Global Optimization with Parametric Function Approximation

Chong Liu [1] **Yu-Xiang Wang** [1]

Abstract

We consider the problem of global optimization with noisy zeroth order oracles — a well-motivated problem useful for various applications ranging from hyper-parameter tuning for deep learning to new material design. Existing work relies on Gaussian processes or other non-parametric family, which suffers from the curse of dimensionality. In this paper, we propose a new algorithm GO-UCB that leverages a parametric family of functions (e.g., neural networks) instead. Under a realizable assumption and a few other mild geometric conditions, we show that GO-UCB achieves a cumulative regret of $\tilde{O}(\sqrt{T})$ where T is the time horizon. At the core of GO-UCB is a carefully designed uncertainty set over parameters based on gradients that allows optimistic exploration. Synthetic and real-world experiments illustrate GO-UCB works better than popular Bayesian optimization approaches, even if the model is misspecified.

1. Introduction

We consider the problem of finding a global optimal solution to the following optimization problem

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

where $f : \mathcal{X} \to \mathbb{R}$ is an unknown non-convex function that is not necessarily differentiable in x.

This problem is well-motivated by many real-world applications. For example, the accuracy of a trained neural network on a validation set is complex non-convex function of a set of hyper-parameters (e.g., learning rate, momentum, weight decay, dropout, depth, width, choice of activation functions ...) that one needs to maximize (Kandasamy et al., 2020). Also in material design, researchers want to synthesize ceramic materials, e.g., titanium dioxide (TiO_2) thin films, using microwave radiation (Nakamura et al., 2017) where the film property is a non-convex function of parameters including temperature, solution concentration, pressure, and processing time. Efficiently solving such non-convex optimization problems could significantly reduce energy cost.

We assume having access to only noisy function evaluations, i.e., at round t, we select a point x_t and receive a noisy function value y_t,

$$y_t = f(x_t) + \eta_t, \tag{1}$$

where η_t for $t = 1, ..., T$ are *independent*, *zero-mean*, *σ-sub-Gaussian* noise. This is known as the *noisy zeroth-order oracle* setting in optimization literature. Let f^* be the optimal function value, following the tradition of Bayesian optimization (see e.g., Frazier (2018) for a review), throughout this paper, we use *cumulative regret* as the evaluation criterion, defined as

$$R_T = \sum_{t=1}^{T} r_t = \sum_{t=1}^{T} f^* - f(x_t),$$

where r_t is called instantaneous regret at round t. An algorithm $\mathcal{A}$ is said to be a no-regret algorithm if $\lim_{T\to\infty} R_T(\mathcal{A})/T = 0$.

Generally speaking, solving a global non-convex optimization is NP-hard (Jain et al., 2017) and we need additional assumptions to efficiently proceed. Bayesian optimization usually assumes the objective function f is drawn from a Gaussian process prior. Srinivas et al. (2010) proposed the GP-UCB approach, which iteratively queries the argmax of an upper confidence bound of the current posterior belief, before updating the posterior belief using the new data point. However, Gaussian process relies on kernels, e.g., squared error kernel or Matérn kernel, which suffer from the curse of dimensionality. A folklore rule-of-thumb is that GP-UCB becomes unwieldy when the dimension is larger than 10.

A naive approach is to passively query T data points uniformly at random, estimate f by $\hat{f}$ using supervised learning, then return the maximizer of the plug-in estimator $\hat{x} = \mathrm{argmax}_{x\in\mathcal{X}} \hat{f}(x)$. This may side-step the curse-of-dimensionality depending on which supervised learning

[1]Department of Computer Science, University of California, Santa Barbara, CA 93106, USA. Correspondence to: Chong Liu <chongliu@cs.ucsb.edu>, Yu-Xiang Wang <yuxiangw@cs.ucsb.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

model we use. The drawback of this passive query model is that it does not consider the structure of the function nor does it quickly "zoom-in" to the region of the space that is nearly optimal. In contrast, an active query model allows the algorithm to iteratively interact with the function. At round t, the model collects information from all previous rounds $1, ..., t-1$ and decides where to query next.

GO-UCB Algorithm. In this paper, we develop an algorithm that allows Bayesian optimization-style active queries to work for general supervised learning-based function approximation. We assume that the supervised learning model $f_w : \mathcal{X} \to \mathbb{R}$ is differentiable w.r.t. its d_w-dimensional parameter vector $w \in \mathcal{W} \subset \mathbb{R}^{d_w}$ and that the function class $\mathcal{F} = \{f_w | w \in \mathcal{W}\}$ is flexible enough such that the true objective function $f = f_{w^*}$ for some $w^* \in \mathcal{W}$, i.e., $\mathcal{F}$ is *realizable*. Our algorithm — *Global Optimization via Upper Confidence Bound* (GO-UCB) — has two phases:

The *GO-UCB* Framework:

- Phase I: Uniformly explore n data points.
- Phase II: Optimistically explore T data points.

The goal of Phase I to sufficiently explore the function and make sure the estimated parameter $\hat{w}_0$ is close enough to true parameter w^* such that exploration in Phase II are efficient. To solve the estimation problem, we rely on a regression oracle that is able to return an estimated $\hat{w}_0$ after n observations. In details, after Phase I we have a dataset $\{(x_j, y_j)\}_{j=1}^n$, then

$$\hat{w}_0 \leftarrow \operatorname*{argmin}_{w \in \mathcal{W}} \sum_{j=1}^{n} (f_w(x_j) - y_j)^2. \tag{2}$$

This problem is known as a *non-linear least square* problem. It is computationally hard in the worst-case, but many algorithms are known (e.g., SGD, Gauss-Newton, Levenberg-Marquardt) to effectively solve this problem in practice. Our theoretical analysis of $\hat{w}_0$ uses techniques from Nowak (2007). See Section 5.1 for details.

In Phase II, exploration is conducted following the principle of "Optimism in the Face of Uncertainty", i.e., the parameter is optimized within an uncertainty region that always contains the true parameter w^*. Existing work in bandit algorithms provides techniques that work when f_w is a linear function (Abbasi-yadkori et al., 2011) or a generalized linear function (Li et al., 2017), but no solution to general differentiable function is known. At the core of our GO-UCB is a carefully designed uncertainty ball Ball_t over parameters based on gradients, which allows techniques from the linear bandit (Abbasi-yadkori et al., 2011) to be adapted for the non-linear case. In detail, the ball is defined to be centered at $\hat{w}_t$ — the solution to a regularized online regression problem after $t-1$ rounds of observations. And the radius of the ball is measured by the covariance matrix of the gradient vectors of all previous rounds. We prove that w^* is always trapped within the ball with high probability.

Contributions. In summary, our main contributions are:

1. We initiate the study of global optimization problem with parametric function approximation and proposed a new optimistic exploration algorithm — GO-UCB.
2. Assuming *realizability* and other mild geometric conditions, we prove that GO-UCB converges to the global optima with cumulative regret at the order of $\tilde{O}(\sqrt{T})$ where T is the time horizon.
3. GO-UCB does not suffer from the curse of dimensionality like Gaussian processes-based Bayesian optimization methods. The unknown objective function f can be high-dimensional, non-convex, non-differentiable, and even discontinuous in its input domain.
4. Synthetic test function and real-world hyperparameter tuning experiments show that GO-UCB works better than all compared Bayesian optimization methods in both realizable and misspecified settings.

Technical novelties. The design of GO-UCB algorithm builds upon the work of Abbasi-yadkori et al. (2011) and Agarwal et al. (2021), but requires substantial technical novelties as we handle a generic nonlinear parametric function approximation. Specifically:

1. LinUCB analysis (e.g., self-normalized Martingale concentration, elliptical potential lemmas (Abbasi-yadkori et al., 2011; Agarwal et al., 2021)) is not applicable for nonlinear function approximation, but we showed that they can be adapted for this purpose if we can *localize* the learner to a neighborhood of w^*.
2. We identify a new set of structural assumptions under which we can localize the learner sufficiently with only $O(\sqrt{T})$ rounds of pure exploration.
3. Showing that w^* remains inside the parameter uncertainty ball $\mathrm{Ball}_t, \forall t \in [T]$ is challenging. We solve this problem by setting regularization centered at the initialization parameter $\hat{w}_0$ and presenting novel inductive proof of a lemma showing $\forall t \in [T], \hat{w}_t$ converges to w^* in ℓ_2-distance at the same rate.

These new techniques could be of independent interest.

2. Related Work

Global non-convex optimization is an important problem that can be found in a lot of research communities and

real-world applications, e.g., optimization (Rinnooy Kan & Timmer, 1987a;b), machine learning (Bubeck et al., 2011; Malherbe & Vayatis, 2017), hyperparameter tuning (Hazan et al., 2018), neural architecture search (Kandasamy et al., 2018; Wang et al., 2020), and material discovery (Frazier & Wang, 2016).

One of the most prominent approaches to this problem is Bayesian Optimization (BO) (Shahriari et al., 2015), in which the objective function is usually modeled by a Gaussian Process (GP) (Williams & Rasmussen, 2006), so that the uncertainty can be updated under the Bayesian formalism. Among the many notable algorithms in GP-based BO (Srinivas et al., 2010; Jones et al., 1998; Bull, 2011; Frazier et al., 2009; Agrawal & Goyal, 2013; Cai & Scarlett, 2021), GP-UCB (Srinivas et al., 2010) is the closest to our paper because our algorithm also selects data points in a UCB (upper confidence bound) style but the construction of the UCB in our paper is different since we are not working with GPs. Scarlett et al. (2017) proves lower bounds on regret for noisy Gaussian process bandit optimization. GPs are highly flexible and can approximate any smooth functions, but such flexibility comes at a price to play — curse of dimensionality. Most BO algorithms do not work well when $d > 10$. Notable exceptions include the work of Shekhar & Javidi (2018); Calandriello et al. (2019); Eriksson et al. (2019); Salgia et al. (2021); Rando et al. (2022) who designed more specialized BO algorithms for high-dimensional tasks.

Besides BO with GPs, other nonparametric families were considered for global optimization tasks, but they, too, suffer from the curse of dimensionality. We refer readers to Wang et al. (2018) and the references therein.

While most BO methods use GP as surrogate models, there are other BO methods that use alternative function classes such as neural networks (Snoek et al., 2015; Springenberg et al., 2016). These methods are different from us in that they use different ways to fit the neural networks and a Monte Carlo sampling approach to decide where to explore next. Empirically, it was reported that they do not outperform advanced GP-based methods that use trust regions (Eriksson et al., 2019).

Our problem is also connected to the bandits literature (Li et al., 2019; Foster & Rakhlin, 2020; Russo & Van Roy, 2013; Filippi et al., 2010). The global optimization problem can be written as a nonlinear bandits problem in which queried points are actions and the function evaluations are rewards. However, no bandits algorithms can simultaneously handle an infinite action space and a generic nonlinear reward function. Here "generic" means the reward function is much more general than a linear or generalized linear function (Filippi et al., 2010). To the best of our knowledge, we are the first to address the infinite-armed bandit problems with a general differentiable value function (albeit with some additional assumptions).

A recent line of work studied bandits and global optimization with neural function approximation (Zhou et al., 2020; Zhang et al., 2020; Dai et al., 2022). The main difference from us is that these results still rely on Gaussian processes with a Neural Tangent Kernel in their analysis, thus intrinsically linear. Their regret bounds also require the width of the neural network to be much larger than the number of samples to be sublinear. In contrast, our results apply to general nonlinear function approximations and do not require overparameterization.

3. Preliminaries

3.1. Notations

We use $[n]$ to denote the set $\{1, 2, ..., n\}$. The algorithm queries n points in Phase I and T points in Phase II. Let $\mathcal{X} \subset \mathbb{R}^{d_x}$ and $\mathcal{Y} \subset \mathbb{R}$ denote the domain and range of f, and $\mathcal{W} \subset [0, 1]^{d_w}$ denote the parameter space of a family of functions $\mathcal{F} := \{f_w : \mathcal{X} \to \mathcal{Y} | w \in \mathcal{W}\}$. For convenience, we denote the bivariate function $f_w(x)$ by $f_x(w)$ when w is the variable of interest. $\nabla f_x(w)$ and $\nabla^2 f_x(w)$ denote the gradient and Hessian of function f w.r.t. w. $L(w) := \mathbb{E}_{x\sim\mathcal{U}}(f_x(w) - f_x(w^*))^2$ denotes the (expected) risk function where $\mathcal{U}$ is uniform distribution. For a vector x, its ℓ_p norm is denoted by $\|x\|_p = (\sum_{i=1}^d |x_i|^p)^{1/p}$ for $1 \le p < \infty$ and its ℓ_∞ norm is denoted by $\|x\|_\infty = \max_{i\in[d_x]} |x_i|$. For a matrix A, its operator norm is denoted by $\|A\|_{\text{op}}$. For a vector x and a square matrix A, define $\|x\|_A^2 = x^\top Ax$. Throughout this paper, we use standard big O notation that hide universal constants; and to improve the readability, we use $\tilde{O}$ to hide all logarithmic factors as well as all polynomial factors in problem-specific parameters except $d_w, 1/\mu, T$. For reader's easy reference, we list all symbols and notations in Appendix A.

3.2. Assumptions

Here we list main assumptions that we will work with throughout this paper. The first assumption says that we have access to a differentiable function family that contains the unknown objective function.

Assumption 3.1 (Realizability)**.** There exists $w^* \in \mathcal{W}$ such that the unknown objective function $f = f_{w^*}$. Also, assume $\mathcal{W} \subset [0, 1]^{d_w}$. This is w.l.o.g. for any compact $\mathcal{W}$.

