] Z. Dai, B. K. H. Low, and P. Jaillet, "Federated Bayesian optimization via thompson sampling," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9687–9699, 2020.

[27] X. Yue, Y. Liu, A. S. Berahas, B. N. Johnson, and R. Al Kontar, "Collaborative and distributed bayesian optimization via consensus," *IEEE Transactions on Automation Science and Engineering*, 2025.

[28] S. Zerefa, Z. Ren, H. Ma, and N. Li, "Distributed thompson sampling under constrained communication," *IEEE Control Systems Letters*, 2025.

[29] M. Seeger, "Gaussian processes for machine learning," *International journal of neural systems*, vol. 14, no. 02, pp. 69–106, 2004.

[30] A. W. van der Vaart, J. H. van Zanten *et al.*, "Reproducing kernel Hilbert spaces of Gaussian priors," *IMS Collections*, vol. 3, pp. 200–222, 2008.

[31] S. Daulton, M. Balandat, and E. Bakshy, "Parallel Bayesian optimization of multiple noisy objectives with expected hypervolume improvement," *Advances in Neural Information Processing Systems*, vol. 34, pp. 2187–2200, 2021.

[32] G. Puccetti, L. Rüschendorf, and S. Vanduffel, "On the computation of wasserstein barycenters," *Journal of Multivariate Analysis*, vol. 176, p. 104581, 2020.

[33] V. Masarotto, V. M. Panaretos, and Y. Zemel, "Procrustes metrics on covariance operators and optimal transportation of gaussian processes," *Sankhya A*, vol. 81, pp. 172–213, 2019.

[34] A. Mallasto and A. Feragen, "Learning from uncertain curves: The 2-Wasserstein metric for Gaussian processes," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[35] X. Yue, R. A. Kontar, A. S. Berahas, Y. Liu, Z. Zai, K. Edgar, and B. N. Johnson, "Collaborative and distributed Bayesian optimization via consensus: Showcasing the power of collaboration for optimal design," *arXiv preprint arXiv:2306.14348*, 2023.

[36] J. Wang, S. C. Clark, E. Liu, and P. I. Frazier, "Parallel Bayesian global optimization of expensive functions," *Operations Research*, vol. 68, no. 6, pp. 1850–1865, 2020.

[37] A. Shah and Z. Ghahramani, "Parallel predictive entropy search for batch global optimization of expensive objective functions," *Advances in neural information processing systems*, vol. 28, 2015.

[38] L. Ding, L. J. Hong, H. Shen, and X. Zhang, "Knowledge gradient for selection with covariates: Consistency and computation," *Naval Research Logistics (NRL)*, vol. 69, no. 3, pp. 496–507, 2022.

[39] M. Balandat, B. Karrer, D. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy, "Botorch: A framework for efficient Monte-Carlo Bayesian optimization," *Advances in neural information processing systems*, vol. 33, pp. 21 524–21 538, 2020.

[40] A. J. Kleywegt, A. Shapiro, and T. Homem-de Mello, "The sample average approximation method for stochastic discrete optimization," *SIAM Journal on optimization*, vol. 12, no. 2, pp. 479–502, 2002.

[41] C. R. Rao, "Differential metrics in probability spaces," *Differential geometry in statistical inference*, vol. 10, pp. 217–240, 1987.

[42] V. M. Panaretos and Y. Zemel, "Statistical aspects of wasserstein distances," *Annual review of statistics and its application*, vol. 6, no. 1, pp. 405–431, 2019.

[43] J. Bect, F. Bachoc, and D. Ginsbourger, "A supermartingale approach to Gaussian process based sequential design of experiments," *Bernoulli*, vol. 25, no. 4A, pp. 2883 – 2919, 2019. [Online]. Available: https://doi.org/10.3150/18-BEJ1074

[44] W. Scott, P. Frazier, and W. Powell, "The correlated knowledge gradient for simulation optimization of continuous parameters using gaussian process regression," *SIAM Journal on Optimization*, vol. 21, no. 3, pp. 996–1026, 2011.

[45] A. W. Van der Vaart, *Asymptotic statistics*. Cambridge university press, 2000, vol. 3.

VII. Proof of Theoretical Results

We prove Theorem 2 in the main text here. The result requires Assumption 1 as stated in the main text and repeated below:

Assumption 1.

1) The feasible set $\mathcal{X}$ is a compact set.

