CoffeeBoost: Gradient Boosting Native Conformal Inference for Bayesian Optimization

Yuanhao Lai[1*], Pengfei Zheng[1*†], Chenpeng Ji[1], Cheng Qiu[1], Tingkai Wang[1], Songhan Zhang[1,2], Zhengang Wang[1], Yunfei Du[1]

[1]Huawei Technologies Co., Ltd.
[2]The Chinese University of Hong Kong, Shenzhen
{laiyuanhao,zhengpengfei18,jichenpeng,qiucheng4,wangtingkai,zhangsonghan2,wangzhengang,duyunfei5}@huawei.com,
222010549@link.cuhk.edu.cn

Abstract

Bayesian optimization (BO) is a key technique for solving black-box optimization problems. This study extends the scope of BO from conventional applications (e.g., AutoML and robotics learning) to automated tuning of software systems. Despite GP (Gaussian Process) implementing a foundation formalism for exploitation and exploration in BO, its limited predictive power and unrealistic assumptions (e.g., continuity and Gaussianity) can severely affect its effectiveness and efficiency in tuning complex software systems. To overcome these limitations, we propose a BO framework CoffeeBoost, which implements exploitation and exploration with a GBDT-native distribution-free probabilistic surrogate model. CoffeeBoost constructs surrogate models via stochastic gradient boosting ensembles (SGBE) and quantifies probabilistic distributions via distribution-free conformal predictive systems. Moreover, CoffeeBoost leverages the residual paths in SGBE to improve the local adaptiveness of the resulting predictive distributions in a GBDT-native manner. Across eight auto-tuning benchmarks for database management systems (DBMS), we evaluate CoffeeBoost and show its superior learnability and optimizability against existing GP-based and tree-ensemble-based BO schemes. Detailed analysis further shows CoffeeBoost's predictive distributions excel in both coverage and tightness.

1 Introduction

Bayesian optimization (BO) has been widely applied to solve sample-efficient black-box optimization problems in various domains (Wang et al. 2023), including chemical synthesis (Korovina et al. 2020; Wang and Dowling 2022), hardware design (Ejjeh et al. 2022; Nardi, Koeplinger, and Olukotun 2019), hyperparameter optimization (Snoek, Larochelle, and Adams 2012; Lindauer et al. 2022) and neural architecture search (Kandasamy et al. 2018; Ru et al. 2020; White, Neiswanger, and Savani 2021). Recent technical trends extend the use of BO towards software system control, in particular, tuning software configurations to maximize performance and resource efficiency (Van Aken et al. 2017; Kanellis et al. 2022; Wang et al. 2022; Lin et al. 2022).

Most prior work is grounded on Gaussian Process regression (GPR)-based surrogate model (Seeger 2004) to guide exploration and exploitation. However, the inherent assumptions made by GPR, such as continuity, Gaussianity, and stationarity can severely limit their applicability to real-world software configuration tuning (Van Aken et al. 2017; Zhang et al. 2022) problems. System measurements are usually high-dimensional (e.g., multiple hundreds), heterogeneous with both discrete and continuous variables, and embedded with non-stationary, heteroscedastic noises of any arbitrary distribution, either of which can fail the learning and optimization process of GPR-based BO algorithms.

To address these limitations, various approaches have been proposed to lift the restrictive assumptions made by GPR-based BO, including the use of non-stationary kernels (Heinonen et al. 2016; Martinez-Cantin 2015), multi-local-surrogates within trust regions (Eriksson et al. 2019), and data warping techniques (Snoek et al. 2014; Cowen-Rivers et al. 2022). Independently, other work tries to avoid the assumption pitfalls by replacing GPR with a wider spectrum of machine learning models, such as random forest (RF) used in SMAC (Lindauer et al. 2022) and gradient boosting decision trees (GBDT) (Friedman 2001) used in scikit-optimize (Head et al. 2021). The practicability of tree-based BO schemes toward software system tuning is evaluated in a recent large-scale experimental study (Zhang et al. 2022) over cloud databases. Results show that the SMAC (Lindauer et al. 2022) significantly outperforms vanilla and advanced GPR-based BO variants.

This study follows the route of tree-based BO to further drive the research of Bayesian optimization under relaxed assumptions. We found that, though BO with random forest surrogates (e.g., SMAC) prevails in building software system auto-tuners (Lindauer et al. 2022; Li et al. 2021; Curino et al. 2020; Thornton et al. 2013), another stream of tree-based models, i.e., gradient boosted decision trees, remains relatively unexplored in BO. This can be attributed to the challenges in uncertainty quantification for GBDTs, which has long existed in the machine learning community.

Existing uncertainty estimation methods for GBDT such as NGBoost (Duan et al. 2020) and PGBM (Sprangers, Schelter, and de Rijke 2021), only account for data uncertainty (Gal et al. 2016) while neglecting knowledge uncer-

*These authors contributed equally.
†Corresponding author.

tainty (Gal et al. 2016) induced by unexplored regions. The lack of knowledge uncertainty can hinder BO schemes from active exploration and cause over-exploitation. To improve this, CatBoost (Dorogush, Ershov, and Gulin 2018) proposes estimating both data and knowledge uncertainty with the virtual ensemble technique (Malinin, Prokhorenkova, and Ustimenko 2021). Nevertheless, NGBoost, PGBM, and virtual ensemble all enforce assuming a parametric predictive distribution (e.g., Gaussian distribution) to produce valid posterior estimates for acquisition functions, e.g., Expected Improvement (EI) (Ament et al. 2024). However, mis-specifying predictive distribution can also distort uncertainty estimation and breach targeted exploration.

In this study, we propose CoffeeBoost, a novel BO algorithm that can inherently overcome the aforementioned limitations of GPR-based BO. Specifically, CoffeeBoost has two features. First, CoffeeBoost achieves accurate probabilistic surrogate modeling by deriving distribution-free conformal predictive distributions (Xu and Xie 2023; Colombo 2023; Vovk et al. 2018) for stochastic gradient boosting ensembles (SGBE) (Malinin, Prokhorenkova, and Ustimenko 2021), which, with empirical evaluation, shows higher modeling accuracy over complex, heterogeneous, system measurements compared to GPR-based and RF-based surrogates. Second, CoffeeBoost proposes a novel GBDT-native conformal predictor to resolve a key problem in conformal inference, i.e., local adaptiveness, which leverages the predictive residual paths within SGBE to construct predictive distributions with superior coverage and tightness (or continuous ranked probability score, i.e., CRPS). Overall, comprehensive experiments across eight database auto-tuning benchmarks show CoffeeBoost significantly higher tuned performance compared with a spectrum of existing BO algorithms, which shed light on a new direction for practical BO.