Realizable parameter class is a common assumption in literature (Chu et al., 2011; Foster et al., 2018; Foster & Rakhlin, 2020), usually the starting point of a line of research for a new problem because one doesn't need to worry about extra regret incurred by misspecified parameter. Although in this paper we only theoretically study the realizable parameter class, our GO-UCB algorithm empirically works well in

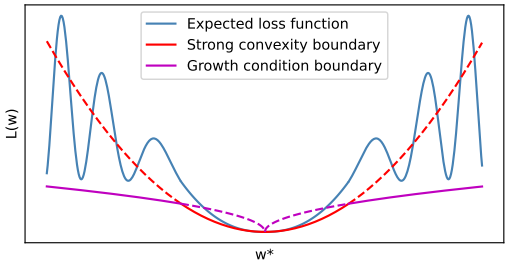


Figure 1. Example of a highly non-convex $L(w)$ satisfying Assumption 3.3. Solid lines denote the actual lower bound by taking min over strong convexity and growth condition. $L(w)$ is strongly convex near w^* but can be highly non-convex away from w^*.

misspecified tasks too.

The second assumption is on properties of the function approximation.

Assumption 3.2 (Bounded, differentiable and smooth function approximation)**.** There exist constants $F, C_g, C_h > 0$ such that $\forall x \in \mathcal{X}, \forall w \in \mathcal{W}$, it holds that $|f_x(w)| \leq F$,

$$\|\nabla f_x(w)\|_2 \leq C_g, \quad \text{and} \quad \|\nabla^2 f_x(w)\|_{\text{op}} \leq C_h.$$

This assumption imposes mild regularity conditions on the smoothness of the function with respect to its parameter vector w.

The third assumption is on the expected loss function over the uniform distribution (or any other exploration distribution) in the Phase I of GO-UCB.

Assumption 3.3 (Geometric conditions on the loss function)**.** $L(w) = \mathbb{E}_{x\sim\mathcal{U}}(f_x(w) - f_x(w^*))^2$ satisfies (τ, γ)-*growth condition* or μ-*local strong convexity* at w^*, i.e., $\forall w \in \mathcal{W}$,

$$\min\left\{\frac{\mu}{2}\|w - w^*\|_2^2, \frac{\tau}{2}\|w - w^*\|_2^\gamma\right\} \leq L(w) - L(w^*),$$

for constants $\mu, \tau > 0, \mu < d_w$ and $0 < \gamma < 2$. Also, $L(w)$ satisfies a c-*local self-concordance* assumption at w^*, i.e., for all w s.t. $\|w - w^*\|_{\nabla^2 L(w^*)} \leq c$,

$$(1-c)^2 \cdot \nabla^2 L(w^*) \preceq \nabla^2 L(w) \preceq (1-c)^{-2} \cdot \nabla^2 L(w^*).$$

We also assume $c \leq 0.5$ for convenience. This is without loss of generality because if the condition holds for $c > 0.5$, then the condition for $c \leq 0.5$ is automatically satisfied.

This assumption has three main components: (global) growth condition, local strong convexity, and local self-concordance.

The global growth condition says that f_w with parameters far away from w^* cannot approximate f well over the distribution $\mathcal{U}$. The local strong convexity assumption requires the local neighborhood near w^* to have quadratic growth.

These two conditions are strictly weaker than global strong convexity because it does not require convexity except in a local neighborhood near the global optimal w^*, i.e., $\{w | \|w - w^*\|_2 \leq (\tau/\mu)^{\frac{1}{2-\gamma}}\}$ and it does not limit the number of spurious local minima, as the global γ-growth condition only gives a mild lower bound as w moves away from w^*. See Figure 1 for an example. Our results work even if γ is a small constant < 1.

Self-concordance originates from a clean analysis of Newton's method (Nesterov & Nemirovskii, 1994). See Example 4 of Zhang et al. (2017) for a list of examples satisfying self-concordance. A localized version of self-concordance is needed in our problem for technical reasons, but again it is only required within a small ball of radius c near w^* for the expected loss under $\mathcal{U}$. Our results work even if c vanishes at $O(T^{-1/4})$.

To avoid any confusion, the three assumptions we made above are only about the expected loss function w.r.t. uniform distribution $\mathcal{U}$ as a function of w, rather than objective function $f_{w^*}(x)$. The problem to be optimized can still be arbitrarily complex in terms of $\mathcal{X}$, e.g., high-dimensional and non-continuous functions. As an example, in Gaussian process-based Bayesian optimization approaches, $f_{w^*}(x)$ belongs to a reproducing kernel Hilbert space, but its loss function is globally convex in its "infinite dimensional" parameter w. Also, we no longer need this assumption in Phase II.

Additional notations. For convenience, we define $\zeta > 0$ such that $\|\nabla^2 L(w^*)\|_{\text{op}} \leq \zeta$. The existence of a finite ζ is implied by Assumption 3.2 and it suffices to take $\zeta = 2C_g^2$ because $\nabla^2 L(w^*) = \mathbb{E}_{x\sim\mathcal{U}}[2\nabla f_x(w^*)\nabla f_x(w^*)^\top]$.

4. Main Results

In Section 4.1, we state our Global Optimization with Upper Confidence Bound (GO-UCB) algorithm and explain key design points of it. Then in Section 4.2, we prove that its cumulative regret bound is at the rate of $\tilde{O}(\sqrt{T})$.

4.1. Algorithm

Our GO-UCB algorithm, shown in Algorithm 1, has two phases. Phase I does uniform exploration in n rounds and Phase II does optimistic exploration in T rounds. In Step 1 of Phase I, n is chosen to be large enough such that the objective function can be sufficiently explored. Step 2-3 are doing uniform sampling. In Step 5, we call regression oracle to estimate $\hat{w}_0$ given all observations in Phase I as in eq. (2). Adapted from Nowak (2007), we prove the convergence rate of $\|\hat{w}_0 - w^*\|_2$ is at the rate of $\tilde{O}(1/\sqrt{n})$. See Theorem 5.2 for details.

The key challenge of Phase II of GO-UCB is to design

an acquisition function to select $x_t, \forall t \in [T]$. Since we are using parametric function to approximate the objective function, we heavily rely on a feasible parameter uncertainty region $\text{Ball}_t, \forall t \in [T]$, which should always contain the true parameter w^* throughout the process. The shape of Ball_t is measured by the covariance matrix Σ_t, defined as

$$\Sigma_t = \lambda I + \sum_{i=0}^{t-1} \nabla f_{x_i}(\hat{w}_i) \nabla f_{x_i}(\hat{w}_i)^\top. \tag{3}$$

Note i is indexing over both x and w, which means that as time t goes from 0 to T, the update to Σ_t is always rank one. It allows us to bound the change of Σ_t from $t = 0$ to T.

Ball_t is centered at $\hat{w}_t$, the newly estimated parameter at round t. In Step 2, we update the estimated $\hat{w}_t$ by solving the following optimization problem:

$$\hat{w}_t = \operatorname*{argmin}_w \frac{\lambda}{2} \|w - \hat{w}_0\|_2^2 + \frac{1}{2} \sum_{i=0}^{t-1} ((w - \hat{w}_i)^\top \nabla f_{x_i}(\hat{w}_i) + f_{x_i}(\hat{w}_i) - y_i)^2. \tag{4}$$

The optimization problem is an online regularized least square problem involving gradients from all previous rounds, i.e., $\nabla f_{x_i}(\hat{w}_i), \forall i \in [T]$. The intuition behind it is that we use gradients to approximate the function since we are dealing with generic objective function. We set the regularization w.r.t. $\hat{w}_0$ rather than 0 because from regression oracle we know how close is $\hat{w}_0$ to w^*. By setting the gradient of objective function in eq. (4) to be 0, the closed form solution of $\hat{w}_t$ is

$$\hat{w}_t = \Sigma_t^{-1} \left(\sum_{i=0}^{t-1} \nabla f_{x_i}(\hat{w}_i)(\nabla f_{x_i}(\hat{w}_i)^\top \hat{w}_i + y_i - f_{x_i}(\hat{w}_i)) \right) + \lambda \Sigma_t^{-1} \hat{w}_0. \tag{5}$$

Now we move to our definition of Ball_t, shown as

$$\text{Ball}_t = \{w : \|w - \hat{w}_t\|_{\Sigma_t}^2 \le \beta_t\}, \tag{6}$$

where β_t is a pre-defined monotonically increasing sequence that we will specify later. Following the "optimism in the face of uncertainty" idea, our ball is centered at $\hat{w}_t$ with β_t being the radius and Σ_t measuring the shape. β_t ensures that the true parameter w^* is always contained in Ball_t w.h.p. In Section 5.2, we will show that it suffices to choose

$$\beta_t = \tilde{O}\left(d_w \sigma^2 + \frac{d_w^3}{\mu^2} + \frac{d_w^3 t}{\mu^2 T} \right), \tag{7}$$

where $\tilde{O}$ hides logarithmic terms in t, T and $1/\delta$ (w.p. $1-\delta$).

Then in Step 5 of Phase II, x_t is selected by joint optimization over $x \in \mathcal{X}$ and $w \in \text{Ball}_t$. Finally, we collect all observations in T rounds and output $\hat{x}$ by uniformly sampling over $\{x_1, ..., x_T\}$.

Algorithm 1 GO-UCB

Input: Time horizon T, uniform exploration phase length n, uniform distribution $\mathcal{U}$, regression oracle Oracle, regularization weight λ, confidence sequence β_t for $t = 1, 2, ..., T$.

Phase I (Uniform exploration)

1: **for** $j = 1, ..., n$ **do**
2: Sample $x_j \sim \mathcal{U}(\mathcal{X})$.
3: Observe $y_j = f(x_j) + \eta_j$.
4: **end for**
5: Estimate $\hat{w}_0 \leftarrow \text{Oracle}(x_1, y_1, ..., x_n, y_n)$.

Phase II (Optimistic exploration)

1: **for** $t = 1, ..., T$ **do**
2: Update Σ_t by eq. (3) with the input λ.
3: Update $\hat{w}_t$ by eq. (5) with the input λ.
4: Update Ball_t by eq. (6) with the input β_t.
5: Select $x_t = \operatorname{argmax}_{x \in \mathcal{X}} \max_{w \in \text{Ball}_t} f_x(w)$.
6: Observe $y_t = f(x_t) + \eta_t$.
7: **end for**

Output: $\hat{x} \sim \mathcal{U}(\{x_1, ..., x_T\})$.

4.2. Regret Upper Bound

Now we present the cumulative regret upper bound of GO-UCB algorithm.

Theorem 4.1 (Cumulative regret of GO-UCB)**.** *Suppose Assumption 3.1, 3.2, & 3.3 hold with parameters $F, C_g, C_h, \zeta, \mu, \gamma, \tau, c$. Assume*

$$T > C d_w^2 F^4 \iota^2 \cdot \max\left\{ \frac{\mu^{\gamma/(2-\gamma)}}{\tau^{2/(2-\gamma)}}, \frac{\zeta}{\mu c^2} \right\}^2, \tag{8}$$

where C is a universal constant and ι is a logarithmic term depending on $n, C_h, 2/\delta$ (both of them from Theorem 5.2). Then Algorithm 1 with parameters $n = \sqrt{T}$, $\lambda = C_\lambda \sqrt{T}$ (for a C_λ logarithmically dependent to T and polynomial in all other parameters) and $\beta_{1:T}$ as in eq. (7) obeys that with probability at least $1 - \delta$,

$$R_{\sqrt{T}+T} = \tilde{O}\left(\sqrt{T} F + \sqrt{T \beta_T d_w + \frac{T \beta_T^2}{\lambda^2}} \right) = \tilde{O}\left(\frac{d_w^2 \sqrt{T}}{\mu} \right).$$

Let us highlight a few interesting aspects of the result.

Remark 4.2. Without Gaussian process assumption, we propose the first algorithm to solve global optimization problem

with $\tilde{O}(\sqrt{T})$ cumulative regret, which is *dimension-free* in terms of its input domain $\mathcal{X}$. GO-UCB is a no-regret algorithm since $\lim_{T\to\infty} R_T/T = 0$, and the output $\hat{x}$ satisfies that $f^* - \mathbb{E}[f(\hat{x})] \leq \tilde{O}(1/\sqrt{T})$, which is also knowns as expected simple regret upper bound. The dependence in T is optimal up to logarithmic factors, as it matches the lower bound for linear bandits (Dani et al., 2008, Theorem 3).

Remark 4.3 (Choice of λ). One important deviation from the classical linear bandit analysis is that we require a regularization that centers around $\hat{w}_0$ and the regularization weight λ to be $C_\lambda\sqrt{T}$, comparing to $\lambda = O(1)$ in the linear case. The choice is to ensure that $\hat{w}_t$ stays within the local neighborhood of $\hat{w}_0$, and to delicately balance different terms that appear in the regret analysis to ensure that the overall regret bound is $\tilde{O}(\sqrt{T})$.

Remark 4.4 (Choice of n). We choose $n = \sqrt{T}$, therefore, it puts sample complexity requirement on T shown in eq. (8). The choice of n plays two roles here. First, it guarantees that the regression result $\hat{w}_0$ lies in the neighboring region of w^* of the loss function $L(w)$ with high probability. The neighboring region of w^* has nice properties, e.g., local strong convexity, which allow us to build the upper bound of ℓ_2-distance between $\hat{w}_0$ and w^*. Second, in Phase I, we are doing uniform sampling over the function so the cumulative regret in Phase I is bounded by $2Fn = 2F\sqrt{T}$ which is at the same $\tilde{O}(\sqrt{T})$ rate as that in Phase II.

5. Proof Overview

In this section, we give a proof sketch of all theoretical results. A key insight of our analysis is that there is more mileage that seminal techniques developed by Abbasi-yadkori et al. (2011) for analyzing linearly parameterized bandits problems in analyzing non-linear bandits, though we need to localize to a nearly optimal region and carefully handle the non-linear components via more aggressive regularization. Other assumptions that give rise to a similarly good initialization may work too and our new proof can be of independent interest in analyzing other extensions of LinUCB, e.g., to contextual bandits, reinforcement learning and other problems.

In detail, first we prove the estimation error bound of $\hat{w}_0$ for Phase I of GO-UCB algorithm, then prove the feasibility of Ball_t. Finally by putting everything together we prove the cumulative regret bound of GO-UCB algorithm. Due to page limit, we list all auxiliary lemmas in Appendix B and show complete proofs in Appendix C.