2) Regarding the kernel function $K(x, x')$, there exists a constant $\tau > 0$ and a continuous function $\rho : \mathbb{R}^d \mapsto \mathbb{R}_+$ such that $K(x, x') = \tau^2 \rho(x - x')$. Moreover,

a) $\rho(|\delta|) = \rho(\delta)$, where $|\cdot|$ means taking the absolute value component-wise;

b) $\rho(\delta)$ is decreasing in δ component-wise for $\delta \geqslant \mathbf{0}$;

c) $\rho(\mathbf{0}) = 1, \rho(\delta) \to 0$ as $\|\delta\| \to \infty$, where $\|\cdot\|$ denotes the Euclidean norm;

d) there exist some $0 < C < \infty$ and $\varepsilon, u > 0$ such that

$$1 - \rho(\delta) \leqslant \frac{C}{|\log(\|\delta\|)|^{1+\varepsilon}},$$

for all δ such that $\|\delta\| < u$.

We first focus on the local GP model. Here we hide the index n of each local GP. Recall that the objective function to optimize $f(x)$ is a GP model with the prior kernel function $K(x, x')$. Specifically, we have the following proposition on the convergence of posterior kernel functions of $f(x)$.

Proposition 4 (Proposition 1 of [38])**.** *If the kernel function $K(x, x')$ satisfies the condition in Assumption 1, then*

$$\lim_{t \to \infty} \tilde{K}(x, x') \xrightarrow{a.s.} K^{\infty}(x, x'),$$

and the convergence is uniform. Here $\tilde{K}(x, x')$ is the posterior kernel function after collecting t observations, as defined in the main text, and $K^{\infty}(x, x')$ is a function that does not depend on t.

Next we provide a corollary regarding the posterior variance

$$\mathrm{Var}\left[f(x) \mid \tilde{\mathcal{S}}\right] = \tilde{K}(x, x).$$

We note that, under Assumption 1, there would be an accumulative point $x^{acc} \in \mathcal{X}$ for each local GP model. We here provide an asymptotic upper bound of $\mathrm{Var}\left[f(x) \mid \tilde{\mathcal{S}}\right]$ within an area centered at this accumulative point.

Lemma 5 (Lemma 6 of [38])**.** *Under Assumption 1, $\forall \epsilon > 0$, we have*

$$\limsup_{t \to \infty} \max_{x \in \mathcal{B}(x^{acc}, \epsilon)} \mathrm{Var}\left[f(x) \mid \tilde{\mathcal{S}}\right] \leqslant \tau^2 \left[1 - \rho^2(2\epsilon \mathbf{1})\right],$$

where $\mathbf{1}$ is the vector of all ones with size $d \times 1$, $\mathcal{B}(x^{acc}, \epsilon)$ is the ball centered at x^{acc} with radius ϵ.

Recall that our Co-KG function is composed of one q-KG function with multiple decision variables as the input and multiple (regular) KG functions with one decision variable as the input. In the main text, we subtract the maximum posterior mean for simplification. That is, an equivalent definition of the KG function is

$$\alpha_{\mathrm{KG}}(x) = \mathbb{E}_{\tilde{\mathcal{S}}}\left[\max_{x' \in \mathcal{X}} \mathbb{E}\left[f(x') \mid \tilde{\mathcal{S}}_x^*\right]\right] - \max_{x' \in \mathcal{X}} \mathbb{E}\left[f(x') \mid \tilde{\mathcal{S}}\right]. \tag{10}$$

Here $\tilde{\mathcal{S}}$ represents the historical observations, and $\tilde{\mathcal{S}}_x^* = \tilde{\mathcal{S}} \cup \{(x, y(x))\}$ is the updated dataset if an additional observation $y(x)$ is decided to be collected at the decision variable x. In this manner, regarding the posterior mean as the approximated objective function, KG represents the increment of the optimal value if an additional sample is collected at x. Since the term

$$\max_{x' \in \mathcal{X}} \mathbb{E}\left[f(x') \mid \tilde{\mathcal{S}}\right]$$

does not involve the decision variable x to be optimized, we do not include it in the main text considering the limited length. A similar definition of the q-KG is in [9]. When we prove the consistency of the collaborative BO procedure with Co-KG in this section, these terms are included. Furthermore, we note that the KG function is non-negative, to see this, we use the Jensen inequality:

$$\begin{aligned}
\mathbb{E}\left[\max_{x' \in \mathcal{X}} \mathbb{E}\left[f(x') \mid \tilde{\mathcal{S}}_x^*\right]\right] &= \mathbb{E}\left[\max_{x' \in \mathcal{X}} \mathbb{E}\left[\tilde{\mu}(x') + \tilde{\sigma}(x, x')\,\xi\right]\right] \\
&\geqslant \max_{x' \in \mathcal{X}} \tilde{\mu}(x') + \mathbb{E}\left[\tilde{\sigma}(x, x')\,\xi\right] \\
&= \max_{x' \in \mathcal{X}} \mathbb{E}\left[f(x') \mid \tilde{\mathcal{S}}\right],
\end{aligned}$$

where $\tilde{\sigma}(x,x') = \tilde{K}(x,x')/\sqrt{\tilde{K}(x,x)+\sigma_\epsilon^2}$.The non-negativity based on the Jensen inequality also holds for the q-KG function with a similar argument [9]. Since our Co-KG function is a weighted summation of a q-KG function and multiple regular KG functions, it is non-negative as well.

