Black-box Optimization with Unknown Constraints via Overparameterized Deep Neural Networks

Dat Phan-Trong[*1] **Hung The Tran**[2] **Sunil Gupta**[1]

[1]Deakin Applied Artificial Intelligence Initiative, Deakin University, Australia
[2]AI Center, VNPT Media , Vietnam

Abstract

Optimizing expensive black-box functions under unknown constraints is a fundamental challenge across a range of real-world domains, such as hyperparameter tuning in machine learning, safe control in robotics, and material or drug discovery. In these settings, each function evaluation may be costly or time-consuming, and the system may need to operate within unknown or difficult-to-specify safety boundaries. We apply the Expected Improvement (EI) acquisition function to select the next samples within a feasible region, determined by Lower Confidence Bound (LCB) conditions for all constraints. The LCB approach guarantees constraint feasibility, while EI efficiently balances exploration and exploitation, especially when the feasible regions are much smaller than the overall search space. To model both the objective function and constraints, we use Deep Neural Networks (DNNs) instead of Gaussian Processes (GPs) to improve scalability and handle complex structured data. We provide a theoretical analysis showing our method's convergence using recent Neural Tangent Kernel (NTK) theory. Under regularity conditions, both cumulative regret and constraint violation are bounded by the maximum information gain, with equivalent upper bounds to GP-based methods. To validate our algorithm, we conduct experiments on synthetic and real-world benchmarks, showing its benefit over recent methods in black-box optimization with unknown constraints.

1 INTRODUCTION

Global optimization of expensive black-box functions (Black-box Optimization, or BO) is a ubiquitous challenge in machine learning, control systems, and material design fields. These tasks often involve non-convex, multi-modal, and costly-to-evaluate functions, necessitating efficient exploration of the search space. Bayesian Optimization has emerged as a widely adopted model-based approach to address this. Bayesian Optimization builds a surrogate model, typically a Gaussian Process (GP), to approximate the unknown objective function from observed data points. This model guides the selection of new points, balancing exploration (sampling uncertain regions) and exploitation (focusing on promising areas). Classical techniques within Bayesian Optimization include Probability of Improvement (PI) [Kushner, 1964], Expected Improvement [Mockus et al., 1978], Gaussian Process Upper Confidence Bound (GP-UCB) [Srinivas et al., 2009], and information-theoretic approaches such as Entropy Search (ES) [Hennig and Schuler, 2012] and Predictive Entropy Search (PES) [Hernández-Lobato et al., 2014].

As real-world problems often involve constraints that are also black-box in nature, Constrained Black-box Optimization (CBO) has become a vital extension of BO. CBO methods adjust the acquisition function to account for these constraints, seeking feasible solutions that satisfy the conditions while optimizing the objective function. A prominent method in CBO is the Expected Improvement with Constraints (cEI), first introduced by Schonlau et al. [1998] and later extended by Gardner et al. [2014] and Gelbart et al. [2014]. cEI integrates feasibility into the acquisition function, directing the optimization process toward regions where feasible solutions are likely. Letham et al. [2019] further improved cEI by using a quasi-Monte Carlo approximation to better manage observation noise, enhancing its effectiveness in noisy environments.

EI-based methods for constrained optimization face several challenges. When no feasible point exists, the EI cannot be computed, leading to modifications that focus solely on finding feasible regions, ignoring the objective function. Additionally, numerical challenges further limit some methods like EVR and IECI to small-dimensional problems.

[*]Corresponding author: d.phantrong@deakin.edu.au

To address this, alternative methods have been proposed. For example, Predictive Entropy Search with Constraints (PESC, Hernández-Lobato et al., 2015) offers a heuristic approach that selects feasible candidates directly from the search space, reducing uncertainty more effectively. However, the computational challenges associated with quadrature calculations during sampling have limited its practical applicability. Recently, Takeno et al. [2022] proposed a Min-Value Entropy Search method that simplifies the sampling process, making it more tractable.

Numerical optimization has also been taken into consideration as an effective tool for solving the unknown constraint problem. The idea is to reformulate constraints into simpler unconstrained problems solved through alternating iterations. The Augmented Lagrangian method is mostly used in this category. For example, Gramacy et al. [2016] with Augmented Lagrangian Bayesian Optimization (ALBO) and its improvement Slack-AL [Picheny et al., 2016] use Augmented Lagrangian Function (ALF) to formulate unconstrained surrogate problems and then solve them using EI as an acquisition function. Recently, ADMMBO [Ariafar et al., 2019] first applied the ADMM technique to transform the constrained problem into an equivalent unconstrained optimization, then solved an augmented Lagrangian relaxation. However, this method requires the introduction of additional variables, leading to increased computational costs.

Recent research has explored penalty functions and primal-dual methods to handle constraint violations during optimization. For example, Lu and Paulson [2022] introduced a penalty-function approach that adds a penalty term for constraint violations to the objective function, transforming the constrained problem into an unconstrained one. Similarly, Zhou and Ji [2022] proposed a primal-dual approach that balances the trade-off between optimizing the objective and minimizing constraint violations. While these methods are promising, their effectiveness is sensitive to the choice of parameters set, often requiring considerable effort in parameter tuning during implementation.

Alongside empirical advancements, recent theoretical works have started to address the absence of formal guarantees in Constrained Black-box Optimization (CBO). For example, Lu and Paulson [2022] introduced a penalty-based regret bound that combines the regret from the objective function with penalties for constraint violations. Xu et al. [2023] expanded this analysis by separately evaluating cumulative regret and constraint violations. In contrast, Nguyen et al. [2023] provided a theoretical performance guarantee for CBO under unknown constraints in a *decoupled* setting, where cumulative regret is calculated as the sum of both objective function regret and constraint violations.

Despite the success of previous works using Gaussian Processes (GPs) to model both objective functions and constraints, GPs struggle with poor computational scalability. The kernel matrix inversion required in GP methods has cubic complexity, which increases significantly as the number of constraints grows. In contrast, Deep Neural Networks (DNNs) have become a popular alternative in various Machine Learning tasks, offering the ability to extract rich features and scale linearly with dataset size, providing a clear advantage over GPs. Recent research has explored the use of DNNs in unconstrained optimization, including black-box function optimization in continuous search spaces [Snoek et al., 2015] and contextual bandit problems in discrete search spaces [Zhou et al., 2020, Zhang et al., 2021]. However, to the best of our knowledge, the challenge of replacing GPs with neural networks for constrained optimization involving black-box, expensive constraints while providing theoretical guarantees remains largely unaddressed.

In this paper, we provide a simple approach for black-box optimization with unknown constraints using deep neural networks. Our contribution can be summarized in three folds:

- We propose a DNN-based black-box optimization algorithm with unknown constraints (Neural-CBO), where both the objective function and constraints are modeled using deep neural networks. We use EI as the acquisition function to find the next samples in a feasible region which is determined using Lower Confidence Bound (LCB) satisfaction conditions to all constraints. Using LCB-based conditions guarantees that the suggested regions encompass the actual feasible regions of the constraints (under our problem setting), while still allowing for constraints exploration. Meanwhile, EI efficiently balances exploration and exploitation in optimizing the objective function, especially when the feasible regions are significantly smaller than the overall search space.
- We provide a theoretical analysis of our proposed Neural-CBO algorithm based on recent advances in NTK theory. Under certain regularity assumptions, we show that cumulative regret as well as cumulative constraint violation has an upper bound of the form $\mathcal{O}(\gamma_T\sqrt{T})$, where γ_T is the maximum information gain. This result is comparable to previous GP-based methods. It is worth noting that, our DNN models only required the network width as $m = \Omega(T)$ for the convergence.
- We conduct benchmarking experiments on synthetic and real-world tasks to prove our algorithm's effectiveness empirically. The numerical results indicate that our algorithm achieves competitive performance with well-known approaches.

2 PROBLEM SETTING

In this paper, we tackle the problem of black-box optimization, where the search space is subject to constraints imposed by other unknown functions. These constraints arise from *real-valued* feedback $c_i(\mathbf{x})$, and the constraint condition $c_i(\mathbf{x})$ is satisfied if and only if $c_i(\mathbf{x}) \leq 0$. Formally, this problem is defined as follows:

$$\min_{\mathbf{x}\in\mathcal{D}} f(\mathbf{x}), \text{ subject to } c_i(\mathbf{x}) \leq 0, \text{ for all } i = 1, \ldots, K,$$

where $\mathcal{D} \subset \mathbb{R}^d$ is a bounded domain, and f and $\{c_i\}_{i=1}^K : \mathbb{R}^d \to \mathbb{R}$ are unknown functions that can be evaluated at specific points. We consider this problem in a **coupled** setting, where both the objective function and constraints are evaluated simultaneously.

3 NEURAL-CBO: NEURAL NETWORK BASED BLACK-BOX OPTIMIZATION WITH UNKNOWN CONSTRAINTS

In this section, we present Neural-CBO, a neural network-based approach to CBO. The complete algorithm is detailed in Algorithm 1. The key innovation of Neural-CBO lies in leveraging neural networks as substitutes for GPs, traditionally used in Bayesian Optimization, to model both the black-box objective function and constraints. We first describe the structure of the neural network surrogate model, followed by our algorithm.

3.1 THE NEURAL NETWORK FOR AN ARBITRARY FUNCTION f_a

Given a black-box, expensive function f_a, we use a fully connected neural network, denoted as $a(\mathbf{x}; \mathbf{W})$, to model f_a:

$$a(\mathbf{x}; \mathbf{W}) = \frac{\mathbf{q}^\top}{\sqrt{m}} \mathbf{D}^{(L)}(\mathbf{x}) \mathbf{W}^{(L)} \ldots \frac{1}{\sqrt{m}} \mathbf{D}^{(1)}(\mathbf{x}) \mathbf{W}^{(1)} \mathbf{x}, \tag{1}$$

where $\mathbf{q} \in \mathbb{R}^m$ is the last layer weight, $\mathbf{W}^{(1)} \in \mathbb{R}^{m\times d}$, $\mathbf{W}^{(l)} \in \mathbb{R}^{m\times m}$ for $2 \leq l \leq L$ is the weight of the l-th hidden layer. The matrix $\mathbf{D}^{(l)}(\mathbf{x})$ is associated with the ReLU activation function and is defined as:

$$\mathbf{D}^{(l)}(\mathbf{x}) = \text{diag}\{\mathbf{1}_{\{\langle w_i^{(l)}, \mathbf{h}^{(l-1)}(\mathbf{x})\rangle \geq 0\}}\} \in \mathbb{R}^{m\times m},$$

with m as the number of neurons in the hidden layer l, and $\mathbf{h}^{(l)}(\mathbf{x})$ is the output of the l-th layer given by

$$\mathbf{h}^{(l)}(\mathbf{x}) = \frac{1}{\sqrt{m}} \mathbf{D}^{(l)}(\mathbf{x}) \mathbf{W}^{(l)} \ldots \frac{1}{\sqrt{m}} \mathbf{D}^{(1)}(\mathbf{x}) \mathbf{W}^{(1)} \mathbf{x},$$

with $\mathbf{h}^{(0)}(\mathbf{x}) = \mathbf{x}$.

At time $t = 0$, each weight matrix $\mathbf{W}^{(l)}, 2 \leq l \leq L$ is initialized as $\begin{pmatrix} \boldsymbol{\Psi} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Psi} \end{pmatrix}$, where $\boldsymbol{\Psi}$ is a Gaussian random matrix with independent and identically distributed (i.i.d.) standard normal entries. Additionally, the outer weights $\mathbf{q} = (\hat{\mathbf{q}}, -\hat{\mathbf{q}})^\top$ are set as random variables, and each entry of $\mathbf{b}$ is set with an equal probability of being either -1 or 1, and remain fixed throughout the training process. This initialization method is commonly employed in the literature, as seen in works like Du et al. [2018], Arora et al. [2019], and it can be verified that, with this initialization scheme, $a(\mathbf{x}; \mathbf{W}_0) = 0$, for all input $\mathbf{x}$.

The neural network is trained by running the stochastic gradient descent on the streaming data in *one pass*. In particular, given the initialization $\{\mathbf{W}_0^{(l)}\}_{l=1}^L$ and last layer weight $\mathbf{q}$, the l-th layer weight matrix at the t-th iteration is updated by minimizing the L_2 loss as:

$$\mathbf{W}_{t+1}^{(l)} = \mathbf{W}_t^{(l)} + \alpha_t (y_t - a(\mathbf{x}_t; \mathbf{W}_t)) \frac{\partial a(\mathbf{x}_t; \mathbf{W}_t)}{\partial \mathbf{W}^{(l)}}, \tag{2}$$

where α_t is the step size, and $\{\mathbf{x}_t, y_t\}$ is the observation at the t-th optimization iteration.

To estimate the uncertainty of the function f_a modeled by $a(\mathbf{x}; \mathbf{W})$, we adopt the variance formula from recent advances in neural contextual bandits research [Zhou et al., 2020, Kassraie and Krause, 2022]:

$$\sigma_{a,t}(\mathbf{x}) = \sqrt{\mathbf{g}_a(\mathbf{x}; \mathbf{W}_0)^\top \mathbf{U}_{a,t-1}^{-1} \mathbf{g}_a(\mathbf{x}; \mathbf{W}_0)}, \tag{3}$$

where

$$\mathbf{g}_a(\mathbf{x}; \mathbf{W}) = \nabla_{\mathbf{W}} a(\mathbf{x}; \mathbf{W}), \text{ and}$$
$$\mathbf{U}_{a,t} = \mathbf{U}_{a,t-1} + \mathbf{g}_a(\mathbf{x}_t; \mathbf{W}_0) \mathbf{g}_a(\mathbf{x}_t; \mathbf{W}_0)^\top \tag{4}$$

3.2 NEURAL TANGENT KERNEL

Definition 3.1. Given an L-layer neural network $a(\mathbf{x}; \mathbf{W})$ with input $\mathbf{x}$ and parameter $\mathbf{W}$ as defined in Equation (1), a Neural Tangent Kernel (NTK) matrix $\mathbf{H}_t$ for a sequence of weights $\mathbf{W}_t$ can be defined as:

$$\mathbf{H}_t[i,j] \coloneqq \left\langle \frac{\partial a(\mathbf{x}_i; \mathbf{W}_t)}{\partial \mathbf{W}}, \frac{\partial a(\mathbf{x}_j; \mathbf{W}_t)}{\partial \mathbf{W}} \right\rangle = \sum_{l=1}^{L} \mathbf{H}_t^{(l)}[i,j],$$

where $\mathbf{H}_t^{(l)}[i,j] \coloneqq \left\langle \frac{\partial a(\mathbf{x}_i; \mathbf{W}_t)}{\partial \mathbf{W}^{(l)}}, \frac{\partial a(\mathbf{x}_j; \mathbf{W}_t)}{\partial \mathbf{W}^{(l)}} \right\rangle$ is the NTK from the l-th hidden layer, for all $1 \leq i, j \leq T$.

Next, we present the common and well-established assumptions. The following assumption indicates the smoothness property of the unknown function f_a.

Assumption 3.2. We assume that $f_a \in \mathcal{H}_{k_a}(\mathcal{D})$, where $\mathcal{H}_{k_a}(\mathcal{D})$ is the Reproducing Kernel Hilbert Space (RKHS) associated with a real-valued function f_a defined on the domain $\mathcal{D}$. This space is induced by the Neural Tangent Kernel

k_a, which arises from a neural network $a(\mathbf{x}; \mathbf{W})$. In particular, the RKHS $\mathcal{H}_{k_a}$ induces an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_{k_a}}$ with the reproducing property: for all $f_a \in \mathcal{H}_{k_a}(\mathcal{D})$, we have $f_a(\mathbf{x}) = \langle f_a, k_a(\cdot, \mathbf{x}) \rangle_{\mathcal{H}_{k_a}}$. The induced norm is bounded and serves as a measure of the smoothness of f_a w.r.t the kernel function k_a: $\|f_a\|_{\mathcal{H}_{k_a}} = \sqrt{\langle f_a, f_a \rangle_{\mathcal{H}_{k_a}}} \leq B_a$.

