Institute for
Data Science in
Mechanical Engineering

RWTH AACHEN UNIVERSITY

The present work was submitted to the Institute for Data Science in Mechanical Engineering.
Diese Arbeit wurde vorgelegt am Institut für Data Science im Maschinenbau.

Learning and Forgetting in Time-Varying Bayesian Optimization

Lernen und Vergessen in Zeitabhängiger Bayes'scher Optimierung

Master Thesis
Masterarbeit

Presented by / Vorgelegt von

Paul Brunzema
Matr.Nr.: 356997

Supervised by / Betreut von	Alexander von Rohr, M.Sc. (Institute for Data Science in Mechanical Engineering)
1st Examiner / 1. Prüfer	Univ.-Prof. Dr. sc. Sebastian Trimpe (Institute for Data Science in Mechanical Engineering)
2nd Examiner / 2. Prüfer	Dr.-Ing. Lorenz Dörschel (Institute of Automatic Control)

Aachen, 6. Dezember 2021

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne die Benutzung anderer als der angegebenen Hilfsmittel selbstständig verfasst habe; die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe des Literaturzitats gekennzeichnet.

Aachen, den 6. Dezember 2021

Abstract

The goal of time-varying Bayesian optimization (TVBO) is to find and track an optimum of a dynamic optimization problem with an unknown objective function. At each time step, a decision-maker needs to decide where to evaluate the objective function next, either exploiting known good queries or exploring uncertain outcomes and capturing the temporal change. However, to learn the temporal changes, we need to place regularity assumptions on them and, ideally, on the properties of the objective functions. Specifically, we are interested in controller tuning problems with convex objective functions. Without any assumptions tracking the optimum becomes infeasible. Instead of using heuristics for the decision-maker, we embed the assumptions directly into a spatio-temporal Gaussian process (GP) model.

We propose modeling the temporal dimension as a Wiener process, retaining past information in the form of its expected value. A Wiener process better captures the expected changes in the objective function of a changing dynamical system and is robust to misspecifications of the prior distribution. To exploit the prior knowledge of a convex objective function, we impose inequality constraints on the second derivative of the GP model. This allows the decision-maker to extrapolate globally using local information and thus avoid undesirable global exploration without additional heuristics.

We demonstrate in extensive synthetic experiments and a controller tuning problem the advantages of including these assumptions into TVBO, as they result in significant performance and robustness benefits compared to the state-of-the-art TVBO approach.

Kurzzusammenfassung

Das Ziel von zeitabhängiger Bayes'scher Optimierung (TVBO) ist es, ein Optimum in einem dynamischen Optimierungsproblem mit einer unbekannten Zielfunktion zu finden und zu verfolgen. In jedem Zeitschritt muss ein Algorithmus entscheiden, wo die Zielfunktion als nächstes auszuwerten ist, indem er entweder Orte wählt, an denen die Funktionsauswertung bekanntlich nah am Optimum liegt, oder unsichere Orte untersucht, um so die zeitliche Veränderung der Zielfunktion zu erfassen. Um zeitliche Veränderungen zu lernen, müssen wir jedoch einige Regularitätsannahmen für diese Veränderungen und idealerweise für die Eigenschaften der Zielfunktionen treffen. Wir sind insbesondere an Problemen der Reglereinstellung mit konvexen Zielfunktionen interessiert. Ohne jegliche Annahmen ist die Verfolgung des Optimums nicht möglich. Statt Heuristiken für den Algorithmus zu verwenden, betten wir die Annahmen in ein räumlich-zeitliches GP-Modell ein.

Wir schlagen vor, die zeitliche Dimension als Wiener-Prozess zu modellieren, der vergangene Informationen in Form ihres Erwartungswerts erhält. Ein Wiener-Prozess erfasst die erwarteten Änderungen in der Zielfunktion eines sich verändernden dynamischen Systems besser und ist robuster gegenüber fehlerhaften A-priori-Verteilungen im Vergleich zum Stand der Technik in TVBO. Um das Wissen über die Konvexität der Zielfunktion zu nutzen, legen wir Ungleichheitsnebenbedingungen für die zweite Ableitung des GP-Modells fest. Dies ermöglicht dem Algorithmus, unter Verwendung lokaler Informationen global zu extrapolieren und hiermit unerwünschte globale Exploration zu vermeiden ohne zusätzliche Heuristiken verwenden zu müssen.

Wir demonstrieren in umfangreichen synthetischen Experimenten und einem Anwendungsbeipiel, dem Einstellen eines Reglers, die Vorteile des Einbeziehen dieser Annahmen für TVBO, aus denen erhebliche Leistungs- und Robustheitsvorteilen im Vergleich zum Stand der Technik resultieren.

Contents

1. Introduction

Bayesian optimization (BO) is a black-box optimization technique used to find an optimum of an unknown objective function utilizing only noisy function evaluations by sequentially querying based on a selection criterion. This makes BO a powerful optimization tool in settings where only the performance depending on the decision variables can be measured. Among other things, BO has been applied to tune an optimal controller in Marco *et al.* [26] by minimizing a linear-quadratic regulator (LQR) cost function using only limited prior knowledge about the dynamics of the system. As in this example, in literature on BO for controller tuning, mainly time-invariant systems have been considered. However, the system dynamics may vary over time for physical systems due to wear, e.g., considering spring and damping constants, or sudden changes, e.g., through an additional mass. An once found optimal controller could therefore become sub-optimal over time. The challenge then arises of not only finding the optimum but also tracking it over time. Solving such a dynamic optimization problem without prior knowledge of the system dynamics combined with concepts from event-triggered learning [43] could result in an autonomous adaptation of physical systems to changing environments and system properties.

Suppose standard BO algorithms are used to solve dynamic optimization problems. In that case, the results may be undesirable, and performance may degrade over time as they treat the information from each iteration as equally informative. However, recently obtained information should be valued higher than information from earlier iterations in a time-varying setting. Hence, the loss of information over time must be explicitly taken into account by the optimization algorithm. For BO, implementing such a notion of forgetting is called time-varying Bayesian optimization (TVBO).

As in standard BO, TVBO uses a Gaussian process (GP) to model the objective function. Defining the GP prior distribution, therefore, implies the prior belief over possible objective functions with respect to smoothness, differentiability, and continuity. Furthermore, in TVBO the forgetting strategy implies the prior belief about the rate of change as well as the type of change of the objective function. In previous studies regarding time-varying regression two different forgetting strategies have been proposed – Back-To-Prior (B2P) forgetting and Uncertainty-Injection (UI) forgetting [47].

B2P forgetting represents the idea that the expectation of the objective function propagates back to the prior distribution over time after observing no more data. Intuitively, as an algorithm gets more uncertain about a measurement, it defaults back to the state of the model before seeing any data. All previous empirical work in GP-based TVBO implicitly defined B2P forgetting as their forgetting strategy. In contrast, UI forgetting expresses model uncertainty over time based on a different assumption. Instead of assuming gradual change back to the prior distribution, UI forgetting assumes gradual change around the measurement taken by maintaining its expected value.

This thesis suggests that this form of modeling temporal change better captures the expected changes in the objective function arising in tasks such as tuning a controller in a time-varying context. Therefore, a novel method, Uncertainty-Injection-in-TVBO (UI-TVBO), is presented using UI forgetting in the context of GP-based TVBO based on modeling the temporal change as a Wiener process. It preserves past information and reduces dependency on the prior distribution compared to the state-of-the-art approach.

Besides embedding assumptions about the temporal change of the objective function, this thesis investigates if embedding further assumptions on its shape into the GP model can improve the performance of TVBO. Therefore, this thesis considers the shape of the objective function to remain convex through time as many real-world problems result in convex objective functions (for example, the mentioned LQR problem in control theory). Furthermore, solving convex function is often easier compared to solving non-convex functions [11] indicating that considering only convex objective functions in TVBO may simplify solving the dynamic optimization problem as, due to the convexity of the function, global exploration for the optimizer is not necessary.

To embed this into TVBO, the novel method Constrained-TVBO (C-TVBO) is proposed imposing convexity constraints on the GP posterior distribution at each time step, thus yielding in a hypothesis space with only convex function. These constraints enable useful extrapolation of local information for the global model. This reduces global exploration which is beneficial for many practical applications as changes in the decision variables should be limited, e.g., if the decision variables represent a parameterized controller. The method can be applied to any modeling approach in TVBO independent of the forgetting strategy. However, this thesis shows that especially by combining the proposed methods C-TVBO and UI-TVBO, the performance and robustness are improved compared to the state-of-the-art approach as they retained more structural information about objective function over time.

1.1. Problem Formulation

The goal throughout this thesis is to find sequential optimal solutions $\mathbf{x}_t^*$ of an unknown time-varying objective function $f\colon \mathcal{X} \times \mathcal{T} \mapsto \mathrm{I\!R}$ with $f_t(\mathbf{x}) \coloneqq f(\mathbf{x}, t)$ as

$$\mathbf{x}_t^* = \arg\min_{\mathbf{x}\in\mathcal{X}} f_t(\mathbf{x}) \tag{1.1}$$

at the discrete time step $t \in \mathcal{T} = \{1, 2, \dots, T\}$ with time horizon T within a feasible set $\mathbf{x} \in \mathcal{X} \subset \mathrm{I\!R}^D$. At each time step an algorithm can query the objective function once at a location $\mathbf{x}_t$ and obtains a noisy observation in the form of

$$y_t = f_t(\mathbf{x}_t) + w \tag{1.2}$$

with zero mean Gaussian noise $w \sim \mathcal{N}\left(0, \sigma_n^2\right)$ which is independent between time steps. The performance of an algorithm in this time-varying setting will be measured in terms of the dynamic cumulative regret measuring the difference between optimal and chosen value. The dynamic cumulative regret will be formally introduced in Section 2.2. The goal on an algorithm is to minimize the dynamic cumulative regret by balancing exploration for capturing the change in f_t, and exploitation to minimize regret.

The regularity assumption on f_t in this thesis is that it is a sample from a GP prior with the kernel defining its smoothness in the spatial dimensions as well as the temporal correlation between consecutive time steps. Furthermore, prior knowledge about the shape of the objective function f_t is that it remains convex through time as formalized in Assumption 1.

Assumption 1. *f_t in (1.1) is at least twice differentiable with respect to $\mathbf{x}$ and the Hessian $\nabla^2_{\mathbf{x}} f_t$ is semi-positive definite $\forall t \in \mathcal{T}$ and $\forall \mathbf{x} \in \mathcal{X}$.*

1.2. Key Contributions

As an overview, the key contributions of this thesis are:

- **UI-TVBO – UI Forgetting in TVBO:** The novel modeling approach UI-TVBO using UI forgetting is proposed retaining relevant information from the past, in contrast to B2P forgetting. It is based on modeling the temporal dimension as a Wiener process and shows a more robust performance than the current state-of-the-art method using B2P forgetting.
- **C-TVBO – Shape Constraints in TVBO:** For objective functions remaining convex over time, this thesis introduces the novel method C-TVBO. It embeds the prior knowledge about convexity through shape constraints on the posterior distribution into TVBO. This reduces the dynamic cumulative regret and its variance, especially in combination with UI-TVBO, as it allows global extrapolation resulting in more informative local exploration and allowing for better exploitation.
- **Extensive Empirical Evaluation:** The resulting methods are evaluated in terms of their performance regarding dynamic cumulative regret with three different types of experiments:
  1. Synthetic experiments created according to the model assumptions with predefined hyperparameters.
  2. Synthetic experiments of a moving parabola inspired by benchmarks introduced by Renganathan *et al.* [37].
  3. An application example in the form of a LQR problem of an inverted pendulum with changing system dynamics.

Structure of the Thesis

The thesis is structured as follows. In Chapter 2, the necessary fundamentals for this thesis are presented. Section 2.1 presents the basics of GPs as well as a method to enforce shape constraints. Afterwards, Section 2.2 introduces BO as well as its extension to time-varying functions in TVBO. After an overview of related work and differentiating the proposed methods from it in Chapter 3, the proposed methods of this thesis are derived and presented in Chapter 4 – the modeling approach UI-TVBO in Section 4.1 and in Section 4.2 the algorithm C-TVBO. Subsequently, practical extensions are presented in Section 4.3 In Chapter 5, the methods are compared with the state-of-the-art approach on various experiments. Finally, concluding remarks and a discussion about interesting future work are presented in Chapter 6.

2. Background

TVBO relies on fundamental knowledge about GPs and BO of which the relevant aspects are discussed in this chapter. Furthermore, to embed and formalize prior knowledge about the cost function staying convex through time, as stated in Assumption 1, shape constraints on GPs will be used. This chapter will introduce a concept of enforcing shape constraints at finite points resulting in a high probability of the GP being convex within the feasible set $\mathcal{X}$.

2.1. Gaussian Process Regression

GPs are widely used for regression and build the foundation of many BO algorithms by modeling its objective function. To introduce GP regression, this section will follow Rasmussen and Williams [36] to which is also referred to for further details.

GPs are a nonparametric Bayesian approach to regression which explicitly incorporate uncertainty. The goal of GP regression is to model the function $f\colon \mathcal{X} \mapsto \mathbb{R}$ with $\mathcal{X} \subset \mathbb{R}^D$ which is corrupted by zero mean Gaussian noise $w \sim \mathcal{N}\left(0, \sigma_n^2\right)$. An observation y from $f(\mathbf{x})$ can therefore be expressed as

$$y = f(\mathbf{x}) + w. \tag{2.1}$$

Taking N observation of the objective function (2.1) at the training points $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in R^{D \times N}$ results in a data set $\mathcal{D} \coloneqq \{(\mathbf{x}_i, y_i) | i = 1, \ldots, N\} = (\mathbf{X}, \mathbf{y})$ with $\mathbf{y} = [y_1, \ldots, y_N] \in \mathbb{R}^N$ as the training targets in vector notation.

A GP can be interpreted as modeling $f(\mathbf{x})$ as a distribution over functions and is fully defined as $f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right)$ with the mean function m and kernel k

as

$$m\colon \mathcal{X} \mapsto \mathrm{I\!R}, \quad m(\mathbf{x}) = \mathbb{E}\left[f(\mathbf{x})\right] \tag{2.2}$$

$$k\colon \mathcal{X} \times \mathcal{X} \mapsto \mathrm{I\!R}, \quad k(\mathbf{x}, \mathbf{x}') = \mathbb{E}\left[\left(f(\mathbf{x}) - m(\mathbf{x})\right)\left(f(\mathbf{x}') - m(\mathbf{x}')\right)\right]. \tag{2.3}$$

The mean function m is often set to be constant as $m(\mathbf{x}) = \mu_0$ with μ_0 either as zero or as the mean of the training targets $\mathbf{y}$. However, more complex mean functions are also possible.

With the information contained in the data set $\mathcal{D}$, predictions $\mathbf{f}_*$ at N_* test locations $\mathbf{X}_* = [\mathbf{x}_1^*, \ldots, \mathbf{x}_{N_*}^*] \in \mathrm{I\!R}^{D \times N_*}$ can be made by setting up the multivariate Gaussian joint distribution over training targets and predictions as

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(\mathbf{X}) \\ m(\mathbf{X}_*) \end{bmatrix}, \begin{bmatrix} K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I} & K(\mathbf{X}, \mathbf{X}_*) \\ K(\mathbf{X}_*, \mathbf{X}) & K(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix}\right). \tag{2.4}$$

Here, the notation is used that $K(\mathbf{X}, \mathbf{X}')$ denotes a matrix with entries $K(\mathbf{X}, \mathbf{X}')_{i,j} = k(\mathbf{X}_i, \mathbf{X}_j')$. The marginal distribution $\mathbf{f}_* \sim \mathcal{N}\left(m(\mathbf{X}_*), K(\mathbf{X}_*, \mathbf{X}_*)\right)$ is the prior prediction over the test locations. Conditioning the joint distribution (2.4) on the data set $\mathcal{D}$ yields the conditioned posterior distribution

$$\mathbf{f}_* | \mathcal{D} \sim \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*) \tag{2.5}$$

which again is a multivariate Gaussian distribution with mean $\boldsymbol{\mu}_*$ and covariance matrix $\boldsymbol{\Sigma}_*$ as

$$\boldsymbol{\mu}_* = m(\mathbf{X}_*) + K(\mathbf{X}_*, \mathbf{X})\left(K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}\right)^{-1} \left(\mathbf{y} - m(\mathbf{X})\right), \tag{2.6}$$

$$\boldsymbol{\Sigma}_* = \underbrace{K(\mathbf{X}_*, \mathbf{X}_*)}_{\text{prior knowledge}} - \underbrace{K(\mathbf{X}_*, \mathbf{X})\left(K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}\right)^{-1} K(\mathbf{X}, \mathbf{X}_*)}_{\text{obtained knowledge}}. \tag{2.7}$$

The calculation of the mean and covariance of the posterior distribution can each be divided into a prior knowledge part defined though the GP prior and an update part to this prior obtained through the information in data set $\mathcal{D}$.

