High-Dimensional Contextual Policy Search with Unknown Context Rewards using Bayesian Optimization

Qing Feng[*] Facebook qingfeng@fb.com

Benjamin Letham[*] Facebook bletham@fb.com

Hongzi Mao MIT hongzi@mit.edu

Eytan Bakshy Facebook ebakshy@fb.com

Abstract

Contextual policies are used in many settings to customize system parameters and actions to the specifics of a particular setting. In some real-world settings, such as randomized controlled trials or A/B tests, it may not be possible to measure policy outcomes at the level of context—we observe only aggregate rewards across a distribution of contexts. This makes policy optimization much more difficult because we must solve a high-dimensional optimization problem over the entire space of contextual policies, for which existing optimization methods are not suitable. We develop effective models that leverage the structure of the search space to enable contextual policy optimization directly from the aggregate rewards using Bayesian optimization. We use a collection of simulation studies to characterize the performance and robustness of the models, and show that our approach of inferring a low-dimensional context embedding performs best. Finally, we show successful contextual policy optimization in a real-world video bitrate policy problem.

1 Introduction

Contextual policies are used in a wide range of applications, such as robotics [22, 30] and computing platforms [9]. Here we consider contextual policies that are a map from a discrete *context* to a set of continuous *parameters*. For example, video streaming and real-time conferencing systems use adaptive bitrate (ABR) algorithms to balance between video quality and uninterrupted playback. The optimal policy for a particular ABR controller may depend on the network—for instance, a stream with large fluctuations in bandwidth will benefit from different ABR parameters than a stream with stable bandwidth. This motivates the use of a contextual policy where ABR parameters are personalized by context variables such as country or network type (2G, 3G, 4G, etc.). Various other systems and infrastructure applications commonly rely on tunable parameters which can benefit from contextualization. For example, optimal job scheduling and load balancing parameters may differ because of workload variations [11]. Cell tower configuration or TCP configurations can benefit from finer-grained information about environmental factors [2]. Contextual policies therefore provide an interpretable, robust approach for personalizing system parameters and improving individual-level outcomes under heterogeneous conditions.

Previous applications of contextual policy optimization (CPO) in the literature, such as those in robotics, have considered the setting where with each evaluation of the policy we observe both the reward and the context. Bayesian optimization (BO) has been successfully applied to this problem [30], using a Gaussian process (GP) to model reward as a function of both parameters and context. If the context is continuous-valued, it can be incorporated directly into the GP along with the parameters;

[*] Equal contribution

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

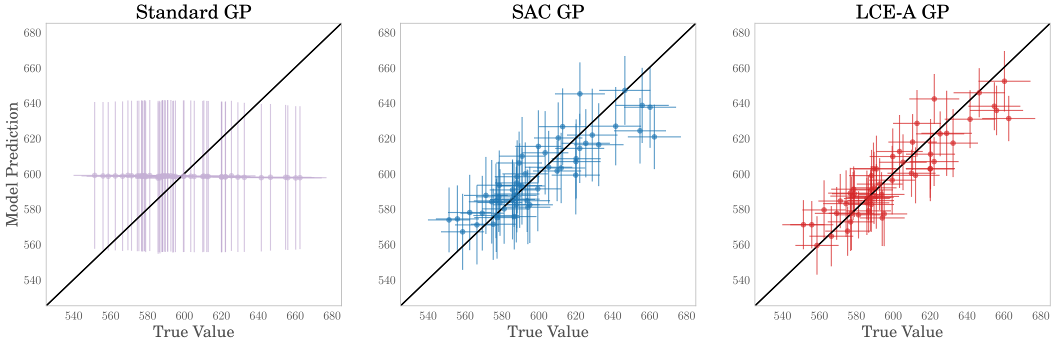

Figure 1: Leave-one-out cross validation predictions (mean, and 95% posterior predictive interval) for standard and contextual GPs fit to the results of a video playback controller experiment conducted at Facebook for a 30-dimensional policy (5 contexts and 6 parameters). The standard GP is unable to learn the high-dimensional response surface, while our proposed SAC and LCE-A models make accurate out-of-sample predictions.

if discrete, it can be included in the GP with a multi-task kernel or similar approaches for handling discrete parameters that are described below.

However, there are important CPO settings where context cannot be observed along with the rewards. In A/B testing platforms, rewards (outcomes) are measured as an aggregate across a large population that spans an entire distribution of contexts. Performing analysis by context is often not feasible, and when it is, such analyses can introduce bias by implicitly conditioning on "post-treatment" variables [31]. Without context-level data, CPO becomes a high-dimensional optimization problem where C contexts each with d parameters requires optimizing a function of $C \times d$ parameters.

In this work, we show that it is possible to optimize a contextual policy and get the benefits of contextualization even when rewards are measured only in aggregate. The contributions of this paper are: (1) We introduce this new, practically-important problem of contextual policy optimization with unknown context rewards. (2) We develop new GP models that take advantage of the problem structure to significantly improve over existing BO approaches. (3) We provide a thorough simulation study that shows how the models scale with factors such as the number of contexts and the population distribution of contexts, considering both aggregate rewards and fairness. (4) We introduce a new real-world problem for CPO, optimizing a contextual ABR policy, and show that our models perform best relative to a wide range of alternative approaches.

Empirically, we have found the proposed contextual GP models work well in practice for a variety of contextual policy optimization use cases, including mobile data retrieval policies, cache eviction policies, and video streaming applications. Fig. 1 gives a preview of how the methods proposed in our paper (SAC and LCE-A) enable inference and optimization of high-dimensional contextual policies using data from a real-world video streaming experiment. For reproducibility, our results in Sec. 5 leverage an open-source video streaming simulator and trace data from video playback sessions on Facebook's Android app to evaluate how these models improve CPO performance. Code for the models and replication materials are available at `https://github.com/facebookresearch/ContextualBO`.

1.1 Background and Related work

Bayesian Optimization BO is a model-based optimization technique for settings with time-consuming function evaluations, such as A/B tests. The model is typically a GP, denoted $f \sim \mathcal{GP}(\mu(\cdot), k(\cdot, \cdot))$, where $\mu(\cdot)$ is the prior mean (usually taken as 0), and $k(\cdot, \cdot)$ is the kernel. BO is highly sample efficient, but struggles in high-dimensional settings with more than 15–20 parameters [44].

High-dimensional BO Additive structure [12] has been used for handling high-dimensional search spaces in BO. [20] assumes the response surface is a sum of low-dimensional functions, each depending on only a subset of the input variables. Additional work on additive BO has developed a variety of approaches for inferring the parameter decomposition [43, 14, 42, 39, 33]. Another set of methods for high-dimensional BO have assumed low-dimensional linear [44, 6, 36, 7, 34, 25] or nonlinear [15, 27, 32] structure to the problem.

Multi-Task BO BO where observations are made for one of several tasks (such as different datasets, or here different context settings) is discussed by [40] and called multi-task BO. They used a GP with the intrinsic coregionalization model (ICM) kernel [3] to borrow strength across tasks. [18] extended the ICM kernel with a new similarity measurement to construct the inter-task covariance matrix. A similar kernel structure has been used for GPs over a mix of discrete and continuous parameters [37].