To ensure that the noise arising from querying unknown function f_a remains bounded and manageable, we impose the following assumption:

Assumption 3.3. We assume the noises $\{\zeta_t\}_{t=1}^T$ where $\zeta_t = o_t - f_a(\mathbf{x}_t)$ are conditionally sub-Gaussian with parameter $R_a > 0$, where $\{\zeta_t\}_{t=1}^T$ is assumed to capture the noises induced by querying the black-box, expensive function $f_a(\cdot)$.

$$\forall t \geq 0,\ \forall \lambda_a \in \mathbb{R},\ \mathbb{E}[e^{\lambda_a \zeta_t} | \mathcal{F}_{a,t-1}] \leq \exp\left(\frac{\lambda_a^2 R_a^2}{2}\right),$$

where $\mathcal{F}_{a,t-1}$ are the σ-algebra generated by the random variables $\{\mathbf{x}_i, \zeta_i\}_{i=1}^{t-1} \cup \{\mathbf{x}_t\}$.

3.3 MAXIMUM INFORMATION GAIN

Assume after t steps, the model $a(\mathbf{x}, \mathbf{W})$ receives an input sequence $\mathcal{X}_t = (\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_t)$ and observes noisy rewards $\mathbf{o}_t = (o_1, o_2, \dots, o_t)$, where $o_i = f_a(\mathbf{x}_i) + \zeta_i$. The *information gain* $I(\mathbf{o}_t; f_a)$ at step t, quantifies the reduction in uncertainty about f_a after observing $\mathbf{o}_t$, defined as the mutual information between $\mathbf{o}_t$ and f_a:

$$I(\mathbf{o}_t; f_a) := H(\mathbf{o}_t) - H(\mathbf{o}_t | f_a),$$

where H denotes the entropy function. Following Srinivas et al. [2009], the maximum information gain for the objective f_a can be calculated as:

$$\gamma_{a,t} = \max_{\mathcal{X}_t \subset \mathcal{D}, |\mathcal{X}_t| = t} \frac{1}{2} \log\det\left(\mathbf{I} + \lambda_a^{-1} \mathbf{H}_t\right),$$

where $\lambda_a > 0$ is a noise variance and $\mathbf{H}_t$ is the kernel matrix. In our case, $\mathbf{H}_t$ can be referred to as the NTK matrix associated with the NTK kernel defined in Section 3.2.

To manage the approximation error, several technical lemmas impose the following condition on the width of the neural network.

Condition 3.4. *Throughout the section, the width of each hidden layer m satisfies is assumed to satisfy:*

$$m \geq d^9 \exp\big(\Omega(\nu L C^L \log T)\big), \tag{5}$$

for some absolute constant C. Besides, the step size $\alpha_t \leq \frac{\nu}{t+1}$, where ν is a parameter and independent of dimension d and width m.

Before going to our main algorithm, we provide the confidence bound, which is a key component in many BO algorithms, to guide algorithm design and ensure theoretical guarantee. The lemma demonstrates that by following the network width condition stated in Condition 3.4, the prediction of the trained neural network $a(\cdot; \mathbf{W}_{t-1})$ is concentrated at the actual value of the function $f_a(\cdot)$.

Lemma 3.5. *Let Assumptions 3.2 and 3.3 hold. Using neural network $a(\mathbf{x}; \mathbf{W})$ satisfied Condition 3.4 to model an arbitrary function f_a. Setting the step size at training step t as $\alpha_t \leq \frac{\nu}{(T+1)^2}$, then for any $\delta \in (0, 1)$, with probability at least $1 - \delta \exp\big(\Omega(C^{-L} m^{1/36})\big)$, the following holds for all $\mathbf{x} \in \mathcal{D}$ and $1 \leq t \leq T$:*

$$|f_a(\mathbf{x}) - a(\mathbf{x}; \mathbf{W}_{t-1})| \leq \beta_{a,t} \sigma_{a,t-1}(\mathbf{x}) + \frac{\mathcal{E}(m)}{T+1},$$
$$\beta_{a,t} = \left(B_a + R_a \sqrt{\gamma_{t,a} + 2 + 2\log(1/\delta)}\right),$$
$$\mathcal{E}(m) = \mathcal{O}(C^{2L} L^{3/2} m^{11/36}).$$

Here, the coefficient $\beta_{a,t}$ control the uncertainty of $a(\mathbf{x}; \mathbf{W}_{t-1})$ about $f_a(\mathbf{x})$ at $\mathbf{x}$, while $\mathcal{E}(m)$ indicates the approximation error when using the neural network's output $a(\mathbf{x}; \boldsymbol{W})$ to learn the underlying function f_a.

To facilitate the following algorithm design and discussion, we introduce the lower confidence and upper confidence bound functions w.r.t the *arbitrary* function f_a:

$$\mathrm{LCB}_{a,t}(\mathbf{x}, \mathbf{W}_t) = a(\mathbf{x}, \mathbf{W}_t) - \beta_{a,t} \sigma_{a,t}(\mathbf{x}) - \frac{\mathcal{E}(m)}{T+1},$$
$$\mathrm{UCB}_{a,t}(\mathbf{x}, \mathbf{W}_t) = a(\mathbf{x}, \mathbf{W}_t) + \beta_{a,t} \sigma_{a,t}(\mathbf{x}) + \frac{\mathcal{E}(m)}{T+1},$$

where $\sigma_{a,t}(\mathbf{x})$ is calculated using the formulate given in Equation (3). Then, with high probability, f_a is bounded by $\mathrm{LCB}_{a,t}(\mathbf{x}, \mathbf{W}_t)$ and $\mathrm{UCB}_{a,t}(\mathbf{x}, \mathbf{W}_t)$ as in the following corollary:

Corollary 3.6. *Let Assumption 3.2, Assumption 3.3 and Condition 3.4 hold. Then with probability at least $1 - \delta \exp\big(\Omega(C^{-L} m^{1/36})\big)$, the following holds for all $\mathbf{x} \in \mathcal{D}$ and $1 \leq t \leq T$:*

$$f_a(\mathbf{x}) \in [\mathrm{LCB}_{a,t}(\mathbf{x}, \mathbf{W}_t), \mathrm{UCB}_{a,t}(\mathbf{x}, \mathbf{W}_t)].$$

3.4 NEURAL-CBO ALGORITHM

In the remaining parts of this paper, we refer to $v(\mathbf{x}; \boldsymbol{\theta})$ and $\{u_{c_i}(\mathbf{x}; \boldsymbol{\omega}_{c_i})\}_{i=1}^K$ as the neural network models for the unknown objective function f and constraints $\{c_i\}_{i=1}^K$, respectively.

Our algorithm starts by initializing the neural networks $v(\mathbf{x}; \boldsymbol{\theta})$ and $\{u_{c_i}(\mathbf{x}; \boldsymbol{\omega}_{c_i})\}_{i=1}^K$ using the initialization scheme described in Section 3.1. We use the EI acquisition function to identify the next samples within the feasible

region, determined by applying LCB-based conditions to all constraints. LCB conditions guarantee that the suggested regions include the true feasible regions of the constraints, allowing for both feasibility and exploration of the constraint boundaries. Meanwhile, EI effectively balances exploration and exploitation in the objective, which is especially important when the feasible region is significantly smaller than the overall search space. (Line 4). At each optimization iteration t, the next evaluation point $\mathbf{x}_t$ is determined by maximizing the acquisition function $\mathrm{EI}_{f,t}(\mathbf{x})$ subject to the lower confidence bound constraints for all unknown constraints $\{c_i(\mathbf{x})\}_{i=1}^K$:

$$\mathrm{LCB}_{c_i,t}(\mathbf{x}, \boldsymbol{\omega}_{c_i,t}) \leq 0, \forall i \in [K].$$

To handle noisy observations, we utilized the standard choice of the incumbent, which is the best value of the mean function so far: $\mu_t^+ = \max_{\mathbf{x}_k \in \mathcal{D}_{t-1}} v(\mathbf{x}_k; \boldsymbol{\theta}_{t-1})$, where the evaluations of both objective function and constraints on $\mathbf{x}_k$ yield noisy observations, such as the objective value $y_k = f(\mathbf{x}_k) + \epsilon_k$ and constraint values $\{z_{c_i,k}\}_{i=1}^K$, with each constraint observation given by $z_{c_i,k} = c_i(\mathbf{x}_k) + \eta_{c_i,k}$ and $\mathcal{D}_{t-1} = \{\mathbf{x}_k, y_k, z_{c_1,k}, \ldots, z_{c_K,k}\}_{k=1}^{t-1}$.

The standard approach for noisy EI formulation considers the difference between the predicted function value $v(\mathbf{x}; \boldsymbol{\theta}_{t-1})$ and the current best value of the mean function so far μ_t^+. However, due to the approximation error of the neural network model, using this standard noisy EI to select next queries may lead to suboptimal decisions. To mitigate this issue, we add $\mathcal{E}(m)$ as a correction term, leading to the modified EI formulation:

$$\mathrm{EI}_{f,t}(\mathbf{x}) = \mathbb{E}[\max\{0, v(\mathbf{x}; \boldsymbol{\theta}_{t-1}) - \mu_t^+ + \mathcal{E}(m)\}],$$

and achieve the closed form expression using similar technique proposed in Tran-The et al. [2022] as:

$$\mathrm{EI}_{f,t}(\mathbf{x}) = \rho(v(\mathbf{x}; \boldsymbol{\theta}_{t-1}) - \mu_t^+ + \mathcal{E}(m), \sigma_{f,t}(\mathbf{x})),$$

$$\text{where } \rho(u, v) = \begin{cases} u\boldsymbol{\Phi}(\frac{u}{v}) + v\phi(\frac{u}{v}), & \text{if } v > 0, \\ \max\{u, 0\}, & \text{if } v = 0. \end{cases}$$

Then, we updated the dataset $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \{\mathbf{x}_t, y_t, z_{c_1,t}, \ldots, z_{c_K,t}\}$. The parameters $\boldsymbol{\theta}$ (for the objective function) and $\{\boldsymbol{\omega}_{c_i}\}_{i=1}^K$ (for the constraints) are then updated separately by minimizing the L_2 loss on the new observation using stochastic gradient descent (SGD) described in Equation (2).

Figure 1 demonstrates the minimization of a 1D objective function $f(x) = \sin(x) + \sin\left(\frac{10x}{3}\right)$ under the constraint $c(x) = (x-7)^2 - 1 \leq 0$ at the 200th iteration. This example highlights the **three key components** of our approach: the neural network surrogate model, LCB-based constraint handling, and the use of EI as the acquisition function for the objective. The top panels of Figures 1a and 1b display the predicted mean and variance of the objective as modeled by a deep neural network, illustrating the behavior of the variance formula in Equation (3) and the operation of Algorithm 1. These plots show that the variance is lower in feasible regions and higher in infeasible regions, supporting the effectiveness of our uncertainty estimation.

Algorithm 1

Input: The input space $\mathcal{D}$, the optimization budget T, the number of constraints N

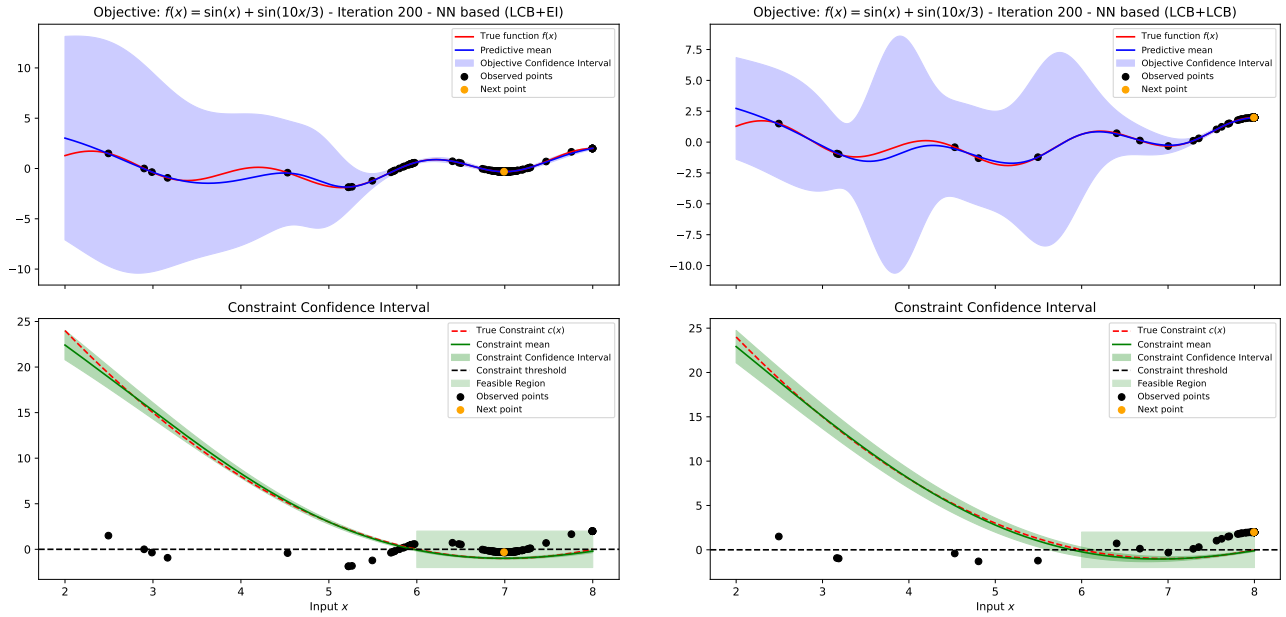
1: Initialize neural network models parameters $\boldsymbol{\theta}_0, \{\boldsymbol{\omega}_{c_i,0}\}_{i=1}^K$.
2: Initialize $\mathbf{U}_{f,0} = \mathbf{I}, \mathbf{U}_{c_i,0} = \mathbf{I}, \forall i \in [1 \ldots K]$,
3: **for** $t = 1$ to T **do**
4: Choose $\mathbf{x}_t = \mathrm{argmin}_{\mathbf{x} \in \mathcal{D}} \mathrm{EI}_{f,t}(\mathbf{x})$ subject to $\mathrm{LCB}_{c_i,t}(\mathbf{x}, \boldsymbol{\omega}_{c_i,t}) \leq 0, \forall i \in [K]$
5: Observe the noisy evaluations of objective function $y_t = f(\mathbf{x}_t) + \epsilon_t$ and constraints $\{z_{c_i,t} = c_i(\mathbf{x}_t) + \eta_{c_i,t}\}_{i=1}^K$.
6: Update observations set $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \{\mathbf{x}_t, y_t, z_{c_1,t}, \ldots, z_{c_K,t}\}$
7: Update the neural network parameters $\boldsymbol{\theta}_0, \{\boldsymbol{\omega}_{c_i,0}\}_{i=1}^K$ using Equation (2).
8: Update $\mathbf{U}_{f,t}$ and $\mathbf{U}_{c_i,t}, \forall i \in [1 \cdots K]$ separately using Equation (3).
9: **end for**

To clarify our use of LCB for constraints, we show the confidence intervals for the constraint function in the lower panels of the figures. These plots demonstrate that the true constraint $c(\mathbf{x})$ remains within the predicted confidence bounds, consistent with the theoretical guarantee in Corollary 3.6. Conditioning on the lower confidence bound allows our algorithm to reliably assess constraint satisfaction and identify feasible solutions. This accurate LCB estimation effectively guides the Expected Improvement (EI) acquisition for the objective, enabling the search to approach the true feasible minimum.

To justify our use of EI for the objective function, we compare Figure 1a (EI for the objective, LCB for constraints) with Figure 1b (LCB for both objective and constraints). These results highlight the advantage of our acquisition strategy: while LCB for constraints reliably guides the search toward feasible regions, applying LCB to the objective leads to overly exploratory behavior. In contrast, using EI for the objective alongside LCB for constraints (LCB-EI) achieves a more effective balance between exploration and exploitation, which is particularly beneficial when the feasible region is much smaller than the overall search space.