Regarding the KG function associated with each local GP model (10), we have

$$\begin{aligned}
\alpha_{\mathrm{KG}}(x) =& \mathbb{E}\left[\max_{x'\in\mathcal{X}} \mathbb{E}\left[f(x') \mid \tilde{\mathcal{S}}_x^*\right]\right] - \max_{x'\in\mathcal{X}} \mathbb{E}\left[f(x') \mid \tilde{\mathcal{S}}\right] \\
=& \mathbb{E}\left[\max_{x'\in\mathcal{X}} \mathbb{E}\left[\tilde{\mu}(x') + \tilde{\sigma}(x,x')\,\xi\right]\right] - \max_{x'\in\mathcal{X}} \mathbb{E}\left[f(x') \mid \tilde{\mathcal{S}}\right] \\
\leqslant& \max_{x'\in\mathcal{X}} \tilde{\mu}(x') + \mathbb{E}\left[\max_{x'\in\mathcal{X}} \tilde{\sigma}(x,x')\,\xi\right] - \max_{x'\in\mathcal{X}} \mathbb{E}\left[f(x') \mid \tilde{\mathcal{S}}\right] \\
=& \mathbb{E}\left[\max_{x'\in\mathcal{X}} \tilde{\sigma}(x,x')\,\xi\right] \\
\leqslant& \mathbb{E}\left[|\xi|\right] \max_{x'\in\mathcal{X}} \tilde{\sigma}(x,x') \\
=& \sqrt{\frac{2}{\pi}} \max_{x'\in\mathcal{X}} \tilde{\sigma}(x,x')
\end{aligned} \tag{11}$$

Furthermore, we have that

$$\begin{aligned}
\tilde{\sigma}(x,x') &= \frac{\tilde{K}(x,x')}{\sqrt{\tilde{K}(x,x)+\sigma_\epsilon^2}} \\
&\leqslant \sqrt{\frac{\tilde{K}(x,x)\,\tilde{K}(x',x')}{\tilde{K}(x,x)+\sigma_\epsilon^2}} \\
&\leqslant \sqrt{\frac{\tau^2\tilde{K}(x,x)}{\sigma_\epsilon^2}},
\end{aligned} \tag{12}$$

where the last inequality comes from the fact that $\tilde{K}(x',x')$ is a non-increasing sequence regarding t and the conditions in Assumption 1. Thus, from (11) and (12), we have

$$\alpha_{\mathrm{KG}}(x) \leqslant \sqrt{\frac{2\tau^2\tilde{K}(x,x)}{\pi\sigma_\epsilon^2}} \tag{13}$$

Regarding the q-KG function, we have

$$\mathbb{E}\left[f^c(x') \mid \tilde{\mathcal{S}}_{\mathbf{x}}^*\right] = \mu^c(x') + \boldsymbol{\sigma}^c(\mathbf{x},x')\,\boldsymbol{\xi},$$

where $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_N)$. Additionally,

$$\boldsymbol{\sigma}^c(\mathbf{x},x') = \boldsymbol{K}^c(\mathbf{x},x')^\top \left(\boldsymbol{D}(\mathbf{x})^\top\right)^{-1},$$

where $\boldsymbol{D}(\mathbf{x})$ is the Cholesky factor of the matrix

$$\boldsymbol{\Sigma}(\mathbf{x}) = \begin{pmatrix} K^c(x_1,x_1) & \dots & K^c(x_1,x_N) \\ \vdots & \ddots & \vdots \\ K^c(x_N,x_1) & \dots & K^c(x_N,x_N) \end{pmatrix} + \sigma_\epsilon^2 \boldsymbol{I}_N.$$

Note that,

$$\boldsymbol{\sigma}^c(\mathbf{x},x')\,\boldsymbol{\xi} \sim \mathcal{N}\left(0, \|\boldsymbol{\sigma}^c(\mathbf{x},x')\|\right).$$

With a similar argument as in (11), the q-KG function is bounded by

$$\alpha_{q\text{-KG}}(\mathbf{x}) \leqslant \sqrt{\frac{2}{\pi}} \max_{x'\in\mathcal{X}} \|\boldsymbol{\sigma}^c(\mathbf{x},x')\|.$$

Furthermore,

$$\begin{aligned}
&\left\|\boldsymbol{\sigma}^c\left(\mathbf{x}, x'\right)\right\|^2 \\
=&\boldsymbol{K}^c\left(\mathbf{x}, x'\right)^\top \boldsymbol{\Sigma}\left(\mathbf{x}\right) \boldsymbol{K}^c\left(\mathbf{x}, x'\right) \\
\leqslant& K^c\left(x', x'\right) \\
=&\left(\frac{1}{N}\sum_{n=1}^N \tilde{K}_n\left(x', x'\right)\right) + \frac{1}{N}\sum_{n=1}^N \left(\tilde{\mu}_n\left(x'\right) - \mu^c\left(x'\right)\right)^2,
\end{aligned} \tag{14}$$

where the last equality comes from the fact that the central GP is a 2-Wasserstein barycenter of local GPs. In this manner, we have connected the terms of the Co-KG function to the variances of local models. We here present a proposition regarding the conditional mean function $\tilde{\mu}\left(x\right)$.