2 Related Work

In the BO literature, a significant amount of research has focused on addressing the limitations of Gaussian Process Regression (GPR), particularly its assumptions of Gaussianity and stationarity. These efforts include both modifications to GPR and the development of new surrogate models.

(a) GPR-based Bayesian optimization include input warping methods such as HEBO (Cowen-Rivers et al. 2022), localized methods such as TuRBO (Eriksson et al. 2019) and BALLET (Zhang et al. 2023), subspace embedding methods such as HESBO (Wang et al. 2016) and BAXUS (Papenmeier, Nardi, and Poloczek 2022), and Conformal-BayesOpt (Stanton, Maddox, and Wilson 2023). Notably, HEBO and TuRBO have demonstrated strong performance in hyperparameter tuning tasks, as evidenced by their success in the NeurIPS 2020 black-box optimization challenge (Turner et al. 2021). Meanwhile, although Conformal-BayesOpt derives a full conformal method for GPR models to enable distribution-free inference, the usage of GPR still induces unrealistic assumptions such as continuity.

(b) Tree-ensemble-based Bayesian optimization methods include methods with a surrogate based on random forests such as SMAC (Lindauer et al. 2022) and methods with a surrogate based on gradient boosting decision trees (GBDT) such as SKOPT(GBQRT) (Head et al. 2021) and Hyperboost (van Hoof and Vanschoren 2021). Additionally, there are state-of-the-art probabilistic GBDTs that have not yet been widely applied to BO, including NGBoost (Duan et al. 2020), PGBM (Sprangers, Schelter, and de Rijke 2021), virtual ensemble gradient boosting (Malinin, Prokhorenkova, and Ustimenko 2021) and IBUG (Brophy and Lowd 2022). IBUG, in particular, features distribution-free properties by constructing a non-parametric distribution based on k-nearest training instances, with distances measured using a supervised tree kernel. However, its high computational complexity for predicting uncertainty makes it infeasible for BO, as predictions will be made hundreds of times during each iteration for acquisition maximization.

3 Methodology

3.1 Preliminaries: Bayesian Optimization

Bayesian Optimization (BO) is a method used to optimize an expensive complex black-box function $y = f(\boldsymbol{x})$ with as few evaluations as possible by solving the problem,

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}\in\boldsymbol{\Lambda}} f(\boldsymbol{x}),$$

where $\boldsymbol{\Lambda}$ represents the search space. The optimization process in BO is iterative. At each evaluation step, a probabilistic surrogate model $\hat{f}(x)$, such as Gaussian Processes (Seeger 2004), is first trained on the historic T-round accumulative evaluated observations $D_T = \{(\boldsymbol{x}_t, y_t)\}_{t=1}^{T}$, to approximate the objective function. An acquisition function that balances exploration and exploitation, such as Expected Improvement (EI) (Ament et al. 2024), identifies the next point to evaluate by optimizing itself with respect to the surrogate model. After each new evaluation, the surrogate model is updated with the new data point, and the process repeats. This iterative cycle continues until a stopping criterion is met, such as a predefined number of evaluations or a satisfactory level of performance. By dynamically balancing exploration and exploitation, BO efficiently navigates the search space, often requiring fewer evaluations than traditional methods like grid search or random search.

3.2 Overview of CoffeeBoost

Enhancing surrogate modeling with stochastic gradient boosting ensembles (SGBE). To tackle the issues of parametric distribution dependency and lack of knowledge uncertainty estimation for existing GBDT surrogates, we construct our CoffeeBoost surrogate model starting from the SGBE (Stochastic Gradient Boosting Ensemble), shown to have a strong point-estimate accuracy and enable knowledge uncertainty estimation (Malinin, Prokhorenkova, and Ustimenko 2021). SGBE is formed by aggregating an ensemble of GBDTs $\{\mathcal{M}_b\}_{b=1}^{B}$, where $\mathcal{M}_b$ is a GBDT trained on a subset D_b randomly sampled from the complete dataset D_T.

Achieving distribution-free uncertainty quantification (predictive intervals or distributions) with conformal predictive systems (CPS). Next, the idea of conformal predictive systems (CPS) (Vovk et al. 2018) is introduced to SGBE for deriving non-parametric predictive distributions

that facilitate BO's acquisition evaluation. CPS offers a robust, distribution-agnostic framework for generating reliable uncertainty estimates in machine learning predictions. Unlike traditional methods that depend on stringent assumptions regarding the underlying data distribution, CPS outputs predictive distributions with refined uncertainty representation, accompanied by statistically valid confidence intervals or prediction regions, with coverage guarantees that hold in finite samples independently of the data distribution.

Conformalized ensemble over GBDTs. The foundation of CPS lies in the concept of nonconformity scores, which measure how atypical or nonconforming a new observation $(\boldsymbol{x}_i, y_i)$ is compared to the training data. Given a model $\hat{f}[\cdot]$, the nonconformity score is defined as $A(\boldsymbol{x}_i, y_i, \hat{f}) = y_i - \hat{f}(\boldsymbol{x}_i)$. Moreover, an estimation of prediction difficulty (Kato, Tax, and Loog 2023; Boström et al. 2017; Johansson, Löfström, and Boström 2023) can be further introduced to $A(\boldsymbol{x}_i, y_i, \hat{f})$ as locally adaptive weights that can gauge the challenge of making accurate predictions for specific instances, resulting in the nonconformity score, defined as,

$$A(\boldsymbol{x}_i, y_i, \hat{f}) = \frac{y_i - \hat{f}(\boldsymbol{x}_i)}{\sigma_i + \beta}, \tag{1}$$

where σ_i represents the difficulty estimation of $\boldsymbol{x}_i$, and β is a small positive constant used to avoid excessively large nonconformity scores that may result from inaccurate difficulty estimates (e.g., $\sigma_i = 0$). Higher nonconformity scores indicate more incredible difficulty and deviations from the model's expectations. This allows CPS to adjust prediction interval widths dynamically, narrowing them for simpler cases and broadening them for more uncertain predictions, thus improving the precision of uncertainty estimates across different instances. Suppose we have $n+1$ new observations forming a calibration data set $D_n^{cal} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, the conformal predictive distribution (i.e., a cumulative distribution) for an input $\boldsymbol{x}$ can then be derived as,

$$\hat{F}(y \mid \boldsymbol{x}, \hat{f}, D_n^{cal}) = \begin{cases} \frac{i+\tau}{n+1}, & \text{if } y \in (C_{(i)}, C_{(i+1)}) \forall i \in \{0, \dots, n\} \\ \frac{i'-1+(i''-i'+2)\tau}{n+1}, & \text{if } y = C_{(i)} \forall i \in \{1, \dots, n\}, \end{cases} \tag{2}$$

where $C_{(i)} = \hat{f}(\boldsymbol{x}) + \sigma_{(i)}\alpha_i$ for $i = 1, \dots, n$ and α_i is the order statistic obtained by sorting $\{A(\boldsymbol{x}_i, y_i, \hat{f})\}_i^n$ in increasing order, $C_{(0)} = -\infty$, $C_{(n+1)} = +\infty$, $\tau \in [0, 1]$ is sampled from the uniform distribution, $i' = \min\{j \mid C_{(j)} = C_{(i)}\}$ and $i'' = \max\{j \mid C_{(j)} = C_{(i)}\}$. Equation (2) follows from (Vovk et al. 2018) that acts similarly to an empirical cumulative distribution function based on nonconformity scores.