5.1. Regression Oracle Guarantee

The goal of Phase I of GO-UCB is to sufficiently explore the unknown objective function with n uniform queries and obtain an estimated parameter $\hat{w}_0$. By assuming access to a regression oracle, we prove the convergence bound of $\hat{w}_0$ w.r.t. w^*, i.e., $\|\hat{w}_0 - w^*\|_2^2$. To get started, we need the following regression oracle lemma.

Lemma 5.1 (Adapted from Nowak (2007))**.** *Suppose Assumption 3.1 & 3.2 hold. There is an absolute constant C', such that after round n in Phase I of Algorithm 1, with probability $> 1 - \delta/2$, regression oracle estimated $\hat{w}_0$ satisfies*

$$\mathbb{E}_{x\sim\mathcal{U}}[(f_x(\hat{w}_0) - f_x(w^*))^2] \leq \frac{C' d_w F^2 \iota}{n},$$

where ι is the logarithmic term depending on $n, C_h, 2/\delta$.

Nowak (2007) proves that expected square error of Empirical Risk Minimization (ERM) estimator can be bounded at the rate of $\tilde{O}(1/n)$ with high probability, rather than $\tilde{O}(1/\sqrt{n})$ rate achieved by Chernoff/Hoeffding bounds. It works with realizable and misspecified settings. Proof of Lemma 5.1 includes simplifying it with regression oracle, Assumption 3.1, and ε-covering number argument over parameter class. Basically Lemma 5.1 says that expected square error of $f_x(\hat{w}_0)$ converges to $f_x(w^*)$ at the rate of $\tilde{O}(1/n)$ with high probability. Based on it, we prove the following regression oracle guarantee.

Theorem 5.2 (Regression oracle guarantee)**.** *Suppose Assumption 3.1, 3.2, & 3.3 hold. There is an absolute constant C such that after round n in Phase I of Algorithm 1 where n satisfies $n \geq C d_w F^2 \iota \cdot \max\left\{\frac{\mu^{\gamma/(2-\gamma)}}{\tau^{2/(2-\gamma)}}, \frac{\zeta}{\mu c^2}\right\}$, with probability $> 1 - \delta/2$, regression oracle estimated $\hat{w}_0$ satisfies*

$$\|\hat{w}_0 - w^*\|_2^2 \leq \frac{C d_w F^2 \iota}{\mu n},$$

where ι is the logarithmic term depending on $n, C_h, 2/\delta$.

Compared with Lemma 5.1, there is an extra sample complexity requirement on n because we need n to be sufficiently large such that the function can be sufficiently explored and more importantly $\hat{w}_0$ falls into the neighboring region (strongly convex region) of w^*. See Figure 1 for illustration. It is also the reason why strong convexity parameter μ appears in the denominator of the upper bound.

5.2. Feasibility of Ball_t

The following lemma is the key part of algorithm design of GO-UCB. It says that our definition of Ball_t is appropriate, i.e., throughout all rounds in Phase II, w^* is contained in Ball_t with high probability.

Lemma 5.3 (Feasibility of Ball_t)**.** *Set $\Sigma_t, \hat{w}_t$ as in eq. (3), (5). Set β_t as*

$$\beta_t = \tilde{O}\left(d_w\sigma^2 + \frac{d_w^3}{\mu^2} + \frac{d_w^3 t}{\mu^2 T}\right). \tag{9}$$

Suppose Assumption 3.1, 3.2, & 3.3 hold and choose $n = \sqrt{T}, \lambda = C_\lambda\sqrt{T}$. Then $\forall t \in [T]$ in Phase II of Algorithm 1,

w.p. $> 1-\delta$,

$$\|\hat{w}_t - w^*\|^2_{\Sigma_t} \leq \beta_t.$$

For reader's easy reference, we write our choice of β_t again in eq. (9). Note this lemma requires careful choices of λ and n because β_t appears later in the cumulative regret bound and β_t is required to be at the rate of $\tilde{O}(1)$. The proof has three steps. First we obtain the closed form solution of $\hat{w}_t$ as in eq. (5). Next we use induction to prove that $\forall t \in [T], \|\hat{w}_t - w^*\|_2^2 \leq \tilde{O}(\tilde{C}/n)$ for some universal constant $\tilde{C}$. Finally we prove $\|\hat{w}_t - w^*\|^2_{\Sigma_t} \leq \beta_t$.

5.3. Regret Analysis

To prove cumulative regrets bound of GO-UCB algorithm, we need following two lemmas of instantaneous regrets in Phase II of GO-UCB.

Lemma 5.4 (Instantaneous regret bound)**.** *Set $\Sigma_t, \hat{w}_t, \beta_t$ as in eq. (3), (5), & (7) and suppose Assumption 3.1, 3.2, & 3.3 hold, then with probability $> 1-\delta$, w^* is contained in Ball_t. Define $u_t = \|\nabla f_{x_t}(\hat{w}_t)\|_{\Sigma_t^{-1}}$, then $\forall t \in [T]$ in Phase II of Algorithm 1,*

$$r_t \leq 2\sqrt{\beta_t} u_t + \frac{2\beta_t C_h}{\lambda}.$$

The first term of the upper bound is pretty standard, seen also in LinUCB (Abbasi-yadkori et al., 2011) and GP-UCB (Srinivas et al., 2010). After we apply first order gradient approximation of the objective function, the second term is the upper bound of the high order residual term, which introduces extra challenge to derive the upper bound.

Technically, proof of Lemma 5.4 requires w^* is contained in our parameter uncertainty ball Ball_t with high probability throughout Phase II of GO-UCB, which has been proven in Lemma 5.3. Later, the proof utilizes Taylor's theorem and uses the convexity of Ball_t twice. See Appendix C.4. The next lemma is an extension of Lemma 5.4, where the proof uses monotonically increasing property of β_t in t.

Lemma 5.5 (Summation of squared instantaneous regret bound)**.** *Set $\Sigma_t, \hat{w}_t, \beta_t$ as in eq. (3), (5), & (7) and suppose Assumption 3.1, 3.2, & 3.3 hold, then with probability $> 1-\delta$, w^* is contained in Ball_t and $\forall t \in [T]$ in Phase II of Algorithm 1,*

$$\sum_{t=1}^{T} r_t^2 \leq 16\beta_T d_w \log\left(1 + \frac{TC_g^2}{d_w \lambda}\right) + \frac{8\beta_T^2 C_h^2 T}{\lambda^2}.$$

Proof of Theorem 4.1 follows by putting everything together via Cauchy-Shwartz inequality $\sum_{t=1}^{T} r_t \leq \sqrt{T\sum_{t=1}^{T} r_t^2}$.

6. Experiments

We compare our GO-UCB algorithm with four Bayesian Optimization (BO) algorithms: GP-EI (Jones et al., 1998), GP-PI (Kushner, 1964), GP-UCB (Srinivas et al., 2010), and Trust Region BO (TuRBO) (Eriksson et al., 2019), where the first three are classical methods and TuRBO is a more advanced algorithm designed for high-dimensional cases.

To run GO-UCB, we choose our parametric function model $\hat{f}$ to be a two linear layer neural network with `sigmoid` function being the activation function:

$$\hat{f}(x) = \texttt{linear2}(\texttt{sigmoid}(\texttt{linear1}(x))),$$

where w_1, b_1 denote the weight and bias of `linear1` layer and w_2, b_2 denote those of `linear2` layer. Specifically, we set $w_1 \in \mathbb{R}^{25\times d_x}, b_1 \in \mathbb{R}^{25}, w_2 \in \mathbb{R}^{25}, b_2 \in \mathbb{R}$, meaning the dimension of activation function is 25. All implementations are based on BoTorch framework (Balandat et al., 2020) and sklearn package (Head et al., 2021) with default parameter settings. To help readers reproduce our results, implementation details are shown in Appendix D.1.

6.1. Synthetic Experiments

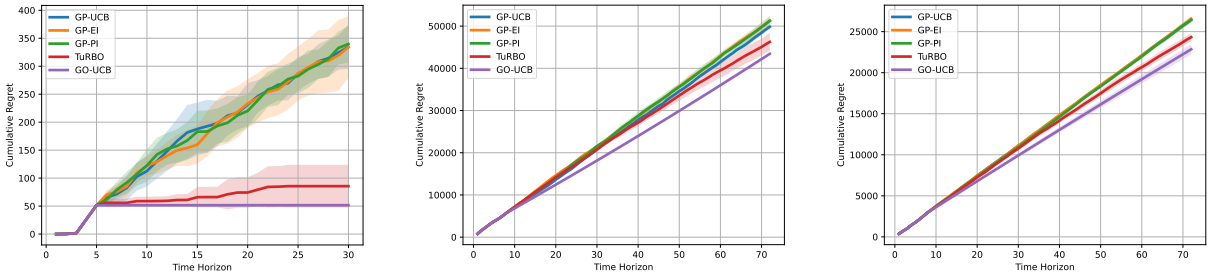
First, we test all algorithms on three high-dimensional synthetic functions defined on $[-5,5]^{d_x}$ where $d_x = 20$, including both realizable and misspecified cases. The first test function f_1 is created by setting all elements in w_1, b_1, w_2, b_2 in $\hat{f}$ to be 1, so f_1 is a realizable function given $\hat{f}$. The second and third test functions f_2, f_3 are Styblinski-Tang function and Rastrigin function, defined as:

$$f_2 = -\frac{1}{2}\sum_{i=1}^{20} x_i^4 - 16x_i^2 + 5x_i,$$

$$f_3 = -200 + \sum_{i=1}^{20} 10\cos(2\pi x_i) - x_i^2,$$

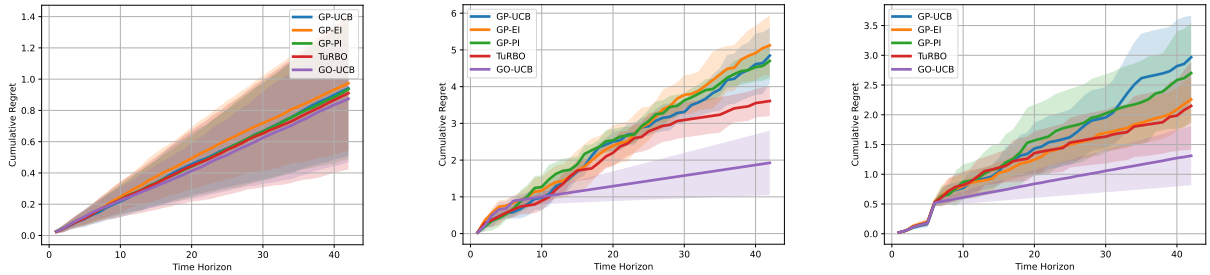
where x_i denotes the i-th element in its 20 dimensions, so f_2, f_3 are misspecified functions given $\hat{f}$. We set $n = 5, T = 25$ for f_1 and $n = 8, T = 64$ for f_2, f_3. To reduce the effect of randomness in all algorithms, we repeat the whole optimization process for 5 times for all algorithms and report mean and error bar of cumulative regrets. The error bar is measured by Wald's test with 95% confidence, i.e., $1.96\nu/\sqrt{5}$ where ν is standard deviation of cumulative regrets and 5 is the number of repetitions.

From Figure 2, we learn that in all tasks our GO-UCB algorithm performs better than all other four BO approaches. Among BO approches, TuRBO performs the best since it is specifically designed for high-dimensional tasks. In Figure 2(a), mean of cumulative regrets of GO-UCB and TuRBO stays the same when $t \geq 22$, which means that both of them have found the global optima, but GO-UCB algorithm is



(a) f_1 (realizable) (b) f_2 (misspecified) (c) f_3 (misspecified)

Figure 2. Cumulative regrets (the lower the better) of all algorithms on 20-dimensional f_1, f_2, f_3 synthetic functions.



(a) Random forest ($d_x = 7$) (b) Multi-layer perceptron ($d_x = 8$) (c) Gradient boosting ($d_x = 11$)

Figure 3. Cumulative regrets (the lower the better) of all algorithms in real-world hyperparameter tuning task on Breast-cancer dataset.

able to find the optimal point shortly after Phase I and enjoys the least error bar. It is well expected since f_1 is a realizable function for $\hat{f}$. Unfortunately, GP-UCB, GP-EI, and GP-PI incur almost linear regrets, showing the bad performances of classical BO algorithms in high-dimensional cases.

In Figure 2(b) and 2(c), all methods are suffering from linear regrets because f_2, f_3 are misspecified functions. The gap between GO-UCB and other methods is smaller in Figure 2(c) than in 2(b) because optimizing f_3 is more challenging than f_2 since f_3 has more local optimal points.

6.2. Real-World Experiments

To illustrate the GO-UCB algorithm works in real-world tasks, we do hyperparameter tuning experiments on three tasks using three classifiers. Three UCI datasets (Dua & Graff, 2017) are Breat-cancer, Australian, and Diabetes, and three classifiers are random forest, multi-layer perceptron, and gradient boosting where each of them has $7, 8, 11$ hyperparameters. For each classifier on each dataset, the function mapping from hyperparameters to classification accuracy is the black-box function that we are maximizing, so the input space dimension $d_x = 7, 8, 11$ for each classifier. We use cumulative regret to evaluate hyperparameter tuning performances, however, best accuracy f^* is unknown ahead of time so we set it to be the best empirical accuracy of each task. To reduce the effect of randomness, we divide each dataset into 5 folds and every time use 4 folds for training and remaining 1 fold for testing. We report mean and error bar of cumulative regrets where error bar is measured by Wald's test, the same as synthetic experiments.

Figure 3 shows results on Breast-cancer dataset. In Figure 3(b)(c) GO-UCB performs statistically much better that all other BO algorithms since there is almost no error bar gap between TuRBO and GO-UCB. It shows that GO-UCB can be deployed in real-world applications to replace BO methods. Also, in Figure 3(b) performance of GO-UCB Phase I is not good but GO-UCB can still perform better than others in Phase II, which shows the effectiveness of Phase II of GO-UCB. In Figure 3(a) all algorithms have similar performances. In Figure 3(b), TuRBO performs similarly as GP-UCB, GP-EI, and GP-PI when $t \leq 23$, but after $t = 23$ it performs better and shows a curved regret line by finding optimal points. Due to page limit, results on Australian and Diabetes datasets are shown in Appendix D.2 where similar algorithm performances can be seen.