Kernels

Selecting a suitable kernel for the regression task is crucial. A commonly chosen kernel is the squared-exponential (SE) kernel as

$$k(\mathbf{x}, \mathbf{x}') = \sigma_k^2 \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \mathbf{\Lambda}^{-1}(\mathbf{x} - \mathbf{x}')\right) \tag{2.8}$$

with the length scales $\mathbf{\Lambda} = \mathrm{diag}(\sigma_{l,1}^2, \ldots, \sigma_{l,D}^2)$ and the output scale σ_k^2 as hyperparameters. It defines a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_k(\mathcal{X})$ with the property that any function $f \in \mathcal{H}_k(\mathcal{X})$ is within the set of infinitely differentiable functions $C^\infty(\mathcal{X})$. Therefore, the mean estimate (2.6) of the GP posterior will also be in $C^\infty(\mathcal{X})$. Furthermore, the SE kernel is a *universal* kernel meaning it can approximate any continuous function. However, since the hypothesis space of the GP model with a SE kernel is restricted to functions in $C^\infty(\mathcal{X})$, choosing the SE kernel can be problematic in some practical applications as this smoothness assumption induces a strong bias and the universal property is only given in the presence of infinite data.

Kernels can be tailed to specific problems as long as they satisfy the conditions of being symmetric and resulting in a positive definite Gram matrix K (for more information see Rasmussen and Williams [36, Chap. 4.1]). For example, in Marco *et al.* [27] a kernel was specifically designed to imply a distribution over LQR cost functions. Valid kernels can also be recombined resulting, e.g., in a product composite kernel as

$$k_{comp}(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') \otimes k_2(\mathbf{x}, \mathbf{x}'). \tag{2.9}$$

This can be beneficial for capturing different characteristic in the data set $\mathcal{D}$ [15]. Furthermore, if valid kernels $k_1(\mathbf{x}_1, \mathbf{x}_1')$ and $k_2(\mathbf{x}_2, \mathbf{x}_2')$ mapping to $\mathbb{R}$ are defined over different input spaces $\mathcal{X}_1$ and $\mathcal{X}_2$, multiplying k_1 and k_2 also results in a valid product composite kernel as

$$k_{comp} \colon \mathcal{X}_1 \times \mathcal{X}_2 \mapsto \mathbb{R}, \quad k_{comp}(\{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_1', \mathbf{x}_2'\}) = k_1(\mathbf{x}_1, \mathbf{x}_1') \otimes k_2(\mathbf{x}_2, \mathbf{x}_2'). \tag{2.10}$$

This property enables the use of different kernels for individual dimensions of the input $\mathbf{x}$ which might have different context [24].

Linear Operators on Gaussian Processes

GPs are closed under linear operators $\mathcal{L}$ such as differentiation [36]. Considering a GP as $f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right)$, this means applying the linear operator to $f(\mathbf{x})$ results again in a GP as

$$\mathcal{L}f(\mathbf{x}) \sim \mathcal{GP}\left(\mathcal{L}m(\mathbf{x}), \mathcal{L}k(\mathbf{x}, \mathbf{x}')\mathcal{L}^T\right) \tag{2.11}$$

using the notation by Agrell [1]. Here, $\mathcal{L}k(\mathbf{x}, \mathbf{x}')$ and $k(\mathbf{x}, \mathbf{x}')\mathcal{L}^T$ indicate the operator $\mathcal{L}$ acting on $\mathbf{x}$ and $\mathbf{x}'$, respectively. The property of staying closed under linear operators is used by Geist and Trimpe [17] and Jidling *et al.* [22] to embed prior knowledge in the form of physical insights into the GP regression task as equality constaints on the GP.

2.1.1. Linear Inequality Constraints

Assumption 1 in the problem formulation states that the objective function remains convex thought time. This means that the Hessian $\nabla_{\mathbf{x}} f_t$ remains positive definite throughout the feasible set at each time step. Therefore, this thesis uses a linear operator $\mathcal{L} = \frac{\partial^2}{\partial x_i^2}$ for every spatial dimension $i \in [1, \dots, D]$ as introduced above and apply inequality constraints on the posterior distribution $\mathbf{f}_*$ in (2.5). In combination with a smooth kernel such as the SE kernel, the positive definiteness of the Hessian of f_t can be approximated. Applying such linear inequality constraints in GP regression has been discussed among others by Agrell [1] and Wang and Berger [50] which build the theoretical background for this section.

A GP under linear inequality constraints requires the GP posterior conditioned on the data set $\mathcal{D}$ to satisfy

$$a(\mathbf{x}) \leq \mathcal{L}f(\mathbf{x}) \leq b(\mathbf{x}) \tag{2.12}$$

for two bounding functions $a(\mathbf{x}), b(\mathbf{x}) \colon \mathbb{R}^D \mapsto (\mathbb{R} \cup \{-\infty, \infty\})$ with $a(\mathbf{x}) < b(\mathbf{x})$, $\forall \mathbf{x} \in \mathcal{X}$. Agrell [1] and Wang and Berger [50] introduce a method to achieve this approximately by considering only a finite set of N_v inputs $\mathbf{X}_v = [\mathbf{x}_1^v, \dots, \mathbf{x}_{N_v}^v] \in \mathbb{R}^{D \times N_v}$, called the virtual observation points (VOPs), at which (2.12) has to hold. Furthermore, in Agrell [1] the assumption is made that the virtual observations $\mathcal{L}f(\mathbf{x}_i^v)$ are corrupted by Gaussian noise $w_{v,i} \sim \mathcal{N}(0, \sigma_v^2)$. The reason for this is numerical stability in calculating the constrained posterior distribution. This yields in a relaxed formulation of (2.12) as

$$a(\mathbf{X}_v) \leq \mathcal{L}f(\mathbf{X}_v) + w_v \leq b(\mathbf{X}_v), \quad w_v \sim \mathcal{N}(\mathbf{0}, \sigma_v^2 \mathbf{I}), \tag{2.13}$$

where the constraints $a(\mathbf{x}), b(\mathbf{x})$ no longer have to hold for all $\mathbf{x} \in \mathcal{X}$. The VOPs do not have to be within the feasible set $\mathcal{X}$ but $\mathbb{R}^D$.

To simplify notation, the corrupted virtual observations at the VOPs will be denoted as $\tilde{C}(\mathbf{X}_v) \coloneqq \mathcal{L}f(\mathbf{X}_v) + w_v$, the Gram matrix $K(\mathbf{X}, \mathbf{X}')$ as $K_{\mathbf{X},\mathbf{X}'}$, and the mean function $m(\mathbf{x})$ as $\mu_{\mathbf{X}}$. Furthermore, $C(\mathbf{X}_v)$ will denote the event of $\tilde{C}$ satisfying (2.13) for all $\mathbf{x}_i^v \in \mathbf{X}_v$.

Since applying a linear operator $\mathcal{L}$ to $f(\mathbf{x})$ results again in a GP as described in Section 2.1, the joint distribution of the predictions $\mathbf{f}_*$, observations $\mathbf{y}$ and virtual observations $\tilde{C}$ can be set up as

$$\begin{bmatrix} \mathbf{f}_* \\ \mathbf{y} \\ \tilde{C} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_{\mathbf{X}_*} \\ \mu_{\mathbf{X}} \\ \mathcal{L}\mu_{\mathbf{X}_v} \end{bmatrix}, \begin{bmatrix} K_{\mathbf{X}_*,\mathbf{X}_*} & K_{\mathbf{X}_*,\mathbf{X}} & K_{\mathbf{X}_*,\mathbf{X}_v}\mathcal{L}^T \\ K_{\mathbf{X},\mathbf{X}_*} & K_{\mathbf{X},\mathbf{X}} + \sigma_n^2\mathbf{I} & K_{\mathbf{X},\mathbf{X}_v}\mathcal{L}^T \\ \mathcal{L}K_{\mathbf{X}_v,\mathbf{X}_*} & \mathcal{L}K_{\mathbf{X}_v,\mathbf{X}} & \mathcal{L}K_{\mathbf{X}_v,\mathbf{X}_v}\mathcal{L}^T + \sigma_v^2\mathbf{I} \end{bmatrix} \right). \tag{2.14}$$

Conditioning the joint distribution on the data set $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ results in

$$\begin{bmatrix} \mathbf{f}_* \\ \tilde{C} \end{bmatrix} \Bigg| \mathcal{D} \sim \mathcal{N}\left(\begin{bmatrix} \mu_{\mathbf{X}_*} + A_2\left(\mathbf{y} - \mu_{\mathbf{X}}\right) \\ \mathcal{L}\mu_{\mathbf{X}_v} + A_1\left(\mathbf{y} - \mu_{\mathbf{X}}\right) \end{bmatrix}, \begin{bmatrix} B_2 & B_3 \\ B_3^T & B_1 \end{bmatrix} \right) \tag{2.15}$$

with

$$A_1 = \left(\mathcal{L}K_{\mathbf{X}_v,\mathbf{X}}\right)\left(K_{\mathbf{X},\mathbf{X}} + \sigma_n^2\mathbf{I}\right)^{-1} \tag{2.16}$$

$$A_2 = K_{\mathbf{X}_*,\mathbf{X}}\left(K_{\mathbf{X},\mathbf{X}} + \sigma_n^2\mathbf{I}\right)^{-1} \tag{2.17}$$

$$B_1 = \mathcal{L}K_{\mathbf{X}_v,\mathbf{X}_v}\mathcal{L}^T + \sigma_v^2\mathbf{I} - A_1 K_{\mathbf{X},\mathbf{X}_v}\mathcal{L}^T \tag{2.18}$$

$$B_2 = K_{\mathbf{X}_*,\mathbf{X}_*} - A_2 K_{\mathbf{X},\mathbf{X}_*} \tag{2.19}$$

$$B_3 = K_{\mathbf{X}_*,\mathbf{X}_v}\mathcal{L}^T - A_2 K_{\mathbf{X},\mathbf{X}_v}\mathcal{L}^T. \tag{2.20}$$

Further conditioning on $\tilde{C}$ yields in the multivariate Gaussian distribution

$$\mathbf{f}_*|\mathcal{D},\tilde{C} \sim \mathcal{N}\left(\mu_{\mathbf{X}_*} + A(\tilde{C} - \mathcal{L}\mu_{\mathbf{X}_v}) + B(\mathbf{y} - \mu_{\mathbf{X}}), \Sigma\right) \tag{2.21}$$

with

$$A = B_3 B_1^{-1} \tag{2.22}$$

$$B = A_2 - AA_1 \tag{2.23}$$

$$\Sigma = B_2 - AB_3^T. \tag{2.24}$$

Looking at the calculation of A in (2.22), the role of the additional virtual observation noise $w_{v,i} \sim \mathcal{N}(0, \sigma_v^2)$ as a numerical regularization for calculating B_1^{-1} becomes clear. An interpretation of σ_v^2 is that the probability of satisfying the constraints at the virtual locations is slightly reduced. In practice, σ_v^2 is set to be very small ($\sigma_v^2 \approx 10^{-6}$).

Up to this point, the marginalization of the virtual observations from (2.15) remained Gaussian as $\tilde{C} \sim \mathcal{N}(\mu_c, \Sigma_c)$. By now conditioning on the event C we define $\mathbf{C} = \tilde{C}|\mathcal{D}, C$ resulting in a *truncated* multivariate normal distribution as

$$\mathbf{C} \sim \mathcal{TN}(\underbrace{\mathcal{L}\mu_{\mathbf{X}_v} + A_1(\mathbf{y} - \mu_{\mathbf{X}})}_{\mu_{\mathcal{TN}} \in \mathbb{R}^{P \times 1}}, \underbrace{B_1}_{\Sigma_{\mathcal{TN}} \in \mathbb{R}^{P \times P}}, a(\mathbf{X}_v), b(\mathbf{X}_v)) \tag{2.25}$$

with $\mathcal{TN}(\mu_{\mathcal{TN}}, \Sigma_{\mathcal{TN}}, a, b)$ as the Gaussian $\mathcal{N}(\mu_{\mathcal{TN}}, \Sigma_{\mathcal{TN}})$ conditioned on the hyperbox $[a_1, b_1] \times \cdots \times [a_{N_v}, b_{N_v}]$. Following Agrell [1, Lemma 1] the resulting constrained posterior distribution is a compound Gaussian with a truncated mean as

$$\mathbf{f}_*|\mathcal{D}, C \sim \mathcal{N}\left(\mu_{\mathbf{X}_*} + A(\mathbf{C} - \mathcal{L}\mu_{\mathbf{X}_v}) + B(\mathbf{y} - \mu_{\mathbf{X}}), \Sigma\right). \tag{2.26}$$

This posterior distribution is guaranteed to satisfy (2.12) at the VOPs for $\sigma_v^2 \to 0$. However, Wang and Berger [50] observed that using a sufficient amount of VOPs throughout $\mathcal{X}$ results in a high probability of $\mathbf{f}_*$ satisfying the constraints in (2.12) in $\mathcal{X}$. Lemma 2 of Agrell [1] provides a numerically more stable implementation of the factors (2.16) to (2.20) based on Cholesky factorization and is summarized in Appendix A.1.

2.1.2. Sampling from the Constrained Posterior Distribution

Even though the posterior distribution (2.26) is Gaussian, it can no longer be calculated in closed form as the mean is truncated. Therefore, the posterior has to be approximated using sampling. This can be achieved following Algorithm 1 proposed by Agrell [1, Algorithm 3].

Algorithm 1 Sampling form the constrained posterior distribution [1]

```
Initialize: Calculate factors $A$, $B$, $\Sigma$, $A_1$, $B_1$
Find a matrix $\mathbf{Q}$ s.t. $\mathbf{Q}^T\mathbf{Q} = \Sigma \in \mathbb{R}^{M \times M}$ using Cholesky decomposition.
Generate $\tilde{\mathbf{C}}_k$, a $P \times k$ matrix where each column is a sample of $\tilde{C}|\mathcal{D}, C$ from the
    truncated multivariate normal distribution (2.25).
Generate $\mathbf{U}_k$, a $M \times k$ matrix with k samples of the multivariate standard normal
    distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_M)$ with $\mathbf{I}_M \in \mathbb{R}^{M \times M}$.
Calculate the $M \times k$ matrix where each column is a sample from the distribution
    $\mathbf{f}_*|\mathcal{D}, C$ in (2.26) as
        $$[\mu_{\mathbf{X}_*} + B(\mathbf{y} - \mu_{\mathbf{X}})] \oplus \left[A(-\mathcal{L}\mu_{\mathbf{X}_v} \oplus \tilde{\mathbf{C}}_k) + \mathbf{Q}\mathbf{U}_k\right] \tag{2.27}$$
    with $\oplus$ representing the operation of adding the $M \times 1$ vector on the left hand
    side to each column of the $M \times k$ matrix on the right hand side.
```

The difficulty in sampling from the posterior lies in sampling from the truncated multivariate normal distribution. An approach to rejection sampling via a minimax tilting method was proposed by Botev [10] resulting in iid samples of (2.25). The algorithm has shown to perform well with minimal error up to a dimension of $P \approx 100$. However, rejection sampling suffers from the curse of dimensionality as the acceptance rate drops exponentially with growing dimensions. Therefore, one has to fall back to approximate sampling using Markov-Chain-Monte-Carlo (MCMC) methods. In the case of sampling from the truncated multivariate normal distribution, Gibbs sampling can be used, as calculating the necessary conditional distributions is possible. For the case of monotonicity constraints on the posterior distribution ($\mathcal{L} = \frac{\partial}{\partial x_i}$) a Gibbs sampling method has been proposed by Wang and Berger [50]. It has been adapted to work with any constraints $a(\cdot), b(\cdot)$ as well as any operator $\mathcal{L}$ as described in this section and is shown in Algorithm 2. The truncated normal distribution from which has to be sampled in line 5, Algorithm 2, is one dimensional. Therefore, the rejection sampling method by Botev [10] can again be used for this sub-task to efficiently generate the iid samples.