4 THEORETICAL ANALYSIS

In this section, we provide a theoretical analysis of our algorithm to offer insights on its convergence and sampling efficiency. Since the constraints are also black-box, we analyze a bound on the constraint violations.



(a) LCB for constraints, EI for objective.

(b) LCB for both constraints and objective.

Figure 1: Minimization results for the 1D objective $f(x) = \sin(x) + \sin\left(\frac{10x}{3}\right)$ under the constraint $c(x) = (x-7)^2 - 1 \leq 0$, using different acquisition strategies.

4.1 METRICS

To evaluate the performance of black-box optimization methods, much of the prior research on unconstrained Bayesian Optimization has focused on minimizing cumulative regret. The cumulative regret after T iterations is defined as $R_T = \sum_{t=1}^{T} r_t$, where $r_t = f(\mathbf{x}_t) - f(\mathbf{x}^*)$ represents the instantaneous regret, quantifying the difference between the value of the unknown function f at the optimal point, $\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{D}} f(\mathbf{x})$, and the value of the function at point $\mathbf{x}_t$, which is selected by the algorithm at iteration t. However, since $f(\mathbf{x}^*)$ represents the optimal value under constraints, the algorithm may sometimes sample infeasible points with lower objective values than $f(\mathbf{x}^*)$. To account for this, following Xu et al. [2023], Nguyen et al. [2023], we inherited the *positive regret* definition as $r_t^+ = [f(\mathbf{x}_t) - f(\mathbf{x}^*)]^+$, where $[\cdot]^+ := \max\{0, \cdot\}$. Additionally, to measure constraint satisfaction, constraint *violation* is defined as $v_{c_i,t} = [c_i(\mathbf{x}_t)]^+$. Then, we introduce the *cumulative positive regret* for the objective function, R_T^+, and the *cumulative violation* for each constraint, $V_{c_i,T}, \forall i \in [K]$. These metrics measure the additional cost incurred due to suboptimal decisions and violations of the constraints over time by running the algorithm.

Definition 4.1 (Cumulative Positive Regret and Cumulative Violation)**.**

$$R_T^+ = \sum_{t=1}^{T} [f(\mathbf{x}_t) - f(\mathbf{x}^*)]^+$$

$$V_{c_i,T} = \sum_{t=1}^{T} [c_i(\mathbf{x}_t)]^+$$

4.2 DETAILED ASSUMPTIONS FOR OBJECTIVE FUNCTION AND CONSTRAINTS

We apply the general assumption stated in the Assumption 3.2 and 3.3 on both objective function and constraints:

- **Objective function**: $f \in \mathcal{H}_{k_f}(\mathcal{D})$, $\|f_a\|_{\mathcal{H}_{k_f}} \leq B$, where k_f is corresponding to $v(\cdot, \boldsymbol{\theta})$. The noisy observation at step t is $y_t = f(\mathbf{x}_t) + \epsilon_t$, where $\{\epsilon_i\}_{i=1}^t$ is sub-Gaussian with parameter R_f and variance λ_f.
- **Constraint**: $c_i \in \mathcal{H}_{k_{c_i}}(\mathcal{D}), \|c_i\|_{\mathcal{H}_{k_{c_i}}} \leq S_i$, where k_{c_i} is corresponding to $u_{c_i}(\cdot, \boldsymbol{\omega}_{c_i}), \forall i = 1, \ldots, K$. The noisy observation at step t is $z_{c_i,t} = c_i(\mathbf{x}_t) + \eta_{c_i,t}$, where $\{\eta_{c_i,t}\}_{i=1}^t$ is sub-Gaussian with parameter R_{c_i} and variance λ_{c_i}.

We can now state our main theorem:

Theorem 4.2. *Under Assumption 3.2, Assumption 3.3 and Condition 3.4, set the step size used to train the neural networks in Algorithm 1 as* $\alpha_t \leq \frac{\nu}{(T+1)^2}$*, then for any* $\delta \in (0,1)$*, with probability at least* $1 - \delta \exp\left(\Omega(C^{-L} m^{1/36})\right)$*, the Cumulative Regret* R_T *and Cumulative Violation* $V_{c_i,T}$ *after* T *iterations are bounded as:*

$$V_{c_i,T} \leq 2\beta_{c_i,T} \sqrt{\frac{S_i T}{\log(S_i+1)}(2\gamma_{c_i,T} + 1)} + 2\mathcal{E}(m),$$

$$R_T \leq R_T^+ \leq 2\beta_{f,T}\sqrt{\frac{BT}{\log(B+1)}(2\gamma_{f,T}+1)} + 2\mathcal{E}(m),$$

where $\mathcal{E}(m) = \mathcal{O}(C^{2L}L^{3/2}m^{11/36})$. *Especially, by choosing* $m = \Omega(d^9 \exp(\nu LC^L \log T)))$, *the Cumulative Regret and Cumulative Violation enjoy the following results:*

$$V_{c_i,T} = \mathcal{O}(\gamma_{c_i,T}\sqrt{T}), \qquad R_T = \mathcal{O}(\gamma_{f,T}\sqrt{T}).$$

Remark 4.3. Unlike previous works [Zhou et al., 2020, Zhang et al., 2021] that require a neural network width of $m = \Omega(T^6)$ for convergence when modeling the objective function, our paper builds on recent analyzes from Xu and Zhu [2024], which show that only a linear condition of $m = \Omega(T)$ is needed. Furthermore, while Xu and Zhu [2024] focus on the input domain $\mathbb{S}^{d-1}$, we can adapt to inputs $\mathbf{x} \in \mathbb{R}^d$ with $0 < n_l < \|\mathbf{x}\| < n_b$ (where n_l and n_b are positive constants) without changing the order of T in the width condition for m. Similar arguments are noted in Du et al. [2018], Cao and Gu [2020].

5 EXPERIMENTAL RESULTS

In this section, we demonstrate the effectiveness of our proposed Neural-CBO algorithm through its application of synthetic benchmark optimization functions as well as real-world optimization problems. Our implementation is available at `https://github.com/phantrdat/neural-cbo`.

5.1 BASELINES

For all experiments, we compared our algorithm with well-known Constrained EI (cEI), the extension of EI into constrained BO from Gardner et al. [2014]. Besides, we also compare our algorithm with recent state-of-the-art algorithms in unknown constrained BO, including ADMMBO [Ariafar et al., 2019], UCB-C [Nguyen et al., 2023] and ConfigOpt [Xu et al., 2023]. For our proposed Neural-CBO algorithm, we employ fully connected deep neural networks as the surrogate models for both objective function and constraints. Due to space constraints, **implementation details of our Neural-CBO algorithm (including the choice of neural networks hyperparameter and other parameters in Algorithm 1) along with baseline implementations, are provided in Appendix A.1**.

5.2 SYNTHETIC BENCHMARK FUNCTIONS

We conducted optimization experiments on four synthetic objective functions: Branin, Ackley, Simionescu and Hartmann. The input dimension of each objective function and the corresponding number of constraints are summarized in Table 1. **We present the expression of each function and its constraints in the Appendix A.2**. The noise in function evaluations follows a normal distribution with zero mean, and the variance is set to 1% of the function range. All experiments reported here are averaged over 20 runs, each with random initialization. We report the (Log10 of) the Best Positive Regret plus Violation in Figure 2. We present justification for the choice of this metric as well as results for other metrics in Appendix A.4.

Table 1: The input dimension and number of constraints for each synthetic objective function.

Obj	Branin	Simionescu	Ackley	Hartmann
Dim	2	2	5	6
Constraints	1	1	2	1

To ensure statistical significance, we performed one-sided t-tests to assess whether a baseline outperforms Neural-CBO in terms of the best positive regret plus violation. The null hypothesis is $H_0 : \mu_{\text{baseline}} \leq \mu_{\text{Neural-CBO}}$, and the alternative hypothesis is $H_a : \mu_{\text{baseline}} > \mu_{\text{Neural-CBO}}$, where μ_{baseline} and $\mu_{\text{Neural-CBO}}$ represent the means of the (Log10 of) Best Positive Regret plus Violation values of the baseline and our proposed Neural-CBO, respectively. Note that lower values indicate better performance. We present the statistical test results for four synthetic benchmark functions and two real-world tasks (described in Section 5.3 and 5.4) in Table 2. Each cell in the table shows the p-value from the t-test as the first value. To account for multiple comparisons, the Benjamini-Hochberg correction was applied, with the corrected value provided as the second value. A result is labeled as "T" if the null hypothesis is rejected, meaning that Neural-CBO is statistically better to the compared baselines. Conversely, a result is labeled "F" if we cannot reject the null hypothesis, meaning that the baselines and the Neural-CBO are comparable. The results in Table 2 indicate that in 18 out of 24 comparisons, Neural-CBO achieves statistically better performance.

We analyze three real-world constrained black-box optimization tasks: gas transmission compressor and speed reducer designs from Kumar et al. [2020], and a third inspired by He et al. [2018]. Details of each task will follow in the upcoming sections.

5.3 GAS TRANSMISSION COMPRESSOR DESIGN

The main objective is to minimize operational costs or energy consumption. This requires identifying the optimal configuration of the compressor by optimizing four design variables. The problem involves $d = 4$ input dimensions and includes $K = 1$ constraint. The detailed mathematic

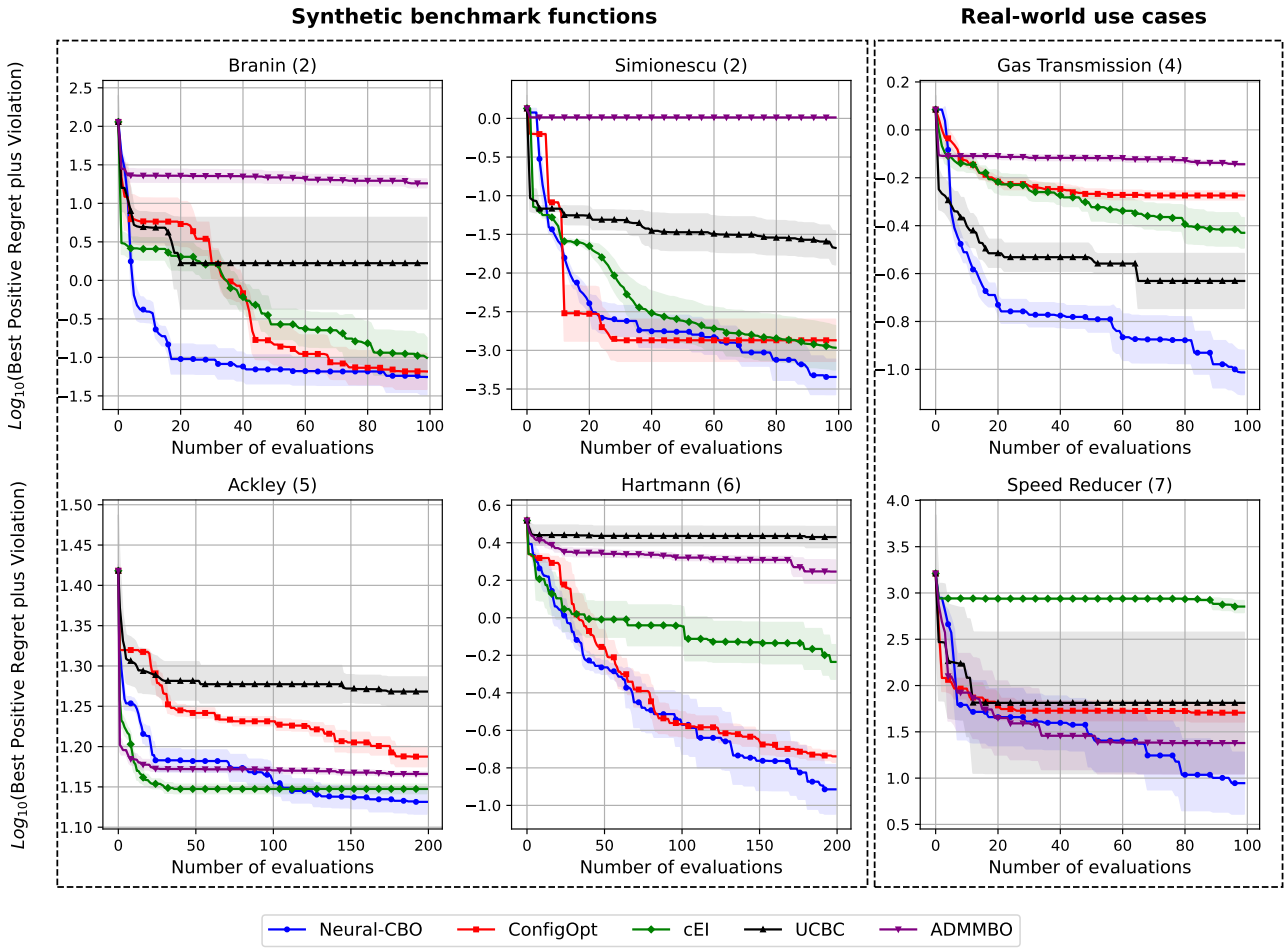


Figure 2: The plots show (Log10 of) the Best Positive Regret plus Violation up to step t, which is $\min_{t\in[T]}[f(\mathbf{x}_t) - f^*]^+ + \sum_{k=1}^{K}[c_k(\mathbf{x}_t)]^+]$, comparing our proposed algorithm and four baselines. The dimension of each objective function is shown in the parenthesis. The left group is four synthetic functions introduced in Section 5.2, while the right group is the optimization results of Gas Transmission Compressor Design and Speed Reducer Design, described in Section 5.3 and 5.4.

formula is provided in Appendix A.3.

5.4 SPEED REDUCER DESIGN

This task involves designing a speed reducer for a small aircraft engine, focusing on minimizing weight while meeting several constraints, including bending stress on gear teeth, surface stress, transverse deflections of shafts, and shaft stresses. The problem includes 7 decision variables and 11 constraints, resulting in an input dimension of $d = 7$ and $K = 11$ constraints. The mathematical formulation is provided in the Appendix A.3. We report numerical results of Section 5.3 and 5.4 in Figure 2 and Table 2.

5.5 DESIGNING SENSITIVE SAMPLES FOR MODEL TAMPERING DETECTION

We examine a scenario where a company offers Machine Learning as a Service (MLaaS) and hosts its model in the cloud. In this context, an attacker with system access could alter the model by changing its weights. To detect such tampering, He et al. [2018] propose generating a set of test inputs, called *Sensitive Samples* $\{v_i\}_{i=1}^n$, whose outputs from the modified model will differ from those of the original. Assuming a pre-trained model $s_\varphi(\mathbf{x})$ may have been altered after being uploaded, the goal is to find sensitive samples by solving the optimization problem:

$$v = \underset{\mathbf{x}}{\operatorname{argmax}} \left\| \frac{\partial s_\varphi(\mathbf{x})}{\partial \varphi} \right\|_F,$$

where $\|\cdot\|_F$ denotes the Frobenius norm. A detection is *successful* if at least one of the N_S sensitive samples shows a

Table 2: One-sided t-tests to evaluate whether the baseline outperforms Neural-CBO in terms of the "best positive regret plus violation" metric.