Note in experiments, we choose parametric model $\hat{f}$ to be a two linear layer neural network. In more real-world experiments, one can choose the model $\hat{f}$ in GO-UCB to be simpler functions or much more complex functions, e.g., deep neural networks, depending on task requirements.

7. Conclusion

Global non-convex optimization is an important problem that widely exists in many real-world applications, e.g., deep learning hyper-parameter tuning and new material design. However, solving this optimization problem in general is NP-hard. Existing work relies on Gaussian process assumption, e.g., Bayesian optimization, or other non-parametric family which suffers from the curse of dimensionality.

We propose the first algorithm to solve such global optimization with parametric function approximation, which shows a new way of global optimization. GO-UCB first uniformly explores the function and collects a set of observation points and then uses the optimistic exploration to actively select points. At the core of GO-UCB is a carefully designed uncertainty set over parameters based on gradients that allows optimistic exploration. Under realizable parameter class assumption and a few mild geometric conditions, our theoretical analysis shows that cumulative regret of GO-UCB is at the rate of $\tilde{O}(\sqrt{T})$, which is dimension-free in terms of function domain $\mathcal{X}$. Our high-dimensional synthetic test shows that GO-UCB works better than BO methods even in misspecified setting. Moreover, GO-UCB performs better than BO algorithms in real-world hyperparameter tuning tasks, which may be of independent interest.

There is μ, the strongly convexity parameter, in the denominator of upper bound in Theorem 4.1. μ can be small in practice, thus the upper bound can be large. Developing the cumulative regret bound containing a term depending on μ but being independent to T remains a future problem.

Acknowledgments

The work is partially supported by NSF Awards #1934641 and #2134214. We thank Ming Yin and Dan Qiao for helpful discussion and careful proofreading of an early version of the manuscript, as well as Andrew G. Wilson for pointing out that Bayesian optimization methods do not necessarily use Gaussian processes as surrogate models. Finally, we thank ICML reviewers and the area chair for their valuable input that led to improvements to the paper.

References

Abbasi-yadkori, Y., Pál, D., and Szepesvári, C. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems 24 (NeurIPS'11)*, 2011.

Agarwal, A., Jiang, N., Kakade, S. M., and Sun, W. Reinforcement learning: Theory and algorithms, 2021.

Agrawal, S. and Goyal, N. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning (ICML'13)*, 2013.

Balandat, M., Karrer, B., Jiang, D., Daulton, S., Letham, B., Wilson, A. G., and Bakshy, E. Botorch: a framework for efficient monte-carlo bayesian optimization. In *Advances in Neural Information Processing Systems 33 (NeurIPS'20)*, 2020.

Bubeck, S., Munos, R., Stoltz, G., and Szepesvári, C. X-armed bandits. *Journal of Machine Learning Research*, 12(46):1655–1695, 2011.

Bull, A. D. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12(10), 2011.

Cai, X. and Scarlett, J. On lower bounds for standard and robust gaussian process bandit optimization. In *International Conference on Machine Learning (ICML'21)*, 2021.

Calandriello, D., Carratino, L., Lazaric, A., Valko, M., and Rosasco, L. Gaussian process optimization with adaptive sketching: Scalable and no regret. In *Conference on Learning Theory (COLT'19)*, 2019.

Chu, W., Li, L., Reyzin, L., and Schapire, R. Contextual bandits with linear payoff functions. In *International Conference on Artificial Intelligence and Statistics (AISTATS'11)*, 2011.

Dai, Z., Shu, Y., Low, B. K. H., and Jaillet, P. Sample-then-optimize batch neural Thompson sampling. In *Advances in Neural Information Processing Systems 35 (NeurIPS'22)*, 2022.

Dani, V., Hayes, T. P., and Kakade, S. M. Stochastic linear optimization under bandit feedback. In *Conference on Learning Theory (COLT'08)*, 2008.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Eriksson, D., Pearce, M., Gardner, J., Turner, R. D., and Poloczek, M. Scalable global optimization via local bayesian optimization. In *Advances in neural information processing systems 32 (NeurIPS'19)*, 2019.

Filippi, S., Cappe, O., Garivier, A., and Szepesvári, C. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems 23 (NeurIPS'10)*, 2010.

Foster, D. and Rakhlin, A. Beyond ucb: Optimal and efficient contextual bandits with regression oracles. In *International Conference on Machine Learning (ICML'20)*, 2020.

Foster, D., Agarwal, A., Dudik, M., Luo, H., and Schapire, R. Practical contextual bandits with regression oracles. In *International Conference on Machine Learning (ICML'18)*, 2018.

Frazier, P., Powell, W., and Dayanik, S. The knowledge-gradient policy for correlated normal beliefs. *INFORMS Journal on Computing*, 21(4):599–613, 2009.

Frazier, P. I. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

Frazier, P. I. and Wang, J. Bayesian optimization for materials design. In *Information science for materials discovery and design*, pp. 45–75. Springer, 2016.

Hazan, E., Klivans, A., and Yuan, Y. Hyperparameter optimization: a spectral approach. In *International Conference on Learning Representations (ICLR'18)*, 2018.

Head, T., Kumar, M., Nahrstaedt, H., Louppe, G., and Shcherbatyi, I. scikit-optimize. https://scikit-optimize.github.io, 2021.

Jain, P., Kar, P., et al. Non-convex optimization for machine learning. *Foundations and Trends® in Machine Learning*, 10(3-4):142–363, 2017.

Jones, D. R., Schonlau, M., and Welch, W. J. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

Kandasamy, K., Neiswanger, W., Schneider, J., Poczos, B., and Xing, E. P. Neural architecture search with bayesian optimisation and optimal transport. In *Advances in neural information processing systems 31 (NeurIPS'18)*, 2018.

Kandasamy, K., Vysyaraju, K. R., Neiswanger, W., Paria, B., Collins, C. R., Schneider, J., Poczos, B., and Xing, E. P. Tuning hyperparameters without grad students: Scalable and robust bayesian optimisation with dragonfly. *Journal of Machine Learning Research*, 21(81):1–27, 2020.

Kushner, H. J. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 8(1):97–106, 1964.

Li, L., Lu, Y., and Zhou, D. Provably optimal algorithms for generalized linear contextual bandits. In *International Conference on Machine Learning (ICML'17)*, 2017.

Li, Y., Wang, Y., and Zhou, Y. Nearly minimax-optimal regret for linearly parameterized bandits. In *Annual Conference on Learning Theory (COLT'19)*, 2019.

Malherbe, C. and Vayatis, N. Global optimization of lipschitz functions. In *International Conference on Machine Learning (ICML'17)*, 2017.

Nakamura, N., Seepaul, J., Kadane, J. B., and Reeja-Jayan, B. Design for low-temperature microwave-assisted crystallization of ceramic thin films. *Applied Stochastic Models in Business and Industry*, 33(3):314–321, 2017.

Nesterov, Y. and Nemirovskii, A. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

Nowak, R. D. Lecture notes: Complexity regularization for squared error loss, 2007. URL https://nowak.ece.wisc.edu/SLT07/lecture12.pdf.

Rando, M., Carratino, L., Villa, S., and Rosasco, L. Ada-bkb: Scalable gaussian process optimization on continuous domains by adaptive discretization. In *International Conference on Artificial Intelligence and Statistics (AISTATS'22)*, 2022.

Rinnooy Kan, A. and Timmer, G. T. Stochastic global optimization methods part i: Clustering methods. *Mathematical programming*, 39(1):27–56, 1987a.

Rinnooy Kan, A. and Timmer, G. T. Stochastic global optimization methods part ii: Multi level methods. *Mathematical Programming*, 39(1):57–78, 1987b.

Russo, D. and Van Roy, B. Eluder dimension and the sample complexity of optimistic exploration. In *Advances in Neural Information Processing Systems 26 (NeurIPS'13)*, 2013.

Salgia, S., Vakili, S., and Zhao, Q. A domain-shrinking based bayesian optimization algorithm with order-optimal regret performance. In *Advances in Neural Information Processing Systems 34 (NeurIPS'21)*, 2021.

Scarlett, J., Bogunovic, I., and Cevher, V. Lower bounds on regret for noisy gaussian process bandit optimization. In *Annual Conference on Learning Theory (COLT'17)*, 2017.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104 (1):148–175, 2015.

Shekhar, S. and Javidi, T. Gaussian process bandits with adaptive discretization. *Electronic Journal of Statistics*, 12(2):3829–3874, 2018.

Sherman, J. and Morrison, W. J. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *Annals of Mathematical Statistics*, 21(1): 124–127, 1950.

Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M., Prabhat, M., and Adams, R. Scalable bayesian optimization using deep neural networks. In *International Conference on Machine Learning (ICML'15)*, 2015.

Springenberg, J. T., Klein, A., Falkner, S., and Hutter, F. Bayesian optimization with robust bayesian neural networks. In *Advances in Neural Information Processing Systems 29 (NeurIPS'16)*, 2016.

Srinivas, N., Krause, A., Kakade, S., and Seeger, M. Gaussian process optimization in the bandit setting: no regret and experimental design. In *International Conference on Machine Learning (ICML'10)*, 2010.

Wang, L., Fonseca, R., and Tian, Y. Learning search space partition for black-box optimization using monte carlo tree search. In *Advances in Neural Information Processing Systems 33 (NeurIPS'20)*, 2020.

Wang, Y., Balakrishnan, S., and Singh, A. Optimization of smooth functions with noisy observations: Local minimax rates. In *Advances in Neural Information Processing Systems 31 (NeurIPS'18)*, 2018.

Williams, C. K. and Rasmussen, C. E. *Gaussian processes for machine learning*. MIT Press, 2006.

Zhang, L., Yang, T., Yi, J., Jin, R., and Zhou, Z.-H. Improved dynamic regret for non-degenerate functions. In *Advances in Neural Information Processing Systems 30 (NeurIPS'17)*, 2017.

Zhang, W., Zhou, D., Li, L., and Gu, Q. Neural thompson sampling. In *International Conference on Learning Representations (ICLR'20)*, 2020.

Zhou, D., Li, L., and Gu, Q. Neural contextual bandits with ucb-based exploration. In *International Conference on Machine Learning (ICML'20)*, 2020.

A. Notation Table

Table 1. Symbols and notations.

Symbol	Definition	Description
$\|A\|_{\text{op}}$		operator norm
Ball_t	eq. (6)	parameter uncertainty region at round t
β_t	eq. (7)	parameter uncertainty region radius at round t
μ		local strong convexity parameter
c		local self-concordance parameter
C, ζ		constants
d_x		domain dimension
d_w		parameter dimension
δ		failure probability
ε		covering number discretization distance
η	σ-sub-Gaussian	observation noise
$f_w(x)$		objective function at x parameterized by w
$f_x(w)$		objective function at w parameterized by x
$\nabla f_x(w)$		1st order derivative w.r.t. w parameterized by x
$\nabla^2 f_x(w)$		2nd order derivative w.r.t. w parameterized by x
F		function range constant bound
γ, τ		growth condition parameters
ι, ι', ι''		logarithmic terms
$L(w)$	$\mathbb{E}[(f_x(w) - f_x(w^*))^2]$	expected loss function
λ		regularization parameter
n		time horizon in Phase I
$[n]$	$\{1, 2, ..., n\}$	integer set of size n
Oracle		regression oracle
r_t	$f_{w^*}(x^*) - f_{w^*}(x_t)$	instantaneous regret at round t
R_T	$\sum_{t=1}^{T} r_t$	cumulative regret after round T
Σ_t	eq. (3)	covariance matrix at round t
T		time horizon in Phase II
$\mathcal{U}$		uniform distribution
w	$w \in \mathcal{W}$	function parameter
w^*	$w^* \in \mathcal{W}$	true parameter
$\hat{w}_0$		oracle-estimated parameter after Phase I
$\hat{w}_t$	eq. (5)	updated parameter at round t
$\mathcal{W}$	$\mathcal{W} \subseteq [0, 1]^{d_w}$	parameter space
x	$x \in \mathcal{X}$	data point
x^*		optimal data point
$\|x\|_\infty$	$(\sum_{i=1}^{d} \|x_i\|^p)^{1/p}$	ℓ_p norm
$\|x\|_p$	$\max_{i \in [d]} \|x_i\|$	ℓ_∞ norm
$\|x\|_A$	$\sqrt{x^\top A x}$	distance defined by square matrix A
$\mathcal{X}$	$\mathcal{X} \subseteq \mathbb{R}^{d_x}$	function domain
$\mathcal{Y}$	$\mathcal{Y} = [-F, F]$	function range

B. Auxiliary Technical Lemmas

In this section, we list auxiliary lemmas that are used in proofs.

Lemma B.1 (Adapted from eq. (5) (6) of Nowak (2007))**.** *Given a dataset $\{x_i, y_i\}_{j=1}^n$ where y_j is generated from eq. (1) and f_0 is the underlying true function. Let $\hat{f}$ be an ERM estimator taking values in $\mathcal{F}$ where $\mathcal{F}$ is a finite set and*

$\mathcal{F} \subset \{f : [0,1]^d \to [-F, F]\}$ for some $F \geq 1$. Then with probability $> 1-\delta$, $\hat{f}$ satisfies that

$$\mathbb{E}[(\hat{f} - f_0)^2] \leq \left(\frac{1+\alpha}{1-\alpha}\right)\left(\inf_{f\in\mathcal{F}} \mathbb{E}[(f - f_0)^2] + \frac{F^2 \log(|\mathcal{F}|)\log(2)}{n\alpha}\right) + \frac{2\log(2/\delta)}{n\alpha},$$

for all $\alpha \in (0, 1]$.