Algorithm 2 Gibbs sampling for the truncated multivariate normal distribution (adapted form Wang and Berger [50, Section 3.1.2])

```
Initialize: Calculate mean $\mu_{\mathcal{TN}}$ and covariance $\Sigma_{\mathcal{TN}}$ of the truncated multivariate
    normal distribution (2.25)
for $k = 1, \ldots, K$ do
    for $i = 1, \ldots, P$ do
        $\mu^k_{(i)} = \mu_{\mathcal{TN},(i)} + \Sigma_{\mathcal{TN},(i,-\mathbf{i})} \Sigma^{-1}_{\mathcal{TN},(-\mathbf{i},-\mathbf{i})} \left(\tilde{\mathbf{C}}^k_{(-\mathbf{i})} - \mu_{\mathcal{TN},(-\mathbf{i})}\right)$
        $\sigma_{(i)} = \Sigma_{\mathcal{TN},(i,i)} - \Sigma_{\mathcal{TN},(i,-\mathbf{i})} \Sigma^{-1}_{\mathcal{TN},(-\mathbf{i},-\mathbf{i})} \Sigma^{T}_{\mathcal{TN},(i,-\mathbf{i})}$
        Draw a sample $\tilde{\mathrm{C}}^{k+1}_{(i)} | \tilde{\mathbf{C}}^k_{(-\mathbf{i})}, \mathcal{D} \sim \mathcal{TN}(\mu^k_{(i)}, \sigma_{(i)}, a_{(i)}, b_{(i)})$
    end for
end for
    with $\tilde{\mathbf{C}}^k_{(-\mathbf{i})} = (\tilde{\mathrm{C}}^{k+1}_{(0)}, \ldots, \tilde{\mathrm{C}}^{k+1}_{(i-1)}, \tilde{\mathrm{C}}^{k}_{(i+1)}, \ldots, \tilde{\mathrm{C}}^{k}_{(N_v)})$, $\mu_{\mathcal{TN},(-\mathbf{i})}$ as the mean vector
    without the $i$th element, and $\Sigma^{-1}_{\mathcal{TN},(-\mathbf{i},-\mathbf{i})}$ as the covariance matrix without the
    $i$th row and $i$th column.
```

On the Virtual Observation Locations

As mentioned, sampling from the truncated multivariate normal distribution (2.25) is the most difficult and computationally demanding part of the presented approach to constrain the GP posterior. Assuming that one linear operator $\mathcal{L}$ for each of the spatial dimensions D of a GP is used, then the dimension of (2.25) would be $P = D \cdot N_v$. Furthermore, the number of VOPs N_v depends exponentially on the dimensions D if the whole hyper cube should be filled equidistantly. Therefore, the dimensions of P are

$$P = D \cdot N_{v/D}^{D}. \tag{2.28}$$

with $N_{v/D}$ as the number of VOPs per dimension This is visualized for different spatial dimensions in Figure 2.1. Depending on P, different sampling methods are suitable. Up to $P \approx 100$ the rejection sampling method by Botev [10] showed to perform well. However, at higher dimensions P the acceptance rate becomes too low and Gibbs sampling as proposed in Algorithm 2 showed to perform well. While Gibbs sampling is not limited by an acceptance rate because every sample is accepted, sampling at in dimensions $P > 250$ showed to be also not feasible as the computational effort is too high since the inner loop of Algorithm 2 scales with P.

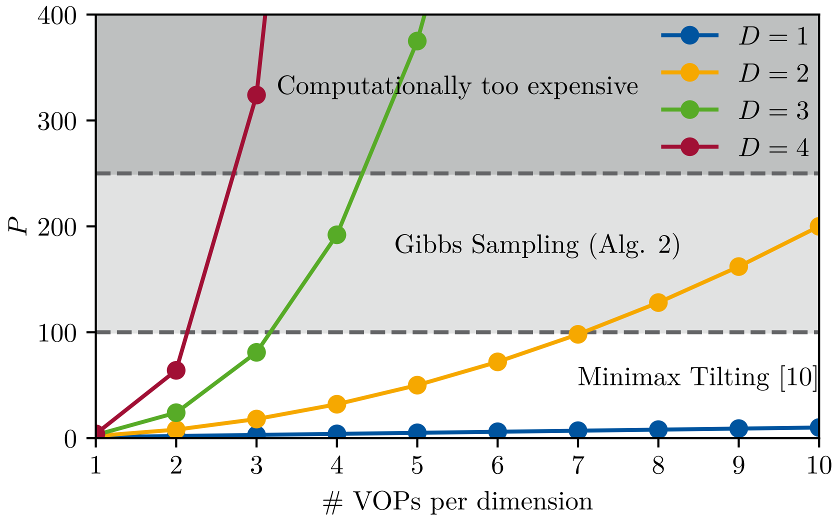

Figure 2.1.: Dependency of the number of VOPs per dimension on the dimension P of the truncated multivariate normal distribution (2.25) for different spatial dimensions D. The displayed bounds are not fixed as sampling strongly depends on the covariance matrix in (2.25).

However, these bounds on the algorithms are not fixed as sampling strongly depends on the covariance matrix of (2.25).

There have been proposals for choosing the VOPs optimally. These methods try to find the locations within the feasible set $\mathcal{X}$ at which the probability of satisfying (2.12) is the lowest. These locations with low probability are then added to the set of VOPs [1][50]. However, this can still result in a large set of VOPs in higher dimensions. Instead, a different approach could be to choose the VOPs based on a sensor placement problem, thus choosing the locations which maximize the probability of satisfying (2.12) for all $\mathbf{x} \in \mathcal{X}$. This would reduce the number of VOPs needed but also increase the computational complexity significantly.

2.1.3. Example: 1D Convexity Constrained Gaussian Process

To show the concept of constraining a GP and the influence of choosing the VOPs, a short one dimensional example is presented. Considering the prior knowledge in Assumption 1 of the objective function staying convex through time, the constrained posterior distribution can be constructed using a linear operator $\mathcal{L} = \frac{\partial^2}{\partial x_i^2}$

and defining the constraint functions $a(\mathbf{x}) = 0$ and $b(\mathbf{x}) = +\infty$ to enforce convexity. Furthermore, nine VOPs are defined in an equidistant grid $\mathbf{X}_v = [-4, \dots, 4] \in \mathbb{R}^9$. To construct the posterior, the Gram matrices with the applied linear operator and the mean with the linear operator have to be constructed. As the mean function is assumed to be constant applying the linear operator results in $\mathcal{L}\mu_{\mathbf{X}_v} = \mathbf{0}$. The Gram matrices are constructed as

$$K_{\mathbf{X},\mathbf{X}_v}\mathcal{L}^T = \left[(K^{1,0}_{\mathbf{X}_v,\mathbf{X}})^T, \dots, (K^{D,0}_{\mathbf{X}_v,\mathbf{X}})^T\right] \tag{2.29}$$

$$K_{\mathbf{X}_*,\mathbf{X}_v}\mathcal{L}^T = \left[(K^{1,0}_{\mathbf{X}_v,\mathbf{X}_*})^T, \dots, (K^{D,0}_{\mathbf{X}_v,\mathbf{X}_*})^T\right] \tag{2.30}$$

$$\mathcal{L}K_{\mathbf{X}_v,\mathbf{X}_v}\mathcal{L}^T = \begin{bmatrix} K^{1,1}_{\mathbf{X}_v,\mathbf{X}_v} & \cdots & K^{1,D}_{\mathbf{X}_v,\mathbf{X}_v} \\ \vdots & \ddots & \vdots \\ K^{D,1}_{\mathbf{X}_v,\mathbf{X}_v} & \cdots & K^{D,D}_{\mathbf{X}_v,\mathbf{X}_v} \end{bmatrix} \tag{2.31}$$

with the notation of $K^{i,0}_{\mathbf{x},\mathbf{x}'} = \frac{\partial^2}{\partial x_i^2}K(\mathbf{x},\mathbf{x}')$ and $K^{i,j}_{\mathbf{x},\mathbf{x}'} = \frac{\partial^4}{\partial x_i^2 x'^2_j}K(\mathbf{x},\mathbf{x}')$ as partial derivatives, which are listed in Appendix A.3 for the SE kernel. The resulting posterior distribution by applying Algorithm 1 is displayed in Figure 2.2. The sampling

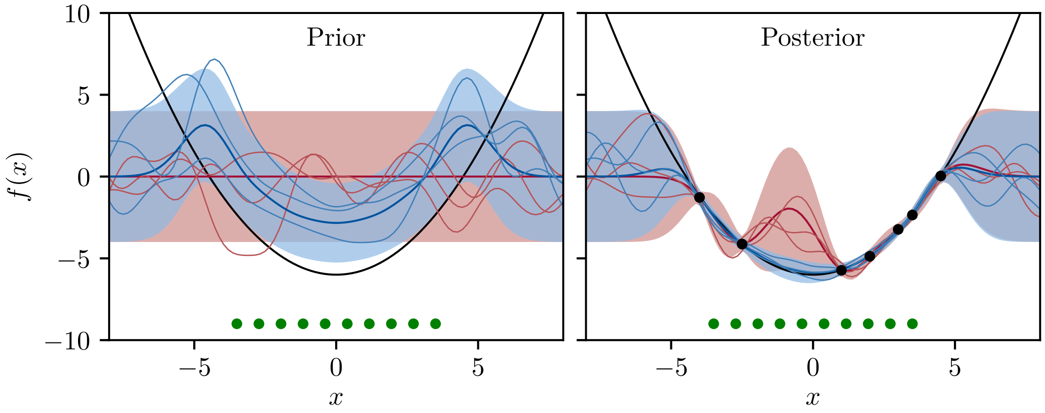

Figure 2.2.: Unconstrained (red) and constrained (blue) prior distribution on the left as well as the unconstrained (red) (2.5) and constrained (blue) (2.26) posterior distribution conditioned on the training data from the objective function (black) on the right. The green points are the VOPs. The thin lines are samples from the corresponding distribution. The hyperparameters are $\mu_0 = 0$, $\sigma_l^2 = 1$, $\sigma_k^2 = 4$, $\sigma_v^2 = 10^{-8}$, $\sigma_n^2 = 0.05^2$.

algorithm for the constrained GP prior distribution as displayed in Figure 2.2 (left) is given in Appendix A.2. It can be observed that at the VOPs the posterior is convex. However, outside of the VOPs the posterior converges back to the unconstrained posterior, highlighting the importance of choosing the VOPs.

2.2. Bayesian Optimization

Optimizing a black-box function $f\colon \mathcal{X} \mapsto \mathbb{R}$ as

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \tag{2.32}$$

is complex, especially if only noisy observations of the form $y = f(\mathbf{x}) + w$ with $w \sim \mathcal{N}(0, \sigma_n^2)$ are available to the optimization algorithm. This is also referred to as bandit feedback. If the function evaluations are cheap, gradients of f can be approximated, and stochastic optimization methods can be applied to find local optima. However, if the function evaluations are expensive, such an approximation of the gradient is not practical. Furthermore, in some applications, it is desirable to find the global optimum. For this setting, BO has been developed as a global optimization method in the case of expensive function evaluation and has been applied, e.g., for optimizing the hyperparameter in deep learning and other machine learning algorithms.

As the objective function is unknown, BO requires a surrogate model which captures the prior believe of the objective function and can be updated from the observations. This model can be a parametric model such as a linear model, however the most common choice in current BO algorithms is the use of a nonparametric model in form of a GP as $f(\mathbf{x}) \sim \mathcal{GP}(m, k)$ as discussed in the previous Section 2.1.

Besides the model, BO requires the definition of an acquisition function in the form of $\alpha(\mathbf{x}|\mathcal{D})\colon \mathcal{X} \mapsto \mathbb{R}$ which maps a query $\mathbf{x}$ to a corresponding value defining the utility of the query. Optimizing the acquisition function and updating the model is performed sequentially as shown in Algorithm 3 up to a terminal condition or after exhausting a predefined observation budget. A few BO steps are also visualized on an example in Figure 2.3.

The sampling efficiency of BO highly depends on choosing a suitable acquisition function for the problem at hand. Different acquisition functions have been

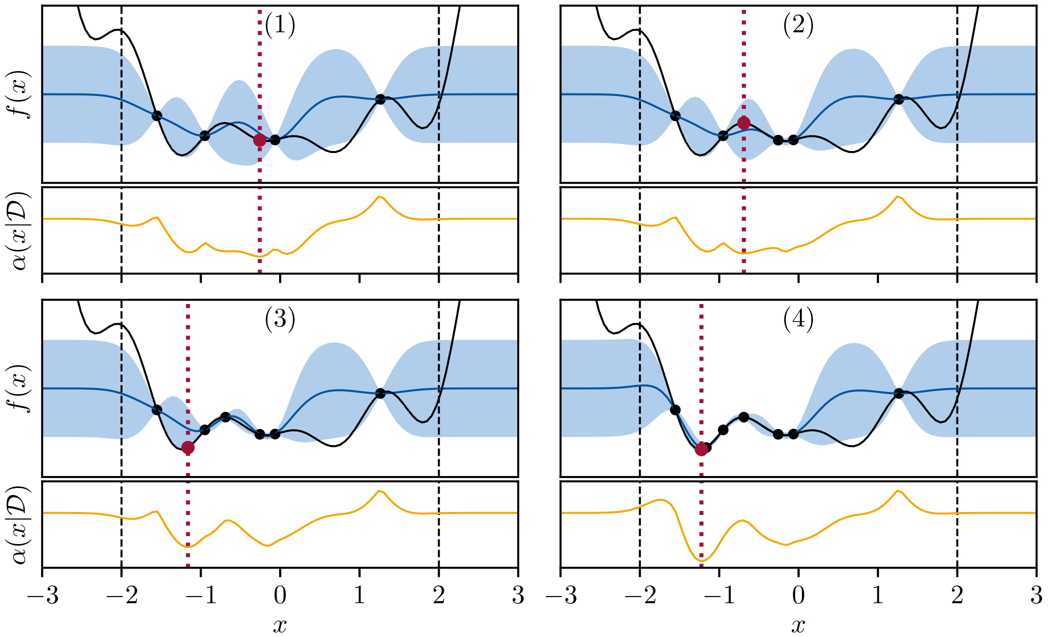

Figure 2.3.: Visualization of BO iterations with the objective function (black line), previous queries (black points) and the chosen query (red points). The query is chosen by minimizing the acquisition function $\alpha(\mathbf{x}|\mathcal{D})$ (orange) based on the current model (blue) within the feasible set $\mathbf{x} \in \mathcal{X} = [-2, 2]$. This is indicated by the red dashed line.

proposed such as probability of improvement (PI)[25], expected improvement (EI) and upper-confidence-bound (UCB)[3] (or in the case of minimizing an objective function lower-confidence-bound (LCB)), each with different characteristics regarding exploration and exploitation. Every acquisition function has to balance the exploration-exploitation trade-off in exploring the objective function by querying at locations with high variance or exploiting the model's mean. The mentioned acquisition functions are myopic, as they try to optimize in a one-step-look-ahead fashion considering only the model's current state as opposed to planing a sequence of queries.

Regret

To evaluate the performance of different BO algorithms a metric called regret is used. It defines the cost of choosing a query at iteration k which deviates from the optimum. The objective of a BO algorithm is then to minimize the cumulative regret.

Definition 1 (Cumulative regret). *Let $\mathbf{x}^*$ be the optimizer to the function $f(\mathbf{x})$ and let $\mathbf{x}_k$ be the queried point by the algorithm at iteration k. The cumulative regret after K iterations is then given by*

$$R_K \coloneqq \sum_{k=1}^{K} (f(\mathbf{x}_k) - f(\mathbf{x}^*)). \tag{2.33}$$

A desirable characteristic of a BO algorithm is to achieve sub-linear regret (also called *no-regret*) as

$$\lim_{T \to \infty} \frac{R_K}{T} = 0. \tag{2.34}$$

It defines that for large T the cumulative regret converges to a constant implying the convergence of the algorithm to the true optimum $\mathbf{x}^*$. Such a no-regret algorithm is GP-LCB as defined in [44] with the acquisition function

$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x} \in \mathcal{X}} \alpha_{GP-LCB}(\mathbf{x}|\mathcal{D}) = \arg\min_{\mathbf{x} \in \mathcal{X}} \mu_k(\mathbf{x}) - \sqrt{\beta_{k+1}}\, \sigma_k(\mathbf{x}) \tag{2.35}$$

with μ_k and σ_k describing the posterior mean and standard deviation of the GP model from the previous iteration, respectively. Hence, β_{k+1} defines the mentioned exploration-exploitation trade-off at the current iteration for GP-LCB. Setting β_{k+1}

Algorithm 3 BO [40]

```
Initialize: prior $f \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$; feasible set $\mathcal{X} \in \mathbb{R}^D$; data set $\mathcal{D}_N = \{y_j, \mathbf{x}_j\}_{j=0}^{N}$
for $k = N, 2, \ldots, K$ do
    Train GP model with $\mathcal{D}_k$
    choose next query $\mathbf{x}_{k+1} = \arg\min_{\mathbf{x} \in \mathcal{X}} \alpha(\mathbf{x}|\mathcal{D})$
    query objective function $y_{k+1} = f(\mathbf{x}_{k+1}) + w$
    update data set $\mathcal{D}_{k+1} = \mathcal{D}_k \cup \{y_{k+1}, \mathbf{x}_{k+1}\}$
end for
```

according to Srinivas *et al.* [44, Theorem 1] results in proven sub-linear regret. For a deeper introduction to BO it is referred to Shahriari *et al.* [40].