	ConfigOpt	cEI	UCB-C	ADMMBO
Branin	(3.76e-01, F)	(2.70e-01, F)	(3.36e-03, T)	(2.08e-12, T)
Simionescu	(0.30e-01, T)	(0.70e-01, F)	(1.42e-07, T)	(8.53e-15, T)
Ackley	(0.21e-03, T)	(0.77e-01, F)	(3.60e-08, T)	(0.16e-02, T)
Hartmann	(3.35e-02, T)	(2.79e-06, T)	(1.80e-11, T)	(2.61e-10, T)
Gas Transmission	(3.51e-10, T)	(2.23e-07, T)	(5.34e-04, T)	(1.84e-11, T)
Speed Reducer	(0.30e-01, F)	(5.83e-08, T)	(0.89e-01, T)	(1.08e-01, F)

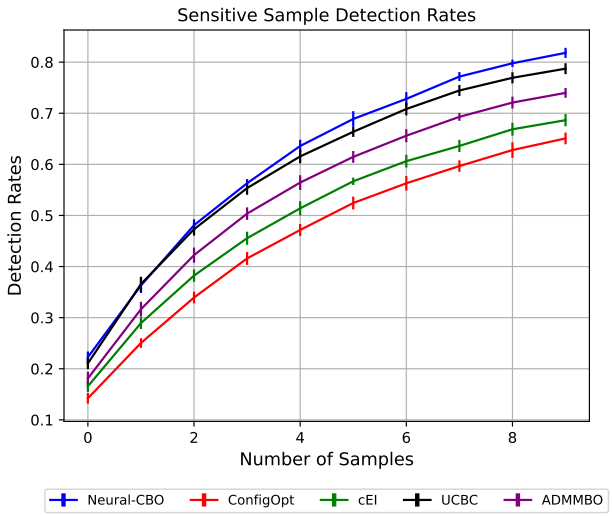


Figure 3: **Detection Rates** w.r.t to the number of samples for the MNIST dataset. As shown in the figure, Neural-CBO can generate sensitive samples that achieve nearly 85% of the detection rate with just 10 samples.

different top-1 prediction between the tampered and original models. To prevent attackers from evading detection, sensitive samples must resemble normal inputs. Therefore, a human-in-the-loop process is employed, where reviewers rate the realism of each sample on a scale of $(0, 1)$; higher scores indicate more realistic samples. These scores serve as constraints in the optimization, where obtaining human feedback can be costly.

We utilized a pre-trained MNIST handwritten digit classification model and compared our method's performance against several baselines based on average detection rates for sensitive samples. Feasible samples were chosen based on their realistic scores. Figure 3 shows the detection rates of (feasible) sensitive samples generated by our method compared to four baselines, demonstrating that our samples achieved higher detection rates. As expected, the detection rate improves with more samples and our method is consistently competitive.

6 CONCLUSION

We propose a novel algorithm for black-box optimization with unknown constraints, utilizing deep neural networks as surrogate models for both the objective function and constraints. Our algorithm leverages the bounded nature of constraint values by applying LCB conditions at each iteration to ensure feasibility. We also employ EI as the acquisition function to balance exploration and exploitation, especially in scenarios where feasible regions are significantly smaller than the search space. Our theoretical analysis shows that, under mild conditions regarding neural network width, our algorithm achieves upper bounds on cumulative regret and constraint violations comparable to previous GP-based methods. We validate our approach through experiments on synthetic and real-world benchmark tasks involving structural data, with results demonstrating competitive performance against state-of-the-art methods.

References

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International conference on machine learning*, pages 242–252. PMLR, 2019.

Setareh Ariafar, Jaume Coll-Font, Dana Brooks, and Jennifer Dy. Admmbo: Bayesian optimization with unknown constraints using admm. *Journal of Machine Learning Research*, 20(123):1–26, 2019.

Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pages 322–332. PMLR, 2019.

Yuan Cao and Quanquan Gu. Generalization error bounds of gradient descent for learning over-parameterized deep relu networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3349–3356, 2020.

Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, pages 844–853. PMLR, 2017.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2018.

Jacob R Gardner, Matt J Kusner, Zhixiang Eddie Xu, Kilian Q Weinberger, and John P Cunningham. Bayesian

optimization with inequality constraints. In *ICML*, volume 2014, pages 937–945, 2014.

Michael A Gelbart, Jasper Snoek, and Ryan P Adams. Bayesian optimization with unknown constraints. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pages 250–259, 2014.

Robert B Gramacy, Genetha A Gray, Sébastien Le Digabel, Herbert KH Lee, Pritam Ranjan, Garth Wells, and Stefan M Wild. Modeling an augmented lagrangian for blackbox constrained optimization. *Technometrics*, 58(1): 1–11, 2016.

Zecheng He, Tianwei Zhang, and Ruby B Lee. Verideep: Verifying integrity of deep neural networks through sensitive-sample fingerprinting. *arXiv preprint arXiv:1808.03277*, 2018.

Philipp Hennig and Christian J Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(6), 2012.

José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. *Advances in neural information processing systems*, 27, 2014.

José Miguel Hernández-Lobato, Michael Gelbart, Matthew Hoffman, Ryan Adams, and Zoubin Ghahramani. Predictive entropy search for bayesian optimization with unknown constraints. In *International conference on machine learning*, pages 1699–1707. PMLR, 2015.

Parnian Kassraie and Andreas Krause. Neural contextual bandits without regret. In *International Conference on Artificial Intelligence and Statistics*, pages 240–278. PMLR, 2022.

Abhishek Kumar, Guohua Wu, Mostafa Z Ali, Rammohan Mallipeddi, Ponnuthurai Nagaratnam Suganthan, and Swagatam Das. A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. *Swarm and Evolutionary Computation*, 56: 100693, 2020.

Harold J Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Fluids Engineering*, 86(1):97–106, 1964.

Benjamin Letham, Brian Karrer, Guilherme Ottoni, and Eytan Bakshy. Constrained bayesian optimization with noisy experiments. *Bayesian Analysis*, 14(2):495–519, 2019.

Congwen Lu and Joel A Paulson. No-regret bayesian optimization with unknown equality and inequality constraints using exact penalty functions. *IFAC-PapersOnLine*, 55(7):895–902, 2022.

Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. The application of bayesian methods for seeking the extremum. *Towards global optimization*, 2(117-129): 2, 1978.

Quoc Phong Nguyen, Wan Theng Ruth Chew, Le Song, Bryan Kian Hsiang Low, and Patrick Jaillet. Optimistic bayesian optimization with unknown constraints. In *The Twelfth International Conference on Learning Representations*, 2023.

Dat Phan-Trong, Hung Tran-The, and Sunil Gupta. Neuralbo: A black-box optimization algorithm using deep neural networks. *Neurocomputing*, 559:126776, 2023.

Victor Picheny, Robert B Gramacy, Stefan Wild, and Sebastien Le Digabel. Bayesian optimization under mixed constraints with a slack-variable augmented lagrangian. *Advances in neural information processing systems*, 29, 2016.

Matthias Schonlau, William J Welch, and Donald R Jones. Global versus local search in constrained optimization of computer models. *Lecture notes-monograph series*, pages 11–25, 1998.

Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Mostofa Patwary, Mr Prabhat, and Ryan Adams. Scalable bayesian optimization using deep neural networks. In *International conference on machine learning*, pages 2171–2180. PMLR, 2015.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

Shion Takeno, Tomoyuki Tamura, Kazuki Shitara, and Masayuki Karasuyama. Sequential and parallel constrained max-value entropy search via information lower bound. In *International Conference on Machine Learning*, pages 20960–20986. PMLR, 2022.

Hung Tran-The, Sunil Gupta, Santu Rana, and Svetha Venkatesh. Regret bounds for expected improvement algorithms in gaussian process bandit optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 8715–8737. PMLR, 2022.

Jiaming Xu and Hanjing Zhu. Overparametrized multilayer neural networks: Uniform concentration of neural tangent kernel and convergence of stochastic gradient descent. *Journal of Machine Learning Research*, 25(94): 1–83, 2024.

Wenjie Xu, Yuning Jiang, Bratislav Svetozarevic, and Colin Jones. Constrained efficient global optimization of expensive black-box functions. In *International Conference on Machine Learning*, pages 38485–38498. PMLR, 2023.

Fengxue Zhang, Zejie Zhu, and Yuxin Chen. Constrained bayesian optimization with adaptive active learning of unknown constraints. *arXiv preprint arXiv:2310.08751*, 2023.

Weitong Zhang, Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural thompson sampling. In *International Conference on Learning Representation (ICLR)*, 2021.

Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with ucb-based exploration. In *International Conference on Machine Learning*, pages 11492–11502. PMLR, 2020.

Xingyu Zhou and Bo Ji. On kernelized multi-armed bandits with constraints. *Advances in neural information processing systems*, 35:14–26, 2022.

Black-box Optimization with Unknown Constraints via Overparameterized Deep Neural Networks
Appendix

Dat Phan-Trong[*1] **Hung The Tran**[2] **Sunil Gupta**[1]

[1]Deakin Applied Artificial Intelligence Initiative, Deakin University, Australia
[2]AI Center, VNPT Media , Vietnam

A ADDITIONAL EXPERIMENTAL RESULTS

A.1 BASELINES

In this section, we briefly describe all baselines and our Neural-CBO implementations:

- **Constrained EI** (cEI) [Gardner et al., 2014] integrates feasibility into the acquisition function by multiplying the probability of feasibility into EI value at every point in the search space.
- **ConfigOpt** [Xu et al., 2023]: Optimize LCB-based acquisition function for the objective, which satisfies LCB-based conditions for constraints. For cEI and ConfigOpt, we used the public implementation provided at GitHub repository: `https://github.com/PREDICT-EPFL/ConfigOPT`.
- **ADMMBO** [Ariafar et al., 2019]: Reformulates the constrained optimization problem into an unconstrained one using the Alternating Direction Method of Multipliers (ADMM) framework. As the official implementation of ADMMBO is written in Matlab and available at `https://github.com/SetarehAr/ADMMBO`, we use our own Python implementation based on the official implementation.
- **UCB-C** [Nguyen et al., 2023]: Similar to ConfigOpt, but using a UCB-based acquisition function for the objective, we utilized the implementation obtained directly from the authors.

Neural-CBO implementation details: As described in Section 3, the network's weights are initialized with independent samples drawn from a normal distribution $\mathcal{N}(0, 1)$. We also initialize fixed outer weight $\mathbf{q}$ to be a symmetric Bernoulli random variable with equal probability to be -1 or 1. To train the surrogate neural network models, we use a Gradient Descent optimizer as described in the main paper, with a default learning rate of $\alpha = 1\text{e}{-4}$. However, α can be tuned within the range $(1\text{e}{-4}, 1\text{e}{-3})$ as needed. The network width depends on the tasks and is set to be $m = T$, where T is the number of optimization iterations. We choose the network depth $L = 2$ by default to reduce computational costs. Following Algorithm 1, we update the neural networks modeling the objective function and constraints using $\mathcal{D}_t$ after each optimization iteration with a single training pass.

In our implementation (as well as in other baselines), we discretized the search space, computed the values of EI (acquisition function for the objective) and LCB (acquisition function for the constraints) for all candidate points, and then selected the next evaluation point satisfying Line 4 in Algorithm 1. More details, we randomly sample 10k points and selecting the one with the highest acquisition function value. This approach was chosen as our paper does not focus on handling high-dimensional cases. Alternatively, a gradient-based approach could optimize the acquisition function using the Lagrangian method to combine the objective acquisition and constraint acquisition functions into a single unconstrained optimization problem.

[*]Corresponding author: d.phantrong@deakin.edu.au
[*]Corresponding author: d.phantrong@deakin.edu.au

To efficiently compute the inversion of matrix $\mathbf{U}_{a,t}$ in variance formula (3), we employ the Sherman-Morrison formula, taking advantage of the low-rank structure in the outer-products that add up to form that matrix.

A.2 SYNTHETIC BENCHMARK FUNCTIONS

We present the mathematical expressions of four synthetic objective functions and their accompanying constraints used for benchmarking in Section 5.2 of the main paper as follows:

Branin: We adopt this function from Letham et al. [2019].

$$\begin{aligned} f(\mathbf{x}) &= \left(\mathbf{x}_2 - \frac{5.1}{4\pi^2}\mathbf{x}_1^2 + \frac{5}{\pi}\mathbf{x}_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(\mathbf{x}_1) + 10, \\ \text{s.t. } c(\mathbf{x}) &= (\mathbf{x}_1 - 2.5)^2 + (\mathbf{x}_2 - 7.5)^2 - 50 \leq 0 \end{aligned}$$

where $\mathbf{x}_1 \in [-5, 10]$ and $\mathbf{x}_2 \in [0, 15]$.

Simionescu:

$$\begin{aligned} f(\mathbf{x}) &= 0.1\mathbf{x}_1\mathbf{x}_2 \\ \text{s.t. } c(\mathbf{x}) &= \mathbf{x}_1^2 + \mathbf{x}_2^2 - \left[1 + 0.2\cos\left(8\arctan\left(\frac{\mathbf{x}_2}{\mathbf{x}_1}\right)\right)\right]^2 \leq 0 \end{aligned}$$

Ackley: We inherited this function from Zhang et al. [2023].

$$\begin{aligned} f(\mathbf{x}) &= -20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d}\mathbf{x}_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi\mathbf{x}_i)\right) + 20 + e \\ \text{s.t. } &\begin{cases} c_1(\mathbf{x}) = 1 - (\|\mathbf{x} - \mathbf{1}\|_2 - 5.5)^2 & \leq 0 \\ c_2(\mathbf{x}) = \|\mathbf{x}\|_\infty^2 - 9 & \leq 0 \end{cases}, \end{aligned}$$

where $\mathbf{x} \in [-5, 3]^5$.

Hartmann This is a constrained version of the standard Hartmann test function that uses $\|\mathbf{x}\|_2 - 1 \leq 0$ as the constraint. This problem comes from Letham et al. [2019].

$$\begin{aligned} f(\mathbf{x}) &= -\sum_{i=1}^{4}\alpha_i \exp\left(-\sum_{j=1}^{6}\mathbf{A}_{ij}(\mathbf{x}_j - \mathbf{P}_{ij})^2\right) \\ \text{s.t. } c(\mathbf{x}) &= \|\mathbf{x}\|_2 - 1 \leq 0 \end{aligned}$$

where $\mathbf{x} \in [0, 1]^6$, and the constants are:

$$\alpha = (1.0, 1.2, 3.0, 3.2), \mathbf{A} = \begin{bmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{bmatrix}, \mathbf{P} = 10^{-4}\times \begin{bmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{bmatrix}.$$

A.3 REAL-WORLD APPLICATIONS:

Gas Transmission Compressor Design: The main objective is to minimize operational costs or energy consumption. This requires identifying the optimal configuration of the compressor by optimizing four design variables. The problem involves $d = 4$ input dimensions and includes $K = 1$ constraint. The mathematics formula for this problem is:

$$\begin{aligned} f(\mathbf{x}) &= 8.61 \times 10^5 \mathbf{x}_1^{1/2}\mathbf{x}_2\mathbf{x}_3^{-2/3}\mathbf{x}_4^{-1/2} + 3.69 \times 10^4\mathbf{x}_3 + 7.72 \times 10^8\mathbf{x}_1^{-1}\mathbf{x}_2^{0.219} - 765.43 \times 10^6\mathbf{x}_1^{-1}, \\ \text{s.t } c(\mathbf{x}) &= \mathbf{x}_4\mathbf{x}_2^{-2} + \mathbf{x}_2^{-2} - 1 \leq 0 \end{aligned}$$

Speed Reducer Design: This task involves designing a speed reducer for a small aircraft engine, focusing on minimizing weight while meeting several constraints, including bending stress on gear teeth, surface stress, transverse deflections of shafts, and shaft stresses. The problem includes 7 decision variables and 11 constraints, resulting in an input dimension of $d = 7$ and $K = 11$ constraints.

$$
\begin{aligned}
f(\mathbf{x}) &= 0.7854\mathbf{x}_2^2\mathbf{x}_1(14.9334\mathbf{x}_3 - 43.0934 + 3.3333\mathbf{x}_3^2) + 0.7854(\mathbf{x}_5\mathbf{x}_7^2 + \mathbf{x}_4\mathbf{x}_6^2) \\
&\quad - 1.508\mathbf{x}_1(\mathbf{x}_7^2 + \mathbf{x}_6^2) + 7.477(\mathbf{x}_7^3 + \mathbf{x}_6^3), \\
\text{s.t.} &\begin{cases}
c_1(\mathbf{x}) = -\mathbf{x}_1\mathbf{x}_2^2\mathbf{x}_3 + 27 & \leq 0 \\
c_2(\mathbf{x}) = -\mathbf{x}_1\mathbf{x}_2^2\mathbf{x}_3^2 + 397.5 & \leq 0 \\
c_3(\mathbf{x}) = -\mathbf{x}_2\mathbf{x}_6^4\mathbf{x}_3\mathbf{x}_4^{-3} + 1.93 & \leq 0 \\
c_4(\mathbf{x}) = -\mathbf{x}_2\mathbf{x}_7^4\mathbf{x}_3\mathbf{x}_5^{-3} + 1.93 & \leq 0 \\
c_5(\mathbf{x}) = 10\mathbf{x}_6^{-3}\sqrt{16.91 \times 10^6 + (745\mathbf{x}_4\mathbf{x}_2^{-1}\mathbf{x}_3^{-1})^2} - 1100 & \leq 0 \\
c_6(\mathbf{x}) = 10\mathbf{x}_7^{-3}\sqrt{157.5 \times 10^6 + (745\mathbf{x}_5\mathbf{x}_2^{-1}\mathbf{x}_3^{-1})^2} - 850 & \leq 0 \\
c_7(\mathbf{x}) = \mathbf{x}_2\mathbf{x}_3 - 40 & \leq 0 \\
c_8(\mathbf{x}) = -\mathbf{x}_1\mathbf{x}_2^{-1} + 5 & \leq 0 \\
c_9(\mathbf{x}) = -\mathbf{x}_1\mathbf{x}_2^{-1} - 12 & \leq 0 \\
c_{10}(\mathbf{x}) = 1.5\mathbf{x}_6 - \mathbf{x}_4 + 1.9 & \leq 0 \\
c_{11}(\mathbf{x}) = 1.1\mathbf{x}_7 - \mathbf{x}_5 + 1.9 & \leq 0
\end{cases},
\end{aligned}
$$

Designing Sensitive Samples for Detection of Model Tampering As described in Section 5.5, we used a pre-trained MNIST digit classification model and compared our method's detection performance with several baselines. The model was tampered with by adding noise to its weights 1,000 times, producing 1,000 distinct versions. While the original model had a top-1 accuracy of 93%, this dropped to $87.73\% \pm 0.08\%$ after tampering. To reduce computational costs, we downscaled the images from 28×28 to 7×7, optimized in this 49-dimensional space, and then restored them to the original resolution to generate sensitive samples.