Lemma B.2 (Sherman-Morrison lemma (Sherman & Morrison, 1950))**.** *Let A denote a matrix and b, c denote two vectors. Then*

$$(A + bc^\top)^{-1} = A^{-1} - \frac{A^{-1}bc^\top A^{-1}}{1 + c^\top A^{-1} b}.$$

Lemma B.3 (Self-normalized bound for vector-valued martingales (Abbasi-yadkori et al., 2011; Agarwal et al., 2021))**.** *Let $\{\eta_i\}_{i=1}^\infty$ be a real-valued stochastic process with corresponding filtration $\{\mathcal{F}_i\}_{i=1}^\infty$ such that η_i is $\mathcal{F}_i$ measurable, $\mathbb{E}[\eta_i|\mathcal{F}_{i-1}] = 0$, and η_i is conditionally σ-sub-Gaussian with $\sigma \in \mathbb{R}^+$. Let $\{X_i\}_{i=1}^\infty$ be a stochastic process with $X_i \in \mathcal{H}$ (some Hilbert space) and X_i being F_t measurable. Assume that a linear operator $\Sigma : \mathcal{H} \to \mathcal{H}$ is positive definite, i.e., $x^\top \Sigma x > 0$ for any $x \in \mathcal{H}$. For any t, define the linear operator $\Sigma_t = \Sigma_0 + \sum_{i=1}^t X_i X_i^\top$ (here $xx^\top$ denotes outer-product in $\mathcal{H}$). With probability at least $1-\delta$, we have for all $t \geq 1$:*

$$\left\|\sum_{i=1}^t X_i \eta_i\right\|^2_{\Sigma_t^{-1}} \leq \sigma^2 \log\left(\frac{\det(\Sigma_t)\det(\Sigma_0)^{-1}}{\delta^2}\right).$$

C. Missing Proofs

In this section, we show complete proofs of all technical results in the main paper. For reader's easy reference, we define ι as a logarithmic term depending on $n, C_h, 2/\delta$ (w.p. $> 1-\delta/2$), ι' as a logarithmic term depending on $t, d_w, C_g, 1/\lambda, 2/\delta$ (w.p. $> 1-\delta/2$), and ι'' as a logarithmic term depending on $t, d_w, C_g, 1/\lambda$.

C.1. Regression Oracle Guarantee

Lemma C.1 (Restatement of Lemma 5.1)**.** *Suppose Assumption 3.1 & 3.2 hold. There is an absolute constant C', such that after round n in Phase I of Algorithm 1, with probability $> 1-\delta/2$, regression oracle estimated $\hat{w}_0$ satisfies*

$$\mathbb{E}_{x\sim\mathcal{U}}[(f_x(\hat{w}_0) - f_x(w^*))^2] \leq \frac{C' d_w F^2 \iota}{n},$$

where ι is the logarithmic term depending on $n, C_h, 2/\delta$.

Proof. The regression oracle lemma establishes on Lemma B.1 which works only for finite function class. In order to work with our continuous parameter class $\mathcal{W}$, we need ε-covering number argument.

First, let $\tilde{w}, \widetilde{\mathcal{W}}$ denote the ERM parameter and finite parameter class after applying covering number argument on $\mathcal{W}$. By Lemma B.1, we find that with probability $> 1-\delta/2$,

$$\begin{aligned}\mathbb{E}_{x\sim\mathcal{U}}[(f_x(\tilde{w}) - f_x(w^*))^2] &\leq \left(\frac{1+\alpha}{1-\alpha}\right)\left(\inf_{w\in\widetilde{\mathcal{W}}\cup\{w^*\}} \mathbb{E}_{x\sim\mathcal{U}}[(f_x(w) - f_x(w^*))^2] + \frac{F^2\log(|\widetilde{\mathcal{W}}|)\log(2)}{n\alpha}\right) + \frac{2\log(4/\delta)}{n\alpha} \\ &\leq \left(\frac{1+\alpha}{1-\alpha}\right)\left(\frac{F^2\log(|\widetilde{\mathcal{W}}|)\log(2)}{n\alpha}\right) + \frac{2\log(4/\delta)}{n\alpha},\end{aligned}$$

where the second inequality is by realizable assumption (Assumption 3.1). Our parameter class $\mathcal{W} \subseteq [0,1]^{d_w}$, so $\log(|\widetilde{\mathcal{W}}|) = \log(1/\varepsilon^{d_w}) = d_w \log(1/\varepsilon)$ and the new upper bound is that with probability $> 1-\delta/2$,

$$\mathbb{E}_{x\sim\mathcal{U}}[(f_x(\tilde{w}) - f_x(w^*))^2] \leq C''\left(\frac{d_w F^2 \log(1/\varepsilon)}{n} + \frac{\log(2/\delta)}{n}\right),$$

where C'' is a universal constant obtained by choosing $\alpha = 1/2$. Note $\tilde{w}$ is the ERM parameter in $\widetilde{\mathcal{W}}$ after discretization, not our target parameter $\hat{w}_0 \in \mathcal{W}$. By $(a+b)^2 \leq 2a^2 + 2b^2$,

$$\begin{aligned}\mathbb{E}_{x\sim\mathcal{U}}[(f_x(\hat{w}_0) - f_x(w^*))^2] &\leq 2\mathbb{E}_{x\sim\mathcal{U}}[(f_x(\hat{w}_0) - f_x(\tilde{w}))^2] + 2\mathbb{E}_{x\sim\mathcal{U}}[(f_x(\tilde{w}) - f_x(w^*))^2] \\ &\leq 2\varepsilon^2 C_h^2 + 2C''\left(\frac{d_w F^2 \log(1/\varepsilon)}{n} + \frac{\log(2/\delta)}{n}\right) \end{aligned} \tag{10}$$

where the second line applies discretization error ε and Assumption 3.2. By choosing $\varepsilon = 1/\sqrt{nC_h^2}$, we get

$$(10) = \frac{2}{n} + \frac{C'' d_w F^2 \log(nC_h^2)}{n} + \frac{2C'' \log(2/\delta)}{n} \leq C' \frac{d_w F^2 \log(nC_h^2) + \log(2/\delta)}{n}$$

where we can take $C' = 2C''$ (assuming $2 < C'' d_w F^2 \log(nC_h^2)$). The proof completes by defining ι as the logarithmic term depending on $n, C_h, 2/\delta$. □

Theorem C.2 (Restatement of Theorem 5.2)**.** *Suppose Assumption 3.1, 3.2, & 3.3 hold. There is an absolute constant C such that after round n in Phase I of Algorithm 1 where n satisfies*

$$n \geq C d_w F^2 \iota \cdot \max\left\{\frac{\mu^{\gamma/(2-\gamma)}}{\tau^{2/(2-\gamma)}}, \frac{\zeta}{\mu c^2}\right\},$$

with probability $> 1 - \delta/2$, regression oracle estimated $\hat{w}_0$ satisfies

$$\|\hat{w}_0 - w^*\|_2^2 \leq \frac{C d_w F^2 \iota}{\mu n},$$

where ι is the logarithmic term depending on $n, C_h, 2/\delta$.

Proof. Recall the definition of expected loss function $L(w) = \mathbb{E}_{x\sim\mathcal{U}}(f_x(w) - f_x(w^*))^2$ and the second order Taylor's theorem, $L(\hat{w}_0)$ at w^* can be written as

$$L(\hat{w}_0) = L(w^*) + (\hat{w}_0 - w^*)\nabla L(w^*) + \frac{1}{2}\|\hat{w}_0 - w^*\|^2_{\nabla^2 L(\tilde{w})},$$

where $\tilde{w}$ lies between $\hat{w}_0$ and w^*. Also, because $\nabla L(w^*) = \nabla E_{x\sim\mathcal{U}}(f_x(w^*) - f_x(w^*))^2 = 0$, then with probability $> 1 - \delta/2$,

$$\frac{1}{2}\|\hat{w}_0 - w^*\|^2_{\nabla^2 L(\tilde{w})} = L(\hat{w}_0) - L(w^*) \leq \frac{C' d_w F^2 \iota}{n}, \tag{11}$$

where the inequality is due to Lemma 5.1.

Next, we prove the following lemma stating after a certain number of n samples, $\|\hat{w}_0 - w^*\|_{\nabla^2 L(w^*)}$ can be bounded by the parameter c from our local-self-concordance assumption.

Lemma C.3. *Suppose Assumption 3.1, 3.2, & 3.3 hold. There is an absolute constant C' such that after round n in Phase I of Algorithm 1 where n satisfies*

$$n \geq 2C' d_w F^2 \iota \cdot \max\left\{\frac{\mu^{\gamma/(2-\gamma)}}{\tau^{2/(2-\gamma)}}, \frac{\zeta}{\mu c^2}\right\},$$

then with probability $> 1 - \delta/2$,

$$\|\hat{w}_0 - w^*\|_{\nabla^2 L(w^*)} \leq c.$$

Proof. First we will prove that when n satisfies the first condition, then $\|\hat{w}_0 - w^*\|_2 \leq (\tau/\mu)^{1/(2-\gamma)}$ by a proof by contradiction.

Assume $\|\hat{w}_0 - w^*\|_2 > (\tau/\mu)^{1/(2-\gamma)}$. Check that under this condition, we have $\frac{\tau}{2}\|\hat{w}_0 - w^*\|_2^\gamma < \frac{\mu}{2}\|\hat{w}_0 - w^*\|_2^2$, therefore the growth-condition (rather than the local strong convexity) part of the Assumption 3.3 is active. By the (τ, γ)-growth condition, we have

$$\frac{\tau}{2}\|\hat{w}_0 - w^*\|_2^\gamma \leq L(\hat{w}_0) - L(w^*) \leq \frac{C' d_w F^2 \iota}{n}.$$

Substituting the first lower bound of n in the assumption, we get

$$\|\hat{w}_0 - w^*\| \leq (\tau/\mu)^{1/(2-\gamma)},$$

thus having a contradiction. This proves that when n satisfies the first condition, $\hat{w}_0$ is within the region where local strong convexity is active.

By the local strong-convexity condition,

$$\frac{\mu}{2}\|\hat{w}_0 - w^*\|_2^2 \leq L(\hat{w}_0) - L(w^*) \leq \frac{C' d_w F^2 \iota}{n}.$$

Then,

$$\|\hat{w}_0 - w^*\|_{\nabla^2 L(w^*)} \leq \sqrt{\zeta}\|\hat{w}_0 - w^*\|_2 \leq \sqrt{\frac{2\zeta C' d_w F^2 \iota}{\mu n}}.$$

Substitute the second lower bound on n that we assumed, we get that

$$\|\hat{w}_0 - w^*\|_{\nabla^2 L(w^*)} \leq \sqrt{\frac{2\zeta C' d_w F^2 \iota}{\mu n}} \leq c.$$

□

Now we continue the proof of Theorem 5.2. Observe that $\|\tilde{w} - w^*\|_{\nabla^2 L(w^*)} \leq \|\hat{w}_0 - w^*\|_{\nabla^2 L(w^*)} \leq c$, since $\tilde{w}$ lies on the line-segment between $\hat{w}_0$ and w^*. It follows that by the c-local self-concordance assumption (Assumption 3.3),

$$(1-c)^2\|\hat{w}_0 - w^*\|_{\nabla^2 L(w^*)}^2 \leq \|\hat{w}_0 - w^*\|_{\nabla^2 L(\tilde{w})}^2.$$

Therefore, by eq. (11)

$$\|\hat{w}_0 - w^*\|_{\nabla^2 L(w^*)}^2 \leq \frac{2C' d_w F^2 \iota}{(1-c)^2 n}.$$

The proof completes by inequality $\|\hat{w}_0 - w^*\|_2^2 \leq \|\hat{w}_0 - w^*\|_{\nabla^2 L(w^*)}^2/\mu$ due to μ-strongly convexity of $L(w)$ at w^* (Assumption 3.3) and defining $C = 2C'/(1-c)^2$. □

C.2. Properties of Covariance Matrix Σ_t

In eq. (3), Σ_t is defined as $\lambda I + \sum_{i=0}^{t-1} \nabla f_{x_i}(\hat{w}_i)\nabla f_{x_i}(\hat{w}_i)^\top$. In this section, we prove three lemmas saying the change of Σ_t as $t \in 1, ..., T$ is bounded in Phase II of GO-UCB. The key observation is that at each round i, the change made to Σ_t is $\nabla f_{x_i}(\hat{w}_i)\nabla f_{x_i}(\hat{w}_i)^\top$, which is only rank one.

Lemma C.4 (Adapted from Agarwal et al. (2021))**.** *Set $\Sigma_t, \hat{w}_t$ as in eq. (3) & (5), suppose Assumption 3.1 & 3.3 hold, and define $u_t = \|\nabla f_{x_t}(\hat{w}_t)\|_{\Sigma_t^{-1}}$. Then*

$$\det \Sigma_t = \det \Sigma_0 \prod_{i=0}^{t-1}(1 + u_i^2).$$

Proof. Recall the definition of $\Sigma_t = \lambda I + \sum_{i=0}^{t-1} \nabla f_{x_i}(\hat{w}_i)\nabla f_{x_i}(\hat{w}_i)^\top$ and we can show that

$$\begin{aligned}
\det \Sigma_{t+1} &= \det(\Sigma_t + \nabla f_{x_t}(w_t)\nabla f_{x_t}(w_t)^\top) \\
&= \det(\Sigma_t^{\frac{1}{2}}(I + \Sigma_t^{-\frac{1}{2}}\nabla f_{x_t}(w_t)\nabla f_{x_t}(w_t)^\top \Sigma_t^{-\frac{1}{2}})\Sigma_t^{\frac{1}{2}}) \\
&= \det(\Sigma_t)\det(I + \Sigma_t^{-\frac{1}{2}}\nabla f_{x_t}(w_t)(\Sigma_t^{-\frac{1}{2}}\nabla f_{x_t}(w_t))^\top) \\
&= \det(\Sigma_t)\det(I + v_t v_t^\top),
\end{aligned}$$

where $v_t = \Sigma_t^{-\frac{1}{2}}\nabla f_{x_t}(w_t)$. Recall u_t is defined as $\|\nabla f_{x_t}(\hat{w}_t)\|_{\Sigma_t^{-1}}$. Because $v_t v_t^\top$ is a rank one matrix, $\det(I + v_t v_t^\top) = 1 + u_t^2$. The proof completes by induction. □

Lemma C.5 (Adapted from Agarwal et al. (2021))**.** *Set Σ_t as in eq. (3) and suppose Assumption 3.1, 3.2, & 3.3 hold. Then*

$$\log\left(\frac{\det \Sigma_{t-1}}{\det \Sigma_0}\right) \le d_w \log\left(1 + \frac{tC_g^2}{d_w\lambda}\right).$$

Proof of Lemma C.4 directly follows definition of Σ_t and proof of Lemma C.5 involves Lemma C.4 and inequality of arithmetic and geometric means. Note C_g is a constant coming from Assumption 3.2. We do not claim any novelty in proofs of these two lemmas which replace feature vector in linear bandit (Agarwal et al., 2021) with gradient vectors.