Motivated by recent advances in combining ensemble learning with conformal inference (Kim, Xu, and Barber 2020; Xu and Xie 2021), we are able to integrate the computation of nonconformity scores with ensemble learning seamlessly as illustrated in Algorithm 1. We use the held-out dataset of each individual GBDT model within SGBE as a calibration set to derive nonconformity scores and aggregate them further to improve the estimation accuracy.

Algorithm 1: Probabilistic surrogate of CoffeeBoost

Input: Training Data $D = \{(\boldsymbol{x}_t, y_t)\}_{t=1}^T$, GBDT algorithm $\mathcal{A}$, and prediction input $\{\boldsymbol{x}_{\text{test},i}\}_{i=1}^n$, number of learners B and constant $\beta > 0$.

Output: Predictive distributions $\{\hat{F}(y \mid \boldsymbol{x}_{\text{test},i}, \hat{f}, \boldsymbol{A})\}_{i=1}^n$.

for $b = 1, \dots, B$ **do**
  Randomly sample with replacement to obtain subdata D_b and its complementary data $\bar{D}_b$. Train a base model $\mathcal{M}_b = \mathcal{A}(D_b)$ and form a ensemble model by $\hat{f}(\boldsymbol{x}) = \sum_{b=1}^B \mathcal{M}_b(\boldsymbol{x})/B$.
end for
Initialize nonconformity score set $\boldsymbol{A} = \{\}$.
for $t = 1, \dots, T$ **do**
  Let $\mathcal{B}_t = \{b \mid (\boldsymbol{x}_t, y_t) \notin D_b\}$ stand for indexes of base models that do not see $(\boldsymbol{x}_t, y_t)$ in the training set and hence facilitate calibration with $(\boldsymbol{x}_t, y_t)$.
  Compute $A_t = y_t - \hat{h}_{-t}(\boldsymbol{x}_t)$ where $\hat{h}_{-t}(\boldsymbol{x}_t) = \sum_{b \in \mathcal{B}_t} \mathcal{M}_b(\boldsymbol{x}_t)/|\mathcal{B}_t|$.
  Update $\boldsymbol{A} = \boldsymbol{A} \cup \{A_t\}$.
end for
Use Equation (4) to derive the GBDT-native difficulty estimator $\hat{\sigma}_{\text{vr}}(\boldsymbol{x})$.
Update $A_t = A_t/(\hat{\sigma}_{\text{vr}}(\boldsymbol{x}_t) + \beta)$ for $A_t \in \boldsymbol{A}$.
Use Equation (2) to compute the predictive distribution $\hat{F}(y \mid \boldsymbol{x}_{\text{test},i}, \hat{f}, \boldsymbol{A})$ for $i = 1, \dots, n$.

The limitation of ERC-based locally adaptive conformal prediction due to improper difficulty measure. So far, we have not discussed how to estimate the difficulty parameter σ_i within the nonconformity scores. This is usually accomplished by using the Error Re-weighted Conformal (ERC) approach (Lei et al. 2018; Colombo 2023), which fits a regression model $\hat{h}_{ERC}(x)$ on the (absolute) residuals within the training set. ERC not only requires the computational overhead of additional model fitting but may also result in an over-optimistic difficulty estimator because of the underestimation of training residual errors. To mitigate this issue, we find that there exists a native solution when GBDT is used as the conformalized model, as discussed next.

Improving locally adaptive conformal inference with boosted residuals in SGBE. The key idea behind GBDT is to iteratively improve the model by focusing on the errors made by previous models. This is achieved by constructing a sequence of n_{tree} decision trees, where each tree is trained to model the residuals of the combined predictions of all previously built trees. At each iteration $t \in \{0, \dots, n_{tree}\}$, the model updates its predictions by adding the output of the new decision tree $h_t(x)$ to the existing model:

$$g_t(\boldsymbol{x}) = g_{t-1}(\boldsymbol{x}) + \gamma h_t(\boldsymbol{x}) \tag{3}$$

where $g_{t-1}(x)$ is the model from the previous iteration, $h_t(\boldsymbol{x})$ is the new decision tree trained on the residuals, and γ is the learning rate that scales the contribution of $h_t(\boldsymbol{x})$.

Equation (3) indicates that each decision tree $h_t(x)$ for $t \geq 1$ can be treated as applying ERC to the training residuals of the previous-step GBDT. Therefore, the value pre-

dicted by each decision tree $h_t(x)$, denoted by boosted residuals, naturally provides an estimate for the prediction difficulty for an instance $\boldsymbol{x}$ and how it changes during the entire training process. Since GBDT tends to overfit data near the final training iteration, their training residual errors may tend to under-estimate the generalization error and hence reduce the effectiveness of ERC. To develop a more robust estimate of prediction difficulty, we thus propose using the aggregation of boosted residuals from different iterations to form a GBDT-native difficulty estimate. Specifically, we use the Frobenius norm for aggregation and define the GBDT-native difficulty estimate of an instance $\boldsymbol{x}$ for the b-th GBDT model $\mathcal{M}_b$ of SGBE as $\hat{\sigma}_{\text{vr},b}(\boldsymbol{x}) = \sqrt{\sum_{t\in\mathcal{T}} h_{b,t}(\boldsymbol{x})^2/|\mathcal{T}|}$, where $\mathcal{T}$ is the selected iteration set, and $h_{b,t}$ is the decision tree at the t-th iteration of $\mathcal{M}_b$ for $b = 1, \ldots, B$. We then further aggregate these estimators among all GBDTs of SGBE to produce a unified difficulty estimator,

$$\hat{\sigma}_{\text{vr}}(\boldsymbol{x}) = \sum_{b=1}^{B} \hat{\sigma}_{\text{vr},b}(\boldsymbol{x}). \tag{4}$$

For the choice of selected iteration set $\mathcal{T}$, we find that CoffeeBoost using 20 equally spaced iterations among the entire training history is computationally efficient yet sufficient to provide an uncertainty estimation as good as the one using all iterations in our experiment (cf. Section 4.5).