A.4 FURTHER RESULTS

Justification for the metric used in the main paper: In the main paper, we report the (Log10 of) the Best Positive Regret plus Violation. To justify our choice of metrics, we consider the definitions of positive regret and violation in the context of a minimization problem and emphasize the following:

- When the selected point lies outside the feasible region and its objective function value exceeds the true optimum, the combined sum of positive regret and violation is large, effectively penalizing both infeasibility and poor performance.
- If the selected point is infeasible but has an objective function value lower than the true optimum, the violation term dominates the sum. This ensures that the penalty remains significant, especially when the point is far from the feasible region.
- When the selected point is within the feasible region, even if its objective function value is suboptimal, it still provides useful guidance by steering the algorithm toward feasible solutions, promoting further exploration in the correct direction.

To provide a more comprehensive evaluation of each method, we report results using three metrics: **Cumulative Positive Regret**, **Cumulative Violation**, and **Best Positive Regret for Feasible Points**. These metrics are illustrated in Figure 4, Figure 5, and Figure 6, respectively. As shown, Neural-CBO consistently achieves strong performance across all metrics, demonstrating both rapid convergence to the optimum and effective constraint satisfaction. In particular, for the best positive regret among feasible points, Neural-CBO reliably identifies feasible minima. Note that the starting points in Figure 6 differ between methods, as each algorithm may encounter the feasible region at different iterations.

Wall-clock Time Comparison: We provide the wall-clock running time of our algorithm and the considered baselines in the following table. We revisit six optimization tasks from our paper but increase the number of iterations for Hartmann ($D = 6$) and Speed Reducer ($D = 7$) to $n_{\text{iter}} = 1000$, to demonstrate the performance of neural networks when run for a

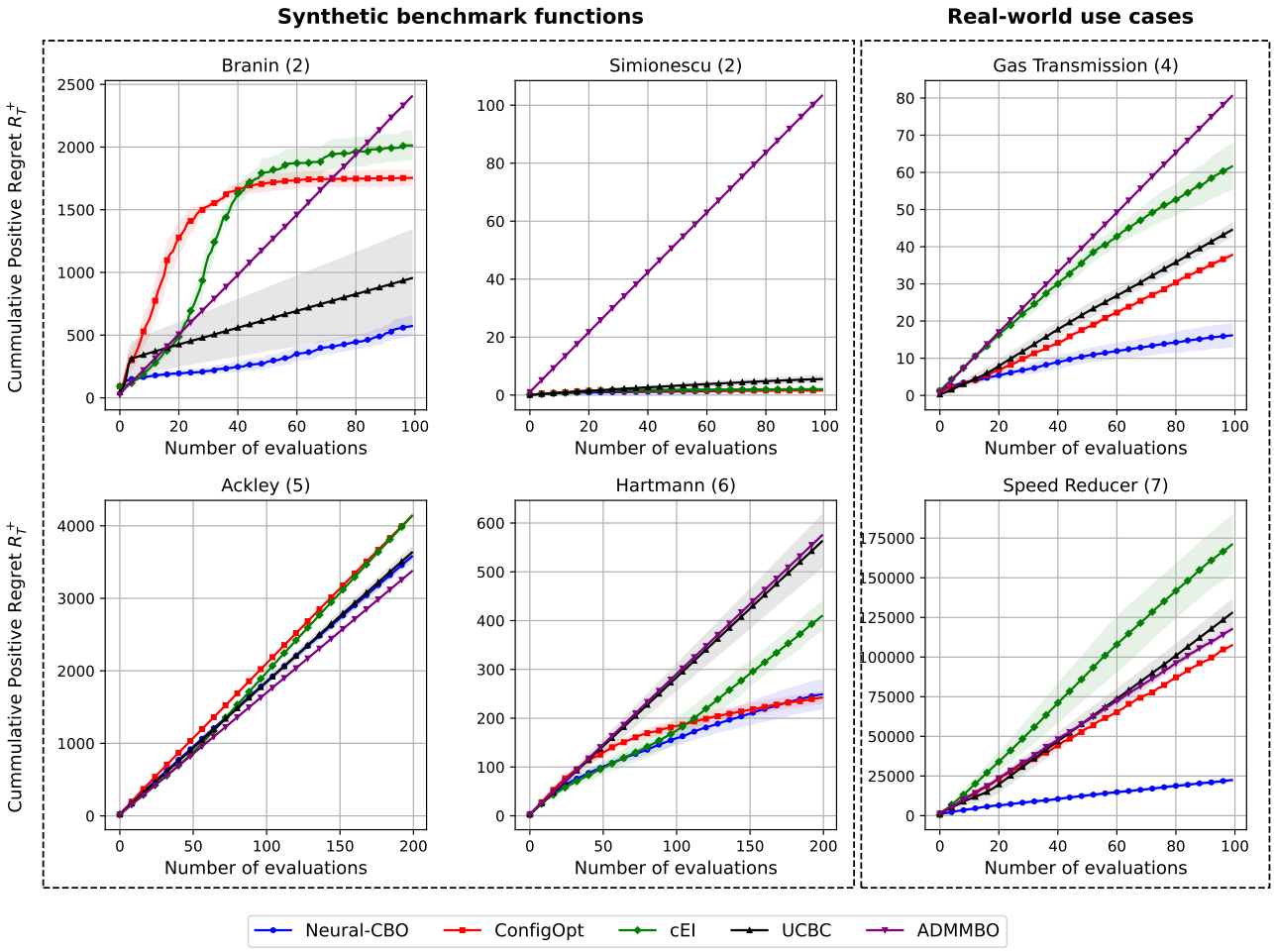


Figure 4: The plots show Cumulative Positive Regret up to step t, which is $\sum_{t=1}^{T}[f(\mathbf{x}_t) - f^*]^+$, comparing our proposed algorithm and four baselines. The dimension of each objective function is shown in the parenthesis. The left group is four synthetic functions introduced in Section 5.2, while the right group is the optimization results of Gas Transmission Compressor Design and Speed Reducer Design, described in Section 5.3 and 5.4.

large number of iterations. While GP-based methods appear more efficient in terms of runtime with a small number of iterations, this advantage diminishes as the number of iterations increases - an often necessary condition in high-dimensional or challenging problems to approach the global optimum. Due to their cubic time complexity with respect to the number of data points, GPs exhibit significant scalability limitations. In contrast, neural networks enjoy nearly linear scaling in runtime, highlighting their suitability and efficiency for higher number of iterations.

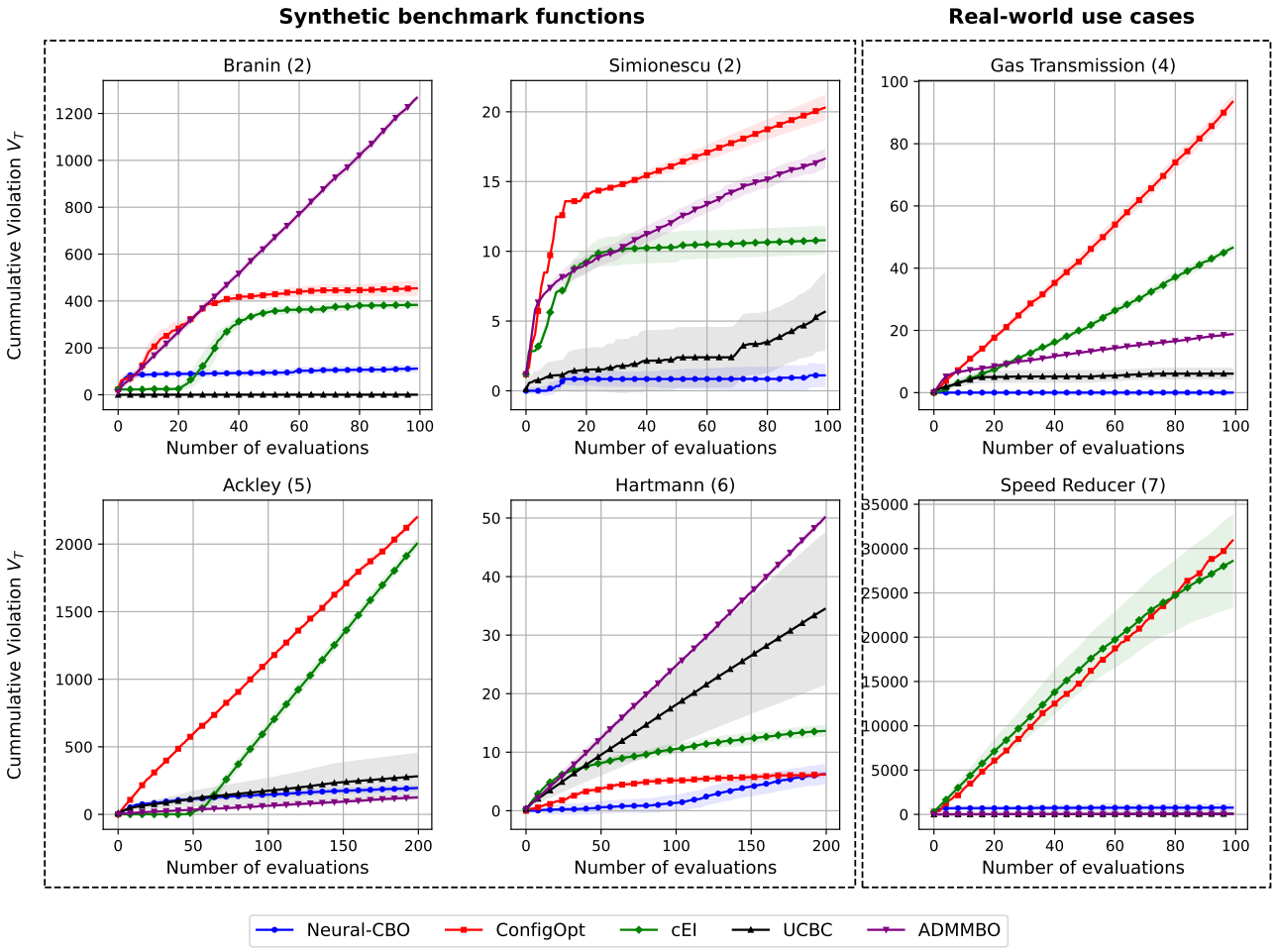


Figure 5: The plots show Cumulative Violation up to step t, which is $\sum_{t=1}^{T}[f(\mathbf{x}_t) - f^*]^+$, comparing our proposed algorithm and four baselines. The dimension of each objective function is shown in the parenthesis. The left group is four synthetic functions introduced in Section 5.2, while the right group is the optimization results of Gas Transmission Compressor Design and Speed Reducer Design, described in Section 5.3 and 5.4.

Objective	**DIM**	**N_ITERS**	**Neural-CBO**	**ConfigOpt**	**cEI**	**UCBC**	**ADMMBO**
Branin	2	100	40.63 ± 1.77	6.76 ± 4.54	6.31 ± 1.14	34.74 ± 5.15	342.24 ± 11.10
Simionescu	2	100	54.41 ± 2.13	18.19 ± 0.52	20.77 ± 7.31	19.94 ± 1.25	253.81 ± 5.45
Gas Transmission	4	200	125.83 ± 1.87	6.66 ± 0.46	8.56 ± 0.43	30.01 ± 1.45	468.67 ± 12.40
Ackley	5	200	85.26 ± 2.80	31.78 ± 4.71	31.44 ± 1.40	117.09 ± 40.86	309.13 ± 411.46
Hartmann	6	1000	561.06 ± 24.00	927.45 ± 45.09	472.24 ± 25.50	2321.70 ± 781.94	7196.00 ± 141.70
Speed Reducer	7	1000	721.44 ± 88.25	1003.93 ± 67.57	615.10 ± 30.52	2545.92 ± 512.82	8162.02 ± 133.70

Table 3: Wall-clock runtime (in seconds) of our algorithm and baselines across optimization tasks.

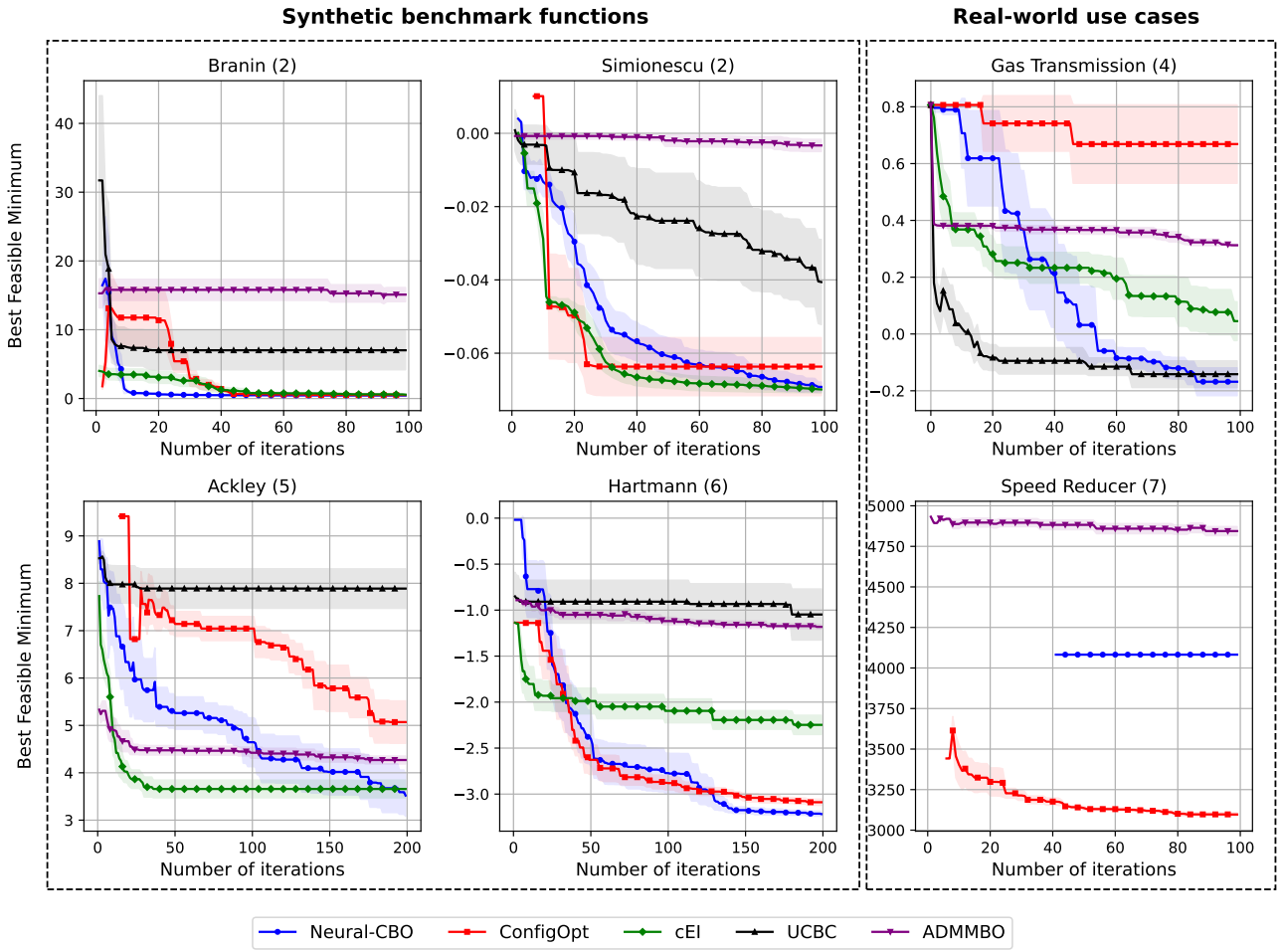


Figure 6: The plots show Best Feasible Minimum up to step t, which is $\min_{1\leq j\leq t} f(\mathbf{x}_j), \forall c_i(\mathbf{x}_j) \leq 0, \forall i = 1 \dots K$ (where K is the number of constraints), comparing our proposed algorithm and four baselines. The dimension of each objective function is shown in the parenthesis. The left group is four synthetic functions introduced in Section 5.2, while the right group is the optimization results of Gas Transmission Compressor Design and Speed Reducer Design, described in Section 5.3 and 5.4.