Time-Varying Bayesian Optimization

The following notation of TVBO is based on the notation in [49]. In TVBO the unknown objective function is time-varying as $f\colon \mathcal{X} \times \mathcal{T} \mapsto \mathrm{I\!R}$ where $\mathcal{T}$ represents the time domain as an increasing sequence $\mathcal{T} = \{1, 2, \ldots, T\}$ with T as the time horizon. To include the time dependency into the GP model, the current state-of-the-art is to use a product composite kernel of $k_S\colon \mathcal{X} \times \mathcal{X} \mapsto \mathrm{I\!R}$ and $k_T\colon \mathcal{T} \times \mathcal{T} \mapsto \mathrm{I\!R}$ resulting in the *spatio-temporal kernel*

$$k\colon \mathcal{X} \times \mathcal{T} \mapsto \mathrm{I\!R}, \quad k(\{\mathbf{x}, t\}, \{\mathbf{x}', t'\}) = k_S(\mathbf{x}, \mathbf{x}') \otimes k_T(t, t') \tag{2.36}$$

with $\otimes$ as the Hadamard product. The kernel k_S embeds the spatial correlations within $\mathcal{X}$ and implies the Bayesian regularity assumptions on $f_t(\mathbf{x})$ as being a sample from a GP prior with kernel k_S at each time step. The spatial kernel k_S is often chosen to be a kernel from the Matérn class such as the SE kernel. The kernel k_T characterizes the temporal correlations and defines how to treat data from the past. As defined in (2.10) the resulting kernel $k(\{\mathbf{x}, t\}, \{\mathbf{x}', t'\})$ is a valid kernel, as long as k_S and k_T are valid kernels in $\mathcal{X}$ and $\mathcal{T}$, respectively. The base algorithm for TVBO is displayed in Algorithm 4.

Algorithm 4 Base TVBO

```
Initialize: prior $\mathcal{GP}(m(\mathbf{x}), k_S(\mathbf{x}, \mathbf{x}') \otimes k_T(t, t'))$ and hyperparameter; feasible set
    $\mathcal{X} \in \mathrm{I\!R}^D$; data set $\mathcal{D}_N = \{y_j, \mathbf{x}_j, t_j\}_{j=0}^N$
t_0 = N  →  $t_0 = N$
for $t = t_0, t_0 + 1, t_0 + 2, \ldots, T$ do
    Train GP model with $\mathcal{D}_t$
    choose next query $\mathbf{x}_{t+1} = \arg\min_{\mathbf{x} \in \mathcal{X}} \alpha(\mathbf{x}, t + 1 | \mathcal{D})$
    query objective function $y_{t+1} = f_{t+1}(\mathbf{x}_{t+1}) + w$
    update data set $\mathcal{D}_{t+1} = \mathcal{D}_t \cup \{y_{t+1}, \mathbf{x}_{t+1}, t + 1\}$
end for
```

In contrast to standard BO, in TVBO the acquisition function is constrained to only choose a query for the next time step $t + 1$, given the GP model of the current time step, even though the GP model is defined over the whole domain $\mathcal{X}\times\mathcal{T}$ through the kernel k. Moreover, in this time-varying environment, there is no longer a single optimizer for the objective function, but an optimizer for each time step, which may vary over time. Therefore, a different notion of regret has to be defined, to capture the performance of a TVBO algorithm. For this purpose, the dynamic cumulative regret metric is introduced (Definition 2).

Definition 2 (Dynamic cumulative regret)**.** *Let $\mathbf{x}_t^*$ be the optimizer to the time-varying function $f_t(\mathbf{x})$ as $\mathbf{x}_t^* = \arg\min_{\mathbf{x}\in\mathcal{X}} f_t(\mathbf{x})$ at time step t and let $\mathbf{x}_t$ be the queried point by the algorithm at time step t. Than the dynamic cumulative regret after T time steps is*

$$R_T \coloneqq \sum_{t=1}^{T}(f_t(\mathbf{x}_t) - f_t(\mathbf{x}_t^*)). \tag{2.37}$$

In the following, the dynamic cumulative regret will only be referred to as regret for convenience. Achieving sub-linear regret in a general time-varying setting is not possible without stating assumptions on the amount of change over the time horizon T [8]. The intuition behind is that it is not possible to track the optimum with arbitrary precision if the objective function changes significantly at each time step [9]. However, when $f_t(\mathbf{x})$ is a function in an RKHS $\mathcal{H}_K$ with a bounded norm and the amount of change is limited and known a-priori within a variation budget P_T [7] as

$$\sum_{t=1}^{T-1} ||f_t(\mathbf{x}) - f_{t+1}(\mathbf{x})||_{\mathcal{H}_K} \leq P_T, \tag{2.38}$$

TVBO algorithms have been developed that have been proven to achieve sub-linear regret [51].

3. Related Work

This thesis proposes the modelling approach UI-TVBO and the method C-TVBO. UI-TVBO is the first model to implement a UI forgetting strategy into the GP model of TVBO and C-TVBO utilizes prior knowledge about the shape of the objective function to improve the sampling efficiency. In the following, related work for these methods is discussed and the resulting research gap is identified which is filled by this thesis.

Embedding Prior Knowledge in Bayesian Optimization

BO has been a powerful optimization framework for optimizing black-box functions in time-invariant environments. Especially GP-based BO has shown to efficiently find global optima of multimodal functions also in higher dimensional spaces [42] and while providing strong convergence guarantees under mild assumptions [44]. However, choosing a suitable kernel is crucial to increase the sample efficiency of BO as the kernel defines the hypothesis space of the unknown objective function. Therefore, the selection of a suitable kernel is a way to incorporate prior knowledge, and Duvenaud [15] developed a method for automatically creating a kernel based on a given data set. Marco *et al.* [27] considered an LQR problem and specifically designed an *LQR kernel* improving sampling efficiency compared to the universal SE kernel.

However, rather than restricting the hypothesis space by the choice of kernel, the proposed method C-TVBO enforces shape constraints on the GP posterior. Jauch and Peña [20] used the method of Wang and Berger [50] with the rejection sampling algorithm of Botev [10] to enforce convexity constraints in standard BO for finding optimal hyperparameters of a SVM. The method of Wang and Berger [50] was also applied by Ospina *et al.* [30] in an online primal-dual optimization algorithm using a convexity constrained GP to model the objective function. Furthermore, Owen *et al.* [31] applied monotonicity constraints as introduced in Riihimäki and Vehtari [38] to actively learn a decision hyper plane through user feedback in psychophysics experiments. Also, recent work by Jeong and Kim [21] restricts the hypothesis space not through the kernel but by conditioning the GP model to consider prior knowledge about lower and upper bounds on the optimum.

Modeling Time-Varying Functions Using Gaussian Processes

Modeling time varying-function in a Bayesian setting has discussed in multiple regression and Bayesian filtering studies [47][48][39][12]. To model spatial and temporal correlations of a time-varying function using a GP, the proposed method UI-TVBO uses a spatio-temporal kernel as in (2.36). This approach of separating spatial and temporal data, each handled in a separate kernel, was also used in Sarkka *et al.* [39] and Carron *et al.* [12].

Modeling time varying-function in a Bayesian setting has discussed in multiple regression and Bayesian filtering studies [47][48][39][12]. To model spatial and temporal correlations of a time-varying function using a GP, the proposed method UI-TVBO uses a spatio-temporal kernel as in (2.36). This approach of separating spatial and temporal data, each handled in a separate kernel, was also used in Sarkka *et al.* [39] and Carron *et al.* [12].

In Van Vaerenbergh *et al.* [47], a kernel recursive least squares tracker was developed to capture nonlinear and time-varying correlations in data by explicitly modeling information loss in the algorithm. Since the informativeness of past data from a time-varying function decreases over time, they propose two methods to model this notion of forgetting at each time step explicitly. The first method is called B2P forgetting and captures the idea that as the informativeness of a measurement decreases, the algorithm should converge back to the prior distribution. A formal definition for B2P forgetting is given in Definition 3.

Definition 3 (Back-To-Prior forgetting, adapted from Van Vaerenbergh *et al.* [47, Section 3.A])). *Given a prior prediction for the expected value at the spatial location* $\mathbf{x}$ *as* $\mu_{0,\mathbf{x}}$ *and let* $\mu_{t,\mathbf{x}}$ *be the expected value at location* $\mathbf{x}$ *at time* t*. Then an algorithm implies B2P forgetting if* $\lim_{t\to\infty} \mu_{t,\mathbf{x}} = \mu_{0,\mathbf{x}}$ *after observing no more data.*

The second presented approach, UI forgetting, expresses forgetting by maintaining the mean estimate but expressing uncertainty by increasing the variance (Definition 4).

Definition 4 (Uncertainty-Injection forgetting, adapted from Van Vaerenbergh *et al.* [47, Section 3.B])). *Let* $\mu_{t_1,\mathbf{x}}$ *be the expected value at location* $\mathbf{x}$ *at time* t_1 *and let* $\sigma^2_{t_1,\mathbf{x}}$ *be the variance at location* $\mathbf{x}$ *at time* t_1. *Then an algorithm implies UI forgetting if* $\mu_{t,\mathbf{x}} = \mu_{t_1,\mathbf{x}}$ *and* $\sigma^2_{t,\mathbf{x}} > \sigma^2_{t_1,\mathbf{x}}$, $\forall t > t_1$ *after observing no more data after* t_1.

The two forgetting strategies will be used to compare modeling approaches in TVBO. To account for less informative data, Meier and Schaal [28] use *drifting GPs* discarding data points after some time steps and performing GP regression in a sliding window on the current data set. Following Definition 3, this can be interpreted as a special case of B2P forgetting as discarding a data point is equivalent to assuming the prior distribution at that location. Therefore, this sliding window approach arises naturally as an approximation strategy to algorithms which imply B2P forgetting and it will be used in Chapter 5 as an approximation method for B2P forgetting TVBO algorithms.

Optimization in Time-Varying Environments with Bandit Feedback

Minimizing regret over finite actions in a time-varying environment is subject to the study of dynamic multi-armed bandits (MABs) or sometimes called restless bandits. On the assumption of gradual changes in the regret at each bandit, Slivkins and Upfal [41] model the regret as Brownian motion. In Brownian motion, a particle is assumed to perform a random walk which can mathematically be described as a Wiener process. A Wiener process has the property that between each time step the variance is increased while the mean remains constant. Therefore, according to Definition 4, the developed algorithm in Slivkins and Upfal [41] implies a UI forgetting strategy. In contrast, Chen *et al.* [13] modeled the regret not as a Wiener process but an autoregressive model of order 1 in

$$X_t = c + \varphi X_{t-1} + \epsilon_t \tag{3.1}$$

of which the Wiener process is a special case with $\varphi = 1$, $c = 0$, and ϵ_t as the standard normal distribution. For $\varphi \in (0, 1)$, the stochastic process in (3.1) converges to 0. If 0 is defined as the expected value of a prior prediction, than the implicit forgetting strategy in Chen *et al.* [13] can be considered as B2P forgetting.

In the dynamic MABs approaches the regret of each bandits is considered to be independent and therefore these algorithms seek to exploit only temporal correlations. In contrast, the problem formulation in Section 1.1 considers an objective function which is defined over infinitely many bandits as $\mathcal{X} \subset \mathbb{R}^D$ correlating the regret also in the spatial dimension. Therefore, an algorithm for this setting should exploit temporal *and* spatial correlations. This is the goal of algorithms in GP-based TVBO.

GP-based TVBO was first discussed in Bogunovic *et al.* [9] under a Bayesian regularity assumption of f_t being a sample from a GP prior at each time step as $f_t \sim \mathcal{GP}(\mathbf{0}, k)$. Another assumption is, that f_t can be modeled as a Markov chain given a sequence of independent samples $g_1, g_2, \dots$ from a zero mean GP prior $g_t \sim \mathcal{GP}(\mathbf{0}, k)$ with kernel k as

$$f_1(\mathbf{x}) = g_1(\mathbf{x}) \tag{3.2}$$

$$f_{t+1}(\mathbf{x}) = \sqrt{1-\epsilon} f_t(\mathbf{x}) + \sqrt{\epsilon} g_{t+1}(\mathbf{x}), \quad \forall t \geq 2. \tag{3.3}$$

Here, $\epsilon \in [0, 1]$ is the forgetting factor, which defines how much the objective function varies at each time step. For $\epsilon = 0$ this formulation is equivalent to standard BO while for $\epsilon = 1$ the objective function is modeled to be independent at each time step. For an objective function which satisfies the modeling assumptions, Bogunovic *et al.* [9] derived regret bounds for two algorithms – R-GP-UCB and TV-GP-UCB. Instead of including the time-varying nature of the objective function into the GP model, R-GP-UCB performes GP-UCB in *blocks* of size $H = \lceil \min(T, 12\epsilon^{-\frac{1}{4}}) \rceil$ (for a SE spatial kernel [9, Corollary 4.1]). The mean μ_t and the variance σ_t of the GP model are reset after every H time steps. In contrast, the presented models in this thesis explicitly model the objective function as time-varying and are therefore more similar to the second algorithm TV-GP-UCB. Here, the objective function is modeled using a GP with a spatio-temporal kernel (2.36) with temporal kernel

$$k_{T,tv}(t, t') = (1-\epsilon)^{\frac{|t-t'|}{2}} \tag{3.4}$$

which is similar to the kernel defined by the stochastic Ornstein-Uhlenbeck process and is a form of B2P forgetting. Applying $k_{T,tv}$ on the temporal dimension is identical to decreasing the *weight* of data from the past to model the loss of information over time [9]. TV-GP-UCB has been used in a few different applications such as controller learning [45], safe adaptive control [23], and online hyperparameter optimization in reinforcement learning [33][32] as well as for developing a non-myopic acquisition function for TVBO [37]. The stationary kernel $k_{T,tv}$ implies B2P forgetting, as the posterior mean (2.6) and covariance (2.7) converge to the prior distribution for large $\Delta t = |t - t'|$ since

$$\lim_{\Delta t \to \infty} k_{T,tv}(t, t') = \lim_{\Delta t \to \infty} (1 - \epsilon)^{\frac{\Delta t}{2}} = 0, \quad \forall \epsilon \in (0, 1). \tag{3.5}$$

In contrast, the proposed method UI-TVBO utilizes a temporal kernel implementing UI forgetting as opposed to B2P forgetting. Such a modeling approach with a spatio-temporal kernel for BO can also be considered as a special case of *contextual* BO [24] with the context time, but the constraint of not being able to sample in the whole context space. In fact, in Krause and Ong [24, Section 5.3 and Section 6.2] a Matérn kernel with $\nu = \frac{5}{2}$ was used as a temporal kernel

$$k_{T,mat\acute{e}rn}(t, t') = \left(1 + \frac{\sqrt{5}\Delta t}{l} + \frac{5\Delta t^2}{3l^2}\right) \exp\left(-\frac{\sqrt{5}}{l}\Delta t\right). \tag{3.6}$$

Like (3.5), $k_{T,mat\acute{e}rn}$ converges to 0 for $\Delta t \to \infty$ implying B2P forgetting. Unlike the above presented approaches to TVBO, Baheri and Vermillion [5] used an additive instead of a product composite kernel as spatio-temporal kernel. The proposed method UI-TVBO uses a product composite kernel as it aims to exploit spatial *and* temporal correlations [36, Section 4.2.4].