Contextual BO Contextual BO is related to the much-studied contextual bandits problem [26], though with a continuous action space rather than discrete arms. [21] used BO for contextual policy search with both continuous and discrete context spaces. They used multiplicative and additive kernels to incorporate continuous context spaces into the GP, and used a multi-task GP (MTGP) for discrete contexts. The model can then be used to identify the best parameter values for any given context. [30] similarly used BO for CPO by augmenting the parameter space with the continuous context space and using a typical GP. [9] used an MTGP for CPO with discrete contexts, which enables sharing information across contexts since context rewards are often correlated. In all of this prior work, context was observed along with the reward (that is, context-level rewards) as opposed to the aggregate rewards setting we consider here.

[41] developed acquisition functions for a robotics setting with the goal of finding a contextual policy that maximizes aggregate rewards, integrated over the context distribution. Each evaluation in the policy optimization phase was of a single context, and that context was selected by the experimenter. The ability to select individual contexts for evaluation is possible in simulation or lab experiments, but not in many real-world settings, such as ours. A related line of work in robust optimization seeks to find a non-contextual policy that optimizes the aggregate rewards [45, 23], again while selecting contexts and observing rewards at the individual context level. Our work here has the same goal of maximizing aggregate rewards, but we directly observe that aggregate and cannot specify or even observe individual contexts for evaluation. Maximizing aggregate rewards using only observations of aggregate rewards allows for CPO when reward is evaluated across heterogeneous environments while avoiding the difficult credit assignment problem.

2 Contextual Policy Search with Multi-Task BO

In this work, we consider the case of discrete contexts. The multi-task models used in prior CPO work with discrete contexts [21, 9] assume that rewards are observed for each context, and so cannot be applied to problems with unknown context rewards. However, multi-task models form the foundation and motivation for the models that we develop for unknown context rewards. We now describe the classic MTGP here, and then develop a novel extension with latent embeddings.

2.1 The Intrinsic Coregionalization Model

When context $c \in \{1, \dots, C\}$ is observed along with each reward, contextual BO can be framed as a multi-task BO problem in which each context setting corresponds to a task. The MTGP extends the GP from a single function to a collection of functions $\mathbf{f} = \{f_c(\mathbf{x})\}_{c=1}^{C}$ with output space $\mathbb{R}^C$ and f_c the response function for context c. The MTGP covariance function models the covariance across both contexts and the parameter space $\mathcal{X} = \mathbb{R}^d$, that is, $k((c, \mathbf{x}), (c', \mathbf{x}')) = \text{cov}[f_c(\mathbf{x}), f_{c'}(\mathbf{x}')]$. The most common choice for $k(\cdot, \cdot)$ is the ICM kernel [8]—see [3] for a comprehensive discussion of *multi-output* GPs.

The ICM kernel assumes separability between the covariance across contexts and across parameters:

$$\text{cov}[f_c(\mathbf{x}), f_{c'}(\mathbf{x}')] = K_{c,c'} k^x(\mathbf{x}, \mathbf{x}'),$$

where $K_{c,c'}$ is a positive semi-definite (PSD) matrix that models the similarities between tasks (the task covariance matrix) and $k^x(\mathbf{x}, \mathbf{x}')$ is the kernel over the parameter space, which is shared across contexts. This is an instance of a special class of multi-output kernel functions called sum of separable (SoS) kernels. A "free-form" task covariance matrix is used in [8] written as $\mathbf{B}$, a $C \times C$ matrix, estimated using empirical Bayes. The kernel matrix for a dataset $\mathbf{X}$ can be efficiently evaluated as

$$K(\mathbf{X}, \mathbf{X}) = \mathbf{B} \otimes K^x(\mathbf{X}, \mathbf{X}).$$

The ICM kernel implicitly models each task function f_c as a linear combination of independent latent functions, $f_c(\mathbf{x}) = \sum_{i=1}^{m} a_{c,i} u_i(\mathbf{x})$, where $u_i \sim \mathcal{GP}(0, k^x(\cdot,\cdot))$. The rank of $\mathbf{B}$ is equal to the number of latent functions m, so when $m < C$, $\mathbf{B}$ is not full rank. A low-rank task covariance matrix can induce helpful regularization [8], and also reduces the number of kernel hyperparameters from $\mathcal{O}(C^2 + d)$ to $\mathcal{O}(Cm + d)$.

2.2 Latent Context Embeddings

An alternative approach for handling contexts is to map them to a low-dimensional Euclidean space. This approach is especially useful when contexts take on many possible values, and is widely used in deep learning, where embeddings of discrete variables (called entity embeddings) are learned as part of the neural network during the standard training process [16].

We use this strategy to create a new kernel for contextual BO for the setting of observed context rewards. The context c is one-hot encoded and passed through an embedding layer that maps it into τ-dimensional Euclidean space. It is then augmented with any external or existing embeddings for the contexts. For instance in the ABR problem one might have an existing embedding of country based on network characteristics. Given an η-dimensional external embedding, each context c is then represented as a point $\mathbf{z}^c \in \mathbb{R}^{(\tau+\eta)}$. The parameters of the embedding layer are jointly fit with the kernel hyperparameters by maximizing the marginal likelihood; the external embedding remains fixed. Covariance across tasks is measured with a PSD kernel function $k^z(\cdot,\cdot)$ over $\mathbb{R}^{(\tau+\eta)}$ (we use an ARD RBF kernel). Letting $E : \{1, \dots, C\} \to \mathbb{R}^{(\tau+\eta)}$ represent the combined embeddings,

$$\mathrm{Cov}[f_c(\mathbf{x}), f_{c'}(\mathbf{x}')] = k^z(E(c), E(c'))k^x(\mathbf{x}, \mathbf{x}'). \tag{1}$$

We call this the *latent context embedding multi-output* (LCE-M) kernel.

LCE-M uses the same separability assumption as ICM, but differs in that it can incorporate external embedding information, and also differs in the way that it implicitly regularizes the function. [3] discusses the connection between GPs and frequentist kernel methods based on reproducing kernel Hilbert spaces (RKHS), and shows that the regularization of a GP prior is related to the norm of function in the corresponding RKHS. For LCE-M and ICM,

$$\|\mathbf{f}\|_K^2 = \sum_{c,c'=1}^{C} \mathbf{B}^*_{c,c'} \langle f_c, f_{c'} \rangle_k$$

where $\mathbf{B}^*$ is the pseudo-inverse of the task covariance matrix. In the ICM kernel, regularization comes from reducing the rank of $\mathbf{B}$. In the LCE-M kernel, the embedding dimension $\tau < C$ provides regularization by restricting $\mathbf{B}$ without reducing its rank, and thus without reducing the number of latent functions being used to model $\mathbf{f}$. The embedding dimensionality τ can be treated as a hyperparameter, or set with a rule of thumb like $C^{0.25}$ [1]. A similar kernel was used by [47] for categorical variables, though without the inclusion of an existing embedding for flexibly incorporating external information.