Proposition 6 (Proposition 2.9 in [43])**.** *Under Assumption 1, the conditional mean function converges to* $\mu^\infty(x) \doteq \mathbb{E}\left[f(x) \mid \mathcal{F}_\infty\right]$ *uniformly in* $x \in \mathcal{X}$ *almost surely (a.s.). That is,*

$$\mathbb{P}\left\{\sup_{x\in\mathcal{X}} |\tilde{\mu}(x) - \mu^\infty(x)| \to 0\right\} = 1$$

as $t \to \infty$.

Here $\mathcal{F}_\infty$ *denotes the filtration of the dataset collection when the number of iterations approaches infinity.*

Next we consider a rescaled Co-KG function:

$$\alpha_{\text{Co-KG}}\left(\mathbf{x}\right) \doteq \mathbb{E}_{\tilde{\mathcal{S}}}\left\{\frac{\alpha^c\left(\mathbf{x}\right)}{\beta_t} + \sum_{n=1}^N \alpha_n\left(x\right)\right\},$$

where $\alpha^c(\mathbf{x})$ is the q-KG function defined on the central GP and $\alpha_n(x)$ is the regular KG function defined on the n-th local GP model. We provide an asymptotic property of Co-KG:

Lemma 7. *Under Assumption 1, the limit inferior of Co-KG is 0. That is,*

$$\liminf_{t\to\infty} \alpha_{\textit{Co-KG}}(\mathbf{x}) = 0 \quad \forall \mathbf{x} \in \mathcal{X}^N.$$

Proof. We first provide the notation here. We denote the sequence of decision variables selected by maximizing Co-KG as

$$\mathbf{x}(t) = \left(x_1\left(t\right), x_2\left(t\right), \ldots, x_N\left(t\right)\right),$$

where $x_n(t)$ denotes the decision variables for the n-th agent in the t-th iteration. In the main text, we did not emphasize the dependence on t for notational simplicity. For each local model, note that the sequence of the selected $x_n(t)$'s for this local GP has an accumulative point x^{acc}. We denote the subsequence as $z_n(1), z_n(2), \ldots, z_n\left(t'\right)$ such that $z_n\left(t'\right) \to x^{acc}$. Based on Lemma 5, we have

$$\limsup_{t\to\infty} \operatorname{Var}\left[f\left(z_n\left(t'\right)\right) \mid \tilde{\mathcal{S}}_n\right] \leqslant \tau^2\left[1 - \rho^2(2\epsilon\mathbf{1})\right].$$

Furthermore, from (13), we have

$$\limsup_{t\to\infty} \alpha(z_n\left(t'\right)) \leqslant \sqrt{\frac{2\tau^2\tilde{K}\left(z_n\left(t'\right), z_n\left(t'\right)\right)}{\pi\sigma_\epsilon^2}} \leqslant \limsup_{t\to\infty} \sqrt{\frac{2\tau^4}{\pi\sigma_\epsilon^2}}\left[1 - \rho^2(2\epsilon\mathbf{1})\right].$$

Let $\epsilon \to 0$, we have that

$$\liminf_{t\to\infty} \alpha_{\text{KG}}\left(x_n(t)\right) \leqslant \liminf_{t\to\infty} \alpha_{\text{KG}}\left(z_n\left(t'\right)\right) \leqslant \limsup_{t\to\infty} \alpha_{\text{KG}}(z_n\left(t'\right)) = 0 \quad \forall n. \tag{15}$$

The last equality holds because the KG function $\alpha(x)$ is non-negative.