The approach of *weighting* the data to incorporate information loss over time as in TV-GP-UCB was also studied under frequentist regularity assumptions by Deng *et al.* [14]. The frequentist regularity assumptions are, that the objective function f_t is not a sample of a GP, but its smoothness is defined at each time step by the RKHS with a bounded norm of the corresponding spatial kernel. The resulting algorithm WGP-UCB [14] uses for each acquired data points $\mathbf{x}_t$ a weight $w_t = \gamma^{-t}$ with $\gamma \in (0, 1)$ as forgetting factor. This yields in an increased weight of each new data point as opposed to decreasing the weight of old data points as in TV-GP-UCB. However, the implicit forgetting strategy remains B2P forgetting as in algorithm data points converge back to a zero mean. R-GP-UCB has also been

Forgetting strategy	**Infinte bandits** (GP-based TVBO)	**Finite bandits** (Dynamic MABs)
B2P forgetting	[9],[49],[14],[51],[29],[19]	[13]
UI forgetting	UI-TVBO, C-UI-TVBO	[41]

Table 3.1.: Research gap in modeling time-varying environments with bandit feedback. In this thesis, UI forgetting for TVBO with (C-UI-TVBO) and without (UI-TVBO) convexity constraints is evaluated.

considered under frequentist regularity assumptions and regret bounds have been derived by Zhou and Shroff [51]. Furthermore, Zhou and Shroff [51] introduced SW-GP-UCB with a similar idea to R-GP-UCB. Instead of performing GP-UCB in blocks, it performs GP-UCB in a sliding window of size W which moves with time. Based on the Markov chain assumption of TV-GP-UCB, Wang *et al.* [49] developed the CE-GP-UCB algorithm for online hyperparameter optimization, which only opts to observe feedback from the objective function if the variance at the query points chosen by TV-GP-UCB is high. Therefore, contrary to Section 1.1, CE-GP-UCB can skip iterations.

All the prior work in TVBO discussed above used discrete time steps $t \in \mathcal{T} = \{1, 2, \ldots, T\}$ and therefore a fixed sampling interval of $\Delta t = 1$ as it is the case in the problem formulation in Section 1.1. However, there have been approaches presented with a varying sampling interval [29][19][35]. Imamura *et al.* [19] considered the same Markov chain model as TV-GP-UCB but accounted for a non-constant evaluation time. They were able to prove lower regret bounds for Bayesian regularity assumptions compared to Bogunovic *et al.* [9] if the evaluation time is explicitly considered. Similarly, in the algorithm ABO-f [29], the time at which the objective function is evaluated can vary. ABO-f uses a SE kernel as a spatial as well as a temporal kernel. Unlike in the problem formulation in Section 1.1, the acquisition function in ABO-f is not limited to choosing a query within the feasible spatial set $\mathcal{X}$ at the current time. It can change the box constraints for optimizing the acquisition function at each iteration based on the learned length scale of the temporal kernel. The acquisition function therefore chooses also the time at which the objective function is to be evaluated as $\{\mathbf{x}, t\} = \arg\min_{\mathbf{x}} \alpha(\mathbf{x}, t|\mathcal{D})$. To validate ABO-f and show its benefits compared to optimizing at fixed time intervals, Nyikosa *et al.* [29] introduced ABO-t as a fixed time variation of ABO-f, which also uses a temporal SE kernel denoted as $k_{T,se}$ hereafter. Since this satisfies the problem formulation, ABO-t will be used as a benchmark. Raj *et al.* [35] extended ABO-f by using a spectral mixture kernel as the temporal kernel to improve the extrapolation properties in the temporal dimension.

In reviewing related work on optimization in time-varying environments considering only bandit feedback, the research gap was identified as shown in Figure 3.1. The proposed method UI-TVBO fills this research gap introducing the first modeling approach using UI forgetting in GP-based TVBO. Furthermore, in C-TVBO, this thesis presents a method to incorporate prior knowledge about the shape of the objective function in TVBO, which, to the best of my knowledge, has not been done in a time-varying setting.

4. Methods

In this chapter, the methods proposed in this thesis are derived and presented on an example to build up intuition. The methods are:

- **UI-TVBO:** A modeling approach to TVBO incorporating UI forgetting.
- **C-TVBO:** An method embedding prior knowledge about the shape of the objective function into TVBO using shape constraints.

Afterwards, extensions which are relevant for the practical application of both methods are discussed.

4.1. Uncertainty-Injection Forgetting in Time-Varying Bayesian Optimization

The current state-of-the-art modeling for forgetting in TVBO is B2P forgetting. According to Definition 3, the expectation about a function value propagates back to the prior belief over time, losing all information. Meanwhile, UI forgetting is expressed by increasing the variance over time and, the key difference compared to B2P forgetting, maintaining structural information of the function value in the form of the posterior mean. Therefore, inspired by Slivkins and Upfal [41], the modeling approach UI-TVBO is introduced and transfers the concept of UI forgetting to the infinite bandit setting in TVBO to retain structural information.

The intuition behind UI forgetting is gradual change. Figure 4.1 shows an example with measurements taken at a fixed spatial coordinate x_1 of an objective function at time steps t_1–t_4. If no prior knowledge about a drift in the objective function is available, the expected value remains constant until the next measurement. However, the variance increases due to the exact value becoming more uncertain over time. As in Slivkins and Upfal [41], this can be formalized as a Wiener process. In

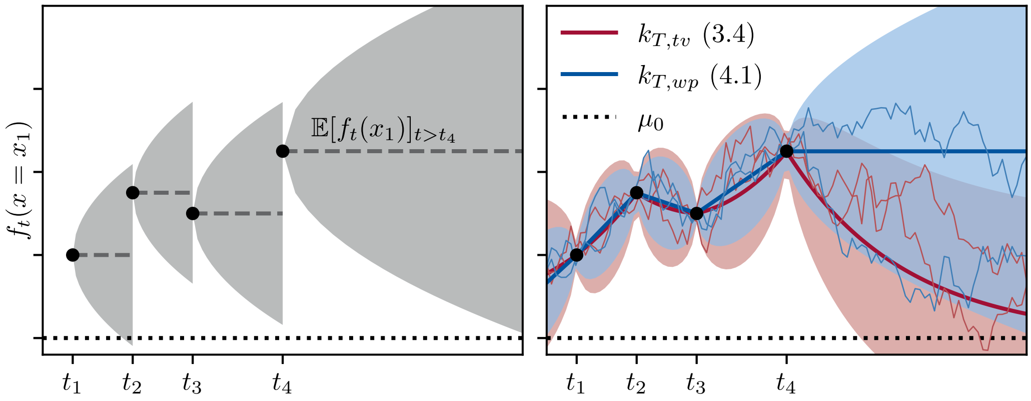

Figure 4.1.: On the left, the intuition behind UI forgetting is displayed. Until a new measurement is conducted, the expected value remains the same but the uncertainty increases. On the right, the Wiener process kernel and the B2P forgetting kernel $k_{T,tv}$ are applied on the temporal data. The thin lines are samples from the posterior.

order to use the Wiener process for describing temporal correlations with a GP, the Wiener process kernel is introduced as

$$k_{T,wp}(t,t') = \sigma_w^2 \left(\min(t,t') - c_0\right) \tag{4.1}$$

with a scaling factor σ_w^2 and a start time parameter c_0. Applying the Wiener process kernel $k_{T,wp}$ as well as the B2P forgetting kernel $k_{T,tv}$ in (3.4) to the described example is displayed on the right in Figure 4.1. The information on the last measurements' function value is preserved with $k_{T,wp}$, while the expected value using $k_{T,tv}$ converges to the prior mean μ_0. Furthermore, B2P forgetting can also be interpreted as assuming gradual changes, but with the bias that these changes are always in the direction of the predefined prior.

To correlate also the spatial with the temporal dimension, UI-TVBO uses a product composite kernel as in (2.36) and previous work [9][29]. Considering a SE kernel for the spatial dimensions and a Wiener process kernel for the temporal dimension the spatio-temporal product kernel k for UI-TVBO is

$$k(\{\mathbf{x},t\},\{\mathbf{x}',t'\}) = \sigma_k^2 \exp\left(-\frac{1}{2}\boldsymbol{\tau}^T\boldsymbol{\Lambda}^{-1}\boldsymbol{\tau}\right) \cdot \sigma_w^2 \left(\min(t,t') - c_0\right) \tag{4.2}$$

with $\boldsymbol{\tau} = \mathbf{x} - \mathbf{x}'$ for notation convenience. This modeling approach is not limited to the SE kernel for the spatial dimension. Other kernels, e.g., from the Matèrn class, are suitable as well. The spatio-temporal kernel in (4.2) possesses $3 + D$ hyperparameters. The output variance σ_k^2 and the D length scales depend on the objective function at each time step leaving the two hyperparameters c_0 and σ_w^2 to characterize the forgetting of the model. However, similar to B2P forgetting kernel $k_{T,tv}$, it is desirable to have only one additional hyperparameter for the temporal dimension that defines the forgetting of the model. Therefore, c_0 and σ_w^2 are correlated with each other resulting in the definition of only one forgetting hyperparameter for UI-TVBO in the following.

By definition, no forgetting has occurred at the time step $t = 0$. Therefore, for inputs with the same spatial location, the output of the kernel in (4.2) should be the output variance σ_k^2. Consequently, the start time parameter c_0 of the Wiener process kernel for $\boldsymbol{\tau} = \mathbf{0}$ is calculated as

$$k(\{\mathbf{x}, 0\}, \{\mathbf{x}, t'\}) = \sigma_k^2 \cdot \sigma_w^2 \left(\min(0, t') - c_0\right) \overset{!}{=} \sigma_k^2 \tag{4.3}$$

$$\Longrightarrow \; c_0 \overset{!}{=} -\frac{1}{\sigma_w^2}, \quad \text{with } \min(0, t') = 0, \, \forall t' \geq 0. \tag{4.4}$$

At a different time step t_1 with $t_1 > 0$ and $t_1 < t'$ the variance should increase with $\sigma_w^2 \cdot t_1$ as it is the case for a Wiener process. The output of k for $\boldsymbol{\tau} = \mathbf{0}$ at t_1 is then

$$\begin{aligned} k(\{\mathbf{x}, t\}, \{\mathbf{x}, t'\}) &= \sigma_k^2 \cdot \sigma_w^2 \left(\min(t_1, t') - c_0\right) && (4.5)\\ &= \sigma_k^2 \cdot \sigma_w^2 \left(t_1 - c_0\right), \quad \text{with } \min(t_1, t') = t_1, \, \forall t' \geq t_1 && (4.6)\\ &= \sigma_k^2 \cdot \sigma_w^2 \left(t_1 + \frac{1}{\sigma_w^2}\right) && (4.7)\\ &= \underbrace{\sigma_k^2 \cdot \sigma_w^2 \cdot t_1}_{\text{variance due to forgetting}} + \underbrace{\sigma_k^2}_{\text{variance of the spatial kernel}} . && (4.8) \end{aligned}$$

However, the increase in variance due to forgetting still depends on the output variance of the spatial kernel σ_k^2. Therefore, the scaling factor of the Wiener process σ_w^2 has to be normalized by σ_k^2 as

$$\sigma_w^2 = \frac{\hat{\sigma}_w^2}{\sigma_k^2} \tag{4.9}$$

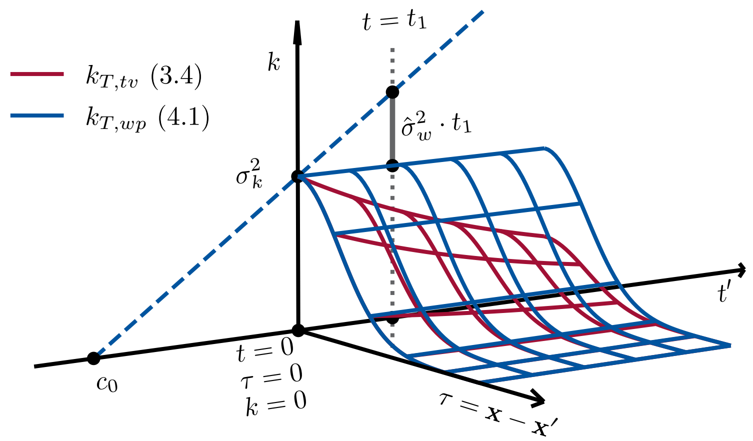

Figure 4.2.: Different temporal kernels with a SE spatial kernel and the resulting output of the spatio-temporal kernel over time and spatial distance, given a fixed first input of k. The dashed line is visual interpretation for deriving the hyperparameters c_0 and σ_w^2 in (4.3)–(4.11).

resulting in

$$\begin{aligned} k(\{\mathbf{x}, t\}, \{\mathbf{x}, t'\}) &= \sigma_k^2 \cdot \sigma_w^2 \left(\min(t_1, t') - c_0\right) && (4.10) \\ &= \underbrace{\hat{\sigma}_w^2 \cdot t_1}_{\text{variance due to forgetting}} + \underbrace{\sigma_k^2}_{\text{variance of the spatial kernel}} && (4.11) \end{aligned}$$

with $\hat{\sigma}_w^2$ as the hyperparameter defining the increase in variance at each time step. It is denoted as the UI forgetting factor hereafter. This normalization has to be performed at each time step, if the output scale of the spatial kernel varies over time, e.g., due to hyperparameter learning. As c_0 is fixed due to the starting conditions as $c_0 = -\sigma_k^2/\hat{\sigma}_w^2$, $\hat{\sigma}_w^2 \in (0, \infty)$ remains as the only hyperparameter of the UI-TVBO model defining the change in objective function over time. A graphical interpretation of the hyperparameters as well as the behavior of different temporal kernels are given in Figure 4.2. It shows that for a fixed first entry of the kernel (e.g. a training point) the output of the kernel remains constant afterwards given the same spatial location of the second input. In contrast, for the kernel $k_{T,tv}$ implying B2P forgetting the output of the spatio-temporal kernel propagates towards $k = 0$. Furthermore, for

$\hat{\sigma}_w^2 \to 0$ the UI-TVBO model converges to a time-invariant setting.

Comparing Back-2-Prior Forgetting and Uncertainty-Injection Forgetting

In Figure 4.3 a comparison between B2P and UI forgetting is conducted. In each of the three sub-figures, a different constant prior mean is defined. An optimistic prior mean is visualized on the left, reflecting an overly positive expectation of the optimum. In contrast, the right side shows a pessimistic prior mean overestimating the optimum. The center shows a well-defined prior. At the time $t = 0$, the objective function

$$f_t(x) = (0.25x)^2 \tag{4.12}$$

was learned with an equidistant grid of twelve training points in all three cases. At each subsequent time step, only the points $\mathbf{X}_t = [-1, 0, 1]$ and the corresponding training targets are added to the data set. The posteriors shown are each at time $t = 3$ (top row) and $t = 50$ (bottom row). For B2P forgetting the pessimistic and optimistic prior mean cause a significant deviation of the posterior mean from the objective function at $t = 50$. In contrast, the mean of UI forgetting is independent of the prior and is able to maintain the information of the objective function in form of the mean.

In the context of TVBO, assuming the acquisition function is of a similar form as GP-LCB [44] in (2.35), an optimistic mean for B2P forgetting will likely result in more exploration. On the other hand, a pessimistic mean will result in more exploitation. Especially in cases with the objective functions' mean changing over time, this sensitivity of B2P forgetting on the prior mean may be undesirable and lead to an increased cumulative regret. Since UI forgetting is independent of the prior mean, if enough data points within $\mathcal{X}$ have been observed, a much more robust performance in terms of cumulative regret is expected with this type of modeling. These assumptions are summarized in Hypothesis 1 and 2. They will be tested empirically in Chapter 5.