3 Contextual Bayesian Optimization with Aggregate Rewards

When the contexts are observed along with the rewards, the contextual policy is constructed by jointly modeling context and parameters. For the ICM kernel, the dimensionality of the model space is only d with C tasks; for LCE-M it is $d + \tau + \eta$. In the aggregate reward setting, we observe a single reward for the entire $(C \times d)$-parameter contextual policy, and have no choice except to solve a high-dimensional optimization problem. We develop two kernels that allow for effective BO in this space by taking advantage of the particular structure of the aggregated CPO problem. This approach fits into the broader framework of grey-box optimization by exploiting the additive structure of the objective function to improve optimization efficiency [4].

3.1 Structural Assumptions for Aggregated Contextual Policy Search

We consider three assumptions that enable modeling in the high-dimensional contextual policy space. We denote the full policy as $\bar{\mathbf{x}} \in \mathbb{R}^{C \times d}$, and let $\mathbf{x}_c \in \mathbb{R}^d$ represent the parameters for context c.

Assumption A1. *No interference*: A policy $\bar{\mathbf{x}}$ affects the rewards of individual units with context c only via $\mathbf{x}_c$.
Assumption A2. *Single spatial kernel*: The (unobserved) response surfaces for each context $c = 1, \ldots, C$ have the same covariance function over the parameters $k^x(\cdot, \cdot)$. In essence, the smoothness of the response surface with respect to the parameters is the same under each context (note that how reward depends on the parameters need not be the same for each context).
Assumption A3. *Stable context distribution*: The distribution of context values is constant. This is required for consistency of the aggregate reward both in time and parameter space.

A1 and A2 are assumptions also made by the multi-task models in Sec. 2: A1 is implied by the kernel separability, and A2 is used when giving the same $k^x(\cdot, \cdot)$ to each task. In A3 we assume the context distribution is stable, but we do not assume it is known, and will infer it in our models here. Under these assumptions, we can derive an appropriate GP kernel for the aggregate reward response. Let $f : \mathbb{R}^{C \times d} \to \mathbb{R}$ be the aggregate reward response, $f_c : \mathbb{R}^d \to \mathbb{R}$ the unobserved unit-level reward for c, and w_c the population frequency of c. Then,

$$f(\bar{\mathbf{x}}) = \sum_{c=1}^{C} w_c f_c(\mathbf{x}_c). \tag{2}$$

3.2 The Structural Additive Contextual Kernel

If we take f_c as i.i.d. $\mathcal{GP}(0, k^x(\cdot, \cdot))$ in (2), we can derive the corresponding kernel which we call the *structural additive contextual* (SAC) kernel:

$$\text{Cov}[f(\bar{\mathbf{x}}), f(\bar{\mathbf{x}}')] = \sum_{c=1}^{C} w_c^2 k^x(\mathbf{x}_c, \mathbf{x}_c').$$

Fitting a typical ARD kernel to the high-dimensional function would require $C \times d + 1$ hyperparameters ($C \times d$ lengthscales and an output scale). For ARD $k^x(\cdot, \cdot)$, the SAC kernel has $C + d$ hyperparameters (d lengthscales and C context weights w_c) which allows incorporating a larger number of contexts.

The SAC kernel is similar to the additive kernels used for generic high-dimensional BO described in Sec. 1.1, such as Add-GP-UCB [20], with a key difference: since those do not have the explicit $C \times d$ block structure that comes with contextualization, they use isotropic kernels across each component. The SAC kernel is able to use an ARD kernel for each context, and share lengthscales for $x_{c,i}$ and $x_{c',i}$ to better borrow strength across contexts. In that regard, the SAC kernel has a similar flavor to the ICM kernel: it is equivalent to summing the outputs of an ICM model with a diagonal task covariance matrix. In practice, the shared kernel is a significant source of ICM generalization across tasks [24], and the SAC kernel brings that same power to contextual BO.

3.3 Latent Context Embeddings with Aggregate Rewards

Just as the SAC kernel extends the ICM structure to the aggregate rewards setting, we can extend the LCE-M model to aggregate rewards to get the benefit of context embeddings. We give the latent context rewards $f_1, \ldots, f_C$ a joint GP prior with the LCE-M kernel. Then, combining (1) and (2),

$$\begin{aligned} \text{Cov}\left[f(\bar{\mathbf{x}}), f(\bar{\mathbf{x}}')\right] &= \text{Cov}\left[\sum_{c=1}^{C} w_c f_c(\mathbf{x}_c), \sum_{c=1}^{C} w_c f_c(\mathbf{x}_c')\right] \\ &= \sum_{c=1}^{C} \sum_{c'=1}^{C} w_c w_{c'} k^z(E(c), E(c')) k^x(\mathbf{x}_c, \mathbf{x}_c'). \end{aligned} \tag{3}$$

We call this the *latent context embedding additive* (LCE-A) kernel. This kernel has $(1 + \tau) \times C + d + \tau + \eta$ hyperparameters. Although it has more hyperparameters than the SAC kernel, the model complexity is not necessarily higher as the kernel $k^z(E(c), E(c'))$ imposes regularization and encourages correlations across component functions. With stronger correlation, there is a larger norm and more regularization, which results in a lower model complexity.

LCE-A maintains the benefits of the SAC kernel sharing a spatial kernel across contexts, while also explicitly modeling and accounting for the correlation in the latent context rewards. As we will see in Sec. 4, this leads to benefits in both average performance and in robustness.

4 Numerical Experiments

We use a series of synthetic problems to study the performance characteristics of our proposed models, including their performance relative to the observed context rewards setting; their ability to scale to large numbers of contexts; their ability to improve outcomes for individual contexts, including rare contexts; and their performance under model misspecification.

The results here focus on a family of synthetic benchmarks based on the Hartmann6 test function. The parameter space is taken to be the first five dimensions of the function ($d = 5$), and the last dimension is used as an unobserved latent context dimension. The latent context values are spaced uniformly across the sixth input dimension. We then compute aggregate rewards with the formula in (2). This benchmark construction allows us to assess BO performance on CPO with aggregate rewards while varying both the number of contexts C and the weight distribution w_c. Because we can compute the true latent f_c, we can also compare with multi-task models that have access to context-level rewards. For BO, we use EI [19] as the acquisition function, though our models are agnostic to the acquisition function. In all benchmarks, we consider at least four methods: quasi-random search (Sobol) [35], BO with a standard ARD Matérn 5/2 kernel [38] (Standard BO), and BO with the SAC and the LCE-A models. We used embedding dimension $\tau = 1$ for LCE-A. Results with alternative synthetic functions are given in the Appendix and are qualitatively the same. All plots show the mean and 95% confidence intervals (two standard errors) across 15 runs.

Scaling with number of contexts. We first consider the setting where each context has equal weight $w_c = \frac{1}{C}$. Fig. 2 shows the results of using BO for CPO with aggregate rewards (StandardGP, SAC, and LCE-A), as well as with fully observed context rewards (ICM and LCE-M). In the latter case, BO was performed separately per context. With aggregate rewards, the dimensionality of the search space grows linearly with the number of contexts C. The performance with the standard GP rapidly degraded while the other methods found significantly better policies across the full range of C. Because the LCE-A is able to borrow strength across contexts, it scales better with the number of contexts relative to the other aggregate reward methods, as indicated by a slower decrease in final reward as C grew.