Proof. Let $\xi_1, ..., \xi_{d_w}$ denote eigenvalues of $\sum_{i=0}^{t-1} \nabla f_{x_i}(w_i)\nabla f_{x_i}(w_i)^\top$, then

$$\sum_{k=1}^{d_w} \xi_k = \operatorname{tr}\left(\sum_{i=0}^{t-1} \nabla f_{x_i}(w_i)\nabla f_{x_i}(w_i)^\top\right) = \sum_{i=0}^{t-1} \|\nabla f_{x_i}(w_i)\|_2^2 \le tC_g^2, \tag{12}$$

where the inequality is by Assumption 3.2. By Lemma C.4,

$$\begin{aligned}
\log\left(\frac{\det \Sigma_{t-1}}{\det \Sigma_0}\right) &\le \log\det\left(I + \frac{1}{\lambda}\sum_{i=0}^{t-1} \nabla f_{x_i}(w_i)\nabla f_{x_i}(w_i)^\top\right) \\
&= \log\left(\prod_{k=1}^{d_w}(1 + \xi_k/\lambda)\right) \\
&= d_w \log\left(\prod_{k=1}^{d_w}(1 + \xi_k/\lambda)\right)^{1/d_w} \\
&\le d_w \log\left(\frac{1}{d_w}\sum_{k=1}^{d_w}(1 + \xi_k/\lambda)\right) \\
&\le d_w \log\left(1 + \frac{tC_g^2}{d_w\lambda}\right),
\end{aligned}$$

where the second inequality is by inequality of arithmetic and geometric means and the last inequality is due to eq. (12). □

Lemma C.6. *Set $\Sigma_t, \hat{w}_t$ as in eq. (3) & (5) and suppose Assumption 3.1, 3.2, & 3.3 hold. Then*

$$\sum_{i=0}^{t-1} \nabla f_{x_i}(\hat{w}_i)^\top \Sigma_t^{-1} \nabla f_{x_i}(\hat{w}_i) \le 2d_w \log\left(1 + \frac{tC_g^2}{d_w\lambda}\right).$$

A trivial bound of LHS in Lemma C.6 could be simply $O(tC_g^2/\lambda)$. Lemma C.6 is important because it saves the upper bound to be $O(\log(tC_g^2/\lambda))$, which allows us to build a feasible parameter uncertainty ball, shown in the next section.

Proof. First, we prove $\forall i \in \{0, 1, ..., t-1\}, 0 < \nabla f_{x_i}(\hat{w}_i)^\top \Sigma_t^{-1} \nabla f_{x_i}(\hat{w}_i) < 1$. Recall the definition of Σ_t, it's easy to see that Σ_t is a positive definite matrix and thus $0 < \nabla f_{x_i}(\hat{w}_i)^\top \Sigma_t^{-1} \nabla f_{x_i}(\hat{w}_i)$. To prove it's smaller than 1, we need to decompose Σ_t and write

$$\begin{aligned}
&\nabla f_{x_i}(\hat{w}_i)^\top \Sigma_t^{-1} \nabla f_{x_i}(\hat{w}_i) \\
&= \nabla f_{x_i}(\hat{w}_i)^\top \left(\lambda I + \sum_{i=0}^{t-1} \nabla f_{x_i}(\hat{w}_i) \nabla f_{x_i}(\hat{w}_i)^\top \right)^{-1} \nabla f_{x_i}(\hat{w}_i) \\
&= \nabla f_{x_i}(\hat{w}_i)^\top \left(\nabla f_{x_i}(\hat{w}_i) \nabla f_{x_i}(\hat{w}_i)^\top - \nabla f_{x_i}(\hat{w}_i) \nabla f_{x_i}(\hat{w}_i)^\top + \lambda I + \sum_{i=0}^{t-1} \nabla f_{x_i}(\hat{w}_i) \nabla f_{x_i}(\hat{w}_i)^\top \right)^{-1} \nabla f_{x_i}(\hat{w}_i).
\end{aligned}$$

Let $A = -\nabla f_{x_i}(\hat{w}_i) \nabla f_{x_i}(\hat{w}_i)^\top + \lambda I + \sum_{i=0}^{t-1} \nabla f_{x_i}(\hat{w}_i) \nabla f_{x_i}(\hat{w}_i)^\top$, and it becomes

$$\nabla f_{x_i}(\hat{w}_i)^\top \Sigma_t^{-1} \nabla f_{x_i}(\hat{w}_i) = \nabla f_{x_i}(\hat{w}_i)^\top (\nabla f_{x_i}(\hat{w}_i) \nabla f_{x_i}(\hat{w}_i)^\top + A)^{-1} \nabla f_{x_i}(\hat{w}_i).$$

By applying Sherman-Morrison lemma (Lemma B.2), we have

$$\begin{aligned}
\nabla f_{x_i}(\hat{w}_i)^\top \Sigma_t^{-1} \nabla f_{x_i}(\hat{w}_i) &= \nabla f_{x_i}(\hat{w}_i)^\top \left(A^{-1} - \frac{A^{-1} \nabla f_{x_i}(\hat{w}_i) \nabla f_{x_i}(\hat{w}_i)^\top A^{-1}}{1 + \nabla f_{x_i}(\hat{w}_i)^\top A^{-1} \nabla f_{x_i}(\hat{w}_i)} \right) \nabla f_{x_i}(\hat{w}_i) \\
&= \nabla f_{x_i}(\hat{w}_i)^\top A^{-1} \nabla f_{x_i}(\hat{w}_i) - \frac{\nabla f_{x_i}(\hat{w}_i)^\top A^{-1} \nabla f_{x_i}(\hat{w}_i) \nabla f_{x_i}(\hat{w}_i)^\top A^{-1} \nabla f_{x_i}(\hat{w}_i)}{1 + \nabla f_{x_i}(\hat{w}_i)^\top A^{-1} \nabla f_{x_i}(\hat{w}_i)} \\
&= \frac{\nabla f_{x_i}(\hat{w}_i)^\top A^{-1} \nabla f_{x_i}(\hat{w}_i)}{1 + \nabla f_{x_i}(\hat{w}_i)^\top A^{-1} \nabla f_{x_i}(\hat{w}_i)} < 1.
\end{aligned}$$

Next, we use the fact that $\forall x \in (0, 1), x \leq 2 \log(1 + x)$, and we have

$$\begin{aligned}
\sum_{i=0}^{t-1} \nabla f_{x_i}(\hat{w}_i)^\top \Sigma_t^{-1} \nabla f_{x_i}(\hat{w}_i) &\leq \sum_{i=0}^{t-1} 2 \log \left(1 + \nabla f_{x_i}(\hat{w}_i)^\top \Sigma_t^{-1} \nabla f_{x_i}(\hat{w}_i) \right) \\
&\leq 2 \log \left(\frac{\det \Sigma_{t-1}}{\det \Sigma_0} \right) \\
&\leq 2 d_w \log \left(1 + \frac{t C_g^2}{d_w \lambda} \right),
\end{aligned}$$

where the last two inequalities are due to Lemma C.4 and C.5. □

C.3. Feasibility of Ball_t

Lemma C.7 (Restatement of Lemma 5.3)**.** *Set $\Sigma_t, \hat{w}_t$ as in eq. (3), (5). Set β_t as*

$$\beta_t = \tilde{O} \left(d_w \sigma^2 + \frac{d_w^3}{\mu^2} + \frac{d_w^3 t}{\mu^2 T} \right).$$

Suppose Assumption 3.1, 3.2, & 3.3 hold and choose $n = \sqrt{T}, \lambda = C_\lambda \sqrt{T}$. Then $\forall t \in [T]$ in Phase II of Algorithm 1, w.p. $> 1 - \delta$,

$$\|\hat{w}_t - w^*\|_{\Sigma_t}^2 \leq \beta_t.$$

Proof. The proof has three steps. First we obtain the closed form solution of $\hat{w}_t$. Next we derive the upper bound of $\|\hat{w}_i - w^*\|_2^2$. Finally we use it to prove that the upper bound of $\|\hat{w}_t - w^*\|_{\Sigma_t}^2$ matches our choice of β_t.

Step 1: Closed form solution of $\hat{w}_t$. The optimal criterion for the objective function in eq. (4) is

$$0 = \lambda(\hat{w}_t - \hat{w}_0) + \sum_{i=0}^{t-1} ((\hat{w}_t - \hat{w}_i)^\top \nabla f_{x_i}(\hat{w}_i) + f_{x_i}(\hat{w}_i) - y_i) \nabla f_{x_i}(\hat{w}_i).$$

Rearrange the equation and we have

$$\begin{aligned}
\lambda(\hat{w}_t - \hat{w}_0) + \sum_{i=0}^{t-1}(\hat{w}_t - \hat{w}_i)^\top \nabla f_{x_i}(\hat{w}_i)\nabla f_{x_i}(\hat{w}_i) &= \sum_{i=0}^{t-1}(y_i - f_{x_i}(\hat{w}_i))\nabla f_{x_i}(\hat{w}_i), \\
\lambda(\hat{w}_t - \hat{w}_0) + \sum_{i=0}^{t-1}(\hat{w}_t - \hat{w}_i)^\top \nabla f_{x_i}(\hat{w}_i)\nabla f_{x_i}(\hat{w}_i) &= \sum_{i=0}^{t-1}(y_i - f_{x_i}(w^*) + f_{x_i}(w^*) - f_{x_i}(\hat{w}_i))\nabla f_{x_i}(\hat{w}_i), \\
\lambda(\hat{w}_t - \hat{w}_0) + \sum_{i=0}^{t-1}\hat{w}_t^\top \nabla f_{x_i}(\hat{w}_i)\nabla f_{x_i}(\hat{w}_i) &= \sum_{i=0}^{t-1}(\hat{w}_i^\top \nabla f_{x_i}(\hat{w}_i) + \eta_i + f_{x_i}(w^*) - f_{x_i}(\hat{w}_i))\nabla f_{x_i}(\hat{w}_i), \\
\hat{w}_t\left(\lambda I + \sum_{i=1}^{t-1}\nabla f_{x_i}(\hat{w}_i)\nabla f_{x_i}(\hat{w}_i)^\top\right) - \lambda\hat{w}_0 &= \sum_{i=0}^{t-1}(\hat{w}_i^\top \nabla f_{x_i}(\hat{w}_i) + \eta_i + f_{x_i}(w^*) - f_{x_i}(\hat{w}_i))\nabla f_{x_i}(\hat{w}_i), \\
\hat{w}_t\Sigma_t &= \lambda\hat{w}_0 + \sum_{i=0}^{t-1}(\hat{w}_i^\top \nabla f_{x_i}(\hat{w}_i) + \eta_i + f_{x_i}(w^*) - f_{x_i}(\hat{w}_i))\nabla f_{x_i}(\hat{w}_i),
\end{aligned}$$

where the second line is by removing and adding back $f_{x_i}(w^*)$, the third line is due to definition of observation noise η and the last line is by our choice of Σ_t (eq. (3)). Now we have the closed form solution of $\hat{w}_t$:

$$\hat{w}_t = \Sigma_t^{-1}\left(\lambda\hat{w}_0 + \sum_{i=0}^{t-1}(\hat{w}_i^\top \nabla f_{x_i}(\hat{w}_i) + \eta_i + f_{x_i}(w^*) - f_{x_i}(\hat{w}_i))\nabla f_{x_i}(\hat{w}_i)\right).$$

Further, $\hat{w}_t - w^*$ can be written as

$$\begin{aligned}
\hat{w}_t - w^* &= \Sigma_t^{-1}\left(\sum_{i=0}^{t-1}\nabla f_{x_i}(\hat{w}_i)(\nabla f_{x_i}(\hat{w}_i)^\top \hat{w}_i + \eta_i + f_{x_i}(w^*) - f_{x_i}(\hat{w}_i))\right) + \lambda\Sigma_t^{-1}\hat{w}_0 - \Sigma_t^{-1}\Sigma_t w^* \\
&= \Sigma_t^{-1}\left(\sum_{i=0}^{t-1}\nabla f_{x_i}(\hat{w}_i)(\nabla f_{x_i}(\hat{w}_i)^\top \hat{w}_i + \eta_i + f_{x_i}(w^*) - f_{x_i}(\hat{w}_i))\right) + \lambda\Sigma_t^{-1}(\hat{w}_0 - w^*) \\
&\quad - \Sigma_t^{-1}\left(\sum_{i=0}^{t-1}\nabla f_{x_i}(\hat{w}_i)\nabla f_{x_i}(\hat{w}_i)^\top\right)w^* \\
&= \Sigma_t^{-1}\left(\sum_{i=0}^{t-1}\nabla f_{x_i}(\hat{w}_i)(\nabla f_{x_i}(\hat{w}_i)^\top (\hat{w}_i - w^*) + \eta_i + f_{x_i}(w^*) - f_{x_i}(\hat{w}_i))\right) + \lambda\Sigma_t^{-1}(\hat{w}_0 - w^*) \\
&= \Sigma_t^{-1}\left(\sum_{i=0}^{t-1}\nabla f_{x_i}(\hat{w}_i)\frac{1}{2}\|w^* - \hat{w}_i\|^2_{\nabla^2 f_{x_i}(\tilde{w})}\right) + \Sigma_t^{-1}\left(\sum_{i=0}^{t-1}\nabla f_{x_i}(\hat{w}_i)\eta_i\right) + \lambda\Sigma_t^{-1}(\hat{w}_0 - w^*), \qquad (13)
\end{aligned}$$

where the second line is again by our choice of Σ_t and the last equation is by the second order Taylor's theorem of $f_{x_i}(w^*)$ at $\hat{w}_i$ where $\tilde{w}$ lies between w^* and $\hat{w}_i$.

