# Time-Varying Gaussian Process Bandit Optimization

**Ilija Bogunovic, Jonathan Scarlett, Volkan Cevher**
Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)
Email: {ilija.bogunovic, jonathan.scarlett, volkan.cevher}@epfl.ch,

## Abstract

We consider the sequential Bayesian optimization problem with bandit feedback, adopting a formulation that allows for the reward function to vary with time. We model the reward function using a Gaussian process whose evolution obeys a simple Markov model. We introduce two natural extensions of the classical Gaussian process upper confidence bound (GP-UCB) algorithm. The first, R-GP-UCB, resets GP-UCB at regular intervals. The second, TV-GP-UCB, instead forgets about old data in a smooth fashion. Our main contribution comprises of novel regret bounds for these algorithms, providing an explicit characterization of the trade-off between the time horizon and the rate at which the function varies. We illustrate the performance of the algorithms on both synthetic and real data, and we find the gradual forgetting of TV-GP-UCB to perform favorably compared to the sharp resetting of R-GP-UCB. Moreover, both algorithms significantly outperform classical GP-UCB, since it treats stale and fresh data equally.

## 1 Introduction

In recent years, there has been a great deal of interest in the theory and methods for bandit optimization problems, where one seeks to sequentially select a sequence of points to optimize an unknown reward function from noisy samples [1, 2, 3]. Such problems have numerous applications, including sensors networks, recommender systems, and finance. A key challenge is to rigorously trade-off between *exploration*, i.e., learning the behavior of the function across the whole domain, and *exploitation*, i.e., selecting points that have previously given high rewards.

In the vast majority of practical applications, the function to be optimized is not static, but varies with time: In sensor networks, measured quantities such as temperature undergo fluctuations; in recommender systems, the users' preferences may change according to external factors; similarly, financial markets are highly dynamic. In such cases, the performance of standard algorithms may deteriorate, since these continue to treat stale data as being equally important as fresh data. The development of algorithms and theory to handle time variations is therefore crucial.

In this paper, we take a novel approach to handling time variations, modeling the reward function as a Gaussian process (GP) that varies according to a simple Markov model.

**Related Work:** Even in time-invariant settings, GP-based models provide a flexible and powerful approach to Bayesian optimization problems [4, 5]. Here, the smoothness properties of the reward function are dictated by a kernel function [6]. A wide variety of works have made use of upper confidence bound (UCB) algorithms, where the selected point maximizes a linear combination of the posterior mean and standard deviation. In particular, Srivinas *et al.* [1] provided regret bounds for the GP-UCB algorithm, and several extensions were given subsequently, including the contextual [7] and high-dimensional [8, 9, 10] settings.

While the study of time-varying models is limited in the GP setting, several such models have been considered in the multi-armed bandit (MAB) setting. Perhaps the most well-known one is the adversarial setting [3, 2, 11], where one typically seeks to compete with the best fixed strategy. Rewards modeled by Markov chains have been considered under the categories of *restless bandits* [12, 13, 14, 15], where the reward for each arm changes at each time step, and *rested bandits* [16, 17], where only the pulled arm changes.

Appearing in Proceedings of the $19^{th}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51. Copyright 2016 by the authors.

Two further related works are those of Slivkins and Upfal [15], who studied a MAB problem with varying rewards based on Brownian motion, and Besbes *et al.*[18], who considered a general MAB setting with time-varying rewards subject to a total budget in the amount of change allowed. Both [15] and [18] demonstrate the need for a *forgetting-remembering trade-off* arising from the fact that using the information from more samples may decrease the variance of the function estimates, while older information may be stale and hence misleading. Both papers present strategies in which the algorithm is reset at regular intervals in order to discard stale data. This is shown to be optimal in the worst case for the function class considered in [18], whereas in [15] it is shown that simple resetting strategies can be suboptimal in more specific scenarios, and alternative approaches are presented.

In contrast to GP-based settings such as ours, the setups of [15] and [18] consider finite action spaces, and assume independence between the rewards associated with different arms. Thus, observing the reward of one arm does not reveal any information about the other ones, and the algorithms are designed to exploit temporal correlations, but not spatial correlations.

**Contributions:** We introduce two algorithms for addressing the fundamental trade-offs inherent in the problem formulation: (i) trading off exploration with exploitation; (ii) differentiating between stale and fresh data in the presence of time variations; (iii) exploiting spatial and temporal correlations present in the reward function. Our main results present regret bounds, first for general kernels and then for the squared-exponential and Mátern kernels, that explicitly characterize the trade-off between the time horizon and the rate at which the function varies. Their proofs require novel techniques to handle difficulties arising from the time variations, such as the maximum function value and its location changing drastically throughout the duration of the time horizon. Moreover, we provide an algorithm-independent lower bound on the cumulative regret. Finally, we demonstrate the utility of our model and algorithms on both synthetic and real-world data

## 2 Problem Statement

We seek to sequentially optimize an unknown reward function $f_t$ over a compact, convex subset $D \subset \mathbb{R}^d$.[1] At time $t$, we can interact with $f_t$ only by querying at some point $x_t \in D$, after which we observe a noisy observation $y_t = f_t(x_t) + z_t$, where $z_t \sim \mathcal{N}(0, \sigma^2)$. We assume that the noise realizations at different time instants are independent. The goal is to maximize the reward via a suitable trade-off between exploration and exploitation. This problem is ill-posed for arbitrary reward functions even in the time-invariant setting, and it is thus necessary to introduce suitable smoothness assumptions. We take the approach of [1], and model the reward function as a sample from a Gaussian process, where its smoothness is dictated by the choice of kernel function.