Acquisition computation with Monte Carlo integration. The Expected Improvement (EI) acquisition function (Jones, Schonlau, and Welch 1998; Ament et al. 2024) is a fundamental tool in Bayesian optimization, guiding the search for the global optimum of a target function. EI balances exploration and exploitation by quantifying the expected improvement over the current best observation if a new point is sampled. Mathematically, for a given point x and a surrogate model f, EI is defined as:

$$\text{EI}(\boldsymbol{x}) = \mathbb{E}[max(0, f(\boldsymbol{x}) - f(\boldsymbol{x}^+))] \tag{5}$$

where $f(\boldsymbol{x}^+)$ represents the best-observed value so far. In principle, there is no analytical form of EI unless a certain distribution (i.e., Gaussian) is assumed. We can approximate the EI accurately without such an assumption via Monte Carlo integration by drawing moderate (e.g., 100) samples from the conformal predictive distribution in Equation (2).

Overall, Algorithm 2 presents the pseudo-code for CoffeeBoost. Compared to a GBDT, the usage of SGBE with B GBDTs will require B multiplies of computational training time and memory usage. To enable building conformal predictive distribution, extra $O(TB)$ computational inference time is required to compute the nonconformity scores and sort them, where T is the size of evaluated samples.

Algorithm 2: Pseudo code of CoffeeBoost

Input: Objective function $f(\boldsymbol{x})$, search space $\boldsymbol{\Lambda}$, budgets N, initialized dataset $D = \{(\boldsymbol{x}_t, y_t)\}_{t=1}^{T_0}$ by Sobol design.
Output: Optimal solution $\boldsymbol{x}_{t^*}$.

1: **for** $i = 1, \ldots, N$ **do**
2: Build a probabilistic surrogate for computing predictive distribution of $\boldsymbol{x}$ based on D by Algorithm 1.
3: Select new $\boldsymbol{x}_{T_0+i} \in \boldsymbol{\Lambda}$ by optimizing the EI acquisition function in Equation (5).
4: Evaluate the objective function $y_{T_0+i} = f(\boldsymbol{x}_{T_0+i})$.
5: Augment dataset $D = \{D, (\boldsymbol{x}_{T_0+i}, y_{T_0+i})\}$.
6: Update the optimal index $t^* = \underset{t=1,\ldots,T_0+i}{\arg\max}\, f(\boldsymbol{x}_t)$.
7: **end for**

4 Experiments

4.1 Experimental Settings

Software Auto-Tuning Benchmarks. We conduct extensive experiments on eight open benchmarks of database performance auto-tuning (Zhang et al. 2022) with respect to different workloads including TWITTER, TATP, VOTER, SMALLBANK, YCSB, TPC-C, SYSBENCH, and JOB. The objective of these benchmarks is to find the optimal configuration input for achieving peak performance. The input dimensions varied, with 196 for SYSBENCH and JOB benchmarks, and 100 for the other benchmarks. Additionally, the number of categorical variables was 76 and 54, respectively. In particular, these benchmark benchmarks contain ML-based (Random Forest) database simulators that were trained on samples of real performance measurements to output the performance metric of a database (MySQL-v5.7) given a configuration input under a specific workload. However, using random forests for building simulators may give an unfair structural advantage to tree-based optimizers. To address this issue, we replace the RF-based simulators with transformer-based simulators trained on the author's published data using SAINT (Somepalli et al. 2021), a state-of-the-art transformer model for tabular data regression.

Evaluation settings and metrics. To ensure a fair comparison, each optimization starts with the same 20 initialization points generated randomly from the Sobol sampler (Sobol' 1967) and proceeds for 80 trials, and we report the average result of each optimizer over seven seeded searches for each benchmark. To evaluate the optimization effectiveness, we consider the performance gain of each optimizer, defined as the difference of final database performances found by an optimizer and Latin Hypercube sampling (LHS).

Baselines and implementations. We compare the performance of CoffeeBoost with fifteen baseline optimizers including four GP-BOs, six tree-ensemble BOs, two conformalized-GBDT BOs and three other commonly used black-box optimization baselines. The GP-BO baselines include (1) **Matern-BO**, a GP-based BO with Matérn-5/2 kernel, (2) **HEBO** (Cowen-Rivers et al. 2022), (3) **TuRBO** (Eriksson et al. 2019), and (4) **HESBO** (Wang et al. 2016). The tree-ensemble BO baselines include (5) **SMAC** (Lindauer et al. 2022) with a surrogate based on random forests, (6) **NGB-BO** with a surrogate based on natural gradient boosting (NGBoost) (Duan et al. 2020), (7) **PGBM-BO** with a surrogate based on the probabilistic gradient boosting machine (PGBM) (Sprangers, Schelter, and de Rijke 2021), (8) **VEGB-BO** with a surrogate based on the virtual ensemble gradient boosting (Malinin, Prokhorenkova, and Ustimenko 2021), (9) **SKOPT(GBQRT)** (Head et al. 2021)

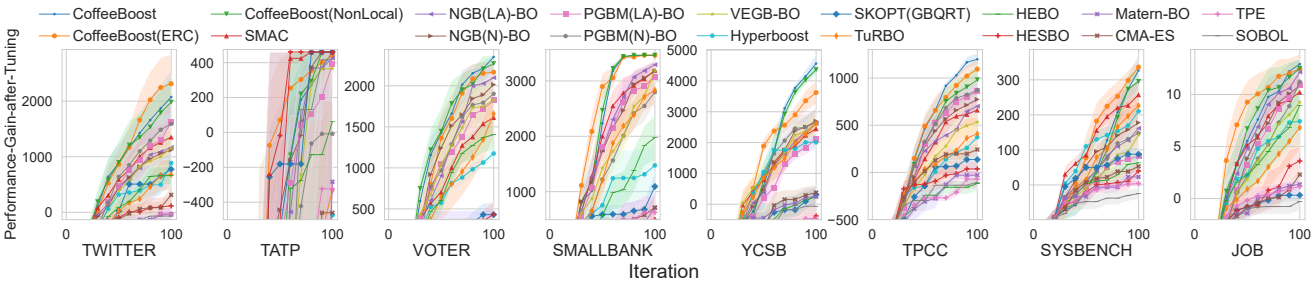

Figure 1: The mean and the one-standard-error range of optimization progress (seven runs each, each run with 100 iterations) over different database tuning benchmarks.