Then we look at the central GP and the associated q-KG function. Recall that, from (14), we have

$$\alpha_{q\text{-KG}}\left(\mathbf{x}\right) \leqslant \sqrt{\frac{2}{\pi}\left(\left(\frac{1}{N}\sum_{n=1}^N \tilde{K}_n\left(x', x'\right)\right) + \frac{1}{N}\sum_{n=1}^N \left(\tilde{\mu}_n\left(x'\right) - \mu^c\left(x'\right)\right)^2\right)} \quad \forall \mathbf{x},$$

where x' denotes some maximizer. Note that $\tilde{K}_n\left(x', x'\right) \leqslant K\left(x', x'\right)$ is bounded, and each $\left(\tilde{\mu}_n\left(x'\right) - \mu^c\left(x'\right)\right)^2$ is bounded as well because of Proposition 6. Thus, as $t \to \infty$,

$$\lim_{t\to\infty} \frac{\alpha_{q\text{-KG}}\left(\mathbf{x}\left(t\right)\right)}{\beta_t}$$

since $\beta_t \to \infty$ from the definition in the main text. This further leads to

$$\liminf_{t\to\infty} \alpha_{\text{Co-KG}}\left(\mathbf{x}\left(t\right)\right) = 0$$

because of (15). Recall that

$$\mathbf{x}\left(t\right) = \arg\max_{\mathbf{x}\in\mathcal{X}^N} \alpha_{\text{Co-KG}}\left(\mathbf{x}\right).$$

Thus, $\forall \mathbf{x}$, we have

$$0 \leqslant \liminf_{t\to 0} \alpha_{\text{Co-KG}}(\mathbf{x}) \leqslant \liminf_{t\to 0} \alpha_{\text{Co-KG}}(\mathbf{x}(t)) = 0$$

□

Now we prove that for all local GP models, the uncertainty at each decision variable (represented by the conditional variance) shrinks to zero. Specifically, we have the following lemma.

Lemma 8. *Under Assumption 1, $\forall n$ and $\forall x \in \mathcal{X}$, we have*

$$\lim_{t\to\infty} \operatorname{Var}\left[f\left(x\right) \mid \tilde{\mathcal{S}}_n\right] = 0 \quad \forall x \in \mathcal{X}, \forall n. \tag{16}$$

Proof. Without loss of generality, we assume that

$$\lim_{t\to\infty} \operatorname{Var}\left[f\left(\tilde{x}\right) \mid \tilde{\mathcal{S}}_1\right] = c > 0. \tag{17}$$

This limit exists because of Lemma 5. Also, since $\tilde{K}\left(x, x'\right)$ is continuous and the convergence in Lemma 5 is uniform, $\operatorname{Var}\left[f\left(x\right) \mid \tilde{\mathcal{S}}_1\right]$ is continuous as well. Regarding this agent, we denote the current posterior function as $\tilde{\mu}\left(x\right) = \mathbb{E}\left[f\left(x\right) \mid \tilde{\mathcal{S}}_1\right]$ and the posterior function with additional x as $\tilde{\mu}^*\left(x'\right) = \mathbb{E}\left[f\left(x'\right) \mid \tilde{\mathcal{S}}_x^*\right]$. Furthermore, we denote $\tilde{x}^* = \arg\max_{x\in\mathcal{X}} \tilde{\mu}(x)$. Also, let

$$\begin{aligned}
a_1 =& \tilde{\mu}\left(\tilde{x}^*\right)\\
b_1 =& \tilde{\sigma}\left(x, \tilde{x}^*\right)\\
a_2 =& \tilde{\mu}\left(x\right)\\
b_2 =& \tilde{\sigma}\left(x, x\right).
\end{aligned}$$

We consider the KG function associated with this agent:

$$\begin{aligned}
&\alpha_{\text{KG}}(x)\\
&= \mathbb{E}\left[\max_{x'\in\mathcal{X}} \mathbb{E}\left[f\left(x'\right) \mid \tilde{\mathcal{S}}_x^*\right]\right] - \max_{x'\in\mathcal{X}} \mathbb{E}\left[f\left(x'\right) \mid \tilde{\mathcal{S}}\right]\\
&= \mathbb{E}\left[\max_{x'\in\mathcal{X}} \tilde{\mu}^*\left(x'\right)\right] - \max\left(\tilde{\mu}\left(\tilde{x}^*\right), \tilde{\mu}(x)\right)\\
&\geqslant \mathbb{E}\left[\max\left(\tilde{\mu}^*\left(\tilde{x}^*\right), \tilde{\mu}^*(x)\right)\right] - \max\left(\tilde{\mu}\left(\tilde{x}^*\right), \tilde{\mu}(x)\right)\\
&= \mathbb{E}\left[\max\left(\tilde{\mu}\left(\tilde{x}^*\right) + \tilde{\sigma}\left(x, \tilde{x}^*\right)\xi, \tilde{\mu}(x) + \tilde{\sigma}\left(x, x\right)\xi\right)\right] - \max\left(\tilde{\mu}\left(\tilde{x}^*\right), \tilde{\mu}(x)\right)\\
&= \mathbb{E}\left[\max\left(a_1 + b_1\xi, a_2 + b_2\xi\right)\right] - \max\left(a_1, a_2\right)\\
&= \begin{cases}
\int_{-\infty}^{\frac{a_2-a_1}{b_1-b_2}} \left(a_2 + b_2\xi\right)\phi(\xi)d\xi + \int_{\frac{a_2-a_1}{b_1-b_2}}^{\infty} \left(a_1 + b_1\xi\right)\phi(\xi)d\xi - \max\left(a_1, a_2\right), & \text{if } b_2 \leqslant b_1\\
\int_{-\infty}^{\frac{a_2-a_1}{b_1-b_2}} \left(a_1 + b_1\xi\right)\phi(\xi)d\xi + \int_{\frac{a_2-a_1}{b_1-b_2}}^{\infty} \left(a_2 + b_2\xi\right)\phi(\xi)d\xi - \max\left(a_1, a_2\right), & \text{if } b_1 < b_2
\end{cases}\\
&= \begin{cases}
a_2\Phi\left(\frac{a_2-a_1}{b_1-b_2}\right) - b_2\phi\left(\frac{a_2-a_1}{b_1-b_2}\right) + a_1\left(1 - \Phi\left(\frac{a_2-a_1}{b_1-b_2}\right)\right) + b_1\phi\left(\frac{a_2-a_1}{b_1-b_2}\right) - \max\left(a_1, a_2\right), & \text{if } b_2 \leqslant b_1\\
a_1\Phi\left(\frac{a_2-a_1}{b_1-b_2}\right) - b_1\phi\left(\frac{a_2-a_1}{b_1-b_2}\right) + a_2\left(1 - \Phi\left(\frac{a_2-a_1}{b_1-b_2}\right)\right) + b_2\phi\left(\frac{a_2-a_1}{b_1-b_2}\right) - \max\left(a_1, a_2\right), & \text{if } b_1 < b_2
\end{cases}\\
&= a_2\Phi\left(\frac{a_2-a_1}{|b_1-b_2|}\right) + a_1\left(1 - \Phi\left(\frac{a_2-a_1}{|b_1-b_2|}\right)\right) + |b_1-b_2|\,\phi\left(\frac{a_2-a_1}{|b_1-b_2|}\right) - \max\left(a_1, a_2\right)\\
&= -|a_2-a_1|\,\Phi\left(\frac{-|a_2-a_1|}{|b_1-b_2|}\right) + |b_1-b_2|\,\phi\left(\frac{|a_2-a_1|}{|b_1-b_2|}\right),
\end{aligned}$$

where Φ is the standard normal distribution function and ϕ is its density function.

Let $g(s,t) := t\phi(s/t) - s\Phi(-s/t)$. Then 1) $g(s,t) > 0$ for all $s \geqslant 0$ and $t > 0$; 2) $g(s,t)$ is strictly decreasing in $s \in [0,\infty)$ and strictly increasing in $t \in (0,\infty)$; and 3) $g(s,t) \to 0$ as $s \to \infty$ or as $t \to 0$. See more details in [38], [44]. By letting

$x = \tilde{x}$, which is defined in (17), we have that $\liminf_{t\to\infty} |b_1 - b_2| \geqslant c'$ for some constant $c' > 0$. Meanwhile, based on Proposition 6, we have $\limsup_{t\to\infty} |a_2 - a_1| \leqslant r'$ for some constant $r' < \infty$. Thus,

$$\liminf_{t\to\infty} \alpha_{\text{KG}}(\tilde{x}) \geqslant g(r', c') > 0.$$

A similar argument is in Theorem 5.6 in [44]. On the other hand, since q-KG and KG are non-negative, we have

$$\liminf_{t\to\infty} \alpha_{\text{Co-KG}}(\mathbf{x}) \geqslant \liminf_{t\to\infty} \alpha_{\text{KG}}(x) > 0,$$

where $\tilde{\mathbf{x}}$ includes $\tilde{x}$ as one component. This provides a contradiction with Lemma 7, which proves the result in (16).

□

Based on Lemma 8, we prove Theorem 2 in the main text.

Proof. For each local GP, we have

$$\mathbb{E}\left[\tilde{\mu}_n(x) - f(x)\right]^2 = \text{Var}\left[f(x) \mid \tilde{\mathcal{S}}_n\right] \to 0$$

from Lemma 8. In addition, based on Proposition 6, we have that

$$\mathbb{P}\left\{\sup_{x\in\mathcal{X}} |\tilde{\mu}_n(x) - \mu_n^{\infty}(x)| \to 0\right\} = 1 \quad \forall n$$

as $t \to \infty$. Thus, $\mu_n^{\infty}(x) \overset{a.s.}{=} f(x)$ and

$$\mathbb{P}\left\{\sup_{x\in\mathcal{X}} |\tilde{\mu}_n(x) - f(x)| \to 0\right\} = 1 \quad \forall n$$

as $t \to \infty$. Since kernel function $K(x, x')$ is continuous, the conditional mean function $\tilde{\mu}_n(x)$, as well as $f(x)$, is continuous as well. Therefore, for each local model, the optimizer submitted satisfies that