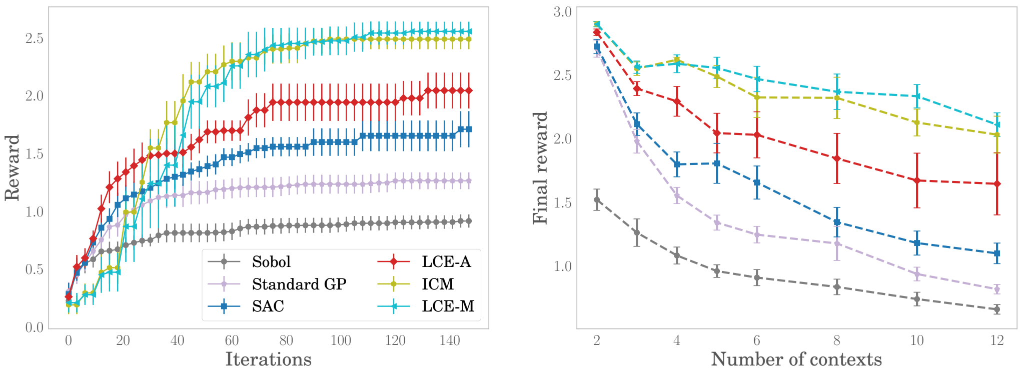

Figure 2: Contexual Hartmann5D with uniform weights. (Left) benchmark traces (average best found across 15 runs, with 95% confidence interval over 150 trials) for each method for $C = 5$ contexts with equal weight. LCE-A performed best among methods using aggregated reward. The gap between LCE-A and LCE-M indicates the value of knowing context-level rewards. (Right) Final reward of 200 trials for different numbers of contexts. The best value found by LCE-A decreased much slower with more contexts, showing the scalability of high-dimensional contextual policy search.

Fig. 2 also demonstrates that observing context-level rewards (ICM and LCE-M) unsurprisingly improves BO performance. However, aggregate-rewards CPO with LCE-A was able to capture much of the benefit of CPO. Furthermore, for $C \geq 5$ the regret gap between LCE-A and multi-task models was constant: aggregate-rewards CPO did not scale worse with C than multi-task CBO. This result shows that, with LCE-A, CPO can be a valuable tool in settings where observing context-level rewards is infeasible.

Robustness to the context distribution. In many applications some contexts make up only a small portion of the total population, which can make it especially challenging to optimize policies in the aggregate reward setting. We assessed CPO performance in this setting by skewing the weight distribution: a small number of contexts were given the majority of the weight (called *dense contexts*), and the remaining (*sparse contexts*) were each given weight $w_c = 0.01$ (where weights sum to 1).

Fig. 3 shows the optimization performance on this problem, for two and five dense contexts, while varying the total number of contexts. LCE-A remained the best-performing method in all scenarios. With a skewed context distribution, the effective dimensionality of the problem is lowered because optimizing for dense contexts alone will capture most of the available reward. Because of this, the best reward found by both the SAC and LCE-A models remained essentially constant when increasing the number of sparse contexts.

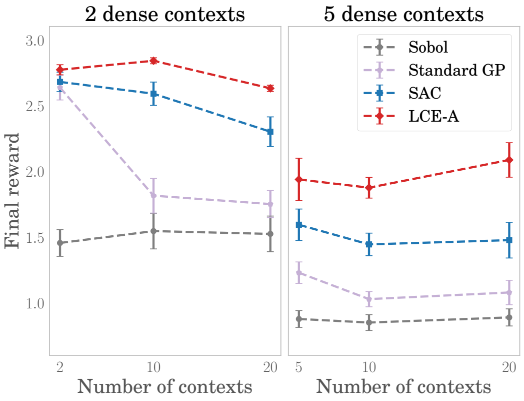

Figure 3: Best reward found for contextual Hartmann5D with a skewed context weight distribution over 100 trials. s sparse contexts were each given weight 0.01, and dense contexts split the remaining $1 - 0.01s$ weight. Increasing the number of sparse contexts did not significantly alter the optimization.

One potential concern with optimizing contextual policies without observing context-level rewards is that the policy for rare contexts could be poorly identified and the contextual policy could degrade their experience. This relates to the issue of fairness in machine learning [10]. Fig. 4 shows optimization traces for one of the sparse-context problems of Fig. 3 in which aggregate reward was optimized with 5 dense and 5 sparse contexts (total weights of 0.95 and 0.05, respectively). The left figure shows performance on the aggregate reward across all 10 contexts (the objective of the optimization), while the right figure shows the reward of just the sparse contexts.

The SAC model did well at optimizing the top-level aggregate reward, but performance on the rare contexts was no better than random. Since the SAC model assumes independence across contexts, the model is unable to infer the latent reward responses for the rare contexts and cannot craft a good policy. The LCE-A model, on the other hand, is able to borrow strength from the dense contexts to the sparse contexts, and performed well even on the sparse contexts. This result shows that, with LCE-A, CPO with aggregated rewards is robust to rare contexts and can produce policies that improve performance for all contexts.

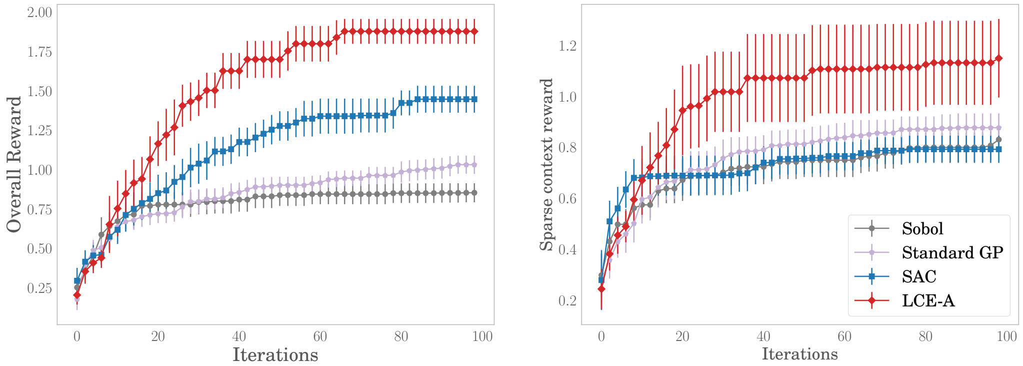

Figure 4: Optimization performance with 5 dense and 5 sparse contexts. (Left) Overall optimization performance over 100 trials. (Right) Performance on the sparse contexts. Compared with other models, LCE-A produced the most fair results and significantly improved the policy for the rare contexts.

Dynamic context distributions. In real-world settings like internet services or mobile applications, policies may cause individuals to change their context. For example, a poor experience with video conferencing over a cellular network may cause individuals to change to wifi when possible. This means context weights may depend on the unobserved context-level rewards, and thus depend on the policy parameters in direct violation of Assumption A3.

To study the impact of this on our models, we consider the case in which the context distribution shifts according to the rewards. Specifically, we set $w_c(\bar{\mathbf{x}}) = f_c(\mathbf{x}_c)/\sum_{c'} f_{c'}(\mathbf{x}_{c'})$, so that weight shifts towards contexts with higher unit-level rewards. Fig. 5 shows the optimization performance with $C = 50$ contexts. Despite the violation of A3, both the SAC and LCE-A models were able to effectively optimize and found policies much better than random and standard BO.