B DETAILED THEORETICAL ANALYSIS

In this section, we will provide the proof of Lemma 3.5 and Theorem 4.2. Before presenting the proofs, we briefly remind the reader of existing terms and introduce new notations for convenience. Remind that $\mathbf{g}_a(\mathbf{x}; \mathbf{W}) = \nabla_{\mathbf{W}} a(\mathbf{x}; \mathbf{W})$. Therefore, $\mathbf{g}_a(\mathbf{x}; \mathbf{W}_0)$ and $\mathbf{g}_a(\mathbf{x}; \mathbf{W}_t)$ will be the gradients of the neural network $a(\mathbf{x}; \mathbf{W})$ (using to model an *arbitrary* function f_a, defined in Equation (1)) at initialization and at iteration t, respectively. Further, let us define terms as follows:

$$
\begin{aligned}
\mathbf{G}_{a,t-1} &= [\mathbf{g}_a(\mathbf{x}_1; \mathbf{W}_0), \dots, \mathbf{g}_a(\mathbf{x}_{t-1}; \mathbf{W}_0)] \\
\bar{\mathbf{G}}_{a,t-1} &= [\mathbf{g}_a(\mathbf{x}_1; \mathbf{W}_{t-1}), \dots, \mathbf{g}_a(\mathbf{x}_{t-1}; \mathbf{W}_{t-1})] \\
\mathbf{U}_{a,t-1} &= \mathbf{I} + \mathbf{G}_{a,t-1}\mathbf{G}_{a,t-1}^\top \\
\mathbf{F}_{a,t-1} &= [f_a(\mathbf{x}_1), \dots, f_a(\mathbf{x}_{t-1})]
\end{aligned}
\tag{6}
$$

Further, it can be verified that $\mathbf{H}_0 = \mathbf{G}_{a,t-1}^\top \mathbf{G}_{a,t-1}$, where $\mathbf{H}_0$ is the NTK matrix at initialization defined in Section 3.2. Now we are ready to bound Lemma 3.5.

B.1 PROOF OF MAIN RESULTS PROVIDED IN SECTION 4

B.1.1 Proof of Lemma 3.5

Lemma 3.5. *Let Assumptions 3.2 and 3.3 hold. Using neural network $a(\mathbf{x};\mathbf{W})$ satisfied Condition 3.4 to model an arbitrary function f_a. Setting the step size at training step t as $\alpha_t \leq \frac{\nu}{(T+1)^2}$, then for any $\delta \in (0,1)$, with probability at least $1-\delta\exp\big(\Omega(C^{-L}m^{1/36})\big)$, the following holds for all $\mathbf{x}\in\mathcal{D}$ and $1\leq t\leq T$:*

$$
\begin{aligned}
&|f_a(\mathbf{x}) - a(\mathbf{x};\mathbf{W}_{t-1})| \leq \beta_{a,t}\sigma_{a,t-1}(\mathbf{x}) + \frac{\mathcal{E}(m)}{T+1},\\
&\beta_{a,t} = \left(B_a + R_a\sqrt{\gamma_{t,a} + 2 + 2\log(1/\delta)}\right),\\
&\mathcal{E}(m) = \mathcal{O}(C^{2L}L^{3/2}m^{11/36}).
\end{aligned}
$$

Proof. To prove Lemma 3.5, we analyze the left-hand side as follows:

$$
\begin{aligned}
&|f_a(\mathbf{x}) - a(\mathbf{x};\mathbf{W}_{t-1})|\\
&\leq \underbrace{|f_a(\mathbf{x}) - \langle \mathbf{g}_a(\mathbf{x}_t;\mathbf{W}_0), \mathbf{U}_{a,t-1}^{-1}\mathbf{G}_{a,t-1}\mathbf{y}_{a,t-1}\rangle|}_{T_1} + \underbrace{|a(\mathbf{x};\mathbf{W}_{t-1}) - \langle \mathbf{g}_a(\mathbf{x}_t;\mathbf{W}_0), \mathbf{U}_{a,t-1}^{-1}\mathbf{G}_{a,t-1}\mathbf{y}_{a,t-1}\rangle|}_{T_2}
\end{aligned}
$$

Here, T_1 represents the difference between the actual function value and the theoretical optimal solution for a linearized network. Meanwhile, T_2 refers to the gap between the neural network's output $a(\mathbf{x};\boldsymbol{W}_{t-1})$ at iteration $t-1$ and the theoretical optimal solution for the same linearized network.

Bound term T_1 :

First, following Assumption 3.2 in the main paper, we assume that f_a is in RKHS $\mathcal{H}_{k_a}$ with NTK kernel k_a, and $\mathbf{g}_a(\mathbf{x};\mathbf{W}_0)$ can be considered as finite approximation of $\varphi(\cdot)$, the feature map of the NTK from $\mathbb{R}^d \to \mathcal{H}_{k_a}$. From Lemma B.5, there exists $f_a^* \in \mathbb{R}^p$ such that $f_a(\mathbf{x}) = \langle \mathbf{g}_a(\mathbf{x};\mathbf{W}_0), f_a^*\rangle = \mathbf{g}_{\mathbf{a}}(\mathbf{x};\mathbf{W}_0)^\top f_a^*$. Then the term T_1 can be bounded as:

$$
\begin{aligned}
T_1 &= \left|f(\mathbf{x}) - \langle \mathbf{g}_a(\mathbf{x};\mathbf{W}_0); \mathbf{U}_{a,t-1}^{-1}\mathbf{G}_{a,t-1}\mathbf{y}_{a,t-1}\rangle\right|\\
&= \left|f(\mathbf{x}) - \mathbf{g}_a(\mathbf{x};\mathbf{W}_0)^\top \mathbf{U}_{a,t-1}^{-1}\mathbf{G}_{a,t-1}\mathbf{y}_{a,t-1}\right|\\
&\leq \left|f(\mathbf{x}) - \mathbf{g}_a(\mathbf{x};\mathbf{W}_0)^\top \mathbf{U}_{a,t-1}^{-1}\mathbf{G}_{a,t-1}\mathbf{f}_{a,t-1}\right| + \left|\mathbf{g}_a(\mathbf{x};\mathbf{W}_0)^\top \mathbf{U}_{a,t-1}^{-1}\mathbf{G}_{a,t-1}\boldsymbol{\epsilon}_{a,t-1}\right|\\
&= \left|\mathbf{g}_a(\mathbf{x};\mathbf{W}_0)^\top f_a^* - \mathbf{g}_a(\mathbf{x};\mathbf{W}_0)^\top \mathbf{U}_{a,t-1}^{-1}\mathbf{G}_{a,t-1}\mathbf{G}_{a,t-1}^\top f_a^*\rangle\right| + \left|\mathbf{g}_a(\mathbf{x};\mathbf{W}_0)^\top \mathbf{U}_{a,t-1}^{-1}\mathbf{G}_{a,t-1}\boldsymbol{\epsilon}_{a,t-1}\right|\\
&= \left|\mathbf{g}_a(\mathbf{x};\mathbf{W}_0)^\top \left(\mathbf{I} - \mathbf{U}_{a,t-1}^{-1}\mathbf{G}_{a,t-1}\mathbf{G}_{a,t-1}^\top\right) f_a^*\right| + \left|\mathbf{g}_a(\mathbf{x};\mathbf{W}_0)^\top \mathbf{U}_{a,t-1}^{-1}\mathbf{G}_{a,t-1}\boldsymbol{\epsilon}_{a,t-1}\right|\\
&= \left|\mathbf{g}_a(\mathbf{x};\mathbf{W}_0)^\top \left(\mathbf{I} - \mathbf{U}_{a,t-1}^{-1}\left(\mathbf{U}_{t-1} - \mathbf{I}\right)\right) f_a^*\right| + \left|\mathbf{g}_a(\mathbf{x};\mathbf{W}_0)^\top \mathbf{U}_{a,t-1}^{-1}\mathbf{G}_{a,t-1}\boldsymbol{\epsilon}_{a,t-1}\right|\\
&= \left|\mathbf{g}_a(\mathbf{x};\mathbf{W}_0)^\top \mathbf{U}_{a,t-1}^{-1}\mathbf{w}\right| + \left|\mathbf{g}_a(\mathbf{x};\mathbf{W}_0)^\top \mathbf{U}_{a,t-1}^{-1}\mathbf{G}_{a,t-1}\boldsymbol{\epsilon}_{a,t-1}\right|\\
&\leq \|f_a^*\|_{k_a}\left\|\mathbf{U}_{a,t-1}^{-1}\mathbf{g}_a(\mathbf{x};\mathbf{W}_0)\right\|_{k_a} + \left|\mathbf{g}_a(\mathbf{x};\mathbf{W}_0)^\top \mathbf{U}_{a,t-1}^{-1}\mathbf{G}_{a,t-1}\boldsymbol{\epsilon}_{a,t-1}\right|\\
&\leq \|f_a^*\|_{k_a}\sqrt{\mathbf{g}_a(\mathbf{x};\mathbf{W}_0)^\top \mathbf{U}_{a,t-1}^{-1}\mathbf{g}_a(\mathbf{x};\mathbf{W}_0)} + \left|\mathbf{g}_a(\mathbf{x};\mathbf{W}_0)^\top \mathbf{U}_{a,t-1}^{-1}\mathbf{G}_{a,t-1}\boldsymbol{\epsilon}_{a,t-1}\right|\\
&\leq \sqrt{2}B_a\sigma_{a,t}(\mathbf{x}) + \sigma_{a,t}(\mathbf{x})R\sqrt{\log\det(\mathbf{I}+\mathbf{H}_0) + 2\log(1/\delta)}\\
&\leq \sqrt{2}B_a\sigma_{a,t}(\mathbf{x}) + R\sqrt{\gamma_{a,t} + 2 + 2\log(1/\delta)}\sigma_{a,t}(\mathbf{x})\\
&= \left(\sqrt{2}B_a + R_a\sqrt{\gamma_{a,t} + 2 + 2\log(1/\delta)}\right)\sigma_{a,t}(\mathbf{x})
\end{aligned}
$$

where the first inequality uses triangle inequality and the fact that $\mathbf{y}_{a,t-1} = \mathbf{f}_{a,t-1} + \boldsymbol{\epsilon}_{a,t-1}$. The second inequality is from the reproducing property of function relying on RKHS, and the fourth equality is from the verification noted in Equation (6). The last inequality directly uses the results from Lemma B.10 and Lemma B.11.

Bound term T_2 To bound term T_2, we again divide $T2$ into two terms:

$$
T_2 = |a(\mathbf{x};\mathbf{W}_{t-1}) - \langle \mathbf{g}_a(\mathbf{x}_t;\mathbf{W}_0), \mathbf{U}_{a,t-1}^{-1}\mathbf{G}_{a,t-1}\mathbf{y}_{a,t-1}\rangle|
$$

$$= \underbrace{|a(\mathbf{x}; \mathbf{W}_{t-1}) - \langle \mathbf{g}_a(\mathbf{x}; \mathbf{W}_0), \mathbf{W}_{t-1} - \mathbf{W}_0 \rangle|}_{T_2'} + \underbrace{|\langle \mathbf{g}_a(\mathbf{x}; \mathbf{W}_0), \mathbf{W}_{t-1} - \mathbf{W}_0 \rangle - \langle \mathbf{g}_a(\mathbf{x}_t; \mathbf{W}_0), \mathbf{U}_{a,t-1}^{-1} \mathbf{G}_{a,t-1} \mathbf{y}_{a,t-1} \rangle|}_{T_2''}$$

$$\leq C^{2L} L^{3/2} m^{11/36}/(T+1) + C_1^{2L} L^{1/2} m^{-1/36}$$

Here, T_2' is the difference between the network output and its linear approximation, while T_2'' indicates the gap between the network's linear approximation and the theoretical optimal solution for a linearized network. The first inequality uses lemma B.8 and Lemma B.9. Combining the bound of term T_1 and T_2, then given any $\delta \in (0, 1)$, with probability at least $1 - \delta \exp\big(\Omega(C^{-L} m^{1/36})\big)$, we have:

$$|f_a(\mathbf{x}) - a(\mathbf{x}; \mathbf{W}_{t-1})| \leq \left(\sqrt{2} B + R \sqrt{\gamma_{t,a} + 2 + 2 \log(1/\delta)} \right) \sigma_{a,t-1}(\mathbf{x}) + \frac{\mathcal{E}(m)}{T+1},$$

where $\mathcal{E}(m) = \mathcal{O}(C^{2L} L^{3/2} m^{11/36})$. □

B.1.2 Proof of Theorem 4.2

Theorem 4.2. *Under Assumption 3.2, Assumption 3.3 and Condition 3.4, set the step size used to train the neural networks in Algorithm 1 as $\alpha_t \leq \frac{\nu}{(T+1)^2}$, then for any $\delta \in (0, 1)$, with probability at least $1 - \delta \exp\big(\Omega(C^{-L} m^{1/36})\big)$, the Cumulative Regret R_T and Cumulative Violation $V_{c_i,T}$ after T iterations are bounded as:*

$$V_{c_i,T} \leq 2\beta_{c_i,T} \sqrt{\frac{S_i T}{\log(S_i + 1)} (2\gamma_{c_i,T} + 1)} + 2\mathcal{E}(m),$$

$$R_T \leq R_T^+ \leq 2\beta_{f,T} \sqrt{\frac{BT}{\log(B+1)} (2\gamma_{f,T} + 1)} + 2\mathcal{E}(m),$$

where $\mathcal{E}(m) = \mathcal{O}(C^{2L} L^{3/2} m^{11/36})$. Especially, by choosing $m = \Omega(d^9 \exp\big(\nu L C^L \log T\big)))$, the Cumulative Regret and Cumulative Violation enjoy the following results:

$$V_{c_i,T} = \mathcal{O}(\gamma_{c_i,T} \sqrt{T}), \qquad R_T = \mathcal{O}(\gamma_{f,T} \sqrt{T}).$$

Proof. We gradually provide the upper bound of the cumulative regret R_T and cumulative violation of each constraint $V_{c_i,T}$ as:

Bound Cumulative Regret R_T: We utilize some results from Tran-The et al. [2022] presented in Lemma B.1 and Lemma B.2 in Section B.2 to bound our cumulative regret. We obtain an upper bound for the Cumulative Regret R_T as follows:

$$\begin{aligned}
R_T \leq R_T^+ &= \sum_{t=1}^{T} r_t^+ \\
&\leq \sum_{t=1}^{T} \left(C + \sqrt{2}\pi(B + \sqrt{2}) \right) (\mathrm{Im}_t(\mathbf{x}) + \beta_{f,t} \sigma_{f,t-1}(\mathbf{x}_t)) \\
&\leq \sum_{t=1}^{T} \left(C + \sqrt{2}\pi(B + \sqrt{2}) \right) \mathrm{Im}_t(\mathbf{x}) + \sum_{t=1}^{T} \left(C + \sqrt{2}\pi(B + \sqrt{2}) \right) \beta_{f,t} \sigma_{f,t-1}(\mathbf{x}_t) \\
&\leq \left(C + \sqrt{2}\pi(B + \sqrt{2}) \right) \sum_{t=1}^{T} \mathrm{Im}_t(\mathbf{x}) + \left(C + \sqrt{2}\pi(B + \sqrt{2}) \right) \beta_{f,T} \sum_{t=1}^{T} \sigma_{f,t-1}(\mathbf{x}) \\
&\leq \left(C + \sqrt{2}\pi(B + \sqrt{2}) \right) \sum_{t=1}^{T} \mathrm{Im}_t(\mathbf{x}) + \left(C + \sqrt{2}\pi(B + \sqrt{2}) \right) \beta_{f,T} \max \left(\sum_{t=1}^{T} \sigma_{f,t-1}(\mathbf{x}), B \right) \\
&\leq 2\beta_{f,T} \sqrt{\frac{BT}{\log(B+1)} (2\gamma_{f,T} + 1)} + 2\mathcal{E}(m)
\end{aligned}$$

The first inequality is from Lemma B.1 and the last inequality is due to Lemma B.2 and Lemma B.12.