Hypothesis 1. *As UI forgetting maintains structural information, the regret will be smaller compared to B2P forgetting if there is a offset in prior mean towards an optimistic mean.*

Hypothesis 2. *UI forgetting and B2P forgetting show similar performance if the*

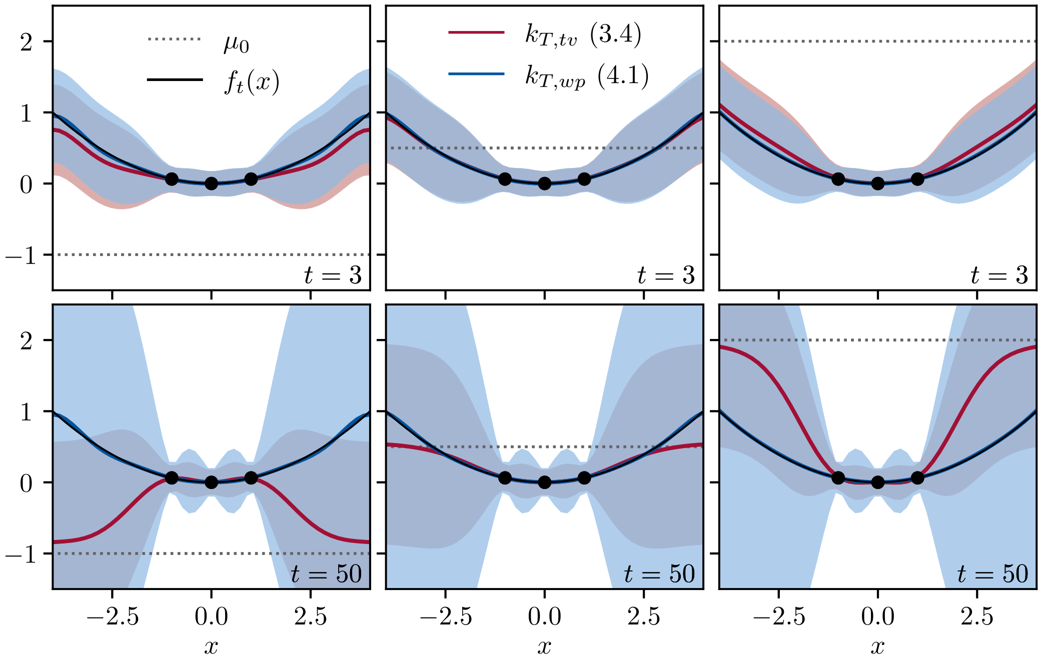

Figure 4.3.: Comparison of the posteriors at $t = 3$ (top row) and $t = 50$ (bottom row) using B2P forgetting and UI forgetting with a optimistic prior (left), well-defined prior (center), and pessimistic prior (right) with forgetting factors $\epsilon = \hat{\sigma}_w^2 = 0.1$.

objective function is mean-reverting and the prior mean is a well-defined prior mean.

Additionally, Figure 4.3 shows that the variance for UI forgetting with $k_{T,wp}$ as temporal kernel does not converge to the output variance of the spatial kernel as it does with B2P forgetting. It diverges in regions where no data is added. Therefore, approaches that limit the variance of UI forgetting could be beneficial in terms of dynamic cumulative regret. One approach would be to include prior knowledge about the shape of the objective function, as discussed in the next section.

4.2. Modeling Convex Objective Functions

Following Assumption 1, the prior knowledge of the objective function staying convex through time is available and embedding prior knowledge in BO has proven to increase sampling efficiency as discussed in Chapter 3. Therefore, the method C-TVBO is proposed to include prior knowledge about the shape of the objective function increasing sampling efficiency and reducing dynamic cumulative regret in a time varying setting.

The objective function remaining convex means that the second derivative in any spatial direction must be greater than zero at each time step. Therefore, shape constraints on the GP are used limit the hypothesis space to only functions with

$$\begin{aligned} \frac{\partial^2 f_t(\mathbf{x})}{\partial x_i^2} \geq 0, \quad & \forall i \in [1, \dots, D] \\ & \forall \mathbf{x} \in \mathcal{X} \\ & \forall t \in \mathcal{T}. \end{aligned} \tag{4.13}$$

In combination with a smooth kernel such as the SE kernel, these constraints approximate the positive definiteness of the Hessian of f_t.

Ideally, the constraints in (4.13) are enforced globally. However, such methods have not been developed yet for GPs. Therefore, the method by Agrell [1] as introduced in Section 2.1.1 is used, which guarantees the convexity only at a finite set of points. However, Wang and Berger [50] as well as Agrell [1] observed, that a sufficient amount of such VOPs results in a high probability of the posterior being convex throughout the feasible set $\mathcal{X}$.

Using the method by Agrell [1] in the time-varying context means applying D linear operators $\mathcal{L}_i = \frac{\partial^2}{\partial x_i^2}$ on the posterior at every time step. As the kernel of the GP is a spatio-temporal kernel, the necessary derivatives in the Gram matrices (2.29)–(2.31) for calculating the factors of the constrained posterior in (2.26) are

$$K_{\mathbf{x},\mathbf{x}'}^{i,0} \coloneqq \frac{\partial^2}{\partial x_i^2} K(\{\mathbf{x}, t\}, \{\mathbf{x}', t'\}) = \left[\frac{\partial^2}{\partial x_i^2} K_S(\mathbf{x}, \mathbf{x}')\right] \otimes K_T(t, t') \tag{4.14}$$

and

$$K_{\mathbf{x},\mathbf{x}'}^{i,j} \coloneqq \frac{\partial^4}{\partial x_i^2 x_j'^2} K(\{\mathbf{x}, t\}, \{\mathbf{x}', t'\}) = \left[\frac{\partial^4}{\partial x_i^2 x_j'^2} K_S(\mathbf{x}, \mathbf{x}')\right] \otimes K_T(t, t'). \tag{4.15}$$

with $\otimes$ as the Hadamard product, $K_S(\mathbf{x}, \mathbf{x}')_{i,j} = k_S(\mathbf{x}_i, \mathbf{x}_j')$, and $K_T(t, t')_{i,j} = k_T(t_i, t_j')$. This is possible because the linear operators act only on the spatial dimensions, of which the temporal kernel is independent. Furthermore, this restricts the spatial kernel k_S to be at least twice differentiable, while any valid kernel can be used as the temporal kernel.

Next, the VOPs $\mathbf{X}_v$ must be placed in the time-variant context. Ideally, the VOPs would be placed dense throughout the domain $\mathcal{X} \times \mathcal{T}$ to ensure convexity. However, as discussed in Section 2.1.1, this is not possible as sampling from the truncated multivariate normal distribution (2.25) becomes infeasible. Therefore, the VOPs are distributed in an equidistant grid only at the current time step ensuring convexity at the time step at which the acquisition function is optimized. The method C-TVBO applied to TVBO is shown in Algorithm 5.

Algorithm 5 TVBO using C-TVBO

```
**Initialize:** prior $\mathcal{GP}(m(\mathbf{x}), k_S(\mathbf{x}, \mathbf{x}') \otimes k_T(t, t'))$ and hyperparameter; feasible set
    $\mathcal{X} \in \mathbb{R}^D$; data set $\mathcal{D}_N = \{y_j, \mathbf{x}_j, t_j\}_{j=0}^N$; number of VOPs per dimension $N_{v/D}$,
    bounding functions $a(\mathbf{X}_v), b(\mathbf{X}_v)$
$t_0 = N$
for $t = t_0, t_0 + 1, t_0 + 2, \dots, T$ do
    Train GP model with $\mathcal{D}_t$
    # Choose VOPs
    Create equidistant grid $X_{v,\mathcal{X}}$ with $N_{v/D}$ VOPs in each spatial dimensions
    $X_v = \{X_{v,\mathcal{X}}, t + 1\}$
    # Calculate constrained posterior
    Calculate Gram matrices (2.29)–(2.31) using (4.14) and (4.15)
    Calculate factors for the posterior                                ▷ Appendix A.1
    Sample from the posterior at $t + 1$ to obtain $\mu_{t+1}, \sigma_{t+1}^2$           ▷ Algorithm 1
    choose next query $\mathbf{x}_{t+1} = \arg\min_{\mathbf{x} \in \mathcal{X}} \alpha(\mathbf{x}, t + 1 | \mu_{t+1}, \sigma_{t+1}^2)$
    query objective function $y_{t+1} = f_{t+1}(\mathbf{x}_{t+1}) + w$
    update data set $\mathcal{D}_{t+1} = \mathcal{D}_t \cup \{y_{t+1}, \mathbf{x}_{t+1}, t + 1\}$
end for
```

In the following, an intuition about C-TVBO is presented based on the example objective function in (4.12). As the posterior distribution should satisfy (4.13) the bounding functions in (2.13) are set to

$$a(\mathbf{X}_v) = 0, \quad b(\mathbf{X}_v) = \infty. \tag{4.16}$$

Applying the proposed method C-TVBO (Algorithm 5) to the example in Figure 4.3 with the bounding functions as in (4.16) is visualized in Figure 4.4. It can be observed that the variance for UI forgetting no longer diverges where the VOPs are placed. Furthermore, the convexity at the VOPs prevents the mean of B2P forgetting from falling back to the optimistic prior mean, as was the case in Figure 4.3. Most noticeable, however, is the mean of UI forgetting changing over time, although this was not observed in the non-constrained case. Even for B2P forgetting with an optimistic prior, the mean initially moves upwards after three time steps, although the opposite is expected. Since the constraints limit the hypothesis space, the constrained prior mean is no longer constant within the VOPs, but takes on different *natural curvatures* depending on the bounding functions in (2.13). Here, *natural curvatures* means the shape of the function that results from the prior distribution

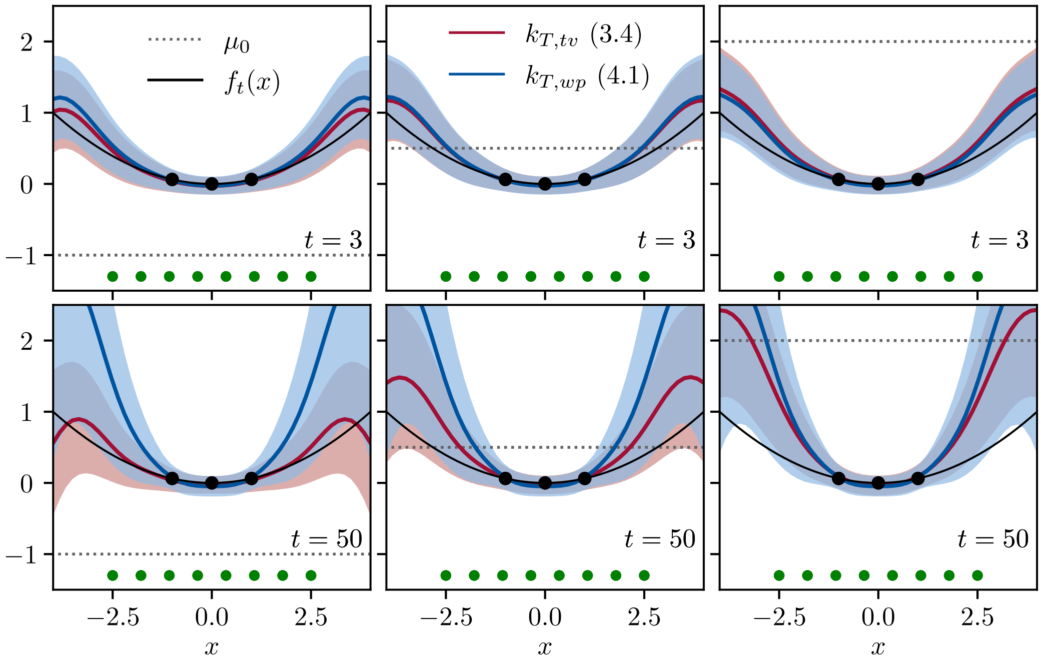

Figure 4.4.: Comparison of the constrained posteriors at $t = 3$ (top row) and $t = 50$ (bottom row) using B2P forgetting and UI forgetting with a optimistic prior (left), well-defined prior (center), and pessimistic prior (right) with forgetting factors $\epsilon = \hat{\sigma}_w^2 = 0.1$ and bounding functions as in (4.16).

over the GP's second derivative. The effect of choosing different bounding functions on the prior distribution is displayed in Figure 4.5. All three of the displayed prior distributions enforce convexity at the VOPs, but vary in the *natural curvature* of the prior mean prediction. In the time-varying setting, the mean of the constrained posterior distribution of UI forgetting converges to this *natural curvature* implied by the bounding functions, since the Wiener process modeling the temporal change is no longer unbiased but biased. For constrained B2P forgetting the bias is also induced but counteracted by the propagation back to the constant prior mean.

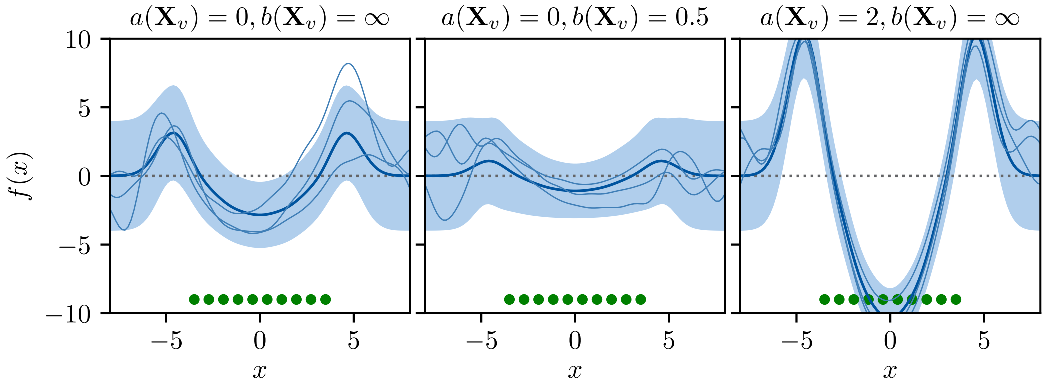

Figure 4.5.: The effect of choosing different bounding functions $a(\mathbf{X}_v)$, $b(\mathbf{X}_v)$ on the constrained prior distribution. The green points denote the VOPs and the dotted line is the unconstrained prior mean. Depending on the bounds on the second derivative, the constrained prior mean has a different *natural curvature*.

If prior knowledge about an upper bound on the second derivative is available, it can be incorporated into C-TVBO thereby influencing the curvature of the prior distribution. As the second derivative is given as $\partial^2 f_t(x)/\partial x^2 = 0.125$, the upper bounding function can be specified as

$$b(\mathbf{X}_v) = 2 \cdot \frac{\partial^2 f_t(\mathbf{x})}{\partial x^2} = 0.25. \tag{4.17}$$

Applying again the C-TVBO method with the adjusted upper bound function is displayed in Figure 4.6. Including the upper bound further refines the induced bias, resulting in a good approximation to the objective function even after 50 time steps. The characteristic behaviors of B2P forgetting and UI forgetting can also be

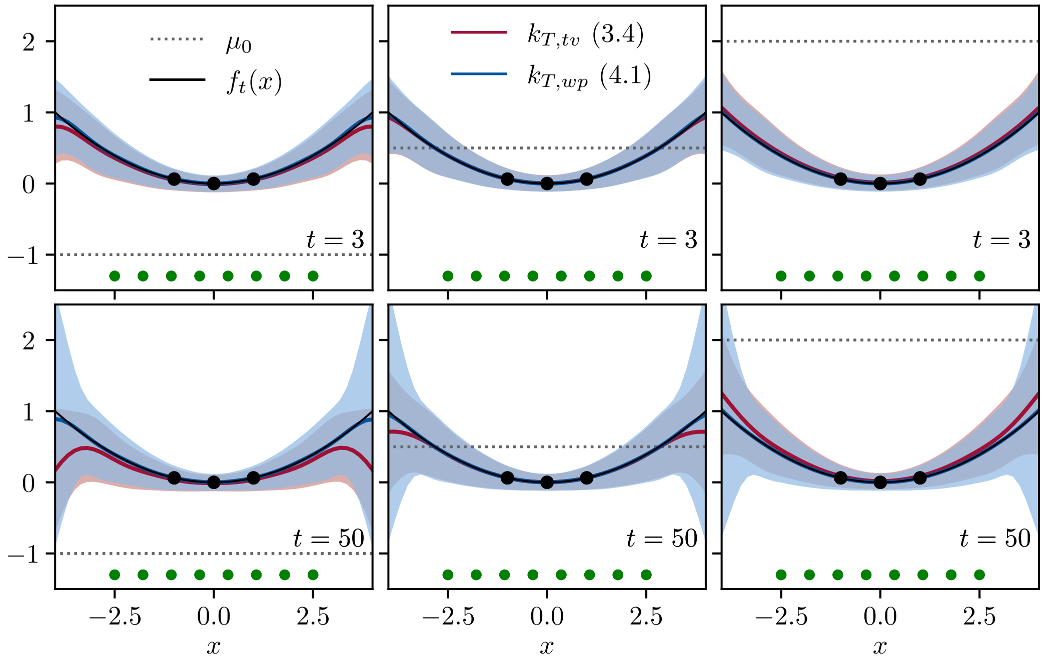

Figure 4.6.: Comparison of the constrained posteriors at $t = 3$ (top row) and $t = 50$ (bottom row) using B2P forgetting and UI forgetting with a optimistic prior (left), well-defined prior (center), and pessimistic prior (right) with forgetting factors $\epsilon = \hat{\sigma}_w^2 = 0.1$ and bounding functions $a(\mathbf{X}_v) = 0$, $b(\mathbf{X}_v) = 0.25$. In contrast, the unconstrained posteriors are displayed in Figure 4.3.

observed, with the mean of B2P forgetting tending towards the constant prior mean μ_0 and the mean of UI forgetting being independent of μ_0. Moreover, the downside of UI forgetting of a diverging variance is neglected by enforced inequality constraints. Away from the VOPs, the posterior distributions behave as in the unconstrained case. The mean of B2P forgetting propagates back to constant prior mean and the variance of UI forgetting diverges.