Step 2: Upper bound of $\|\hat{w}_i - w^*\|_2^2$. Note eq. (13) holds $\forall i \in [T]$ because all $\hat{w}_i$ are obtained through the same optimization problem, which means

$$\hat{w}_i - w^* = \Sigma_i^{-1}\left(\sum_{\rho=0}^{i-1}\nabla f_{x_\rho}(\hat{w}_\rho)\frac{1}{2}\|w^* - \hat{w}_\rho\|^2_{\nabla^2 f_{x_\rho}(\tilde{w})}\right) + \Sigma_i^{-1}\left(\sum_{\rho=0}^{i-1}\nabla f_{x_\rho}(\hat{w}_\rho)\eta_\rho\right) + \lambda\Sigma_i^{-1}(\hat{w}_0 - w^*).$$

By inequality $(a+b+c)^2 \le 4a^2 + 4b^2 + 4c^2$ and definition of Σ_i, we take the square of both sides and get

$$\|\hat{w}_i - w^*\|_2^2 \le \frac{4}{\lambda}\left\|\sum_{\rho=0}^{i-1}\nabla f_{x_\rho}(\hat{w}_\rho)\eta_\rho\right\|^2_{\Sigma_i^{-1}} + 4\|\hat{w}_0 - w^*\|_2^2 + \frac{1}{\lambda}\left\|\sum_{\rho=0}^{i-1}\nabla f_{x_\rho}(\hat{w}_\rho)\|w^* - \hat{w}_\rho\|^2_{\nabla^2 f_{x_\rho}(\tilde{w}_\rho)}\right\|^2_{\Sigma_i^{-1}}. \qquad (14)$$

Now we use induction to prove the convergence rate of $\|\hat{w}_i - w^*\|_2^2, \forall i \in [T]$. Recall at the very beginning of Phase II, by Theorem 5.2 (check that the condition on n is satisfied due to our condition on T and the choice of $n = \sqrt{T}$), with probability $> 1 - \delta/2$,

$$\|\hat{w}_0 - w^*\|_2^2 \leq \frac{C d_w F^2 \iota}{\mu n}.$$

To derive a claim based on induction, formally, we suppose at round i, there exists some universal constant $\tilde{C}$ such that with probability $> 1 - \delta/2$,

$$\|\hat{w}_i - w^*\|_2^2 \leq \frac{\tilde{C} d_w F^2 \iota}{\mu n}.$$

Our task is to prove that at round $i+1$ with probability $> 1 - \delta/2$,

$$\|\hat{w}_{i+1} - w^*\|_2^2 \leq \frac{\tilde{C} d_w F^2 \iota}{\mu n}.$$

Note $\tilde{C}$ is for induction purpose, which can be different from C.

From eq. (14), at round $i+1$ we can write

$$\begin{aligned}
\|\hat{w}_{i+1} - w^*\|_2^2 &\leq \frac{4\sigma^2}{\lambda} \log\left(\frac{\det(\Sigma_i)\det(\Sigma_0)^{-1}}{\delta_i^2}\right) + \frac{4 C d_w F^2 \iota}{\mu n} + \frac{1}{\lambda}\left\|\sum_{\rho=0}^{i} \nabla f_{x_\rho}(\hat{w}_\rho)\|w^* - \hat{w}_\rho\|^2_{\nabla^2 f_{x_\rho}(\tilde{w}_\rho)}\right\|^2_{\Sigma_{i+1}^{-1}} \\
&\leq \frac{4\sigma^2}{\lambda}\left(d_w \log\left(1 + \frac{i C_g^2}{d_w \lambda}\right) + \log\left(\frac{\pi^2 i^2}{3\delta}\right)\right) + \frac{4 C d_w F^2 \iota}{\mu n} \\
&\quad + \frac{1}{\lambda}\left\|\sum_{\rho=0}^{i} \nabla f_{x_\rho}(\hat{w}_\rho)\|w^* - \hat{w}_\rho\|^2_{\nabla^2 f_{x_\rho}(\tilde{w}_\rho)}\right\|^2_{\Sigma_{i+1}^{-1}} \\
&\leq \frac{4 d_w \sigma^2 \iota'}{\lambda} + \frac{4 C d_w F^2 \iota}{\mu n} + \frac{1}{\lambda}\left\|\sum_{\rho=0}^{i} \nabla f_{x_\rho}(\hat{w}_\rho)\|w^* - \hat{w}_\rho\|^2_{\nabla^2 f_{x_\rho}(\tilde{w}_\rho)}\right\|^2_{\Sigma_{i+1}^{-1}},
\end{aligned}$$

where the first inequality is due to self-normalized bound for vector-valued martingales (Lemma B.3 in Appendix B) and Theorem 5.2, the second inequality is by Lemma C.5 and our choice of $\delta_i = 3\delta/(\pi^2 i^2)$, and the last inequality is by defining ι' as the logarithmic term depending on $i, d_w, C_g, 1/\lambda, 2/\delta$ (with probability $> 1 - \delta/2$). The choice of δ_i guarantees the total failure probability over t rounds is no larger than $\delta/2$. Now we use our assumption $\|\hat{w}_i - w^*\|_2^2 \leq \frac{\tilde{C} d_w F^2 \iota}{\mu n}$ to bound the last term.

$$\begin{aligned}
\|\hat{w}_{i+1} - w^*\|_2^2 &\leq \frac{4 d_w \sigma^2 \iota'}{\lambda} + \frac{4 C d_w F^2 \iota}{\mu n} + \frac{\tilde{C}^2 C_h^2 d_w^2 F^4 \iota^2}{\mu^2 \lambda n^2}\left(\sum_{\rho=0}^{i} \sqrt{\nabla f_{x_\rho}(\hat{w}_\rho)^\top \Sigma_{i+1}^{-1} \nabla f_{x_\rho}(\hat{w}_\rho)}\right)^2 \\
&\leq \frac{4 d_w \sigma^2 \iota'}{\lambda} + \frac{4 C d_w F^2 \iota}{\mu n} + \frac{\tilde{C}^2 C_h^2 d_w^2 F^4 \iota^2}{\mu^2 \lambda n^2}\left(\sum_{\rho=0}^{i} 1\right)\left(\sum_{\rho=0}^{i} \nabla f_{x_\rho}(\hat{w}_\rho)^\top \Sigma_{i+1}^{-1} \nabla f_{x_\rho}(\hat{w}_\rho)\right) \\
&\leq \frac{4 d_w \sigma^2 \iota'}{\lambda} + \frac{4 C d_w F^2 \iota}{\mu n} + \frac{\tilde{C}^2 C_h^2 d_w^3 F^4 i \iota'' \iota^2}{\mu^2 \lambda n^2},
\end{aligned}$$

where the first inequality is due to smoothness of loss function in Assumption 3.3 and triangular inequality, the second inequality is by Cauchy-Schwarz inequality, and the last inequality is because of Lemma C.6 and defining ι'' as logarithmic term depending on $i, d_w, C_g, 1/\lambda$.

What we need is that there exists some universal constant $\tilde{C}$ such that

$$\frac{4 d_w \sigma^2 \iota'}{\lambda} + \frac{4 C d_w F^2 \iota}{\mu n} + \frac{\tilde{C}^2 C_h^2 d_w^3 F^4 i \iota^2 \iota''}{\lambda \mu^2 n^2} \leq \frac{\tilde{C} d_w F^2 \iota}{\mu n}.$$

Note the LHS is monotonically increasing w.r.t i so the inequality must hold when $i = T$, i.e.,

$$\frac{4d_w\sigma^2\iota'}{\lambda} + \frac{4Cd_wF^2\iota}{\mu n} + \frac{\tilde{C}^2C_h^2d_w^3F^4T\iota^2\iota''}{\lambda\mu^2n^2} \leq \frac{\tilde{C}d_wF^2\iota}{\mu n}.$$

Recall the range of our function is $[-F, F]$, given any distribution, the variance σ^2 can always be upper bounded by $F^2/4$, so we just need to show that

$$\begin{aligned}
\frac{d_wF^2\iota'}{\lambda} + \frac{4Cd_wF^2\iota}{\mu n} + \frac{\tilde{C}^2C_h^2d_w^3F^4T\iota^2\iota''}{\lambda\mu^2n^2} &\leq \frac{\tilde{C}d_wF^2\iota}{\mu n}, \\
\mu^2n^2\iota' + 4\lambda\mu nC\iota + \tilde{C}^2C_h^2d_w^2F^2T\iota^2\iota'' &\leq \lambda\mu n\tilde{C}\iota, \\
\tilde{C}^2C_h^2d_w^2F^2T\iota^2\iota'' - \tilde{C}\lambda\mu n\iota + \mu^2n^2\iota' + 4\lambda\mu nC\iota &\leq 0,
\end{aligned}$$

where the second and third lines are by rearrangement. A feasible solution on $\tilde{C}$ requires

$$\begin{aligned}
\lambda^2\mu^2n^2\iota^2 - 4C_h^2d_w^2F^2T\iota^2\iota''(\mu^2n^2\iota' + 4\lambda\mu nC\iota) &\geq 0, \\
\lambda^2\mu^2n - 4C_h^2d_w^2F^2T\iota''(\mu^2n\iota' + 4\lambda\mu C\iota) &\geq 0,
\end{aligned} \tag{15}$$

where the second line is by rearrangement. Substitute our choices of $\lambda = C_\lambda\sqrt{T}, n = \sqrt{T}$ and solve the quadratic inequality for C_λ; we get that it suffices to choose

$$C_\lambda = 4\sqrt{C_h^2d_w^2F^2\iota'\iota'' + \frac{16C^2C_h^4d_w^4F^4\iota^2\iota''^2}{\mu^2}} = \tilde{O}\left(\frac{d_w^2}{\mu}\right), \tag{16}$$

with assumption $d_w > \mu$. Check that C_λ depends only logarithmically on T and that it ensures eq. (15) holds, therefore certifying that a universal constant $\tilde{C}$ exists. Therefore, by induction, we prove that $\forall i \in [T]$ there exists a universal constant $\tilde{C}$ such that with probability $> 1 - \delta/2$,

$$\|\hat{w}_i - w^*\|_2^2 \leq \frac{\tilde{C}d_wF^2\iota}{\mu n}.$$

With this result, now we are ready to move to **Step 3**.

Step 3: Upper bound of $\|\hat{w}_t - w^*\|_{\Sigma_t}^2$. Multiply both sides of eq. (13) by $\Sigma_t^{\frac{1}{2}}$ and we have

$$\Sigma_t^{\frac{1}{2}}(\hat{w}_t - w^*) \leq \frac{1}{2}\Sigma_t^{-\frac{1}{2}}\left(\sum_{i=0}^{t-1}\nabla f_{x_i}(\hat{w}_i)\|w^* - \hat{w}_i\|_{\nabla^2 f_{x_i}(\tilde{w})}^2\right) + \Sigma_t^{-\frac{1}{2}}\left(\sum_{i=0}^{t-1}\nabla f_{x_i}(\hat{w}_i)\eta_i\right) + \lambda\Sigma_t^{-\frac{1}{2}}(\hat{w}_0 - w^*).$$

Take square of both sides and by inequality $(a+b+c)^2 \leq 4a^2 + 4b^2 + 4c^2$ we obtain

$$\|\hat{w}_t - w^*\|_{\Sigma_t}^2 \leq 4\left\|\sum_{i=0}^{t-1}\nabla f_{x_i}(\hat{w}_i)\eta_i\right\|_{\Sigma_t^{-1}}^2 + 4\lambda^2\|\hat{w}_0 - w^*\|_{\Sigma_t^{-1}}^2 + \left\|\sum_{i=0}^{t-1}\nabla f_{x_i}(\hat{w}_i)\|w^* - \hat{w}_i\|_{\nabla^2 f_{x_i}(\tilde{w})}^2\right\|_{\Sigma_t^{-1}}^2.$$