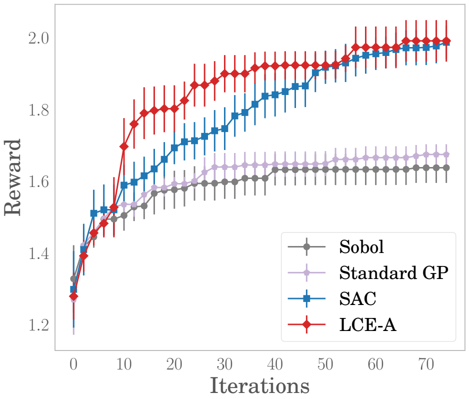

Figure 5: Optimization with 50 contexts and a dynamic context distribution. LCE-A performed well despite the violation of Assumption A3.

5 Contextual Policy Search in a Realistic Environment

We now study contextual policy search in a realistic environment: ABR video playback policies in heterogeneous mobile environments, using de-identified trace data from the Facebook Android mobile app. ABR algorithms determine the bitrate of each video chunk (here a one second segment of video) based on observations from the network (e.g., bandwidth fluctuation) and the video stream (e.g., playhead location). The goal is to maximize the quality of experience (QoE). Following the literature on model predictive control for video streaming [46], we consider an operationalization of QoE that is a weighted sum of the quality of the video chunks, stall time, and inter-temporal variation in quality. With this quality function, the goal of the playback controller is to request the highest quality chunks while ensuring an experience that is free of stalls.

We consider a contextual ABR policy that varies the parameters of an ABR controller based on the device's current *connection quality*. We evaluate our method with a modified video streaming simulator based on the *Park* platform [29]. The simulator mimics real-world network conditions based on de-identifed measurements from video playback sessions on Facebook's mobile android client. This allows us to consider the connection quality, fluctuations in available bandwidth, latency, and compression rates for each playback session.

The contextual policy is specified as a 48-dimensional vector, which specifies four of parameters of a linear controller for 12 possible contexts (see Appendix S1.1 for details). Playback sessions were classified using a 12-category clustering of connection quality based on initial measurements of throughput, latency, and connection type (cell or WiFi).

Fig. 6 shows optimization performance on the task of finding a contextual policy that maximizes QoE. In addition to the methods in Sec. 4, we compared a broad collection of existing methods: additive kernel methods Add-GP-UCB [20] and Ensemble BO (EBO) [43]; linear subspace BO methods ALEBO [25], HeSBO [34], and REMBO [44]; trust-region method TuRBO [13]; and the evolutionary strategy CMA-ES [17]. See the Appendix for the details for each comparison method. We also compared to doing regular BO on the 4-d non-contextual policy, that uses the same controller for all 12 contexts. All BO benchmarks were initialized with 8 quasi-random (Sobol) points. The benchmark results were averaged over 25 runs.

Fig. 6 (left) shows that the LCE-A method learned the best ABR controller policy. Sobol and Standard GP failed to handle the 48-dimensional search space. The linear embedding methods performed better than random, however there is no linear low-dimensional structure to this parameter space, so their final best QoE was 25% worse then LCE-A. Add-GP-UCB and TuRBO had better performance, and achieved QoE 6% less than LCE-A. By leveraging the problem structure and sharing information among contexts, LCE-A and SAC demonstrated higher efficiency than Add-GP-UCB. We excluded CMA-ES from the figure because it was unable to achieve a QoE better than the non-contextual policy

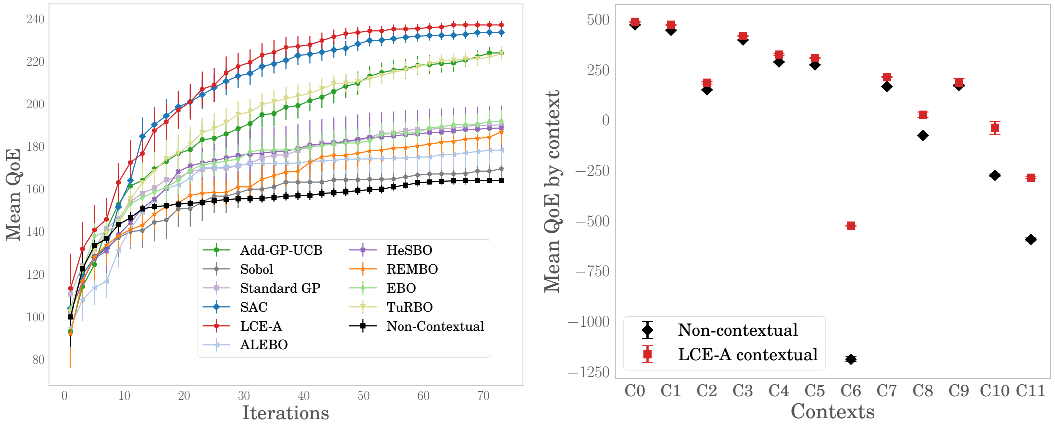

Figure 6: (Left) Average optimization performance for ABR policies with 4 parameters and 12 network connection contexts (error bars show two standard errors of the mean). LCE-A produced the contextual policies with the highest rewards (Right). LCE-A was able to improve the policy for each context (connection quality cluster) relative to the non-contextual policy. Contexts are sorted by mean latency of the cluster, and show that contextualization provided significant benefit for devices with high latency like C11

(i.e 150 ± 10). There was significant benefit to contextualization, with LCE-A producing a 50% increase in QoE relative to the non-contextual policy. Importantly, LCE-A also produced significant improvements for each context individually (Fig. 6, right). For example, the context C6 had the worst QoE under the non-contextual policy. This context had low bandwidth and produced only 1.8% of the total trace population, so it made only a small contribution to the aggregate reward and was not well-supported by the non-contextual optimization. LCE-A was able to significantly improve the outcomes in this context, despite its small weight. Even with aggregate rewards, contextual policy search with LCE-A is able to improve outcomes for all contexts.

6 Conclusion

We have shown that it is possible to deploy and optimize contextual policies even when rewards cannot be measured at the level of context. This enables the use of contextual BO in a broad range of applications where rewards are measured only in aggregate, as is common in A/B testing platforms. The LCE-A model makes it possible to optimize in high-dimensional policy spaces by leveraging plausible inductive biases for contextual policies. This not only improves top-level aggregate rewards relative to non-contextual policies, but also improves the fairness of the policy by improving outcomes across all contexts. The LCE-A model in particular is robust to the presence of rare contexts and can improve outcomes for small populations.

Our work aims to highlight a new, empirically-motivated problem and a class of structured solutions to this problem. There are many plausible extensions to this work, including more flexible models that include hierarchical priors for lengthscales across tasks. For instance, we can relax Assumption A2 by extending to the linear model of coregionalization (LMC) and determine model complexity adaptively. We consider a data regime under which few policies can be evaluated, but we expect that more function evaluations could enable modeling of population shifts caused by the policy (e.g., w_c is a function of $\bar{\mathbf{x}}$). Finally, we hope that future work can consider leveraging pre-trained, unsupervised representations of contexts to reduce the burden of learning good embeddings of contexts from scratch, which would further enable our method to scale to a very large number of contexts.