**Model for the Reward Functions:** Let $k : D \times D \to \mathbb{R}_+$ be a kernel function, and let $\mathcal{GP}(\mu, k)$ be a Gaussian process [6] with mean $\mu \in \mathbb{R}^d$ and kernel $k$. As in [1], we assume bounded variance: $\forall x \in D, k(x, x) \leq 1$. Two common kernels are squared exponential (SE) and Matérn, defined as

$$k_{\text{SE}}(x, x') = \exp\left( -\frac{\|x - x'\|^2}{2l^2} \right) \tag{1}$$

$$k_{\text{Matérn}}(x, x') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left( \frac{\sqrt{2\nu}\|x - x'\|}{l} \right)^{\nu} \times B_\nu \left( \frac{\sqrt{2\nu}\|x - x'\|}{l} \right), \tag{2}$$

where $l > 0$ and $\nu > 0$ are hyperparameters, and $B_\nu$ denotes the modified Bessel function.

Letting $g_1, g_2, \dots$ be independent random functions on $D$ with $g_i \sim \mathcal{GP}(0, k)$, the reward functions are modeled as follows:

$$f_1(x) = g_1(x) \tag{3}$$

$$f_{t+1}(x) = \sqrt{1 - \epsilon}\, f_t(x) + \sqrt{\epsilon}\, g_{t+1}(x) \quad \forall t \geq 2, \tag{4}$$

where $\epsilon \in [0, 1]$ quantifies how much the function changes after every time step. If $\epsilon = 0$ then we recover the standard time-invariant model [1], whereas if $\epsilon = 1$ then the reward functions are independent between time steps. Importantly, for any choice of $\epsilon$ we have for all $t$ that $f_t \sim \mathcal{GP}(0, k)$. See Figure 1 for an illustration.

From a practical perspective, this model has the desirable property of only having one additional hyperparameter $\epsilon$ compared to the standard GP model, thus facilitating the learning process. It serves as a suitable model for reward functions that vary at a steady rate, though we will see numerically in Section 5 that the resulting algorithms are also effective more generally.

As noted in *regression* studies in [19, 20], our model is equivalent to a spatiotemporal kernel model with temporal kernel $(1 - \epsilon)^{|t_1 - t_2|/2}$. We expect our techniques to apply similarly to other temporal kernels, particularly *stationary* kernel functions that depend only on the time difference $|t_1 - t_2|$, but we focus on (3)–(4) for

[1] Finite domains were also handled in the time-invariant setting [1], and all of our upper bounds have counterparts for such scenarios that are in fact simpler to obtain compared to the compact case.

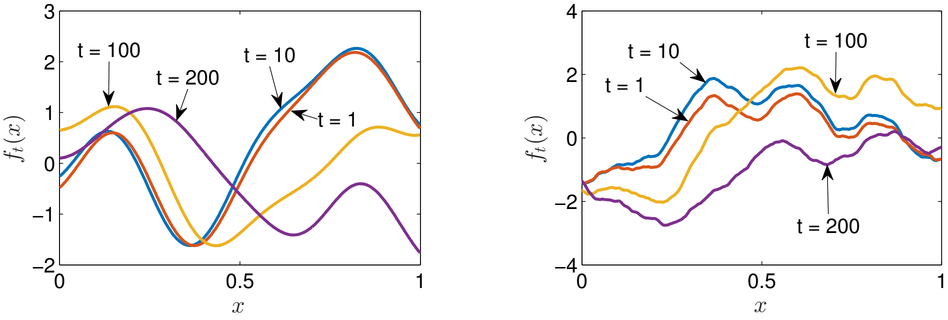


Figure 1: Two examples of GP functions when $\epsilon = 0.01$: (Left) Squared exponential kernel ($l = 0.2$); (Right) Matérn kernel ($l = 0.2$, $\nu = 1.5$). Note that the location of the maximum changes significantly at distant times.

concreteness. Spatiotemporal kernels can also be considered in the contextual bandit setting [7], but to our knowledge, no regret bounds have been given that explicitly characterize the dependence on the function's rate of variation, as is done in our main result.

**Regret:** Let $x_t^*$ denote a maximizer of $f_t$ at time $t$, i.e., $x_t^* = \arg\max_x f_t(x)$, and suppose that our choice at time $t$ is $x_t$. Then the *instantaneous regret* we incur at time $t$ is $r_t = f_t(x_t^*) - f_t(x_t)$. We are interested in minimizing the *cumulative regret* $R_T = \sum_{t=1}^T r_t$.

These definitions naturally coincide with those for the time-invariant setting when $\epsilon = 0$. Note that we do not aim to merely compete with fixed strategies, but instead to track the maximum of $f_t$ for all $t$. In our setting, a notion of regret based on competing with a fixed strategy would typically lead to a negative cumulative regret. In other words, *all fixed strategies perform poorly.*

In time-invariant scenarios, as well as several time-varying scenarios, algorithms are typically designed to achieve *sublinear regret.* In our setting, we will show that for *fixed* $\epsilon$, the cumulative regret $R_T$ must in fact be $\Omega(T)$ (*cf.*, Theorem 4.1). Intuitively, this is because if the function changes significantly at each time step, one cannot expect to track its maximum to arbitrary precision. However, we emphasize that what is really of interest is the *joint* dependence of $R_T$ on $T$ and $\epsilon$, and we thus seek regret bounds of the form $\tilde{\mathcal{O}}(T\psi(\epsilon))$ for some function $\psi(\epsilon)$ that vanishes as $\epsilon \to 0$.[2] Our approach is analogous to Slivkins and Upfal [15], who considered another time-varying setting with unavoidable $\Omega(T)$ regret for any fixed function variation parameter, and focused on the behavior in the implied constant in the limit as that parameter vanishes.

For the squared exponential and Matérn kernels, we obtain regret bounds of the form $\tilde{\mathcal{O}}(T\epsilon^\alpha)$ for some $\alpha > 0$ (*cf.*, Corollary 4.1), which can be viewed as being sublinear whenever $\epsilon = \mathcal{O}(T^{-c})$ for some $c > 0$. We observe that when $c < 1$, the correlation between $f_1(x)$ and $f_T(x)$ is negligible, meaning that the corresponding maximum may (and typically will) change drastically over the duration of the time horizon, e.g., see Figure 1.