$$\lim_{t\to\infty} f(\hat{x}_n^*) = \max_{x\in\mathcal{X}} f(x).$$

We refer to [45] for a detailed proof. In this manner, we have the consistency of the collaborative BO procedure with Co-KG (as summarized in Algorithm 1 in the main text):

$$\lim_{t\to\infty} f(\hat{x}^*) = \max_{x\in\mathcal{X}} f(x).$$

□

VIII. Additional Experiments

We present additional experiments here, including 1) the effects of different selections of β_t in the Co-KG function, 2) the effects of the discretization of the feasible set $\mathcal{X}$, 3) the comparison between the Co-KG procedure with different numbers of agents. Our experiments were conducted with Botorch [39] and Python 3.9 on a computer equipped with two AMD Ryzen Threadripper 3970X 32-Core Processors, 128 GB memory, and a Nvidia GeForce RTX A6000 GPU with 48GB of RAM. The implementation will be released once accepted.

A. Feasible Set Discretization

We discuss the impacts of feasible set discretization here. Regarding the hyperparameter, we set $\beta_t = \log(2t + 1)$. We have $N = 4$ agents. We normalize the feasible set to $\mathcal{X} = [0, 1]^2$ and discretize the feasible set using 1) 10×10, 2) 20×20, and 3) 30×30 uniform mesh grids. We also include the results associated with the parallel BO approach without data privacy concerns (q-KG) for comparison, and the procedure is indicated by "Data Communication".

Regarding the black-box optimization problem, we consider minimizing the validation loss of training a neural network. Specifically, the decision variable is the learning rate and the hidden layer node size of the neural networks. The unknown objective function is the validation loss we would minimize. The dataset is about California housing[4], where the neural network is learned to predict median value of houses in different districts given demographic attributes. We do not impose noise on the observations in the real dataset.

The experimental results are included in Figure 7, and provide the following insights: First, increasing the granularity of mesh grid enhances the performance of the collaborative BO approach with Co-KG, with more accurate central model construction and more flexible decision variable selections. Second, when the grid is precise enough (20×20 and 30×30), our approach is comparable with that without data privacy concerns, which is also supported by the experiments in the main text. Lastly,

[4]https://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.html

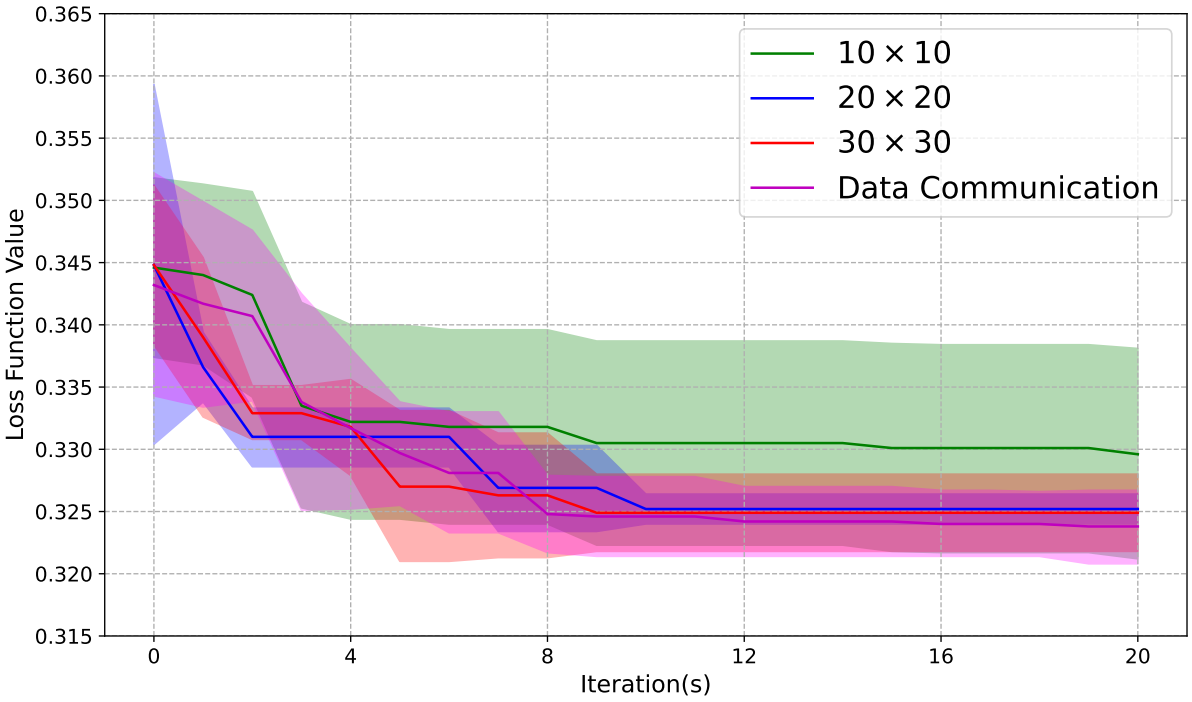

Fig. 7. Validation loss in iterations with different discretization sizes on California Housing Dataset.