Bound Cumulative Violation $V_{c_i,T}$:

$$
\begin{aligned}
V_{c_i,T} &= \sum_{t=1}^{T} [c_i(\mathbf{x}_t)]^+ \\
&= \sum_{t=1}^{T} [c_i(\mathbf{x}_t) - \mathrm{LCB}_{c_i,t}(\mathbf{x}, \boldsymbol{\omega}_{c_i,t}) + \mathrm{LCB}_{c_i,t}(\mathbf{x}, \boldsymbol{\omega}_{c_i,t})]^+ \\
&\leq \sum_{t=1}^{T} [c_i(\mathbf{x}_t) - \mathrm{LCB}_{c_i,t}(\mathbf{x}, \boldsymbol{\omega}_{c_i,t})]^+ + \sum_{t=1}^{T} [\mathrm{LCB}_{c_i,t}(\mathbf{x}, \boldsymbol{\omega}_{c_i,t})]^+ \\
&= \sum_{t=1}^{T} [c_i(\mathbf{x}_t) - \mathrm{LCB}_{c_i,t}(\mathbf{x}, \boldsymbol{\omega}_{c_i,t})]^+ \\
&\leq \sum_{t=1}^{T} [\mathrm{UCB}_{c_i,t}(\mathbf{x}, \boldsymbol{\omega}_{c_i,t}) - \mathrm{LCB}_{c_i,t}(\mathbf{x}, \boldsymbol{\omega}_{c_i,t})]^+ \\
&\leq 2\beta_{c_i,t} \sum_{t=1}^{T} \sigma_{c_i,t}(\mathbf{x}) + \frac{2\mathcal{E}(m)}{T+1} \\
&\leq 2\beta_{c_i,T} \max \left(\sum_{t=1}^{T} (\sigma_{c_i,t}(\mathbf{x}), S_i) + \frac{2\mathcal{E}(m)}{T+1} \right) \\
&\leq 2 \left(S_i + R_a \sqrt{\gamma_{a,T} + 2 + 2\log(1/\delta)} \right) \sqrt{\frac{S_i T}{\log(S_i+1)} (2\gamma_{c_i,T} + 1)} + 2\mathcal{E}(m)
\end{aligned}
$$

□

The first inequality follows by the fact that $[a+b]^+ \leq [a]^+ + [b]^+, \forall a, b \in \mathbb{R}$. The second equality is from the feasibility condition in Algorithm 1. The second inequality is from Corollary 3.6 and the last inequality is from Lemma 3.5.

B.2 TECHNICAL LEMMAS

The following lemmas provide the upper bound of simultaneous regret r_t and the upper bound on cumulative of improvement $\mathrm{Im}_t(\mathbf{x})$ function:

Lemma B.1 (Lemma 10, [Tran-The et al., 2022]). *There exist constant $C > 0$ such that*

$$
r_t \leq r_t^+ \leq \left(C + \sqrt{2}\pi(B + \sqrt{2}) \right) \left(Im_t(\mathbf{x}) + \beta_{f,t}\sigma_{f,t-1}(\mathbf{x}_t) \right),
$$

where $Im_t(\mathbf{x}) = \max(0, f(\mathbf{x}_t) - \mu_t^+ + \mathcal{E}(m))$, and $\mu_t^+ = \max_{\mathbf{x}_k \in \mathcal{D}_{t-1}} v(\mathbf{x}_k; \boldsymbol{\theta}_{t-1})$ is the best value of the mean objective function so far.

Lemma B.2 (Lemma 14, [Tran-The et al., 2022]). *Pick $\delta \in (0,1)$. Then with probability at least $1-\delta$ we have that:*

$$
\sum_{t=1}^{T} Im_t(\mathbf{x}_t) = \mathcal{O}(\beta_T \sqrt{T\gamma_T}) + \mathcal{E}(m).
$$

The following lemma gives the concentration of NTK at the initialization of the neural network introduced in Equation (1).

Lemma B.3 (Theorem 1, [Xu and Zhu, 2024]). *Under Gaussian initialization, for $m \geq Cd^2 \exp(L^2)$ for some constant C, there exist constants C_1, C_2 and C_3 such that, with probability at least $1 - \exp(-C_1 m^{1/3})$,*

$$
\left\| \mathbf{H}_0^{(l)} - \boldsymbol{\Phi}^{(l)} \right\|_\infty \leq C_2 \left(\frac{C_3^L}{m^{1/6}} + \sqrt{\frac{dL\log m}{m}} \right), \forall 1 \leq l \leq L,
$$

where $\boldsymbol{\Phi}^{(l)}$ is a deterministic kernel matrix. For more details about the recursive definition of $\boldsymbol{\Phi}^{(l)}$, see Section 4.1 of Xu and Zhu [2024].

The next lemma shows the bound of difference between the gradient of neural network $a(\mathbf{x};\mathbf{W})$ at step t and initialization.

Lemma B.4 (Proposition 9, [Xu and Zhu, 2024])**.** *Consider the neural network introduced in Equation* (1) *and assume that the condition 3.4 holds. With probability* $1-\exp\big(\Omega(C^{-L}m^{1/36})\big)$*, for any sample path* $\{\mathbf{x}_s,y_s\}_{s=0}^{T-1}$*, all* $t\leq T$*, we have*

$$\sup_{\mathbf{x}}\|\mathbf{g}_a(\mathbf{x},\mathbf{W}_t)-\mathbf{g}_a(\mathbf{x};\mathbf{W}_0)\|_2\leq C_1^{2L}L^{1/2}m^{-1/36}$$
$$\|\mathbf{g}_a(\mathbf{x};\mathbf{W}_0)\|_2\leq C_2^{L}L^{1/2},$$

for some constants C_1 *and* C_2*.*

The next lemma shows the reproducing property of function f_a being assumed to belong to RKHS induced by NTK kernel k_a of the neural network $a(\mathbf{x};\mathbf{W})$ introduced in Equation (1).

Lemma B.5. *Let* f_a *be a member of* $\mathcal{H}_{k_a}$ *with bounded RKHS norm* $\|f_a\|_{\mathcal{H}_{k_a}}\leq B_a$*. Assume that the network width of the model used to estimate function* $f_a(\cdot)$ *satisfies the Condition 3.4, then* $\forall\mathbf{x}\in D$*, there exists* $f_a^*\in\mathbb{R}^p$*, where* $p=md+m^2(L-2)+m$ *such that:*

$$f_a(\mathbf{x})=\langle\mathbf{g}_a(\mathbf{x};\mathbf{W}_0),f_a^*\rangle=\mathbf{g}_a(\mathbf{x};\mathbf{W}_0)^\top f_a^*$$

Proof of Lemma B.5. Due to Lemma B.3, with probability at least $1-\exp\big(-C_1m^{1/3}\log L\big)$, we have:

$$\|\mathbf{H}_0-\mathbf{\Phi}\|_\infty\leq C_2L\left(\frac{C_3^L}{m^{1/6}}+\sqrt{\frac{dL\log m}{m}}\right).$$

It is noted that following our definition, $\mathbf{H}_0=\mathbf{G}_{a,t}^\top\mathbf{G}_{a,t}$. That leads to:

$$\begin{aligned}\frac{1}{\sqrt{m}}\big\|\mathbf{G}_{a,t}^\top\mathbf{G}_{a,t}-\mathbf{\Phi}\big\|_F&\leq\frac{t}{\sqrt{m}}\big\|\mathbf{G}_{a,t-1}^\top\mathbf{G}_{a,t-1}-\mathbf{\Phi}\big\|_\infty\\&\leq\frac{t}{C_2\sqrt{m}L}\left(\frac{C_3^L}{m^{1/6}}+\sqrt{\frac{dL\log m}{m}}\right)\leq\lambda_0,\end{aligned}$$

Where λ_0 is a constant which is independent of m. The second inequality is from the choice of m in Condition 3.4. Then, we have:

$$\begin{aligned}\frac{1}{\sqrt{m}}\mathbf{G}_{a,t}^\top\mathbf{G}_{a,t}&\succcurlyeq\frac{1}{\sqrt{m}}\Big(\mathbf{\Phi}-\big\|\mathbf{G}_{a,t}^\top\mathbf{G}_{a,t}-\mathbf{\Phi}\big\|_F\mathbf{I}\Big)\\&\succcurlyeq\frac{1}{\sqrt{m}}\left(\mathbf{\Phi}-\lambda_0\mathbf{I}\right)\succ 0,\end{aligned}$$

suggests that $\mathbf{G}_{a,t}^\top\mathbf{G}_{a,t}$ is positive definite. Thus, suppose the singular value decomposition of $\mathbf{F}_{a,t-1}$ is $\mathbf{G}_{a,t}=\mathbf{P}_{a,t}\mathbf{A}_{a,t}\mathbf{Q}_{a,t}^\top$, then by choosing $f_a^*=\mathbf{P}_{a,t}\mathbf{A}_{a,t}\mathbf{Q}_{a,t}^\top\mathbf{F}_{a,t}$, we have

$$\mathbf{G}_{a,t-1}^\top f_a^*=\mathbf{Q}_{a,t}\mathbf{A}_{a,t}\mathbf{P}_{a,t}^\top\mathbf{P}_{a,t}\mathbf{A}_{a,t}\mathbf{Q}_{a,t}^\top\mathbf{F}_{a,t}=\mathbf{F}_{a,t},$$

which indicates that for any $\mathbf{x}$, $\langle g(\mathbf{x};\mathbf{W}_0),f_a^*\rangle=f_a(\mathbf{x})$. □

Let $\mathbf{z}_t^{(l)}(\mathbf{x})$ measure the sensitivity of the output from the l-th hidden layer and defined as:

$$\begin{aligned}[\mathbf{z}_t^{(l)}(\mathbf{x})]^\top&=\left[\frac{\partial a(\mathbf{x};\mathbf{W}_t)}{\partial\mathbf{h}_t^{(l)}(\mathbf{x})}\right]^\top\\&=\mathbf{q}^\top\frac{1}{\sqrt{m}}\mathbf{D}_t^{(L)}(\mathbf{x})\mathbf{W}_t^{(L)}\cdots\frac{1}{\sqrt{m}}\mathbf{D}_t^{(l+1)}(\mathbf{x})\mathbf{W}_t^{(l+1)},\end{aligned}$$

Then the following lemma provides the bound of the difference between $\mathbf{z}_t^{(l)}(\mathbf{x})$ and $\mathbf{z}_0^{(l)}(\mathbf{x})$:

Lemma B.6 (Lemma 12, [Xu and Zhu, 2024]). *Consider the neural network introduced in Equation* (1) *and assume that the condition 3.4 holds. With probability* $1-\exp\big(\Omega(C_1^{-L+l}m^{1/36})\big)$*, for layer* l *and any sample path* $\{\mathbf{x}_s, y_s\}_{s=0}^{T-1}$*, with all* $t \leq T$*, we have:*

$$\sup_{\mathbf{x}} \left\|\mathbf{z}_t^{(l)}(\mathbf{x}) - \mathbf{z}_0^{(l)}(\mathbf{x})\right\|_2 \leq \mathcal{O}(C_1^{2L-l}m^{17/36})$$
$$\sup_{\mathbf{x}} \left\|\mathbf{z}_0^{(l)}(\mathbf{x})\right\|_2 \leq C_2^{L-l}\sqrt{m}$$
$$\sup_{\mathbf{x}} \left\|\mathbf{z}_t^{(l)}(\mathbf{x})\right\|_2 \leq C_3^{2L-l-1}\sqrt{m}$$

for some absolute constant C_1, C_2, C_3*.*

The following lemma provides the bound on the difference between neural network weights and output at initialization and at step t:

Lemma B.7 (Lemma 10, [Xu and Zhu, 2024]). *Consider the neural network introduced in Equation* (1) *and assume that the condition 3.4 holds. Setting the step size at training step* t *as* $\alpha_t \leq \frac{\nu}{(T+1)^2}$*, then with probability* $1-\exp\big(\Omega(C^{-L}m^{1/36})\big)$*, for any sample path* $\{\mathbf{x}_s, y_s\}_{s=0}^{T-1}$*, all* $t \leq T$*, we have:*

$$\left\|\mathbf{W}_t^{(l)} - \mathbf{W}_0^{(l)}\right\|_2 \leq \frac{m^{1/3}L^{1/2}}{T+1}$$
$$\left\|\mathbf{W}_0^{(l)}\right\|_2, \left\|\mathbf{W}_t^{(l)}\right\|_2 \leq \mathcal{O}(\sqrt{m})$$
$$\sup_{\mathbf{x}} \left\|\mathbf{h}_t^{(l)}(\mathbf{x}) - \mathbf{h}_0^{(l)}(\mathbf{x})\right\|_2 \leq \frac{C_3^l}{m^{1/6}},$$

for some absolute constant C_3*.*

The following lemmas provide bound on the technical terms used in Lemma 3.5.