**Limitations of GP-UCB:** We briefly recall the GP-UCB algorithm from [1], in which at each time step the selected point maximizes a function of the form $\mu_{t-1}(x) + \sqrt{\beta_t}\sigma_{t-1}(x)$. Here, defining $\mathbf{K}_t = \big[k(x,x')\big]_{x,x'\in\mathbf{x}_t}$ and $\mathbf{k}_t(x) = \big[k(x_i,x)\big]_{i=1}^t$, the quantities

$$\mu_{t+1}(x) := \mathbf{k}_t(x)^T\big(\mathbf{K}_t + \sigma^2\mathbf{I}_t\big)^{-1}\mathbf{y}_t \tag{5}$$

$$\sigma_{t+1}(x,x)^2 := k(x,x) - \mathbf{k}_t(x)^T\big(\mathbf{K}_t + \sigma^2\mathbf{I}_t\big)^{-1}\mathbf{k}_t(x), \tag{6}$$

are the posterior mean and variance of the time-invariant GP $f(x)$, respectively, given the previous samples $\mathbf{x}_t = [x_1,\dots,x_t]$ and corresponding observations $y_1,\dots,y_t$. Intuitively, one seeks points with a high mean $\mu_t$ to favor exploitation, but with a high standard deviation $\sigma_t$ to favor exploration.

In the time-invariant setting, GP-UCB is known to achieve sublinear regret under mild assumptions [1]. As mentioned above, the problem with using it in our setting is that it treats all of the previous samples as being equally important, whereas according to our model, the samples become increasingly stale with time. We now proceed to describing our algorithms that account for this fact.

## 3 Algorithms

We first introduce an algorithm R-GP-UCB that takes a conceptually simple approach to handling the forgetting-remembering trade-off, namely, running the GP-UCB algorithm within blocks of size $N$, and applying resetting at the start of each block. Some insight on how to choose $N$ is given by our bounds in

[2] Here and subsequently, the notation $\tilde{\mathcal{O}}(\cdot)$ denotes asymptotics up to logarithmic factors.

the following section. The pseudo-code is shown in Algorithm 1.

**Algorithm 1** GP-UCB with Resetting (R-GP-UCB)

```
Input: Domain $D$, GP prior ($\mu_0$, $\sigma_0$, $k$), block size $N$
for $t = 1, 2...$ do
    if $t$ mod $N = 1$ then
        Reset $\mu_{t-1}(x) = \mu_0(x)$ and $\sigma_{t-1}(x) = \sigma_0(x)$ for each $x$
    Choose $x_t = \arg\max_{x \in D} \mu_{t-1}(x) + \sqrt{\beta_t}\sigma_{t-1}(x)$
    Sample $y_t = f_t(x_t) + z_t$
    Perform Bayesian update as in (5)–(6), using only the samples $\{x_t\}$ and $\{y_t\}$ obtained since the most recent reset, to obtain $\mu_t$ and $\sigma_t$
```

Our second algorithm, TV-GP-UCB, instead forgets in a “smooth” fashion, by using a posterior update rule obtained via the time-varying model (3)–(4). In analogy with (5)–(6), the mean and variance of $f_t$ given the previous samples $\mathbf{x}_t = (x_1, \dots, x_t)$ and corresponding observations $y_1, \dots, y_t$ are given by

$$\tilde{\mu}_{t+1}(x) := \widetilde{\mathbf{k}}_t(x)^T \big(\widetilde{\mathbf{K}}_t + \sigma^2 \mathbf{I}_t\big)^{-1} \mathbf{y}_t \tag{7}$$

$$\tilde{\sigma}^2_{t+1}(x, x') := k(x, x) - \widetilde{\mathbf{k}}_t(x)^T \big(\widetilde{\mathbf{K}}_t + \sigma^2 \mathbf{I}_t\big)^{-1} \widetilde{\mathbf{k}}_t(x), \tag{8}$$

where $\widetilde{\mathbf{K}}_t = \mathbf{K}_t \circ \mathbf{D}_t$ with $\mathbf{D}_t = \big[(1-\epsilon)^{|i-j|/2}\big]_{i,j=1}^T$, and $\widetilde{\mathbf{k}}_t(x) = \mathbf{k}_t(x) \circ \mathbf{d}_t$ with $\mathbf{d}_t = \big[(1-\epsilon)^{(T+1-i)/2}\big]_{i=1}^T$. Here $\circ$ is the Hadamard product, and $\mathbf{I}_k$ is the $k \times k$ identity matrix.

The derivation of (7)–(8) is given in the supplementary material. Using these updates, the pseudo-code for the TV-GP-UCB algorithm is given in Algorithm 2. The idea is that the older a sample is, the smaller the value in the corresponding entries of $\mathbf{d}_t$ and $\mathbf{D}_t$ defined following (8), and hence the less it contributes to the final values of $\tilde{\mu}_t(x)$ and $\tilde{\sigma}_t(x)$. This algorithm can in fact be considered a special case of contextual GP-UCB [7] with a spatio-temporal kernel, but our analysis (Section 4) goes far beyond that of [7] in order to explicitly characterize the dependence on $T$ and $\epsilon$.

**Algorithm 2** Time-Varying GP-UCB (TV-GP-UCB)

```
Input: Domain $D$, GP prior ($\tilde{\mu}_0$, $\tilde{\sigma}_0$, $k$) and parameter $\epsilon$
for $t = 1, 2...$ do
    Choose $x_t = \arg\max_{x \in D} \tilde{\mu}_{t-1}(x) + \sqrt{\beta_t}\tilde{\sigma}_{t-1}(x)$
    Sample $y_t = f_t(x_t) + z_t$
    Perform Bayesian update as in (7)–(8) to obtain $\tilde{\mu}_t$ and $\tilde{\sigma}_t$
```

**Computational Complexity:** As it is presented above, TV-GP-UCB has an identical computational complexity to GP-UCB, i.e. the complexity of the sequential Bayesian update is $\mathcal{O}(T^2)$ [21]. R-GP-UCB is less complex, since the matrix operations are on matrices of size $N$ rather than the overall time horizon $T$. In practice, however, one could further modify TV-GP-UCB to improve the efficiency by occasionally resetting and/or discarding stale data [21].