and (10) **Hyperboost** (van Hoof and Vanschoren 2021), both of which build a surrogate based on gradient boosting quantile regression technique. The conformalized-GBDT BOs include (11) **CoffeeBoost(ERC)** where we replace the implied-difficulty estimator of CoffeeBoost by Error Reweighted Conformal approach (ERC) (Papadopoulos, Gammerman, and Vovk 2008), and (12) **CoffeeBoost(NonLocal)** where we apply the non-locally-adaptive conformal ensemble method EnbPI (Xu and Xie 2021) to GBDT to form a surrogate. The remaining baselines include (13) **Tree Parzen Estimator (TPE)** (Bergstra et al. 2011), (14) **Covariance Matrix Adaptation Evolution Strategy (CMA-ES)** (Hansen 2006) and (15) **Sobol sampling**. To implement the above methods, we use the authors' released implementations except for NGB-BO, PGBM-BO, VEGB-BO and CoffeeBoost, where we use the authors' implementation of NGBoost, PGBM and virtual ensemble together with lightgbm (Ke et al. 2017) and SMAC3 (Lindauer et al. 2022) to implement the surrogate with an EI acquisition. In particular, we use NGB(N)-BO, NGB(GA)-BO, NGB(LN)-BO, NGB(LA)-BO, and NGB(T3)-BO to indicate NGB-BOs that assume outputs distributed respectively as Normal, Gamma, Log-Normal, Laplace and student's t of degree 3 in order to facilitate acquisition computation. Similar notations also apply to PGBM-BO. We use default hyperparameters for all baseline methods except for GBDT-based methods. We set the number of boosting iterations to 100, the learning rate to 0.05, the maximum tree depth to 7, the number of SGBE's base learners to 20, the conformal-related hyperparameters $\beta = 0.01$ and the set $\mathcal{T}$ of 20 equally spaced iterations among the entire training history for GBDT-native difficulty estimator from Equation (4).

4.2 Performance Comparison

BOs with conformalized ensemble of GBDTs outperforms BOs with other surrogate models for database tuning. Figure 1 shows that CoffeeBoost, CoffeeBoost(ERC) and CoffeeBoost(NonLocal) remain the lead during the optimization process in all benchmarks. Figure 2 further shows the detailed comparison of final-tuned performance gain for each optimizer. On average, CoffeeBoost achieves higher final-tuned performance gain improvement relative to SMAC compared to GP-BOs (+109.7%), tree-ensemble BOs (+58.5%), conformalized-GBDT BOs (+7.6%), TPE (+150.2%), CMA-ES (133.6%), and Sobol sampling (131.0%) across all eight benchmarks.

Tree-ensemble-based BOs outperform Gaussian Process-based BOs in most benchmarks. Figure 2 shows that TuRBO achieves the best performance compared to the other GP-BOs including Matern-BO, HEBO and HESBO in most benchmarks with an average 40.3% performance gain over SMAC. However, it is surpassed by tree-ensemble BOs with an average 21.0% drop in performance gain improvement relative to SMAC. In particular, all of SMAC, VEGB-BO, PGBM-BO and NGB-BO outperform TuRBO in six out of eight benchmarks. This observation agrees with Zhang et al. (2022) and reinforces the necessity of switching BO's surrogate from GP to tree-ensemble models for software configuration tuning benchmarks.

Without conformalized ensemble, BO with GBDT surrogates do not consistently outperform BO with RF surrogate model (i.e., SMAC) across different benchmarks. Figure 2 shows that CoffeeBoost consistently outperforms SMAC by 41.1% relative performance gain improvement on average across all benchmarks, whereas none of the BOs based on existing GBDTs are able to outperform SMAC in the SYSBENCH benchmark. In particular, the quantile regression methods, Hyperboost and SKOPT(GBQRT), even fail to compete with SMAC in all benchmarks with an average 58.6% decrease of relative performance gain improvement. The ensemble-based method, VEGB-BO, outperforms SMAC only in the VOTER and YCSB benchmarks by 13.3% and 6.68% respectively, and outperformed SMAC with an average 45.7% performance gain improvement in the other benchmarks. In contrast, both of the multi-parameter GBDT-based optimizers, NGB(N)-BO and PGBM(N)-BO, can outperform SMAC in benchmarks except for the SYSBENCH, TWITTER, TATP and SmallBank benchmarks. Moreover, their performance can be boosted further respectively by an average 3.1% and 2.78% increase of relative performance gain improvement in the TWITTER, VOTER, SmallBank and JOB benchmarks by replacing their Normal distributional assumption with Laplace distributions. This may indicate there exists a dominated distributional assumption that can lead to outperforming SMAC in all benchmarks. However, our experiments (cf., Sections 4.3 and 4.4) show that there is no such silver bullet regarding the tuning performance and surrogates' uncertainty estima-

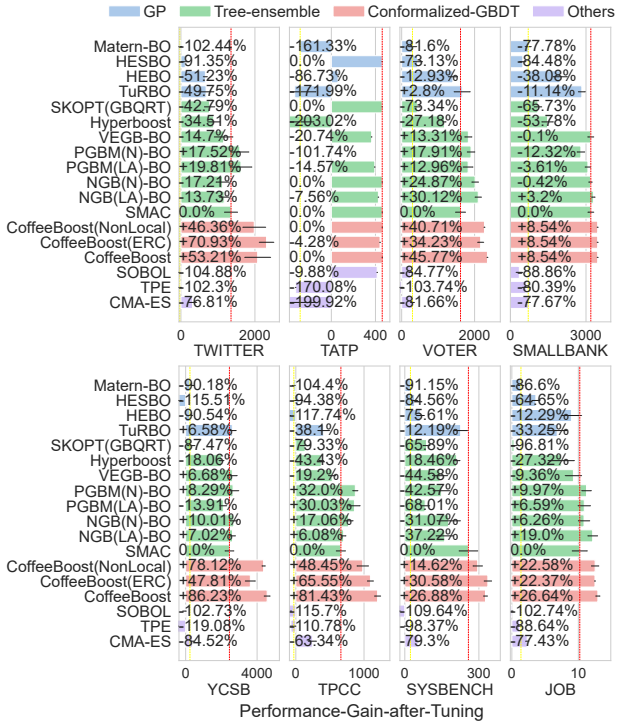

Figure 2: Average of final-tuned performance gain across eight database tuning benchmarks. Percentage numbers show relative improvements over SMAC. One-standard-error ranges are shown at the end of each bar. The prior distribution families for PGBM and NGB are specified within the parentheses, where "N" means Normal and "LA" means Laplace. See entire tuning trajectories in Figure 1.