The tendencies of the forgetting strategies and the discussed bias introduced by constraining the posterior also become apparent when considering the propagation of the distribution of one function value taken at $t = 0$ of the objective function in (4.12) at the location $x_1 = 2.75$ over 50 time steps. This is displayed in Figure 4.7. In the unconstrained case (left), the posterior distribution of B2P forgetting at

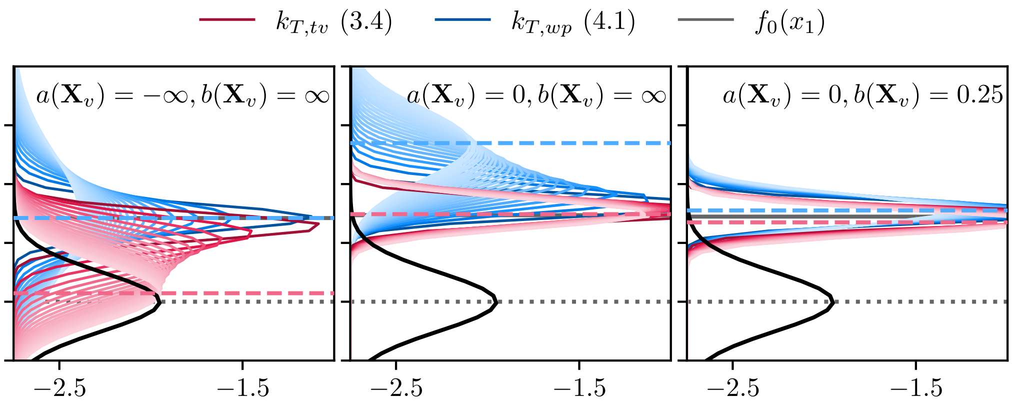

Figure 4.7.: Propagation of the distribution of a measurement taken at $t = 0$ over 50 time steps. Black denotes the prior distribution and the dotted gray line the prior mean. The colored dashed lines are the means at time step $t = 50$. On the left the unconstrained case is visualized, in the middle the constrained case with bounds on the second derivative $\partial^2 f_t(\mathbf{x})/\partial x^2 \in [0, \infty]$, and on the right the constrained case with bounds $\partial^2 f_t(\mathbf{x})/\partial x^2 \in [0, 0.25]$.

x_1 propagates back to the prior distribution, both the mean and the variance. In contrast, UI forgetting maintains the structural information by keeping the expected value of $f_t(x_1)$ constant in the form of a constant posterior mean over time. The subfigure in the middle shows the induced bias with the mean propagating towards the *natural curvature* as a result of truncating the distribution on the second derivative of f_t with the bounding functions as in Figure 4.4. The mean of B2P forgetting seems to remain constant as a result of forgetting and induced bias counteracting each other. However, this exact compensation is not always the case and depends on the objective function, the forgetting factor, as well as μ_0. Lastly, the right subfigure shows the distribution propagation for also including the upper bound on the second derivative as in Figure 4.6, refining the induced bias and limiting the variance of the posterior.

By presenting an example with the proposed method C-TVBO, an intuition about the behavior of the posterior distributions for the different forgetting strategies was built. A bias is induced by truncating the second derivative to be only positive, which affects the posterior mean over time as it converges to the shape of the *natural curvature*. An additional upper bound significantly refined the bias, resulting in a better function approximation. Therefore, if more information about the second derivative is available, such as an estimate of its magnitude in each dimension, it can be incorporated as an upper bound, as shown in the example. If this is not the case, the presented method can still be applied, and it is recommended to set the bounds as in (4.16).

Convexity constraints on the posterior distribution ensure that the posterior mean at the predicted optimum of f_t is always the smallest within the feasible set. Therefore, they reduce the probability that the acquisition function selects queries further away from the predicted optimum, preventing undesired exploration. Consequently, it is expected, that C-TVBO performs better in terms of dynamic cumulative regret compared to standard TVBO as stated in Hypothesis 3 if an objective function satisfies Assumption 1.

Hypothesis 3. *By incorporating prior knowledge, C-TVBO performs better than standard TVBO in terms of dynamic cumulative regret, regardless of the type of forgetting.*

On Hyperparameter Estimation

The hyperparameters of the spatial kernel are estimated using a marginal maximum likelihood approach of the unconstrained GP. In Bachoc *et al.* [4], the influence of including the constraints on the GP into the marginal maximum likelihood estimate was studied and an advantage for small data sets was shown. Since the constraints in the time-varying environment prevent the model from taking global queries as discussed in the previous Section 4.2, the correlation within the data set is high, making the effective size of the data set smaller. Therefore, including the constraints in the hyperparameter estimation could be beneficial. However, the added computational effort caused by considering the constraints must also be taken into account. Therefore, as in Agrell [1], the presented approach does not consider the constraints during hyperparameter estimation.

4.3. Numerical and Practical Considerations

In the following, extensions to the proposed methods UI-TVBO and C-TVBO for the practical application are discussed regarding scalability in terms of run time of each optimization iteration as well as an infinite time horizon.

Local Approximation

The bottleneck of the proposed method C-TVBO is its limitation regarding the number of VOPs due to sampling from the truncated multivariate normal distribution. The number of VOPs depends on the size of the feasible set relative to the spatial length scale in each dimension. If the feasible set is large relative to the spatial length scale, a large number of VOPs would be required to enforce convexity throughout the feasible set approximately. In order to still be able to use the proposed method in such scenarios, it is additionally assumed that the optimum between consecutive time steps only changes within a length scale as described in Assumption 2.

Assumption 2. *The optimizer* $\mathbf{x}_t^*$ *does not change more than one length scale within one time step therefore* $|\mathbf{x}_{t,i}^* - \mathbf{x}_{t-1,i}^*| \leq \mathbf{\Lambda}_{ii}$, $\forall i \in [1, \dots, D]$ *holds for all for all* $t \in \mathcal{T}$.

If Assumption 2 is satisfied, the bounds of the feasible set $\mathcal{X}$ for the optimization of the acquisition function can be adjusted for every sequential time step $t+1$, depending on the predicted optimum of the previous time step $\hat{\mathbf{x}}_t^*$ as shown in Figure 4.8. The upper and lower bounds in each dimension $i \in [1, \dots, D]$ of the new feasible set $\tilde{\mathcal{X}}_{t+1} \subseteq \mathcal{X}$ are set as

$$\tilde{\mathcal{X}}_{t+1,i,lb} = \begin{cases} \hat{\mathbf{x}}_{t,i}^* - \mathbf{\Lambda}_{ii}\,, & \text{if } \hat{\mathbf{x}}_{t,i}^* - \mathbf{\Lambda}_{ii} \geq \mathcal{X}_{i,lb} \\ \mathcal{X}_{i,lb} & \text{otherwise} \end{cases} \tag{4.18}$$

$$\tilde{\mathcal{X}}_{t+1,i,ub} = \begin{cases} \hat{\mathbf{x}}_{t,i}^* + \mathbf{\Lambda}_{ii}\,, & \text{if } \hat{\mathbf{x}}_{t,i}^* + \mathbf{\Lambda}_{ii} \leq \mathcal{X}_{i,ub} \\ \mathcal{X}_{i,ub} & \text{otherwise.} \end{cases} \tag{4.19}$$

The optimization of the acquisition function (Algorithm 5, line 12) then changes to

$$\mathbf{x}_{t+1} = \arg \min_{x \in \tilde{\mathcal{X}}_{t+1}} \alpha(\mathbf{x}, t+1 | \mu_{t+1}, \sigma_{t+1}^2). \tag{4.20}$$

The VOPs are not restricted to be within the feasible set of the acquisition function. Therefore, they are distributed in an equidistant grid around the predicted optimum $\hat{\mathbf{x}}_t^*$. The lower and upper bound in each spatial dimension for the grid of

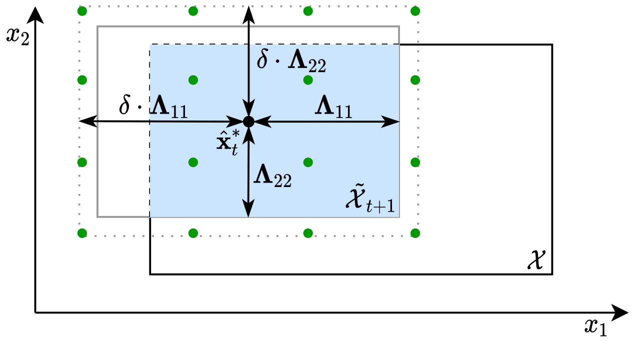
x_2
$\delta \cdot \mathbf{\Lambda}_{22}$
$\delta \cdot \mathbf{\Lambda}_{11}$
$\mathbf{\Lambda}_{11}$
$\hat{\mathbf{x}}_t^*$
$\mathbf{\Lambda}_{22}$
$\tilde{\mathcal{X}}_{t+1}$
$\mathcal{X}$
x_1

Figure 4.8.: Visualization of the local approximation at every time step depending on the predicted optimum of the previous time step $\hat{\mathbf{x}}_t^*$ and the length scales resulting in the new feasible set $\tilde{\mathcal{X}}_{t+1}$ (blue). The green dots denote the VOPs placed in an equidistant grid around $\hat{\mathbf{x}}_t^*$.

VOPs are

$$\text{bounds for VOPs in each dimension} = [\hat{\mathbf{x}}_{t,i}^* - \delta \cdot \mathbf{\Lambda}_{ii}, \, \hat{\mathbf{x}}_{t,i}^* + \delta \cdot \mathbf{\Lambda}_{ii}] \tag{4.21}$$

with $\delta \geq 1$ as hyperparameter to enforce convexity also beyond the bounds of $\tilde{\mathcal{X}}_{t+1}$ accounting for the spatial correlation. If an objective function does not satisfy Assumption 2, C-TVBO with local approximation might induce a delay in tracking the optimum as $\tilde{\mathcal{X}}_{t+1}$ may be too restrictive for the acquisition function. However, due to the convexity of the objective function it is likely, that C-TVBO will still outperform TVBO in terms of dynamic cumulative regret.

Data Selection

TVBO is intended as an algorithm running online to make decisions at equidistant time steps, such as choosing the parameters of a controller. Here, the question of what happens in the case of a very large or infinite time horizon since GPs scale cubical as $\mathcal{O}(N^3)$ in the number of training points ariises. Therefore, for such a scenario with $T \to \infty$, data selection strategies are needed, introducing sparsity into the TVBO algorithm.

Data Selection for Back-2-Prior Forgetting

In B2P forgetting, data points from the past propagate back to the prior distribution. Therefore, discarding data points after a fixed amount of time depending on the forgetting factor, arises naturally as a data selection method for B2P forgetting. This results in a sliding window approach similar to Meier and Schaal [28]. The size of the sliding window W can be calculated for the temporal kernel $k_{T,tv}$ as

$$k_{T,tv} = (1-\epsilon)^{\frac{W}{2}} \leq p \;\; \text{(correlation threshold)} \tag{4.22}$$

$$\Leftrightarrow W \geq 2\frac{\ln p}{\ln(1-\epsilon)} \implies W = \left\lceil 2\frac{\ln p}{\ln(1-\epsilon)} \right\rceil . \tag{4.23}$$

The correlation threshold $p \in (0, 1]$ is a design parameter which determines the time step after which the data points can be discarded. If the sliding window size $W = T$, the posterior at each time step will be exact.

Data Selection for Uncertainty-Injection Forgetting

A sliding window approach can not be applied to UI-TVBO as the primary motivation of the UI forgetting strategy is to maintain important structural information from the past. Discarding old training points after a fixed amount of time might lead to the loss of this structural information. Therefore, a data selection method for UI forgetting based on binning is presented and displayed in Figure 4.9. Each

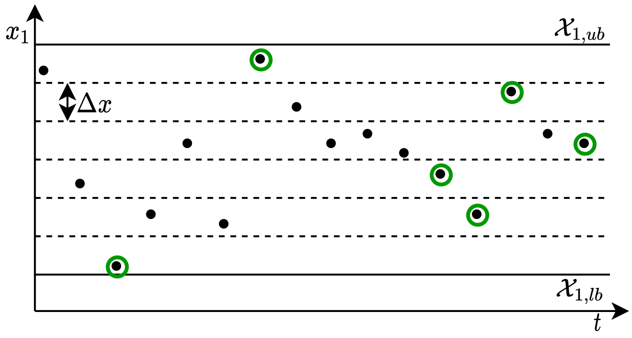

Figure 4.9.: Selection of data points for UI forgetting. The black dots denote the data points observed over time. The feasible set is divided into equal sized bins and only the last data point (circled in green) is added to the active data set for TVBO.

spatial dimension is divided into bins of equal width, and each data point is assigned to one bin depending on its spatial coordinates. In each bin, only the last observed point remains in the data set. The intuition behind this is that in UI forgetting data points at the same spatial coordinate are overwritten over time as in Figure 4.1, making the previous data point obsolete. With the introduction of bins, it is now assumed that new data points not only overwrite the information at the same coordinate but also overwrite information within the width of their bin Δx, since k_S correlates the data points spatially. This allows the algorithm to consider also data points further in the past and maintain their structural information. For a bin width $\Delta x \to 0$, the posterior approximation becomes exact. For better empirical performance, this binning approach can be combined with a small sliding window to include the last n observed data points resulting in a locally more exact approximation of the posterior. At the same time, the remaining bins maintain the global structural information.

As binning suffers from the curse of dimensionality, other data selection strategies or approximation methods have to be applied at higher dimensions. An adaptive grid with varying bin sizes can be an option. Desirable would be a method similar to Titsias [46] which would find inducing points approximating the posterior at the current time step by minimizing the Kullback-Leibler (KL) divergence. However, such approximation methods introduce additional computation effort, and the number of inducing points needed also scales with the spatial dimensions D, compared to the sliding window approach for B2P forgetting, which only depends on the forgetting factor and the correlation threshold. Therefore, in this thesis the binning strategy is used as a data selection strategy for UI forgetting.

5. Results

In this chapter, the proposed methods are evaluated empirically and the Hypotheses 1–3 are tested on different experiments. For this purpose, the proposed modeling approach UI-TVBO is compared to TV-GP-UCB[1]by Bogunovic *et al.* [9] using standard TVBO as well as the proposed method C-TVBO, both with and without data selection. This results in the different variations summarized in Table 5.1.

Variation	TVBO Algorithm	Data Selection	Forgetting Strategy
UI-TVBO	standard TVBO	–	UI
B UI-TVBO	standard TVBO	binning	UI
TV-GP-UCB[1]	standard TVBO	–	B2P
SW TV-GP-UCB[1]	standard TVBO	sliding window	B2P
C-UI-TVBO	C-TVBO	–	UI
B C-UI-TVBO	C-TVBO	binning	UI
C-TV-GP-UCB[1]	C-TVBO	–	B2P
SW C-TV-GP-UCB[1]	C-TVBO	sliding window	B2P

Table 5.1.: Variations which are evaluated in this chapter. Standard TVBO denotes Algorithm 4, C-TVBO denotes the proposed method in Algorithm 5. The data selection strategies are as discussed in Section 4.3.

Furthermore, to test Hypothesis 1, each experiment is performed with a well-defined and an optimistic prior mean. An optimistic prior mean investigates the robustness of the variations regarding a misspecified prior distribution. Such robustness is desirable, especially regarding real-world applications where the mean of an objective function changes over time.

[1] Note that only the *model* as introduced by Bogunovic *et al.* [9] will be used for comparison, not the UCB algorithm.

The acquisition function throughout this chapter will be LCB as

$$\alpha(\mathbf{x}, t+1|\mathcal{D}) = \mu_{t+1}(\mathbf{x}) - \sqrt{\beta_{t+1}}\,\sigma_{t+1}(\mathbf{x}) \tag{5.1}$$

with a constant exploration-exploitation factor of $\beta_{t+1} = 2$. Different choices for β_{t+1} may be appropriate for different variations in Table 5.1, however, the exploration-exploitation also depends on the forgetting factors. Fixing the acquisition function increases the emphasis on the modeling approaches and algorithms.