# 4 Theoretical Bounds

In this section, we provide our main theoretical upper and lower bounds on the regret. We assume throughout this section that hyperparameters are known, i.e. both spatial kernel hyperparameters and $\epsilon$; in the numerical section (Section 5) we will address real-world problems where these are unknown. All proofs are given in the supplementary material.

## 4.1 Preliminary Definitions and Results

**Smoothness Assumptions:** Each of our results below will assume that the kernel $k$ is such that a (strict) subset of the following statements hold for some $(a_i, b_i)$ and all $L \geq 0$:

$$\Pr\left(\sup_{x \in D} \big|f(x)\big| > L\right) \leq a_0 e^{-(L/b_0)^2} \tag{9}$$

$$\Pr\left(\sup_{x \in D} \Big|\frac{\partial f}{\partial x^{(j)}}\Big| > L\right) \leq a_1 e^{-(L/b_1)^2}, \quad j = 1, \dots, d \tag{10}$$

$$Pr\left(\sup_{x \in D} \Big|\frac{\partial^2 f}{\partial x^{(j_1)} \partial x^{(j_2)}}\Big| > L\right) \leq a_2 e^{-(L/b_2)^2}, \quad j_1, j_2 = 1, \dots, d, \tag{11}$$

where $f \sim \mathcal{GP}(0, k)$. Assumption (9) is mild, since $f(x)$ is Gaussian and thus has exponential tails. Assumption (10) was used in [1], and ensures that the behavior of the GP is not too erratic. It is satisfied for the SE kernel, as well as the Matérn kernel with $\nu > 2$ [1], though for other kernels (e.g., Ornstein-Uhlenbeck) it can fail. Assumption (11) is used only for our lower bound; it is again satisfied by the SE kernel, as well as the Matérn kernel with $\nu > 4$.

**Mutual Information:** It was shown in [1] that a key quantity governing the regret bounds of GP-UCB in the time-invariant setting is the mutual information

$$I(\mathbf{f}_T; \mathbf{y}_T) = \frac{1}{2} \log\det\big(\mathbf{I}_T + \sigma^{-2}\mathbf{K}_T\big), \tag{12}$$

where $\mathbf{f}_T := \mathbf{f}_T(\mathbf{x}_T) = (f(x_1), \dots, f(x_T))$ for the time-invariant GP $f$. The corresponding maximum over any set of points $\mathbf{x}_T = (x_1, \dots, x_T)$ is given by

$$\gamma_T := \max_{x_1, \dots, x_T} I(\mathbf{f}_T; \mathbf{y}_T). \tag{13}$$

In our setting, the analogous quantities are as follows:

$$\tilde{I}(\mathbf{f}_T; \mathbf{y}_T) = \frac{1}{2} \log\det\big(\mathbf{I}_T + \sigma^{-2}\widetilde{\mathbf{K}}_T\big), \tag{14}$$

$$\tilde{\gamma}_T := \max_{x_1,\dots,x_T} \tilde{I}(\mathbf{f}_T; \mathbf{y}_T), \tag{15}$$

where $\mathbf{f}_T := \mathbf{f}_T(\mathbf{x}_T) = (f_1(x_1), \dots, f_T(x_T))$. While these take the same form as (12)–(13), they can behave significantly differently when $\epsilon > 0$. In particular, the time-varying versions are typically much higher due to the fact $\mathbf{f}_T$ represents the points of $T$ different random functions, as opposed to a single function at $T$ different points.

**Algorithm-Independent Lower Bound:** The following result gives an asymptotic lower bound for any bandit optimization algorithm under fairly mild assumptions, expressed in terms of the time horizon $T$ and parameter $\epsilon$.

**Theorem 4.1.** *Suppose that the kernel is such that $f \sim \mathcal{GP}(0,k)$ is almost surely twice continuously differentiable and satisfies* (10)*–*(11) *for some $(a_1, b_1, a_2, b_2)$. Then, any GP bandit optimization algorithm incurs expected regret $\mathbb{E}[R_T] = \Omega(T\epsilon)$.*

The proof reveals that this result holds true even in the full information (as opposed to bandit) setting, and is based on the fact that at each time step, there is a non-zero probability that the maximum value and its location change by an amount proportional to $\epsilon$. As discussed above, this lower bound motivates the study of the *joint* dependence on the regret of $T$ and $\epsilon$, and in particular, the highest possible constant $\alpha$ such that the regret behaves as $\tilde{\mathcal{O}}(T\epsilon^\alpha)$.

### 4.2 Main Results

We now present our main general bounds on the algorithms introduced in Section 3. The two provide regret bounds of a similar form, but we will shortly apply these to specific kernels and find that the bounds for TV-GP-UCB yield better scaling laws.

**General Regret Bounds:** The following theorems provide regret bounds for R-GP-UCB and TV-GP-UCB, respectively. We will simplify these bounds below to obtain scaling laws for specific kernels.