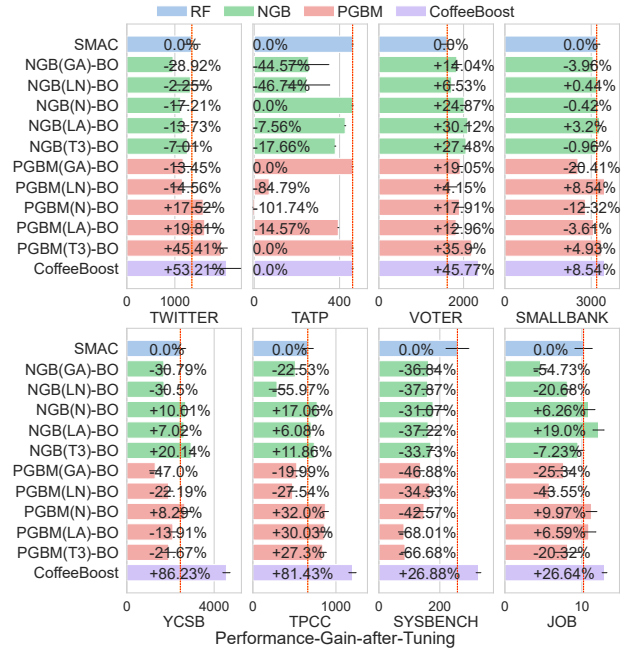

Figure 3: Comparing CoffeeBoost against tree-ensemble-based BO baselines under different prior distribution families specified within the parentheses, where "GA" means Gamma, "LN" means Log-Normal, "N" means Normal, "LA" means Laplace, and "T3" means student's t of degree 3. Percentage numbers are w.r.t. that of SMAC. One-standard-error ranges are shown at the end of each bar.

tion, and therefore a distribution-free approach is necessary for the generalization to different benchmarks.

CoffeeBoost on average achieves the best-tuned database performance among the variants of BO with conformalized gradient boosting. Figure 2 shows that CoffeeBoost achieves consistently better performance than CoffeeBoost(ERC) and CoffeeBoost(NonLocal) except for the TWITTER and SYSBENCH benchmarks, where CoffeeBoost is outperformed by CoffeeBoost(ERC) with 17.7% and 3.7% relative performance gain respectively. On average, CoffeeBoost outperforms CoffeeBoost(ERC) and CoffeeBoost(NonLocal) by respectively 6.6% and 8.7% increases in the relative performance gain over SMAC in all eight benchmarks. Such superiority can be accounted by the better uncertainty quantification of the proposed implied difficulty estimator, as discussed in Section 4.4.

4.3 Comparing Against NGBoost and PGBM Under Varying Distribution Families

Mis-specified Gaussian posterior distribution of NGBoost and PGBM-based BOs. Next, we study whether NGB-BO and PGBM-BO can benefit from specific distributional assumptions for database auto-tuning benchmarks. Other than the Normal distribution, we examine two heavy-tailed symmetric distributions (i.e., Laplace and student's t of degree 3) and two asymmetric heavy-tailed distributions (i.e., Gamma and Log-Normal). As shown in Figure 3, changing the assumption of Normal distributions to the others has substantial influence on the tuning performance. On average through all benchmarks, NGB(N)-BO is quite robust by achieving higher relative performance gain compared to NGB(LA)-BO (+0.3%), NGB(T3)-BO (+2.1%), NGB(GA)-BO (+27.2%), and NGB(LN)-BO (+24.6%), while PGBM(N)-BO achieve higher relative performance gain compared to PGBM(GA)-BO (+10.4%), PGBM(LN)-BO (+18.0%) but lower relative performance gain compared to PGBM(LA)-BO (-5.0%), and PGBM(T3)-BO (-9.5%). The preference for PGBM-BO to symmetric heavy-tailed distributions may indicate that its uncertainty estimation is under-estimated with the normal assumption.

No pre-specified posterior distribution works consistently well across different benchmarks and the need of distribution-freeness. Figure 3 also shows that there is not a specific distributional assumption that can assure either NGB-BO or PGBM-BO outperforms SMAC. Assuming the ideal case where the NGB-BO and PGBM-BO have prior knowledge of the best distributional assumption for each benchmark, the NGB-BO and PGBM-BO can only outperform SMAC by 7.1% and 13.1% performance gain respectively on average across all benchmarks. In contrast, Cof-

feeBoost produces 28% more performance gain improvement relative to SMAC. Therefore, it is critical to develop a distribution-free approach for GBDT-based BOs.

4.4 Surrogate Evaluation

Surrogate performance evaluation settings. To shed light on why CoffeeBoost works, we analyze the predictive performance for both the point estimate and the probabilistic estimate of the underlying surrogate model of each optimizer. We construct an evaluation dataset by merging configuration-performance paired samples from the tuning history from the previous YCSB benchmark experiments. The evaluation dataset is then randomly split into a training set of 100 samples and a test set of the remaining, used to train a surrogate and test its predictive performance. The point estimate is assessed by the coefficient of determination, denoted R^2, and the probabilistic estimate is assessed by the continuous ranked probability score (Gneiting and Katzfuss 2014), denoted by (CRPS). A normalized CRPS (NCRPS) is then obtained by NCRPS $= (\text{CRPS}_{base} - \text{CRPS})/\text{CRPS}_{base}$, where CRPS_{base} is the CRPS of a Normal distribution with constant mean and standard deviation estimated from the sample metrics of a training data set. By doing so, both R^2 and NCRPS are optimized for maximum values. We compute and report the averaged R^2 and NCRPS over 20 evaluations on random train-test splits.

Estimated uncertainty (predictive distribution), i.e., NCRPS, is more decisive for the final-tuned database performance. As shown by Figure 4, the value of NCRPS is highly consistent with the performance gain. In particular, it is hard to predict which method within Conformalized-GBDT BOs will outperform the others in tuning performance gain based on R^2 since they have the same values of R^2. In contrast, they can be discriminated according to NCRPS, which is more comprehensive.

CoffeeBoost outperforms existing tree-ensemble BOs and GP-BOs w.r.t. both point-estimate accuracy (R^2) and uncertainty-estimate accuracy (NCRPS). As shown by Figure 4, CoffeeBoost achieves the highest R^2 and NCRPS among all optimizers while GP-BOs achieve the lowest surrogate performance. In particular, CoffeeBoost increases R^2 and NCRPS by 35.2% and 101.7% respectively compared to PGBM-BOs. When compared to NGB-BOs, CoffeeBoost increases R^2 and NCRPS by 10.1% and 13.1% respectively. It is interesting to observe that the non-locally-adaptive conformal method, CoffeeBoost(NonLocal), outperform CoffeeBoost(ERC) with respect to both NCRPS and performance gain, which may be caused by the under-estimation problem of ERC. These observations reassure the superiority of CoffeeBoost and our proposed difficulty estimator.