although increasing the granularity from 20×20 to 30×30 enhances the performance of Co-KG, the enhancement is not significant. On the other hand, increasing the granularity significantly increases the computational burdens, we include the average running time per iteration in Table I. From the table, we observe that the time complexity for discretization is nearly $O(N^2)$, leading to heavy computational cost if we set a dense discretization for feasible set. Furthermore, we admit that the discretization could be time-varying regarding different iterations and adaptive to the collaborative optimization procedure, while the detailed discussions are left for future work.

Discretization mesh grid	Computational time (s)
10×10	2.02
20×20	6.68
30×30	16.20

TABLE I
COMPUTATIONAL TIME WITH DIFFERENT DISCRETIZATION STRATEGIES.

B. Agent Number Comparison

We compare the effects of the number of agents. The experimental settings are the same in Section VIII-A with the mesh grid fixed to be 20×20. We consider $N = 2, 4, 8$ in our experiments.

The experimental results in Figure 8 reveal the following insights. First, when the number of agents is low ($N = 2$), the performance of Co-KG is suboptimal. Second, comparing the results between $N = 4$ and $N = 8$, we observe that having more agents does not necessarily lead to better performance, especially during the initial iterations. This is because, with more agents, some may be initialized in less promising regions, which negatively impacts collaboration. The Co-KG function currently assigns equal weights to all agents, which is sensitive to suboptimal observations collected by some agents. Future work could explore assigning different weights in the collaborative acquisition function to enhance algorithm robustness. Finally, as the number of iterations increases, the procedure with $N = 8$ agents slightly outperforms that with $N = 4$, but the difference is not substantial. Both procedures tend to stabilize without significant improvements, due to the effects of discretization. This suggests that the optimal number of agents may also depend on the level of discretization, a topic that is left for future discussion.

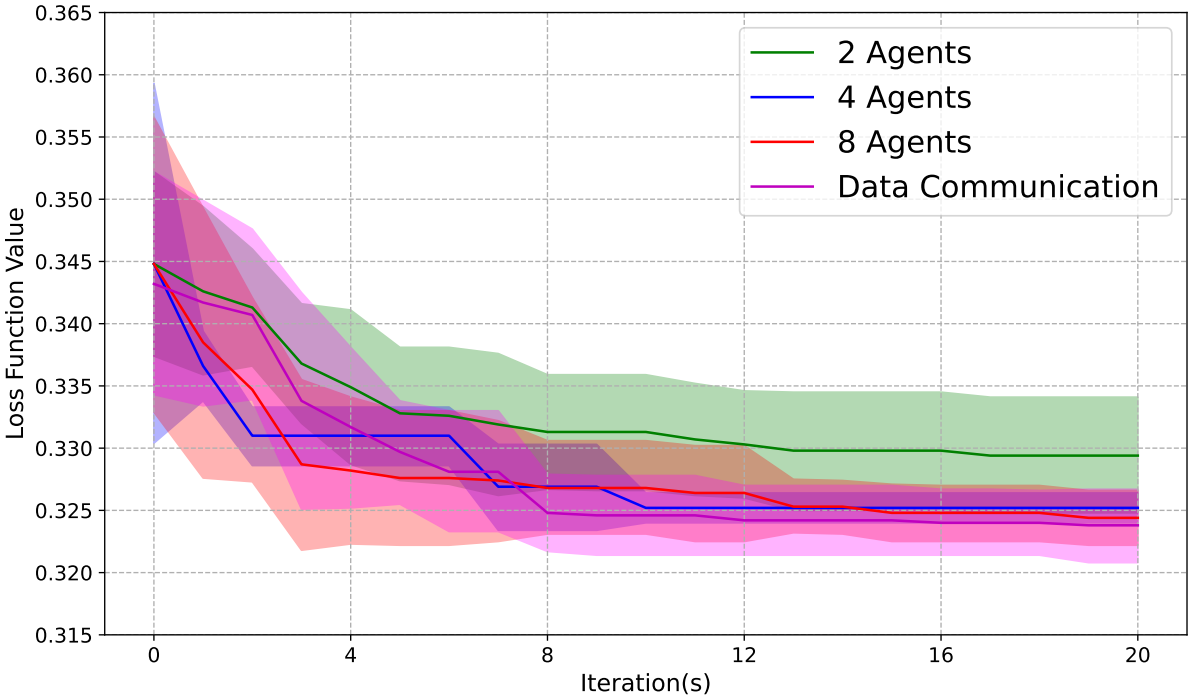

Fig. 8. Validation loss in iterations with different number of agents on California Housing Dataset.