**Theorem 4.2.** *Let the domain $D \subset [0,r]^d$ be compact and convex, and suppose that the kernel is such that $f \sim \mathcal{GP}(0,k)$ is almost surely continuously differentiable and satisfies* (9)*–*(10) *for some $(a_0, b_0, a_1, b_1)$. Fix $\delta \in (0,1)$, and set*

$$\beta_T = 2\log\frac{2\pi^2T^2}{3\delta} + 2d\log\left(rdbT^2\sqrt{\log\frac{2da\pi^2T^2}{3\delta}}\right). \tag{16}$$

*Defining $C_1 = 8/\log(1+\sigma^{-2})$, the R-GP-UCB algorithm satisfies the following after $T$ time steps:*

$$R_T \le \sqrt{C_1 T\beta_T\left(\frac{T}{N}+1\right)\gamma_N} + 2 + T\psi_T(N,\epsilon) \tag{17}$$

*with probability at least $1-\delta$, where*

$$\psi_T(N,\epsilon) := \sqrt{\beta_T\big(3\sigma^{-2}+\sigma^{-4}\big)N^3\epsilon} + \big(\sigma^{-2}+\sigma^{-4}\big)N^3\epsilon(2+b_0)\sqrt{\log\frac{2(1+a_0)\pi^2T^2}{3\delta}}. \tag{18}$$

The proof of Theorem 4.2 departs from regular Bayesian optimization proofs such as [1] in the sense that the posterior updates (5)–(6) assumed by the algorithm differ from the true posterior described by (7)–(8), thus requiring a careful handling of the effect of the mismatch.

**Theorem 4.3.** *Let the domain $D \subset [0,r]^d$ be compact and convex, and suppose that the kernel is such that $f \sim \mathcal{GP}(0,k)$ is almost surely continuously differentiable and satisfies* (10) *for some $(a_1, b_1)$. Fix $\delta \in (0,1)$, and set*

$$\beta_T = 2\log\frac{\pi^2T^2}{2\delta} + 2d\log\left(rdbT^2\sqrt{\log\frac{da\pi^2T^2}{2\delta}}\right). \tag{19}$$

*Defining $C_1 = 8/\log(1+\sigma^{-2})$, the TV-GP-UCB algorithm satisfies the following after $T$ time steps:*

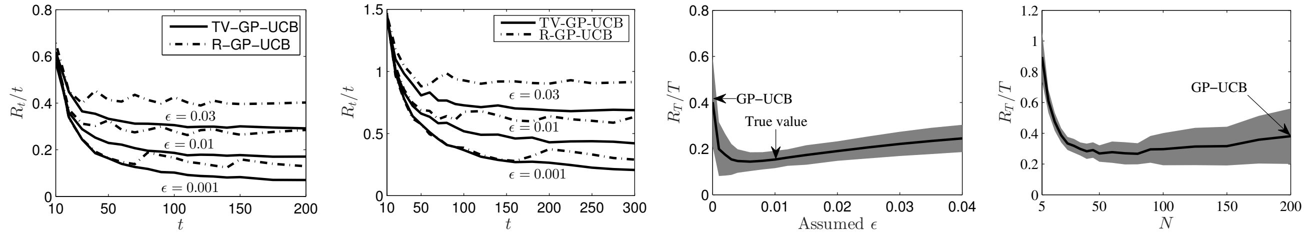
$$R_T \le \sqrt{C_1 T\beta_T\tilde{\gamma}_T} + 2 \tag{20}$$

$$\le \sqrt{C_1 T\beta_T\left(\frac{T}{\tilde{N}}+1\right)\left(\gamma_{\tilde{N}} + \tilde{N}^3\epsilon\right)} + 2 \tag{21}$$

*with probability at least $1-\delta$, where* (21) *holds for any $\tilde{N} \in \{1,\dots,T\}$.*

The step in (20) is obtained using techniques similar to those of [1, 7], whereas the step in (21) is non-trivial and new. This step is key to our analysis, bounding the maximum mutual information $\tilde{\gamma}_T$ for the time varying case in terms of the analogous quantity $\gamma_{\tilde{N}}$ from the time-invariant setting. The idea in doing this is to split the block $\{1,\dots,T\}$ into smaller blocks of size $\tilde{N}$ within which the overall variation in $f_t$ is not too large. This is in contrast with R-GP-UCB (and [18]), where the algorithm takes the block length $N$ as a parameter and explicitly resets the algorithm every $N$ time steps. For TV-GP-UCB, the length $\tilde{N}$ is only introduced as a tool in the analysis.

**Applications to Specific Kernels:** Specializing the above results to the squared exponential and Matérn



(a) SE kernel (b) Matérn52 kernel (c) Mismatched TV-UCB (d) Mismatched R-GP-UCB

Figure 2: Numerical performance of upper confidence bound algorithms on synthetic data.

kernels, using the corresponding bounds on $\gamma_N$ from [1], and optimizing $N$ as a function of $T$ and $\epsilon$, we obtain the following.

**Corollary 4.1.** *Under the conditions of Theorems 4.2 and 4.3, we have the following for any fixed d:*

1. *For the squared exponential kernel, $R_T = \tilde{\mathcal{O}}(\max\{\sqrt{T}, T\epsilon^{1/8}\})$ for R-GP-UCB with $N = \Theta(\min\{T, \epsilon^{-1/4}\})$, and $R_T = \tilde{\mathcal{O}}(\max\{\sqrt{T}, T\epsilon^{1/6}\})$ for TV-GP-UCB.*
2. *Consider the Matérn kernel with parameter $\nu > 2$, and set $c = \frac{d(d+1)}{2\nu + d(d+1)} \in (0,1)$. We have $R_T = \tilde{\mathcal{O}}(\max\{\sqrt{T^{1+c}}, T\epsilon^{\frac{1}{2}\frac{1-c}{4-c}}\})$ for R-GP-UCB with $N = \Theta(\min\{T, \epsilon^{-\frac{1}{4-c}}\})$, and $R_T = \tilde{\mathcal{O}}(\max\{\sqrt{T^{1+c}}, T\epsilon^{\frac{1}{2}\frac{1-c}{3-c}}\})$ for TV-GP-UCB.*

Observe that, upon substituting $\epsilon = 0$, the preceding $\tilde{\mathcal{O}}(\cdot)$ terms are dominated by the first terms in the maxima, and the bounds for both algorithms reduce to those in [1]. In the case that $\epsilon$ vanishes more slowly (e.g., as $1/\sqrt{T}$), the regret bounds for TV-GP-UCB are strictly better than those of R-GP-UCB. The worsened bounds for the latter arise due to the above-mentioned mismatch in the update rules.