Lemma B.8. *Let* $a(\mathbf{x};\mathbf{W})$ *is the neural network defined in Equation* (1)*. Then, with probability* $1-\exp\big(\Omega(C^{-L}m^{1/36})\big)$*, we have:*

$$|a(\mathbf{x},\mathbf{W}_{t-1}) - a(\mathbf{x},\mathbf{W}_0) - \langle \mathbf{g}_a(\mathbf{x},\mathbf{W}_0), \mathbf{W}_{t-1}-\mathbf{W}_0\rangle| \leq \mathcal{O}(C^{2L}L^{3/2}m^{11/36})$$

Proof of Lemma B.8. Remind that

$$\mathbf{h}^{(l)}(\mathbf{x}) = \frac{1}{\sqrt{m}}\mathbf{D}^{(l)}(\mathbf{x})\mathbf{W}^{(l)}\ldots\frac{1}{\sqrt{m}}\mathbf{D}^{(1)}(\mathbf{x})\mathbf{W}^{(1)}\mathbf{x},$$

Then, by direct calculation, we have

$$\begin{aligned}\frac{\partial a(\mathbf{x};\mathbf{W}_0)}{\partial \mathbf{W}^{(l)}} &= \frac{\mathbf{q}^\top}{\sqrt{m}}\mathbf{D}_0^{(L)}(\mathbf{x})\mathbf{W}_0^{(L)}\ldots\frac{1}{\sqrt{m}}\mathbf{D}_0^{(l)}(\mathbf{x})\left[\mathbf{h}_0^{(l-1)}(\mathbf{x})\right]^\top \\ &= \frac{1}{\sqrt{m}}[\mathbf{z}_0^{(l)}(\mathbf{x})]^\top\mathbf{D}_0^{(l)}(\mathbf{x})\left[\mathbf{h}_0^{(l-1)}(\mathbf{x})\right]^\top,\end{aligned}$$

and

$$\mathbf{g}_a(\mathbf{x};\mathbf{W}_0) = \left[\frac{\partial a(\mathbf{x};\mathbf{W}_0)}{\partial \mathbf{W}^{(1)}}, \frac{\partial a(\mathbf{x};\mathbf{W}_0)}{\partial \mathbf{W}^{(2)}}, \ldots, \frac{\partial a(\mathbf{x};\mathbf{W}_0)}{\partial \mathbf{W}^{(L)}}\right]$$

We also rewrite $a(\mathbf{x};\mathbf{W}_{t-1})$ and $a(\mathbf{x};\mathbf{W}_0)$ as:

$$a(\mathbf{x};\mathbf{W}_{t-1}) = \frac{\mathbf{q}^\top}{\sqrt{m}}\mathbf{D}_{t-1}^{(L)}(\mathbf{x})\mathbf{W}_{t-1}^{(L)}\ldots\frac{1}{\sqrt{m}}\mathbf{D}_{t-1}^{(1)}(\mathbf{x})\mathbf{W}_{t-1}^{(1)}(\mathbf{x})$$

$$
\begin{aligned}
&= \frac{1}{\sqrt{m}}\mathbf{z}_{t-1}^{(l)}(\mathbf{x})\mathbf{D}_{t-1}^{(l)}(\mathbf{x})\mathbf{W}_{t-1}\mathbf{h}_{t-1}^{(l-1)}(\mathbf{x}),\\
a(\mathbf{x};\mathbf{W}_0) &= \frac{\mathbf{q}^\top}{\sqrt{m}}\mathbf{D}_0^{(L)}(\mathbf{x})\mathbf{W}_0^{(L)}\cdots\frac{1}{\sqrt{m}}\mathbf{D}_0^{(1)}(\mathbf{x})\mathbf{W}_0^{(1)}(\mathbf{x})\\
&= \frac{1}{\sqrt{m}}\mathbf{z}_0^{(l)}(\mathbf{x})\mathbf{D}_0^{(l)}(\mathbf{x})\mathbf{W}_0^{(l)}\mathbf{h}_0^{(l-1)}(\mathbf{x})
\end{aligned}
$$

Using the technique in the proof of Lemma 8.2 in Allen-Zhu et al. [2019], there exist diagonal matrices $\widehat{\mathbf{D}}^{(l)}(\mathbf{x}) = \mathbf{D}_{t-1}^{(l)}(\mathbf{x}) - \mathbf{D}_0^{(l)}(\mathbf{x}) \in \mathbb{R}^{m\times m}, \forall 1 \le l \le L$ with entries in $[-1,1]$ such that:

$$
\begin{aligned}
&a(\mathbf{x},\mathbf{W}_{t-1}) - a(\mathbf{x},\mathbf{W}_0)\\
&= \frac{1}{\sqrt{m}}\sum_{l=1}^{L}\left[\prod_{r=l+1}^{L}\left(\widehat{\mathbf{D}}^{(r)}(\mathbf{x}) + \mathbf{D}_{t-1}^{(r)}(\mathbf{x})\right)\mathbf{W}_{t-1}^{(r)}\right]\left(\widehat{\mathbf{D}}^{(l)}(\mathbf{x}) + \mathbf{D}_{t-1}^{(l)}(\mathbf{x})\right)(\mathbf{W}_{t-1}^{(l)} - \mathbf{W}_0^{(l)})\mathbf{h}_0^{(l-1)}(\mathbf{x})\\
&= \frac{1}{\sqrt{m}}\sum_{l=1}^{L}\widehat{\mathbf{z}}_{t-1}^{(l)}\left(\widehat{\mathbf{D}}^{(l)}(\mathbf{x}) + \mathbf{D}_{t-1}^{(l)}(\mathbf{x})\right)\left(\mathbf{W}_{t-1}^{(l)} - \mathbf{W}_0^{(l)}\right)\mathbf{h}_0^{(l-1)}(\mathbf{x})
\end{aligned}
$$

Furthermore, we have

$$
\langle \mathbf{g}_a(\mathbf{x},\mathbf{W}_0), \mathbf{W}_{t-1} - \mathbf{W}_0\rangle = \frac{1}{\sqrt{m}}\sum_{l=1}^{L}\mathbf{z}_0^{(l)}(\mathbf{x})\mathbf{D}_0^{(l)}(\mathbf{x})\left(\mathbf{W}_{t-1}^{(l)} - \mathbf{W}_0^{(l)}\right)\mathbf{h}_0^{(l-1)}(\mathbf{x})
$$

Replacing all below expressions, we get

$$
\begin{aligned}
&|a(\mathbf{x},\mathbf{W}_{t-1}) - a(\mathbf{x},\mathbf{W}_0) - \langle \mathbf{g}_a(\mathbf{x},\mathbf{W}_0), \mathbf{W}_{t-1} - \mathbf{W}_0\rangle|\\
&= \frac{1}{\sqrt{m}}\sum_{l=1}^{L}\widehat{\mathbf{z}}_{t-1}^{(l)}(\mathbf{x})\left(\widehat{\mathbf{D}}^{(l)}(\mathbf{x}) + \mathbf{D}_{t-1}^{(l)}(\mathbf{x})\right)\left(\mathbf{W}_{t-1}^{(l)} - \mathbf{W}_0^{(l)}\right)\mathbf{h}_0^{(l-1)}(\mathbf{x})\\
&- \frac{1}{\sqrt{m}}\sum_{l=1}^{L}\mathbf{z}_0^{(l)}(\mathbf{x})\mathbf{D}_0^{(l)}(\mathbf{x})\left(\mathbf{W}_{t-1}^{(l)} - \mathbf{W}_0^{(l)}\right)\mathbf{h}_0^{(l-1)}(\mathbf{x})\\
&\le \frac{1}{\sqrt{m}}\sum_{l=1}^{L}\left\|\widehat{\mathbf{z}}_{t-1}^{(l)} - \mathbf{z}_0^{(l)}(\mathbf{x})\right\|_2\left\|\mathbf{W}_{t-1}^{(l)} - \mathbf{W}_0^{(l)}\right\|_2\left\|\mathbf{h}_0^{(l-1)}\right\|\\
&\le Lm^{-1/2}L^{1/2}m^{1/3}C^{2L}m^{17/36}/(T+1)\\
&\le (C^{2L}L^{3/2}m^{11/36})/(T+1).
\end{aligned}
$$

□

The first inequality uses triangle inequality. The second inequality is from Lemma B.6 and Lemma B.7.

Lemma B.9. *Let $a(\mathbf{x};\mathbf{W})$ is the neural network defined in Equation* (1)*. Then, with probability $1 - \exp\big(\Omega(C^{-L}m^{1/36})\big)$, we have:*

$$
|\langle \mathbf{g}_a(\mathbf{x};\mathbf{W}_0), \mathbf{W}_{t-1} - \mathbf{W}_0 - \mathbf{U}_{a,t-1}^{-1}\mathbf{G}_{a,t-1}\mathbf{y}_{t-1}\rangle| \le C_1^{2L}L^{1/2}m^{-1/36}.
$$

Proof of Lemma B.9. Using the model update formula given in Equation (2), we have

$$
\begin{aligned}
\mathbf{W}_{t-1} - \mathbf{W}_0 &= (\mathbf{W}_{t-1} - \mathbf{W}_{t-2}) + (\mathbf{W}_{t-2} - \mathbf{W}_{t-3}) + \cdots + (\mathbf{W}_1 - \mathbf{W}_0)\\
&= \sum_{i=1}^{t-1}(\mathbf{W}_i - \mathbf{W}_{i-1})\\
&= \sum_{i=1}^{t-1}\alpha_i\left(y_i - a(\mathbf{x}_i,\mathbf{W}_{i-1})\right)\nabla_{\mathbf{W}}a(\mathbf{x}_i,\mathbf{W}_{i-1})
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{i=1}^{t-1} \alpha_i \left(y_i - a(\mathbf{x}_i, \mathbf{W}_{i-1})\right) \mathbf{g}_{a,i-1}(\mathbf{x}_i, \mathbf{W}_{i-1}) \\
&= \alpha \bar{\mathbf{G}}_{a,t-1}(\mathbf{y}_{t-1} - \mathbf{A}_{t-1}),
\end{aligned}
$$

where $\mathbf{A}_{t-1} = [a(\mathbf{x}_1, \mathbf{W}_1), \ldots, a(\mathbf{x}_{t-1}, \mathbf{W}_{t-1})] \in \mathbb{R}^{t-1}$. Then we have:

$$
\begin{aligned}
&|\mathbf{W}_{t-1} - \mathbf{W}_0 - \mathbf{U}_{a,t-1}^{-1} \mathbf{G}_{a,t-1} \mathbf{y}_{t-1}| \\
&= |\alpha \bar{\mathbf{G}}_{a,t-1}(\mathbf{y}_{t-1} - \mathbf{A}_{t-1}) - \mathbf{U}_{a,t-1}^{-1} \mathbf{G}_{a,t-1} \mathbf{y}_{t-1}| \\
&= |\alpha(\bar{\mathbf{G}}_{a,t-1} - \mathbf{G}_{a,t-1})(\mathbf{y}_{t-1} - \mathbf{A}_{t-1}) + \alpha \mathbf{G}_{a,t-1}(\mathbf{y}_{t-1} - \mathbf{A}_{t-1}) - (\mathbf{I} + \mathbf{G}_{a,t-1}\mathbf{G}_{a,t-1}^\top)^{-1} \mathbf{G}_{a,t-1} \mathbf{y}_{t-1}| \\
&= |\alpha(\bar{\mathbf{G}}_{a,t-1} - \mathbf{G}_{a,t-1})(\mathbf{y}_{t-1} - \mathbf{A}_{t-1}) + \alpha \mathbf{G}_{a,t-1}(\mathbf{y}_{t-1} - \mathbf{A}_{t-1}) - \mathbf{G}_{a,t-1}(\mathbf{I} + \mathbf{G}_{a,t-1}^\top \mathbf{G}_{a,t-1})^{-1} \mathbf{y}_{t-1}| \\
&\leq \left\|\alpha(\bar{\mathbf{G}}_{a,t-1} - \mathbf{G}_{a,t-1})(\mathbf{y}_{t-1} - \mathbf{A}_{t-1})\right\|_2 + \alpha \|\mathbf{G}_{a,t-1}\|_2 \left\|\mathbf{y}_{t-1}\left[\mathbf{I} - (\alpha \mathbf{I} + \alpha \mathbf{G}_{a,t-1}^\top \mathbf{G}_{a,t-1})^{-1}\right] - \mathbf{A}_{t-1}\right\|_2 \\
&\leq |\alpha|\sqrt{t}\left\|(\bar{\mathbf{G}}_{a,t-1} - \mathbf{G}_{a,t-1})\right\|_2 + |\alpha|\sqrt{t}\|(\mathbf{G}_{a,t-1})\|_2 \\
&\leq C_1^{2L} L^{1/2} m^{-1/36}
\end{aligned}
$$

The first inequality is from the triangle inequality and the last inequality is due to the choice of $\alpha = \frac{\nu}{(T+1)^2}$, where ν is a parameter and independent of dimension d. Therefore, we have:

$$
\begin{aligned}
&|\langle \mathbf{g}_a(\mathbf{x}; \mathbf{W}_0), \mathbf{W}_{t-1} - \mathbf{W}_0 - \mathbf{U}_{a,t-1}^{-1} \bar{\mathbf{G}}_{a,t-1} \mathbf{y}_{t-1} \rangle| \\
&\leq \|\mathbf{g}_a(\mathbf{x}; \mathbf{W}_0)\|_2 \left\|\mathbf{W}_{t-1} - \mathbf{W}_0 - \mathbf{U}_{a,t-1}^{-1} \bar{\mathbf{G}}_{a,t-1} \mathbf{y}_{t-1}\right\|_2 \\
&\leq C_1^{2L} L^{1/2} m^{-1/36}
\end{aligned}
$$

□

Lemma B.10 (Theorem 1, Chowdhury and Gopalan [2017])**.** *Let $\{\boldsymbol{\epsilon}_{a,t}\}_{t=1}^{\infty}$ be a real-valued stochastic process such that for some $R \geq 0$ and for all $t \geq 1$, $\boldsymbol{\epsilon}_{a,t}$ is $\mathcal{F}_{a,t-1}$-measurable and R-sub-Gaussian conditioned on $\mathcal{F}_{a,t-1}$. Recall $\mathbf{H}_0$ defined in Equation* (6)*. For a given $\eta > 0$, with probability $1 - \delta$, the following holds for all t:*

$$
\boldsymbol{\epsilon}_{a,t}^\top ((\mathbf{H}_0 + \eta \mathbf{I})^{-1} + \mathbf{I})^{-1} \boldsymbol{\epsilon}_{a,t} \leq R_a^2 \log\det((1+\eta)\mathbf{I} + \mathbf{H}_0) + 2R_a^2 \log(1/\delta).
$$

Lemma B.11. *Let $\delta \in (0,1)$. If the network width m satisfies Condition 3.4, then with probability at least $1 - \delta$, the following holds for every $t \in [T]$:*

$$
\log\det(\mathbf{I} + \mathbf{H}_0) \leq 2\gamma_{a,t} + 1,
$$

where $\gamma_{a,t}$ is the maximum information gain associated with the NTK kernel k_a.

Proof of Lemma B.11. From the definition of $\mathbf{H}_0$ and Lemma B.7 Zhou et al. [2020], we have that

$$
\begin{aligned}
\log\det(\mathbf{I} + \mathbf{H}_0) &= \log\det\left(\mathbf{I} + \sum_{i=1}^{T} \mathbf{g}(\mathbf{x}_t; \boldsymbol{\theta}_0) \mathbf{g}(\mathbf{x}_t; \boldsymbol{\theta}_0)^\top\right) \\
&= \log\det(\mathbf{I} + \mathbf{H}_0 + (\boldsymbol{\Phi} - \mathbf{H}_0)) \\
&\leq \log\det(\mathbf{I} + \mathbf{H}_0) + \langle (\mathbf{I} + \mathbf{H}_0)^{-1}, (\boldsymbol{\Phi} - \mathbf{H}_0) \rangle \\
&\leq \log\det(\mathbf{I} + \mathbf{H}_0) + \left\|(\mathbf{I} + \mathbf{H}_0)^{-1}\right\|_F \|(\boldsymbol{\Phi} - \mathbf{H}_0)\|_F \\
&\leq 2\gamma_{a,t} + 1,
\end{aligned}
$$

where the first equality is from the definition of $\mathbf{K}_t$ in Definition 6, the first inequality is from the convexity of $\log\det(\cdot)$ function, and the second inequality is from the fact that $\langle \mathbf{A}, \mathbf{B} \rangle \leq \|\mathbf{A}\|_F \|\mathbf{B}\|_F$. The third inequality is from the choice of m in Condition 3.4, combined with Lemma B.3 and Lemma 3 in Chowdhury and Gopalan [2017]. □

Lemma B.12 (Lemma 8, Phan-Trong et al. [2023])**.** *Consider the neural network $a(\cdot; \boldsymbol{\theta})$ introduced in Equation* (1) *and suppose the width of the neural network m satisfies Condition 3.4. Then*

$$
\sum_{i=1}^{T} \min(\sigma_{a,t}(\mathbf{x}_t), B) \leq \sqrt{\frac{BT}{\log(B+1)}(2\gamma_{a,T} + 1)}.
$$