4.5 The Impact of Selection for Boosted Residuals

We next study how the selection of the iteration set for the proposed GBDT-native difficulty estimator in Equation (4) can affect the surrogate's uncertainty estimation. We examine four strategies of iteration selections. In particular, "CoffeeBoost " uses the default strategy which selects 20 equally spaced iterations among the entire training history. "CoffeeBoost(Complete)" selects all iterations of GB-

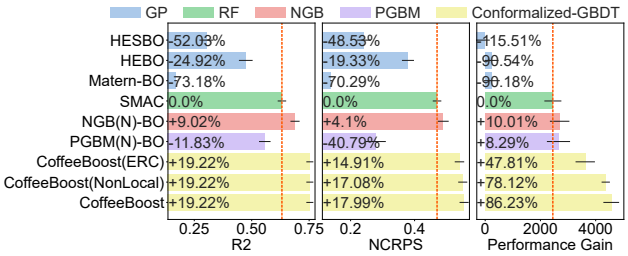

Figure 4: Detailed evaluation of surrogate models' point-estimate accuracy (R2), predicted probabilistic distribution (uncertainty estimation, NCRPS) and tuned performance gain (using YCSB benchmark as an example). Percentage numbers are w.r.t. that of SMAC. One-standard-error ranges are shown at the end of each bar.

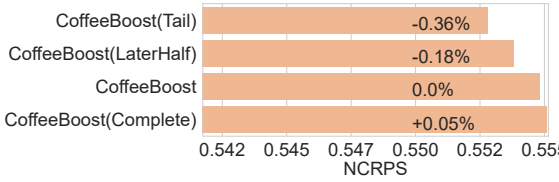

Figure 5: Probabilistic performance (NCRPS) for CoffeeBoost with different iteration selection schema for boosted residuals estimated from the YCSB benchmark. Percentage numbers show relative improvements over CoffeeBoost.

DTs, "CoffeeBoost(LaterHalf)" selects the equally spaced 20 iterations of GBDTs during the later half training process, and "CoffeeBoost(Tail)" only selects 3 iterations at the end of the training process. The performance of their resulted surrogate is summarized in Figure 5, showing that the NCRPS's of using non-symmetric iteration selection (i.e., "CoffeeBoost(LaterHalf)" and "CoffeeBoost(Tail)") are slightly outperformed by "CoffeeBoost(Complete)" and "CoffeeBoost". Moreover, using 20 equally spaced iterations of GBDTs suffices to obtain an equally-well difficulty estimator as using all iterations with less computation.

5 Conclusion

In this paper, we propose CoffeeBoost, a novel GBDT-based ensemble surrogate model with conformal inference for estimating predictive distributions in a distribution-free manner, addressing the limitations of existing GP-based BOs and tree-ensemble-based BOs. We further propose a GBDT-native difficulty estimator to facilitate locally adaptive conformal inference, achieving better CRPS compared to existing difficulty estimators. The comprehensive empirical results across eight database auto-tuning benchmarks validate the effectiveness of the proposed method, demonstrating a better option for the surrogate model of BO used for software auto-tuning and similar tasks. However, it is important to note that the use of GBDT-based ensembles in our method results in higher computational time compared to a single GBDT, which is a trade-off for improved performance.

References

Ament, S.; Daulton, S.; Eriksson, D.; Balandat, M.; and Bakshy, E. 2024. Unexpected improvements to expected improvement for bayesian optimization. *Advances in Neural Information Processing Systems*, 36.

Bergstra, J.; Bardenet, R.; Bengio, Y.; and Kégl, B. 2011. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24.

Boström, H.; Linusson, H.; Löfström, T.; and Johansson, U. 2017. Accelerating difficulty estimation for conformal regression forests. *Annals of Mathematics and Artificial Intelligence*, 81: 125–144.

Brophy, J.; and Lowd, D. 2022. Instance-based uncertainty estimation for gradient-boosted regression trees. *Advances in Neural Information Processing Systems*, 35: 11145–11159.

Colombo, N. 2023. On training locally adaptive CP. In *Conformal and Probabilistic Prediction with Applications*, 384–398. PMLR.

Cowen-Rivers, A. I.; Lyu, W.; Tutunov, R.; Wang, Z.; Grosnit, A.; Griffiths, R. R.; Maraval, A. M.; Jianye, H.; Wang, J.; Peters, J.; and Bou-Ammar, H. 2022. HEBO: Pushing The Limits of Sample-Efficient Hyper-parameter Optimisation. *J. Artif. Int. Res.*, 74.

Curino, C.; Godwal, N.; Kroth, B.; Kuryata, S.; Lapinski, G.; Liu, S.; Oks, S.; Poppe, O.; Smiechowski, A.; Thayer, E.; et al. 2020. MLOS: An infrastructure for automated software performance engineering. In *Proceedings of the Fourth International Workshop on Data Management for End-to-End Machine Learning*, 1–5.

Dorogush, A. V.; Ershov, V.; and Gulin, A. 2018. CatBoost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363*.

Duan, T.; Anand, A.; Ding, D. Y.; Thai, K. K.; Basu, S.; Ng, A.; and Schuler, A. 2020. Ngboost: Natural gradient boosting for probabilistic prediction. In *International conference on machine learning*, 2690–2700. PMLR.

Ejjeh, A.; Medvinsky, L.; Councilman, A.; Nehra, H.; Sharma, S.; Adve, V.; Nardi, L.; Nurvitadhi, E.; and Rutenbar, R. A. 2022. HPVM2FPGA: Enabling true hardware-agnostic FPGA programming. In *2022 IEEE 33rd International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, 1–10. IEEE.

Eriksson, D.; Pearce, M.; Gardner, J.; Turner, R. D.; and Poloczek, M. 2019. Scalable global optimization via local Bayesian optimization. *Advances in neural information processing systems*, 32.

Friedman, J. H. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.

Gal, Y.; et al. 2016. Uncertainty in deep learning.

Gneiting, T.; and Katzfuss, M. 2014. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1: 125–151.

Hansen, N. 2006. The CMA evolution strategy: a comparing review. *Towards a new evolutionary computation: Advances in the estimation of distribution algorithms*, 75–102.

Head, T.; Kumar, M.; Nahrstaedt, H.; Louppe, G.; and Shcherbatyi, I. 2021. scikit-optimize/scikit-optimize.

Heinonen, M.; Mannerström, H.; Rousu, J.; Kaski, S.; and Lähdesmäki, H. 2016. Non-stationary gaussian process regression with hamiltonian monte carlo. In *Artificial Intelligence and Statistics*, 732–740. PMLR.

Johansson, U.; Löfström, T.; and Boström, H. 2023. Conformal predictive distribution trees. *Annals of Mathematics and Artificial Intelligence*, 1–14.

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

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

Kanellis, K.; Ding, C.; Kroth, B.; Müller, A.; Curino, C.; and Venkataraman, S. 2022. LlamaTune: sample-efficient DBMS configuration tuning. *arXiv preprint arXiv:2203.05128*.

Kato, Y.; Tax, D. M.; and Loog, M. 2023. A review of nonconformity measures for conformal prediction in regression. *Conformal and Probabilistic Prediction with Applications*, 369–383.

Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; and Liu, T.-Y. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.

Kim, B.; Xu, C.; and Barber, R. 2020. Predictive inference is free with the jackknife+-after-bootstrap. *Advances in Neural Information Processing Systems*, 33: 4138–4149.

Korovina, K.; Xu, S.; Kandasamy, K.; Neiswanger, W.; Poczos, B.; Schneider, J.; and Xing, E. 2020. Chembo: Bayesian optimization of small organic molecules with synthesizable recommendations. In *International Conference on Artificial Intelligence and Statistics*, 3393–3403. PMLR.

Lei, J.; G'Sell, M.; Rinaldo, A.; Tibshirani, R. J.; and Wasserman, L. 2018. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523): 1094–1111.

Li, Y.; Shen, Y.; Zhang, W.; Chen, Y.; Jiang, H.; Liu, M.; Jiang, J.; Gao, J.; Wu, W.; Yang, Z.; et al. 2021. Openbox: A generalized black-box optimization service. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 3209–3219.

Lin, C.; Zhuang, J.; Feng, J.; Li, H.; Zhou, X.; and Li, G. 2022. Adaptive code learning for spark configuration tuning. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 1995–2007. IEEE.

Lindauer, M.; Eggensperger, K.; Feurer, M.; Biedenkapp, A.; Deng, D.; Benjamins, C.; Ruhkopf, T.; Sass, R.; and Hutter, F. 2022. SMAC3: A versatile Bayesian optimization package for hyperparameter optimization. *The Journal of Machine Learning Research*, 23(1): 2475–2483.

Malinin, A.; Prokhorenkova, L.; and Ustimenko, A. 2021. Uncertainty in Gradient Boosting via Ensembles. In *International Conference on Learning Representations*.

Martinez-Cantin, R. 2015. Locally-biased bayesian optimization using nonstationary gaussian processes. In *Neural Information Processing Systems (NIPS) workshop on Bayesian Optimization*, volume 7, 4.

Nardi, L.; Koeplinger, D.; and Olukotun, K. 2019. Practical design space exploration. In *2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 347–358. IEEE.

Papadopoulos, H.; Gammerman, A.; and Vovk, V. 2008. Normalized nonconformity measures for regression conformal prediction. In *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA 2008)*, 64–69.

Papenmeier, L.; Nardi, L.; and Poloczek, M. 2022. Increasing the scope as you learn: Adaptive Bayesian optimization in nested subspaces. *Advances in Neural Information Processing Systems*, 35: 11586–11601.

Ru, B.; Wan, X.; Dong, X.; and Osborne, M. 2020. Interpretable Neural Architecture Search via Bayesian Optimisation with Weisfeiler-Lehman Kernels. In *International Conference on Learning Representations*.

Seeger, M. 2004. Gaussian processes for machine learning. *International journal of neural systems*, 14(02): 69–106.

Snoek, J.; Larochelle, H.; and Adams, R. P. 2012. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25.

Snoek, J.; Swersky, K.; Zemel, R.; and Adams, R. 2014. Input warping for Bayesian optimization of non-stationary functions. In *International conference on machine learning*, 1674–1682. PMLR.

Sobol', I. M. 1967. On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 7(4): 784–802.

Somepalli, G.; Goldblum, M.; Schwarzschild, A.; Bruss, C. B.; and Goldstein, T. 2021. Saint: Improved neural networks for tabular data via row attention and contrastive pretraining. *arXiv preprint arXiv:2106.01342*.

Sprangers, O.; Schelter, S.; and de Rijke, M. 2021. Probabilistic Gradient Boosting Machines for Large-Scale Probabilistic Regression. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, 1510–1520. New York, NY, USA: Association for Computing Machinery. ISBN 9781450383325.

Stanton, S.; Maddox, W.; and Wilson, A. G. 2023. Bayesian optimization with conformal prediction sets. In *International Conference on Artificial Intelligence and Statistics*, 959–986. PMLR.

Thornton, C.; Hutter, F.; Hoos, H. H.; and Leyton-Brown, K. 2013. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 847–855.

Turner, R.; Eriksson, D.; McCourt, M.; Kiili, J.; Laaksonen, E.; Xu, Z.; and Guyon, I. 2021. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In *NeurIPS 2020 Competition and Demonstration Track*, 3–26. PMLR.

Van Aken, D.; Pavlo, A.; Gordon, G. J.; and Zhang, B. 2017. Automatic database management system tuning through large-scale machine learning. In *Proceedings of the 2017 ACM international conference on management of data*, 1009–1024.

van Hoof, J.; and Vanschoren, J. 2021. Hyperboost: Hyperparameter optimization by gradient boosting surrogate models. *arXiv preprint arXiv:2101.02289*.

Vovk, V.; Nouretdinov, I.; Manokhin, V.; and Gammerman, A. 2018. Cross-conformal predictive distributions. In *conformal and probabilistic prediction and applications*, 37–51. PMLR.

Wang, K.; and Dowling, A. W. 2022. Bayesian optimization for chemical products and functional materials. *Current Opinion in Chemical Engineering*, 36: 100728.

Wang, R.; Wang, Q.; Hu, Y.; Shi, H.; Shen, Y.; Zhan, Y.; Fu, Y.; Liu, Z.; Shi, X.; and Jiang, Y. 2022. Industry practice of configuration auto-tuning for cloud applications and services. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 1555–1565.

Wang, X.; Jin, Y.; Schmitt, S.; and Olhofer, M. 2023. Recent advances in Bayesian optimization. *ACM Computing Surveys*, 55(13s): 1–36.

Wang, Z.; Hutter, F.; Zoghi, M.; Matheson, D.; and De Feitas, N. 2016. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55: 361–387.

White, C.; Neiswanger, W.; and Savani, Y. 2021. Bananas: Bayesian optimization with neural architectures for neural architecture search. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 10293–10301.

Xu, C.; and Xie, Y. 2021. Conformal prediction interval for dynamic time-series. In *International Conference on Machine Learning*, 11559–11569. PMLR.

Xu, C.; and Xie, Y. 2023. Conformal prediction for time series. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhang, F.; Song, J.; Bowden, J. C.; Ladd, A.; Yue, Y.; Desautels, T.; and Chen, Y. 2023. Learning regions of interest for Bayesian optimization with adaptive level-set estimation. In *International Conference on Machine Learning*, 41579–41595. PMLR.

Zhang, X.; Chang, Z.; Li, Y.; Wu, H.; Tan, J.; Li, F.; and Cui, B. 2022. Facilitating database tuning with hyperparameter optimization: a comprehensive experimental evaluation. *Proc. VLDB Endow.*, 15(9): 1808–1821.