For both kernels, the optimized block length $N$ of R-GP-UCB increases as $\epsilon$ decreases; this is to be expected, as it means that older samples are more correlated with the present function. We also observe that $N$ increases as the function becomes smoother (by increasing $\nu$ for the Matérn kernel, or by switching from Matérn to squared exponential).

## 5 Experiments

In this section, we test our algorithms on both synthetic and real data, as well as studying the effect of mismatch with respect to the algorithm parameters $\epsilon$ and $N$.

**Practical considerations:** While (16) and (19) give explicit choices for $\beta_t$, these usually tend to be too conservative in practice. For good empirical performance, we rely only on the *scaling* $\beta_t = O(d \log t)$ dictated by these choices, letting $\beta_t = c_1 \log(c_2 t)$ (similarly to [1, 22], for example). We found $c_1 = 0.8$ and $c_2 = 4$ to be suitable for trading off exploration and exploitation, and we therefore use these in all of our synthetic experiments.

Our theoretical analysis assumed that we know the hyperparameters of both spatial and temporal kernel. Having perfect knowledge of $\epsilon$ and other hyperparameters is typically not realistic. The GP perspective allows us to select them in a principled way by maximizing the GP marginal likelihood [6]. In our real-world experiments below, we select $\epsilon$ in such manner, using the approach from [19], outlined in Appendix B. In our synthetic experiments, we consider both the cases of perfect and imperfect knowledge of $\epsilon$.

**Baseline Comparisons:** We are not aware of any algorithms other than those in Section 3 that exploit both spatial and temporal correlations. In both our synthetic and real-data experiments, we found it crucial to handle both of these in order to obtain reasonable values for the cumulative regret, thus drastically limiting the number of reasonable baselines. Nevertheless, we also consider GP-UCB (which exploits spatial but not temporal correlations), and in the real-world experiments, we consider a completely random selection (thus corresponding to a choice that we should hope to beat significantly).

### 5.1 Synthetic Data

We consider a two-dimensional setting and quantize the decision space $D = [0,1]^2$ into $50 \times 50$ equally-spaced points. We generate our data according to the time-varying model (4), considering both the squared exponential and Matérn kernels. For the former we set $l = 0.2$, and for the latter we set $\nu = 2.5$ and $l = 0.2$. We set the sampling noise variance $\sigma^2$ to 0.01, i.e. 1% of the signal variance.

**Matched Case:** We first consider the case that the algorithm parameters are "matched". Specifically, the parameter $\epsilon$ for TV-GP-UCB is the true parameter for the model, and the parameter $N$ for R-GP-UCB is chosen in accordance with Corollary 4.1: $N = \lceil \min \{T, 12\epsilon^{-1/4}\} \rceil$ for the squared exponential kernel, and $N = \lceil \min \{T, 24\epsilon^{-\frac{1}{4-c}}\} \rceil$ for the Matérn kernel, where the constants were found via cross-validation.

In Figures 2a and 2b, we show the average regret $\frac{R_T}{T}$ of TV-GP-UCB and R-GP-UCB for $\epsilon \in \{0.001, 0.01, 0.03\}$. For each time shown, we average the performance over 200 independent trials. We observe that for all values of $\epsilon$ and for both kernel functions, TV-GP-UCB outperforms R-GP-UCB, which is consistent with the theoretical bounds we obtained in the previous section. Furthermore, we see that the curves for R-GP-UCB have an oscillatory behavior, resulting from the fact that one tends to incur more regret just after a reset is done. In contrast, the curves for TV-GP-UCB are more steady, since the algorithm forgets in a "smooth" fashion.

**Mismatch and Robustness:** We consider the stability of TV-GP-UCB when there is mismatch between the true $\epsilon$ and the one used in TV-GP-UCB. We focus on the squared exponential kernel, and we set $\epsilon = 0.01$ and $T = 200$. From Figure 2c, we see that the performance of TV-GP-UCB is robust with respect to the mis-specification of $\epsilon$. In particular, the increase in regret as $\epsilon$ is increasingly over-estimated is slow. In contrast, while slightly under-estimating $\epsilon$ is not harmful, the regret increases rapidly beyond a certain point. In particular, using 0 instead of the true $\epsilon$ corresponds to simply running the standard GP-UCB algorithm, and gives the worst performance within the range shown. Note that the shaded area corresponds to a standard deviation from the mean.

Next, we study R-GP-UCB on the same model to determine the robustness with respect to the choice of $N$; the results are shown in Figure 2d. Values of $N$ that are too small are problematic, since the algorithm resets too frequently. While the *mean* of the regret is robust with respect to increasing $N$, we observe that the corresponding standard deviation also steadily increases. GP-UCB is again recovered as a special case, corresponding to $N = T$.

### 5.2 Real Data

We use temperature data collected from 46 sensors deployed at Intel Research, Berkeley. The dataset contains 5 days of measurements collected at 10-minute intervals. The goal of the spatiotemporal monitoring problem (see [7] for details) is to activate a sensor at every time step that reports a high temperature. Hence, $f_t$ consists of the set of all sensor temperature reportings at time $t$. A single sensor is activated every 10 minutes, and the regret is measured as the temperature difference between reporting of the activated sensor and the one that reports the maximum temperature at that particular time. Figure 3b plots each of the 46 functions with respect to time.