Broader Impact

The methods introduced in this paper expand the scope of problems to which contextual Bayesian optimization can be applied, and are especially important for settings where policies are evaluated with A/B tests. We expect this work to be directly beneficial in this setting, for instance for improving services at Internet companies as in the ABR example that we described in the paper. We are including our complete code for all of the models introduced in this paper, so the work will be immediately useful. As shown in the paper, contextualization improves not only the top-line performance of policies, but also improves the fairness of policies by improving outcomes specifically for small populations that do not achieve good performance under an existing non-contextual policy. This work will directly benefit these currently under-served populations.

Acknowledgements

We would like to thank Denise Noyse for her guidance on the ABR application and help with data, and Shaun Singh for his help with deploying the live experiments which validated the value of the methods proposed here. Finally, we'd like to thank David Arbour for his feedback on the paper.

References

[1] Introducing tensorflow feature columns. `https://developers.googleblog.com/2017/11/introducing-tensorflow-feature-columns.html`. Accessed: 2020-06-05.

[2] Mohan R Akella, Rajan Batta, Moises Sudit, Peter Rogerson, and Alan Blatt. Cellular network configuration with co-channel and adjacent-channel interference constraints. *Computers & Operations Research*, 35(12):3738–3757, 2008.

[3] Mauricio A. Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012.

[4] Raul Astudillo and Peter I Frazier. Bayesian optimization of composite functions. *arXiv preprint arXiv:1906.01537*, 2019.

[5] Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. In *Advances in Neural Information Processing Systems 33*, NeurIPS, 2020.

[6] Mickaël Binois, David Ginsbourger, and Olivier Roustant. A warped kernel improving robustness in Bayesian optimization via random embeddings. In *Proceedings of the International Conference on Learning and Intelligent Optimization*, LION, pages 281–286, 2015.

[7] Mickaël Binois, David Ginsbourger, and Olivier Roustant. On the choice of the low-dimensional domain for global optimization via random embeddings. *Journal of Global Optimization*, 76(1):69–90, 2020.

[8] Edwin V. Bonilla, Kian Ming A. Chai, and Christopher K. I. Williams. Multi-task Gaussian process prediction. In *Advances in Neural Information Processing Systems 20*, NIPS, pages 153–160, 2007.

[9] Ian Char, Youngseog Chung, Willie Neiswanger, Kirthevasan Kandasamy, Andrew Oakleigh Nelson, Mark Boyer, Egemen Kolemen, and Jeff Schneider. Offline contextual Bayesian optimization. In *Advances in Neural Information Processing Systems 32*, NeurIPS, pages 4627–4638, 2019.

[10] Sam Corbett-Davies and Sharad Goel. The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint arXiv:1808.00023*, 2018.

[11] Jeffrey Dean and Luiz André Barroso. The tail at scale. *Communications of the ACM*, 56(2):74–80, February 2013.

[12] David K. Duvenaud, Hannes Nickisch, and Carl E. Rasmussen. Additive Gaussian processes. In *Advances in Neural Information Processing Systems 24*, NIPS, pages 226–234, 2011.

[13] David Eriksson, Kun Dong, Eric Lee, David Bindel, and Andrew G. Wilson. Scaling Gaussian process regression with derivatives. In *Advances in Neural Information Processing Systems 31*, NIPS, pages 6867–6877, 2018.

[14] Jacob Gardner, Chuan Guo, Kilian Q. Weinberger, Roman Garnett, and Roger Grosse. Discovering and exploiting additive structure for Bayesian optimization. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, AISTATS, pages 1311–1319, 2017.

[15] Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018.

[16] Cheng Guo and Felix Berkhahn. Entity embeddings of categorical variables. *arXiv preprint arXiv:1604.06737*, 2016.

[17] Nikolaus Hansen, Sibylle D. Müller, and Petros Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary computation*, 11(1):1–18, 2003.

[18] Kohei Hayashi, Takashi Takenouchi, Ryota Tomioka, and Hisashi Kashima. Self-measuring similarity for multi-task Gaussian process. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pages 145–153, 2012.

[19] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[20] Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. High dimensional Bayesian optimisation and bandits via additive models. In *International Conference on Machine Learning*, ICML, pages 295–304, 2015.

[21] Andreas Krause and Cheng S. Ong. Contextual Gaussian process bandit optimization. In *Advances in Neural Information Processing Systems 24*, NIPS, pages 2447–2455, 2011.

[22] Andras Gabor Kupcsik, Marc Peter Deisenroth, Jan Peters, and Gerhard Neumann. Data-efficient generalization of robot skills with contextual policy search. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, AAAI, pages 1401–1407, 2013.

[23] Jeffrey Scott Lehman. *Sequential design of computer experiments for robust parameter design*. PhD thesis, The Ohio State University, 2002.

[24] Benjamin Letham and Eytan Bakshy. Bayesian optimization for policy search via online-offline experimentation. *Journal of Machine Learning Research*, 20(145):1–30, 2019.

[25] Benjamin Letham, Roberto Calandra, Akshara Rai, and Eytan Bakshy. Re-examining linear embeddings for high-dimensional Bayesian optimization. In *Advances in Neural Information Processing Systems 33*, NeurIPS, 2020.

[26] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, WWW, pages 661–670, 2010.

[27] Xiaoyu Lu, Javier González, Zhenwen Dai, and Neil Lawrence. Structured variationally auto-encoded optimization. In *Proceedings of the 35th International Conference on Machine Learning*, ICML, pages 3267–3275, 2018.

[28] Hongzi Mao, Shannon Chen, Drew Dimmery, Shaun Singh, Drew Blaisdell, Yuandong Tian, Mohammad Alizadeh, and Eytan Bakshy. Real-world video adaptation with reinforcement learning. *arXiv preprint arXiv:2008.12858*, 2020.

[29] Hongzi Mao, Parimarjan Negi, Akshay Narayan, Hanrui Wang, Jiacheng Yang, Haonan Wang, Ryan Marcus, Ravichandra Addanki, Mehrdad Khani, Songtao He, Vikram Nathan, Frank Cangialosi, Shaileshh Bojja Venkatakrishnan, Wei-Hung Weng, Song Han, Tim Kraska, and Mohammad Alizadeh. Park: An open platform for learning-augmented computer systems. In *Advances in Neural Information Processing Systems 32*, NeurIPS, pages 2490–2502, 2019.

[30] Jan Hendrik Metzen, Alexander Fabisch, and Jonas Hansen. Bayesian optimization for contextual policy search. In *Proceedings of the Second Machine Learning in Planning and Control of Robot Motion Workshop, IROS Workshop*, MLPC, 2015.

[31] Jacob M. Montgomery, Brendan Nyhan, and Michelle Torres. How conditioning on posttreatment variables can ruin your experiment and what to do about it. *American Journal of Political Science*, 62(3):760–775, 2018.

[32] Riccardo Moriconi, K. S. Sesh Kumar, and Marc P. Deisenroth. High-dimensional Bayesian optimization with manifold Gaussian processes. *arXiv preprint arXiv:1902.10675*, 2019.