The remaining proof closely follows **Step 2**, i.e.,

$$\begin{aligned}
\|\hat{w}_t - w^*\|_{\Sigma_t}^2 &\leq 4d_w\sigma^2\iota' + \frac{4\lambda Cd_wF^2\iota}{\mu n} + \frac{\tilde{C}^2C_h^2d_w^2F^4\iota^2}{\mu^2n^2}\left(\sum_{i=0}^{t-1}\sqrt{\nabla f_{x_i}(\hat{w}_i)^\top\Sigma_t^{-1}\nabla f_{x_i}(\hat{w}_i)}\right)^2 \\
&\leq 4d_w\sigma^2\iota' + \frac{4\lambda Cd_wF^2\iota}{\mu n} + \frac{\tilde{C}^2C_h^2d_w^2F^4\iota^2}{\mu^2n^2}\left(\sum_{i=0}^{t-1}1\right)\left(\sum_{i=0}^{t-1}\nabla f_{x_i}(\hat{w}_i)^\top\Sigma_t^{-1}\nabla f_{x_i}(\hat{w}_i)\right) \\
&\leq 4d_w\sigma^2\iota' + \frac{4\lambda Cd_wF^2\iota}{\mu n} + \frac{\tilde{C}^2C_h^2d_w^3F^4t\iota''\iota^2}{\mu^2n^2} \\
&\leq \tilde{O}\left(d_w\sigma^2 + \frac{d_w^3}{\mu^2} + \frac{d_w^3t}{\mu^2T}\right),
\end{aligned}$$

where the last inequality is by our choices of $\lambda = C_\lambda \sqrt{T}, n = \sqrt{T}$. Therefore, our choice of

$$\beta_t = \tilde{O}\left(d_w \sigma^2 + \frac{d_w^3}{\mu^2} + \frac{d_w^3 t}{\mu^2 T}\right)$$

guarantees that w^* is always contained in Ball_t with probability $1-\delta$. □

C.4. Regret Analysis

Lemma C.8 (Restatement of Lemma 5.4)**.** *Set $\Sigma_t, \hat{w}_t, \beta_t$ as in eq. (3), (5), & (7) and suppose Assumption 3.1, 3.2, & 3.3 hold, then with probability $> 1-\delta$, w^* is contained in Ball_t. Define $u_t = \|\nabla f_{x_t}(\hat{w}_t)\|_{\Sigma_t^{-1}}$, then $\forall t \in [T]$ in Phase II of Algorithm 1,*

$$r_t \leq 2\sqrt{\beta_t} u_t + \frac{2\beta_t C_h}{\lambda}.$$

Proof. By definition of instantaneous regret r_t,

$$r_t = f_{x^*}(w^*) - f_{x_t}(w^*).$$

Recall the selection process of x_t and define $\tilde{w} = \text{argmax}_{w \in \text{Ball}_t} f_{x_t}(w)$,

$$r_t \leq f_{x_t}(\tilde{w}) - f_{x_t}(w^*) = (\tilde{w} - w^*)^\top \nabla f_{x_t}(\dot{w}),$$

where the equation is by first order Taylor's theorem and $\dot{w}$ lies between $\tilde{w}$ and w^* which means $\dot{w}$ is guaranteed to be in Ball_t since Ball_t is convex. Then, by adding and removing terms,

$$\begin{aligned}
r_t &= (\tilde{w} - \hat{w}_t + \hat{w}_t - w^*)^\top (\nabla f_{x_t}(\hat{w}_t) - \nabla f_{x_t}(\hat{w}_t) + \nabla f_{x_t}(\dot{w})) \\
&\leq \|\tilde{w} - \hat{w}_t\|_{\Sigma_t} \|\nabla f_{x_t}(\hat{w}_t)\|_{\Sigma_t^{-1}} + \|\hat{w}_t - w^*\|_{\Sigma_t} \|\nabla f_{x_t}(\hat{w}_t)\|_{\Sigma_t^{-1}} + (\tilde{w} - \hat{w}_t)^\top (\nabla f_{x_t}(\dot{w}_t) - \nabla f_{x_t}(\hat{w}_t)) \\
&\quad + (\hat{w}_t - w^*)^\top (\nabla f_{x_t}(\dot{w}) - \nabla f_{x_t}(\hat{w}_t)),
\end{aligned}$$

where the last inequality is due to Holder's inequality. By definitions of β_t in Ball_t and $u_t = \|\nabla f_{x_t}(\hat{w}_t)\|_{\Sigma_t^{-1}}$,

$$r_t \leq 2\sqrt{\beta_t} u_t + (\tilde{w} - \hat{w}_t)^\top (\nabla f_{x_t}(\dot{w}) - \nabla f_{x_t}(\hat{w}_t)) + (\hat{w}_t - w^*)^\top (\nabla f_{x_t}(\dot{w}) - \nabla f_{x_t}(\hat{w}_t)).$$

Again by first order Taylor's theorem where $\ddot{w}$ lies between $\dot{w}$ and $\hat{w}$ and thus $\ddot{w}$ lies in Ball_t,

$$\begin{aligned}
r_t &\leq 2\sqrt{\beta_t} u_t + (\tilde{w} - \hat{w}_t)^\top \Sigma_t^{\frac{1}{2}} \Sigma_t^{-\frac{1}{2}} \nabla^2 f_{x_t}(\ddot{w}) \Sigma_t^{-\frac{1}{2}} \Sigma_t^{\frac{1}{2}} (\dot{w} - \hat{w}_t) + (\hat{w}_t - w^*)^\top \Sigma_t^{\frac{1}{2}} \Sigma_t^{-\frac{1}{2}} \nabla^2 f_{x_t}(\ddot{w}) \Sigma_t^{-\frac{1}{2}} \Sigma_t^{\frac{1}{2}} (\dot{w} - \hat{w}_t) \\
&\leq 2\sqrt{\beta_t} u_t + \|(\tilde{w} - \hat{w}_t)^\top \Sigma_t^{\frac{1}{2}}\|_2 \|\Sigma_t^{-\frac{1}{2}} \nabla^2 f_{x_t}(\ddot{w}) \Sigma_t^{-\frac{1}{2}}\|_{\text{op}} \|\Sigma_t^{\frac{1}{2}} (\dot{w} - \hat{w}_t)\|_2 \\
&\quad + \|(\hat{w}_t - w^*)^\top \Sigma_t^{\frac{1}{2}}\|_2 \|\Sigma_t^{-\frac{1}{2}} \nabla^2 f_{x_t}(\ddot{w}) \Sigma_t^{-\frac{1}{2}}\|_{\text{op}} \|\Sigma_t^{\frac{1}{2}} (\dot{w} - \hat{w}_t)\|_2 \\
&\leq 2\sqrt{\beta_t} u_t + \frac{2\beta_t C_h}{\lambda},
\end{aligned}$$

where the second inequality is by Holder's inequality and the last inequality is due to definition of β_t in Ball_t, Assumption 3.2, and our choice of Σ_t. □

Lemma C.9 (Restatement of Lemma 5.5)**.** *Set $\Sigma_t, \hat{w}_t, \beta_t$ as in eq. (3), (5), & (7) and suppose Assumption 3.1, 3.2, & 3.3 hold, then with probability $> 1-\delta$, w^* is contained in Ball_t and $\forall t \in [T]$ in Phase II of Algorithm 1,*

$$\sum_{t=1}^{T} r_t^2 \leq 16\beta_T d_w \log\left(1 + \frac{TC_g^2}{d_w \lambda}\right) + \frac{8\beta_T^2 C_h^2 T}{\lambda^2}.$$

Proof. By Lemma 5.4 and inequality $(a+b)^2 \leq 2a^2 + 2b^2$,

$$\begin{aligned}\sum_{t=1}^{T} r_t^2 &\leq \sum_{t=1}^{T} 8\beta_t u_t^2 + \frac{8\beta_t^2 C_h^2}{\lambda^2} \\ &\leq 8\beta_T \sum_{i=1}^{T} u_t^2 + \frac{8\beta_T^2 C_h^2 T}{\lambda^2} \\ &\leq 16\beta_T d_w \log\left(1 + \frac{TC_g^2}{d_w \lambda}\right) + \frac{8\beta_T^2 C_h^2 T}{\lambda^2},\end{aligned}$$

where the second inequality is due to β_t is increasing in t and the last inequality is by Lemma C.6. □

By putting everything together, we are ready to prove the main cumulative regret theorem.

Proof of Theorem 4.1. By definition of cumulative regret including both Phase I and II,

$$\begin{aligned}R_{\sqrt{T}+T} &= \sum_{j=1}^{\sqrt{T}} r_j + \sum_{t=1}^{T} r_t \\ &\leq 2\sqrt{T}F + \sqrt{T\sum_{t=1}^{T} r_t^2} \\ &\leq 2\sqrt{T}F + \sqrt{16T\beta_T d_w \log\left(1 + \frac{TC_g^2}{d_w\lambda}\right) + \frac{8T^2\beta_T^2 C_h^2}{\lambda^2}} \\ &\leq \tilde{O}\left(\sqrt{T}F + \sqrt{T\beta_T d_w + \frac{T^2\beta_T^2}{\lambda^2}}\right),\end{aligned}$$

where the first inequality is due to function range and Cauchy-Schwarz inequality, the second inequality is by Lemma 5.5 and the last inequality is obtained by setting $\lambda = C_\lambda\sqrt{T}, n = \sqrt{T}$ as required by Lemma 5.3 where C_λ is in eq. (16).

Recall that β_t is defined in eq. (7), so

$$\beta_T = \tilde{O}\left(\frac{d_w^3}{\mu^2}\right).$$

The proof completes by plugging in upper bound of β_T. □

D. Additional Experimental Details

In addition to Experiments section in main paper, in this section, we show details of algorithm implementation and and real-world experiments.

D.1. Implementation of GO-UCB

Noise parameter $\sigma = 0.01$. Regression oracle in GO-UCB is approximated by stochastic gradient descent algorithm on our two linear layer neural network model with mean squared error loss, 2000 iterations and 10^{-11} learning rate. Exactly solving optimization problem in Step 5 of Phase II may not be computationally tractable, so we use iterative gradient ascent algorithm over x and w with 2000 iterations and 10^{-4} learning rate. β_t is set as $d_w^3 F^4 t/T$. λ is set as $\sqrt{T}\log^2 T$.

D.2. Real-world Experiments

Hyperparameters can be continuous or categorical, however, in order to fairly compare GO-UCB with Bayesian optimization methods, in all hyperparameter tuning tasks, we set function domain to be $[0, 10]^{d_x}$, a continuous domain. If a hyperparameter

is categorical, we allocate equal length domain for each hyperparameter. For example, the seventh hyperparameter of random forest is a bool value, True or False and we define $[0, 5)$ as True and $[5, 10]$ as False. If a hyperparameter is continuous, we set linear mapping from the hyperparameter domain to $[0, 10]$. For example, the sixth hyperparameter of multi-layer perceptron is a float value in $(0, 1)$ thus we multiply it by 10 and map it to $(0, 10)$.

Hyperparameters in hyperparameter tuning tasks. We list hyperparameters in all three tasks as follows.

Classification with Random Forest.

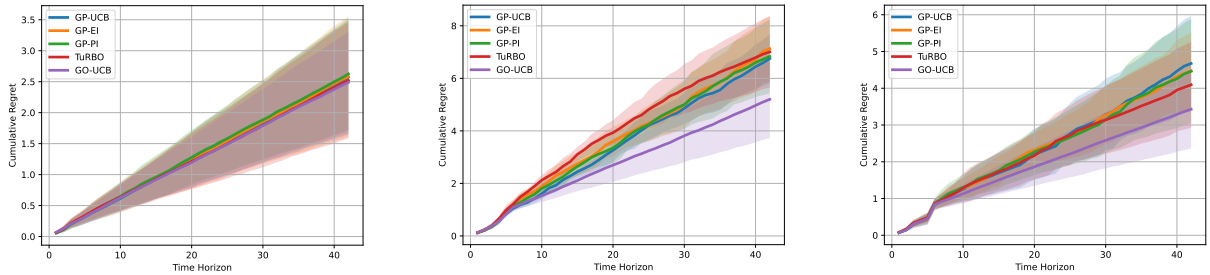
1. Number of trees in the forest, (integer, [20, 200]).
2. Criterion, (string, "gini", "entropy", or "logloss").
3. Maximum depth of the tree, (integer, [1, 10]).
4. Minimum number of samples required to split an internal node, (integer, [2, 10]).
5. Minimum number of samples required to be at a leaf node, (integer, [1, 10]).
6. Maximum number of features to consider when looking for the best split, (string, "sqrt" or "log2").
7. Bootstrap, (bool, True or False).

Classification with Multi-Layer Perceptron.

1. Activation function (string, "identity", "logistic", "tanh", or "relu").
2. Strength of the L2 regularization term, (float, $[10^{-6}, 10^{-2}]$).
3. Initial learning rate used, (float, $[10^{-6}, 10^{-2}]$).
4. Maximum number of iterations, (integer, [100, 300]).
5. Whether to shuffle samples in each iteration, (bool, True or False).
6. Exponential decay rate for estimates of first moment vector, (float, (0, 1)).
7. Exponential decay rate for estimates of second moment vector (float, (0, 1)).
8. Maximum number of epochs to not meet tolerance improvement, (integer, [1, 10]).

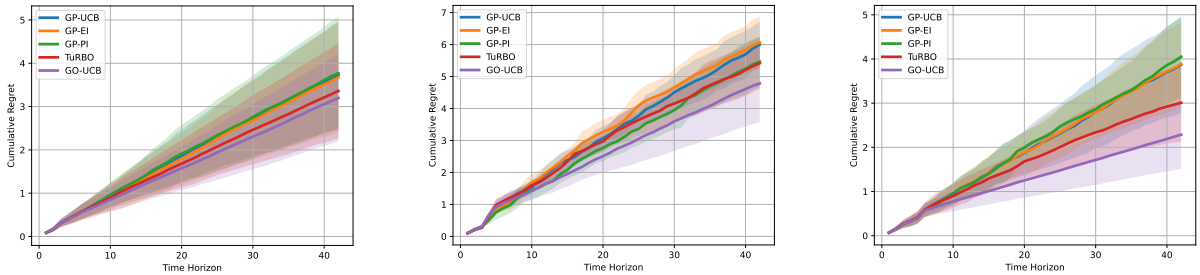
Classification with Gradient Boosting.

1. Loss, (string, "logloss" or "exponential").
2. Learning rate, (float, (0, 1)).
3. Number of estimators, (integer, [20, 200]).
4. Fraction of samples to be used for fitting the individual base learners, (float, (0, 1)).
5. Function to measure the quality of a split, (string, "friedman mse" or "squared error").
6. Minimum number of samples required to split an internal node, (integer, [2, 10]).
7. Minimum number of samples required to be at a leaf node, (integer, [1, 10]).
8. Minimum weighted fraction of the sum total of weights, (float, (0, 0.5)).
9. Maximum depth of the individual regression estimators, (integer, [1, 10]).
10. Number of features to consider when looking for the best split, (float, "sqrt" or "log2").



(a) Random forest ($d_x = 7$) (b) Multi-layer perceptron ($d_x = 8$) (c) Gradient boosting ($d_x = 11$)

Figure 4. Cumulative regrets (the lower the better) of all algorithms in real-world hyperparameter tuning task on Australian dataset.



(a) Random forest ($d_x = 7$) (b) Multi-layer perceptron ($d_x = 8$) (c) Gradient boosting ($d_x = 11$)

Figure 5. Cumulative regrets (the lower the better) of all algorithms in real-world hyperparameter tuning task on Diabetes dataset.

11. Maximum number of leaf nodes in best-first fashion, (integer, [2, 10]).

Results on Australian and Diabetes datasets. Due to page limit of the main paper, we show experimental results of hyperparameter tuning tasks on Australian and Diabetes datasets in Figure 4 and Figure 5. Our proposed GO-UCB algorithm performs consistently better than all other algorithms, which is the same as on Breast-cancer dataset in main paper.