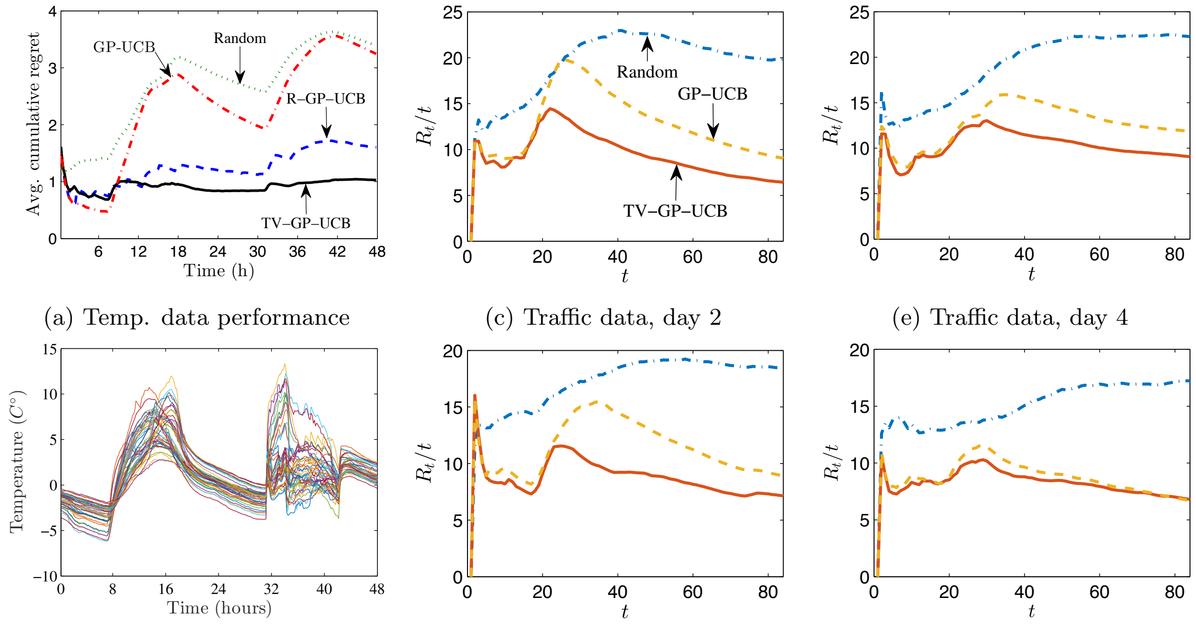
As a base comparison, we consider an algorithm that simply picks the sensors uniformly at random. We also consider the standard GP-UCB algorithm [1], even though it is unsuitable here since the reward function is varying with time.[3] Although it is not shown, we note that the RExp3 algorithm [18] (Exp3 with resetting) performed comparably to GP-UCB for this data set, suffering from the fact that it does not exploit correlations between the sensors.

We use the first three days of measurements for learning our algorithms' parameters. First, we compute the empirical covariance matrix from these days and use it as the kernel matrix in all of the algorithms. Next, using the same three training days, we obtain $\epsilon = 0.03$ by maximizing the marginal likelihood [19], and we obtain $N = 15$ by cross-validation. The algorithms are run on the final two days of the data.The results ($c_1 = 0.8, c_2 = 0.4, \sigma^2 = 0.5$ or 5% of the signal variance) are shown in Figure 3a. We observe that GP-UCB performs well for a short time, but then starts to suffer from the stale data, eventually becoming barely better than a random guess. Once again, TV-GP-UCB improves over R-GP-UCB, with the gap generally increasing over the duration of the experiment.

Next, we use traffic speed data from 357 sensors deployed along the highway I-880 South (California). The dataset contains one month of measurements, where 84 measurements were made on every day in between 6 AM and 11 AM. Our goal is to identify the least congested part of the highway by tracking the point of maximum speed. We use two thirds of the dataset to compute the empirical covariance matrix (and set it as the kernel matrix), and to learn $\epsilon$ by maximizing the marginal likelihood for all the training days [19], treating each day as being statistically independent. The last 10 days were used for testing. Due to the small time horizon $T = 84$ in comparison to the number of sensors, we restrict the domain to contain 50 sensors, chosen randomly from the 357. Our results ($\epsilon = 0.04$, $\sigma^2 = 5.0$ or 5% of the signal variance, $T = 84$, $c_1 = 0.2$, $c_2 = 0.4$) were averaged over 20 different initially activated sensors.

In Figure 3, in final two columns, we show the outcome of the experiment for 4 testing days (for the results on the rest of the days see Appendix G). TV-GP-UCB outperforms GP-UCB on most testing days, with the

[3] In [1], the same data was used to test GP-UCB in a different way; in each experiment, the function $f(x)$ was taken to be the set of temperatures at a single time.



(a) Temp. data performance

(c) Traffic data, day 2

(e) Traffic data, day 4

(b) Temperature data

(d) Traffic data, day 8

(f) Traffic data, day 10

Figure 3: Numerical performance of upper confidence bound algorithms on real data.

two being comparable for a few of the days (e.g., see Figure 3f). The latter situation arises when the indices of the best sensors do not change drastically over the time horizon, which is not always the case. In general, both algorithms suffer a large regret when sensors that were reporting high speeds suddenly change and start to report small speeds. However, TV-GP-UCB recovers more quickly from this compared to GP-UCB, due to its forgetting mechanism.

Note that we have omitted R-GP-UCB from this experiment, since we found it to be unsuitable due to the small time horizon. Moreover, this is the same reason that GP-UCB performs reasonably, unlike the temperature sensor example. Essentially, GP-UCB suffers more with a longer time horizon due to the larger amount of stale data.

## 6 Conclusion

We have studied the bandit optimization problem with time-varying rewards, taking a new approach based on a GP that evolves according to a simple Markov model. We introduced the R-GP-UCB and TV-GP-UCB algorithms, which, in contrast to previous algorithms, simultaneously trade off forgetting and remembering while also exploiting both spatial and temporal correlations. Our regret bounds for these algorithms provide, to our knowledge, the first explicit characterizations of the trade-off between the time horizon $T$ and rate at which the function varies $\epsilon$ in a bandit setting. We also provided an algorithm-independent bound revealing that a linear dependence on $T$ for fixed $\epsilon$ is unavoidable. Despite the simplicity of our theoretical model, we saw that the algorithms performed well on real world data sets that need not be matched to this model.