[33] Mojmír Mutný and Andreas Krause. Efficient high dimensional Bayesian optimization with additivity and quadrature Fourier features. In *Advances in Neural Information Processing Systems 31*, NIPS, pages 9005–9016, 2018.

[34] Amin Nayebi, Alexander Munteanu, and Matthias Poloczek. A framework for Bayesian optimization in embedded subspaces. In *Proceedings of the 36th International Conference on Machine Learning*, ICML, pages 4752–4761, 2019.

[35] Art B Owen. Scrambling Sobol'and Niederreiter–Xing points. *Journal of Complexity*, 14(4):466–489, 1998.

[36] Hong Qian, Yi-Qi Hu, and Yang Yu. Derivative-free optimization of high-dimensional non-convex functions by sequential random embeddings. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, IJCAI, pages 1946–1952, 2016.

[37] Peter Z. G. Qian, Huaiqing Wu, and C. F. Jeff Wu. Gaussian process models for computer experiments with qualitative and quantitative factors. *Technometrics*, 50(3):383–396, 2008.

[38] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge, Massachusetts, 2006.

[39] Paul Rolland, Jonathan Scarlett, Ilija Bogunovic, and Volkan Cevher. High-dimensional Bayesian optimization via additive models with overlapping groups. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, AISTATS, pages 298–307, 2018.

[40] Kevin Swersky, Jasper Snoek, and Ryan P. Adams. Multi-task Bayesian optimization. In *Advances in Neural Information Processing Systems 26*, NIPS, pages 2004–2012, 2013.

[41] Matthew Tesch, Jeff Schneider, and Howie Choset. Adapting control policies for expensive systems to changing environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, IROS, pages 357–364, 2011.

[42] Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. Batched large-scale Bayesian optimization in high-dimensional spaces. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, AISTATS, 2018.

[43] Zi Wang, Chengtao Li, Stefanie Jegelka, and Pushmeet Kohli. Batched high-dimensional Bayesian optimization via structural kernel learning. In *Proceedings of the 34th International Conference on Machine Learning*, ICML, pages 3656–3664, 2017.

[44] Ziyu Wang, Frank Hutter, Masrour Zoghi, David Matheson, and Nando de Feitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016.

[45] Brian J. Williams, Thomas J. Santner, and William I. Notz. Sequential design of computer experiments to minimize integrated response functions. *Statistica Sinica*, 10(4):1133–1152, 2008.

[46] Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli. A control-theoretic approach for dynamic adaptive video streaming over HTTP. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM, pages 325–338, 2015.

[47] Yichi Zhang, Daniel W. Apley, and Wei Chen. Bayesian optimization for materials design with mixed quantitative and qualitative variables. *Scientific Reports*, 10(4924), 2020.

Supplemental Materials: High-Dimensional Contextual Policy Search with Unknown Context Rewards using Bayesian Optimization

S1 Additional Experiment Results

S1.1 Video Playback Experiment

In our experiment (Sec. 5), the ABR controller determines the video bitrate as a function of the current playback buffer and past network observations. Specifically, it computes a bitrate value v_t for each video chunk t with

$$v_t = p_b \tilde{b}_t + p_o o_t + p_c, \tag{S1}$$

where $\tilde{b}_t$ is the bandwidth estimation to download chunk t, o_t is the playback buffer occupancy and p_b, p_o and p_c are parameters of the model for BO to optimize. A bandwidth estimator computes $\tilde{b}_t$ with an exponential average over the past observations:

$$\tilde{b}_t = \sum_{t'=0}^{t} \exp\left[-p_w(\tau_t - \tau_{t'})\right] b_{t'} / \sum_{t'=0}^{t} \exp\left[-p_w(\tau_t - \tau_{t'})\right], \tag{S2}$$

where $b_{t'}$ is the bandwidth measurement at t', τ_t is the wall time when chunk t is downloaded and p_w is a weighting parameter for the exponential average.

With the value v_t, the controller selects the actual bitrate with a thresholding method similar to [28]. Specifically, it picks the maximum bitrate index i such that its corresponding threshold h_i is below the value v_t. Each threshold level is computed with a linear function $h_i = p_s + p_h i$, where p_s and p_h are set to be 1 and 3.

In total, the controller contains 4 parameters to search over, per context. The specific range of the parameters are in Table S1. In our simulator, we adapted the Park [29] to utilize 1912 traces of video playback sessions from the Facebook Android app, and estimated chunk sizes for each chunk of each playback session. Additional latency was added to chunk requests using context-level means from pre-defined connection quality clusters. This setup allows us to capture real-world heterogeneity in available bandwidth, as well as any differences in compression rates that may be correlated with network latency. Traces were classified according to a clustering of connection characteristics. Summary statistics about these clusters are given in Table S2. Clusters are sorted and labeled according to the mean latency for readability. One can see from the table that clusters vary in size, vary by an order of magnitude in mean bandwidth (BW), and vary significantly by latency (latency tends to be higher on clusters that include a higher proportion of connections on cell networks).

p_b	p_o	p_c	p_w
$[0, 1]$	$[0, 3]$	$[0, 1]$	$[0.0001, 0.25]$

Table S1: Parameter ranges.

S1.2 Method Implementations

The LCE-A and SAC models were implemented using BoTorch, a framework for BO in PyTorch [5]. The linear embedding methods (REMBO, HeSBO, and ALEBO) used the implementation of [25], and a linear subspace dimension of $d_e = 8$ for all three methods. The methods Add-GP-UCB, TuRBO, EBO, CMA-ES used reference implementations from their authors with default settings. See the reproduction code at `https://github.com/facebookresearch/ContextualBO` for the exact calls used for each method.

S1.3 Additional numerical experimental results

Independent latent context functions In this simulation, we consider the case where latent context functions are independent with each other, so there is no value to sharing information across contexts. The parameter space was 5-dimensional and the latent functions f_c were drawn from the same

Context	% Traces	On wifi (%)	Mean latency (ms)	Mean BW (Mbps)	Median BW (Mbps)
C0	10.7	100.0	47	3.09	2.85
C1	15.6	100.0	58	2.75	2.37
C2	11.9	100.0	59	0.96	0.85
C3	4.0	80.8	75	2.48	2.05
C4	5.1	40.8	82	1.42	1.02
C5	18.8	0.0	91	1.83	1.56
C6	1.8	84.4	92	0.20	0.18
C7	11.6	100.0	96	1.03	0.79
C8	7.2	26.4	129	0.66	0.43
C9	3.1	84.7	139	1.81	1.31
C10	2.9	66.5	157	0.63	0.29
C11	7.3	63.1	210	0.39	0.26

Table S2: Summary statistics of *connection quality clusters* used as context variables in the ABR simulation.

GP prior with zero mean function and an ARD Matérn 5/2 kernel. The outputscale was 1.0 and, lengthscales ranged from 0.25 to 0.5, and there were 8 contexts with equal weight. Fig. S1 shows the benchmark traces. Both SAC and LCE-A obtained the best values with a not-statistically-significant difference. This shows that LCE-A can continue to perform as well as SAC even when there is no correlation across contexts. The result also indicates that the performance gap between SAC and LCE-A decreases as context correlation goes to zero.