An immediate direction for future research is to determine to what extent the dependence on $\epsilon$ can be improved in our upper and lower bounds. Moreover, one could move to the non-Bayesian setting and consider classes of time-varying functions whose smoothness is dictated by an RKHS norm; see [1] for the time-invariant counterpart. Furthermore, while our time-varying model is primarily suited to handling steady changes, it could potentially be made even more effective by explicitly handling *sudden* changes, e.g., by a combination of our techniques with those from previous works studying changepoint detection [23, 24].

## Acknowledgment

We thank Andreas Krause for sharing his datasets with us for use in Section 5.

This work was supported in part by the European Commission under Grant ERC Future Proof, SNF 200021-146750 and SNF CRSII2-147633, and 'EPFL Fellows' Horizon2020 grant 665667.

## References

[1] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, "Information-theoretic regret bounds for Gaussian process optimization in the bandit setting," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 3250–3265, May 2012.

[2] S. Bubeck and N. Cesa-Bianchi, *Regret Analysis of Stochastic and Nonstochastic Multi-Armed Bandit Problems*, ser. Found. Trend. Mach. Learn. Now Publishers, 2012.

[3] T. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Adv. App. Math.*, vol. 6, no. 1, pp. 4 – 22, 1985.

[4] E. Brochu, V. M. Cora, and N. de Freitas, "A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," 2010, http://arxiv.org/abs/1012.2599.

[5] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Adv. Neur. Inf. Proc. Sys.*, 2012.

[6] C. E. Rasmussen, "Gaussian processes for machine learning." MIT Press, 2006.

[7] A. Krause and C. S. Ong, "Contextual Gaussian process bandit optimization," in *Adv. Neur. Inf. Proc. Sys.* Curran Associates, Inc., 2011, pp. 2447–2455.

[8] J. Djolonga, A. Krause, and V. Cevher, "High-dimensional Gaussian process bandits," in *Adv. Neur. Inf. Proc. Sys.*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 1025–1033.

[9] J. Snoek, K. Swersky, R. Zemel, and R. P. Adams, "Input warping for Bayesian optimization of non-stationary functions," in *Int. Conf. Mach. Learn.*, 2014.

[10] Z. Wang, M. Zoghi, F. Hutter, D. Matheson, and N. de Freitas, "Bayesian optimization in high dimensions via random embeddings," in *Int. Joint. Conf. Art. Int.*, 2013.

[11] S. Bubeck, O. Dekel, T. Koren, and Y. Peres, "Bandit convex optimization: $\sqrt{T}$ regret in one dimension," 2015, http://arxiv.org/abs/1502.06398.

[12] P. Whittle, "Restless bandits: Activity allocation in a changing world," *J. App. Prob.*, vol. 25, pp. 287–298, 1988.

[13] D. Bertsimas and J. Nio-Mora, "Restless bandits, linear programming relaxations, and a primal-dual index heuristic," *Operations Research*, vol. 48, no. 1, pp. 80–90, 2000.

[14] R. Ortner, D. Ryabko, P. Auer, and R. Munos, "Regret bounds for restless Markov bandits," in *Algorithmic Learning Theory.* Springer Berlin Heidelberg, 2012, pp. 214–228.

[15] A. Slivkins and E. Upfal, "Adapting to a changing environment: the Brownian restless bandits," in *Conf. Learn. Theory*, 2008.

[16] C. Tekin and M. Liu, "Online learning of rested and restless bandits," *IEEE Trans. Inf. Theory*, vol. 58, no. 8, pp. 5588–5611, Aug. 2012.

[17] H. Liu, K. Liu, and Q. Zhao, "Learning in a changing world: Restless multiarmed bandit with unknown dynamics," *IEEE Trans. Inf. Theory*, vol. 59, no. 3, pp. 1902–1916, March 2013.

[18] O. Besbes, Y. Gur, and A. Zeevi, "Stochastic multi-armed-bandit problem with non-stationary rewards," in *Adv. Neur. Inf. Proc. Sys.*, 2014, pp. 199–207.

[19] S. Van Vaerenbergh, M. Lázaro-Gredilla, and I. Santamaría, "Kernel recursive least-squares tracker for time-varying regression," *IEEE Trans. Neur. Net. Learn. Sys.*, vol. 23, no. 8, pp. 1313–1326, 2012.

[20] S. Van Vaerenbergh, I. Santamaría, and M. Lázaro-Gredilla, "Estimation of the forgetting factor in kernel recursive least squares," in *IEEE. Int. Workshop Mach. Learn. SIg. Proc.*, 2012, pp. 1–6.

[21] M. A. Osborne, S. Roberts, A. Rogers, S. Ramchurn, and N. R. Jennings, "Towards real-time information processing of sensor network data using computationally efficient multi-output gaussian processes," in *Proc. Int. Conf. Inf. Proc. Sens. Net.*, 2008, pp. 109–120.

[22] K. Kandasamy, J. G. Schneider, and B. Póczos, "High dimensional Bayesian optimisation and Bandits via additive models," 2015, http://arxiv.org/abs/1503.01673.

[23] R. P. Adams and D. J. MacKay, "Bayesian online changepoint detection," 2007, http://arxiv.org/abs/0710.3742.

[24] R. Garnett, M. A. Osborne, and S. J. Roberts, "Sequential bayesian prediction in the presence of changepoints," in *Proc. Inf. Conf. Mach. Learn.*, 2009.

[25] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, Inc., 2001.

[26] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. New York, NY, USA: Cambridge University Press, 2012.

[27] R. Bhatia, *Matrix Analysis*. Springer, 1997.