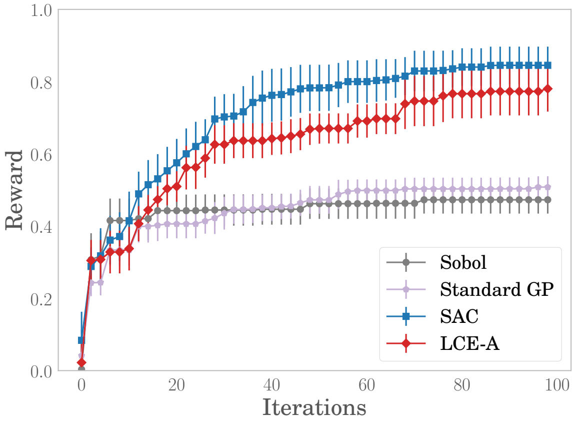

Figure S1: Benchmark trace (average best found across 10 runs, with 95% confidence interval) across methods for $C = 8$ contexts with equal weight. Each latent function was independently sampled from the same GP prior. SAC and LCE-A reached a similar final reward. Due to the independence across contexts, SAC converged to the optimum faster.

Dynamic context weights With the same generative model described in Sec. 4, we evaluated the methods across 10 and 20 contexts. As in Fig. 5, the LCE-A method performed well even with the violation of Assumption A3.

Branin2D function In these simulations we used as the latent functions the 2-dimensional Branin function, scaled by an exponential function $\exp(-u_c \exp(\frac{5+x_{c0}+x_{c1}}{30}))$, in which u_c is the 1-d dimension context embedding spaced evenly between $[0, 1]$ and $\mathbf{x_c}$ is the 2-d parameter vector. In Fig. S3, we show the results of optimization evaluations with 5, 10, and 15 contexts with equal context weights. The LCE-A models consistently performed best. As the number of contexts increased, the gaps in final reward between LCE-A and other methods grew.

Fig. S4 uses a skewed weight distribution with 5 dense contexts and 5 sparse contexts (each with weight 0.01, as in Sec. 4). The left plot shows the benchmark traces of aggregate rewards. LCE-A obtained a significantly better final reward than other methods. Since the dimensionality of the parameter space is low, standard BO method performed reasonably well at optimizing aggregate

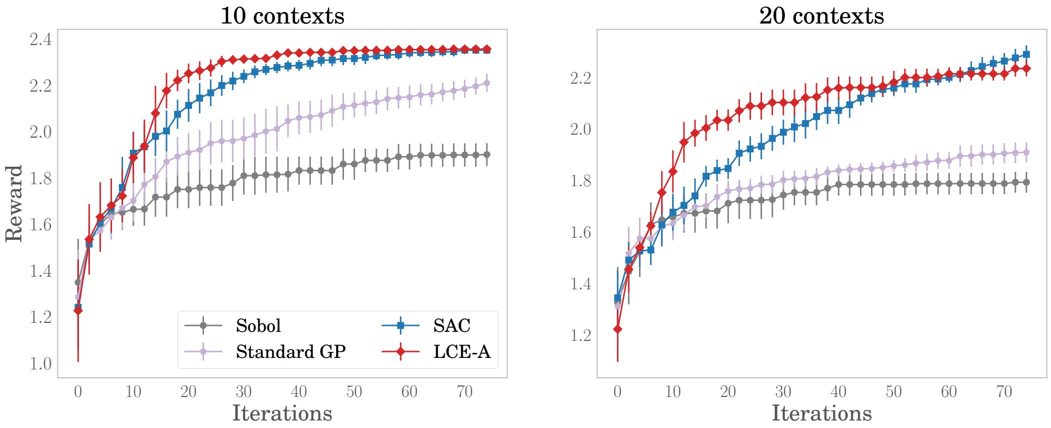

Figure S2: Contextual Hartmann3D with dynamic weights. Benchmark trace (average best found across 15 runs, with 95% confidence interval) across methods for 10 and 20 contexts.

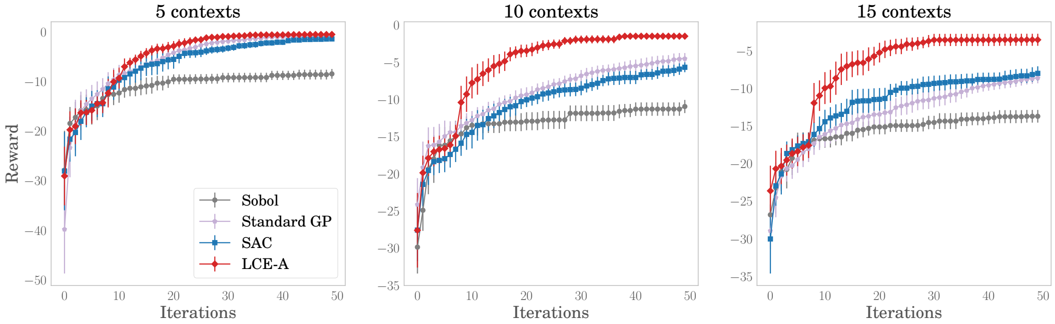

Figure S3: Contextual Branin2D for 5, 10 and 15 contexts with equal weight. Benchmark traces (average best found across 15 runs) show LCE-A outperformed other methods in all three cases.

reward. The right trace plot shows the reward just for sparse contexts. LCE-A performed well for these rare contexts while SAC and standard GP were similar to random search. This again highlights the robustness of LCE-A in handling rare contexts, besides improving overall reward.

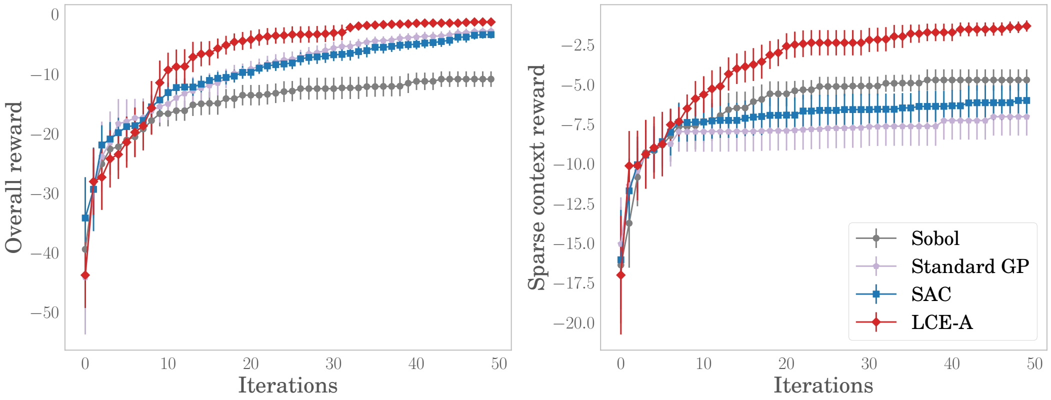

Figure S4: Contextual Branin2D for 10 contexts with a skewed weight distribution (5 dense contexts and 5 sparse contexts). The left plot shows average best found across 15 runs. LCE-A found the best aggregate reward. The right plot shows the trace for sparse contexts only. The LCE-A method performed robustly in optimizing rewards even of the sparse contexts.