The variations are implemented in Python and are based on PyTorch [34] using GPyTorch [16] for modeling the GP and BoTorch [6] for optimizing the acquisition function.

5.1. Synthetic Experiments

To compare the different variations in Table 5.1 different synthetic experiments are performed. First, the variations are compared qualitatively in the within-model as well as out-of-model comparison. These investigate the behavior of the proposed methods if all assumptions are satisfied by the objective function. Then quantitative comparisons are conducted using benchmarks similar to those of Renganathan *et al.* [37]. The comparisons are performed both one-dimensional ($D = 1$) and two-dimensional ($D = 2$). For the method C-TVBO $\delta = 1.5$ (4.21) is chosen and for $D = 1$ the number of VOPs per dimension is set to $N_{v/D} = 10$, for $D = 2$ to $N_{v/D} = 5$. For the data selection strategies, the number of bins per dimension is set to 20, and the sliding window size is set to $W = 30$ for $D = 1$ and $W = 80$ for $D = 2$. The sliding window sizes were chosen to account for the same amount of training data as binning since approximately 80 bins were filled in the two-dimensional experiments. The same number of training points allows for a straightforward comparison.

5.1.1. Within-Model Comparison

The first experiments conducted are within-model comparisons motivated by Hennig and Schuler [18] and previous work by Bogunovic *et al.* [9] where the objective function is generated according to the model assumptions at hand. In this thesis, the assumptions are the objective function staying convex through time captured by the proposed method C-TVBO as well as temporal change according to a Wiener process as embedded in UI-TVBO. For the within-model comparisons, the hyperparameters are known a-priori to the algorithm. Therefore no hyperparameter optimization is performed.

Algorithm 6 Generate Within-Model Objective Function

```
Initialize: prior $\mathcal{GP}(m(\mathbf{x}), k_S(\mathbf{x}, \mathbf{x}') \otimes k_T(t, t'))$ and hyperparameter; feasible set
    $\mathcal{X} \in \mathrm{I\!R}^D$; number of VOPs per dimension $N_{v/D}$, truncation bounds $a(\cdot), b(\cdot)$
Sample $f_0$ from constrained prior distribution        ▷ Appendix A.2
$f = [f_0]$
for $t = 0, 1, \ldots, T$ do
    Learn previous sample $f_t$
    Place VOPs at time steps $t$ and $t + 1$
    Sample $f_{t+1}$ from the posterior at $t + 1$
    $f = [f; f_{t+1}]$
end for
Output: $f$
```

Usually, the within-model objective function is generated by sampling from the GP prior distribution with fixed hyperparameters. However, the number of VOPs needed to span across the whole domain $\mathcal{X} \times \mathcal{T}$ to generate one sample is too high as discussed in Section 4.2. Therefore, the objective function is generated in an iterative fashion according to Algorithm 6 with the temporal kernel $k_{T,wp}$ and a fixed UI forgetting factor $\hat{\sigma}_w^2$. The generated output is a time-varying objective function satisfying Assumption 1 and the temporal change of a Wiener process which are assumptions relevant for real-world applications.

For the one-dimensional within-model objective functions, the samples at each time step are generated on the feasible set $\mathcal{X} = [-5, 9]$ with a length scale $\mathbf{\Lambda}_{11} = 3$, output variance of $\sigma_k^2 = 1$, and a prior mean of $m(\mathbf{x}) = \mathbf{0}$. Furthermore, the UI forgetting factor is set to $\hat{\sigma}_w^2 = 0.03$, the number of VOPs per dimension is set to $N_{v/D} = 10$, and the bounding functions are $a(\mathbf{X}_v) = 0$ and $b(\mathbf{X}_v) = 1$. For creating the two-dimensional within-model objective functions, the same settings are chosen except with $N_{v/D} = 6$, the feasible set as $\mathcal{X} = [-7, 7]^2$, and $\mathbf{\Lambda}_{11} = \mathbf{\Lambda}_{22} = 3$.

The generated objective functions are not *within-model* for the variations using B2P forgetting which has to be accounted for. The UI forgetting factor $\hat{\sigma}_w^2$ implies the increase in variance after one time step. This is also implied by ϵ in the Markov chain model of TV-GP-UCB in (3.3) as shown in Appendix A.4. Therefore, ϵ is set to be $\epsilon = \hat{\sigma}_w^2$. Furthermore, the Wiener process kernel $k_{T,wp}$ causes the output variance of the composite kernel to increase with $\hat{\sigma}_w^2$ at each time step (see (4.11)). Therefore, to allow a comparison, at each time step, the output variance of the models using B2P forgetting is also increased by $\hat{\sigma}_w^2$ as

$$\sigma_{k,t}^2 = \sigma_k^2 + \hat{\sigma}_w^2 \cdot t \text{ (for B2P forgetting)} \tag{5.2}$$

with $\sigma_{k,t}^2$ as the output variance at time step t. Nevertheless, caution is needed when quantitatively comparing the forgetting strategies. Here, the synthetic examples in Sections 5.1.3 and 5.1.4 are more appropriate. However, qualitative trends between the forgetting strategies can be highlighted from the within-model comparisons.

The variations of the Table 5.1 were evaluated on five different objective functions for $D = 1$ and $D = 2$ generated according to Algorithm 6. Five simulations are performed on each objective function using different initializations of $N = 15$ initial training points. The initializations were consistent for each variation. Figure 5.1 shows the results for the one-dimensional within-model comparison.

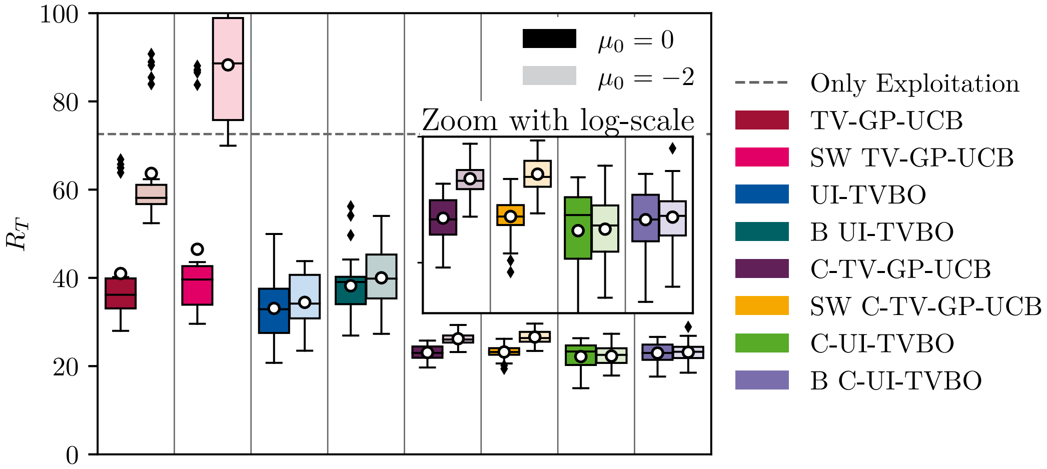

Figure 5.1.: Results for the one-dimensional within-model comparison. The white circles represent the mean of each variation. The darker shades show the performance with a well-defined mean, while the lighter shades show the performance with an optimistic mean. C-TVBO significantly reduces the regret and the sensitivity regarding an optimistic prior.

The dashed line indicates the regret obtained if the minimum of the posterior mean after the initialization had been chosen as the query for the whole time horizon. Figure 5.1 shows that all variations, except SW TV-GP-UCB with an optimistic prior mean, fall below this regret. It indicates the sliding window of 30 being too small for the chosen forgetting factor. However, it can be stated that regardless of the forgetting method and algorithm, it is worth considering the time-varying nature of the objective function. The proposed method C-TVBO reduces the regret for both B2P forgetting and UI forgetting compared to normal TVBO.

Furthermore, by taking into account the prior knowledge, the variance is reduced. Additionally, the proposed modeling approach UI-TVBO shows only minor differences when changing the prior mean. In contrast, the variations with B2P forgetting strongly respond to an optimistic mean with an increased exploratory behavior and thus higher regret. This behavior was expected and stated in Hypothesis 1. Nevertheless, C-TVBO restricts the explorative behavior and, therefore, reduces the effect of the optimistic prior compared to standard TVBO.

The data selection strategies consistently result in worse regret compared to the variations using all queried data. This behavior is expected as the posterior at each time step is only an approximation. However, it should be noted that the binning approach for UI forgetting has only a diminutive impact on regret and is, therefore, a suitable data selection strategy for UI forgetting. For the sliding window approach for B2P forgetting, a larger window size W could improve the regret because the temporal correlations after 30 time steps are still significant using a forgetting factor of $\epsilon = 0.03$. Combining the proposed methods UI-TVBO and C-TVBO results in the best performance in terms of cumulative regret, both for a well-defined and optimistic prior mean.

In the results of the two-dimensional within-model comparisons in Figure 5.2, similar trends can be observed, however, they are not as distinct as in the one-dimensional case. Again, all variations perform better compared to only choosing the minimum of the posterior mean after the initialization. Furthermore, it can be observed that also in the two-dimensional case, the method C-TVBO reduces the regret as well as its variance compared to standard TVBO, which further strengthens Hypothesis 3.

The sensitivity of the forgetting strategies to a shifted prior mean are the same for C-TVBO as in the one-dimensional case. UI forgetting reacts only slightly to the shifted mean compared to B2P forgetting. Contrary to expectations, the sensitivity

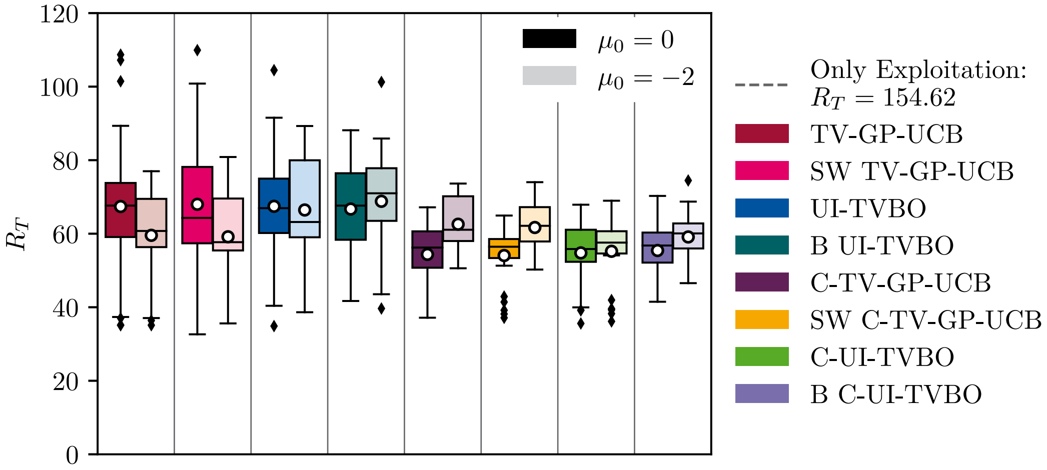

Figure 5.2.: Results for the two-dimensional within-model comparison. It shows lower regret and a smaller regret variance for C-TVBO and similar regret for B2P and UI forgetting. The formatting is as in Figure 5.1.

to a shifted mean is not evident for B2P forgetting with the standard TVBO algorithm. Here, the regret even decreases and thus weakens Hypothesis 1. One possible explanation is that the exploration behavior for $\mu_0 = 0$ was too limited, resulting in the two variations, TV-GP-UCB and SW TV-GP-UCB, exploiting too much and not capturing the change in the objective function sufficiently. This yields in higher regret. The explorative behavior increases again through the optimistic prior in $\mu_0 = -2$, and changes in the objective function can be tracked, thus reducing the regret. This assumption is supported by Figure 5.3.

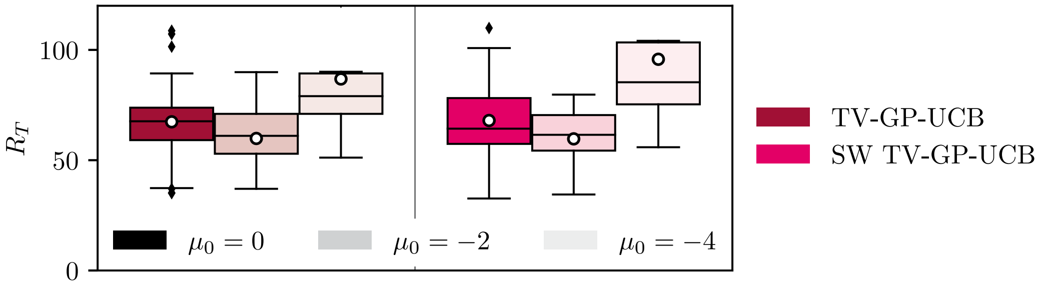

Figure 5.3.: Influence of different optimistic means on B2P forgetting in the two-dimensional within-model comparison.

Here, additional simulations with an optimistic prior mean of $\mu_0 = -4$ were performed, showing the high sensitivity of B2P forgetting regarding the prior mean as in the one-dimensional within-model comparison. This high sensitivity can also complicate the tuning of hyperparameters since the exploration behavior depends not only on the forgetting factor and β_{t+1} but also on the prior mean. If the mean changes, all parameters have to be readjusted to have a desirable exploration-exploitation trade-off.

5.1.2. Out-Of-Model Comparison

For the out-of-model comparison, the same models as in the previous Section 5.1.1 are used, however the length scales are no longer known a-priori. Therefore, at each time step a hyperparameter optimization is performed. A prior on the length scales in form of a Gamma distribution $\mathbf{\Lambda}_{ii} \sim \mathcal{G}(\alpha, \beta)$ as by Marco *et al.* [26] with the probability density function as

$$p(x|\alpha, \beta) = \begin{cases} \frac{\beta^\alpha}{\Gamma(\alpha)} \cdot x^{\alpha-1} \exp\left(-\beta \cdot x\right), & x > 0 \\ 0, & x \leq 0 \end{cases} \tag{5.3}$$

with $\alpha = 11$ and $\beta = \frac{10}{3}$ is chosen. Furthermore, bounds on the length scales are chosen as $\mathbf{\Lambda}_{ii} \in [2, 5]$. The other hyperparameter setting are identical to the within-model comparison in Section 5.1.1. Figure 5.4 shows the results for the one-dimensional objective functions. All variations except SW TV-GP-UCB with an optimistic prior mean outperform exploiting the mean after the initialization.

Furthermore, applying C-TVBO to UI forgetting with the proposed modeling approach UI-TVBO results in the lowest regret. The variations using UI forgetting are more robust to the change in prior mean compared to the variations using B2P forgetting, supporting Hypothesis 1. With a well-defined prior mean, B2P and UI forgetting show similar mean regret when using standard TVBO as stated in Hypothesis 2. However, the main deviation from the within-model comparisons is that B2P forgetting with C-TVBO no longer shows an advantage compared to standard TVBO if no data selection strategy is applied. A reason for this could be that the learned length scales result in a posterior, which is very flat, thus increasing the sampling radius around the optimum in the constrained case.

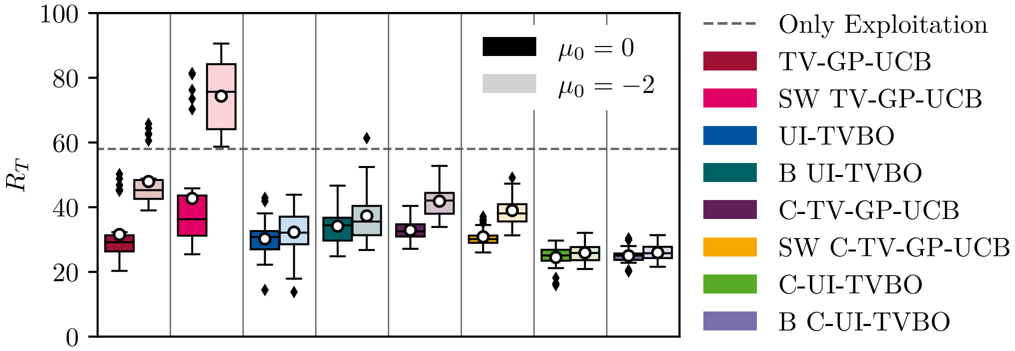

Figure 5.4.: Results for the one-dimensional out-of-model comparison. It shows lower regret and a smaller regret variance for C-TVBO using UI forgetting. The formatting is as in Figure 5.1.

Figure 5.5 shows the mean learned length scales over time. All variations learn length scales, which are greater than the true length scale of the objective function. This is expected, as the bounds on the second derivative of the objective functions where $\partial^2 f_t / \partial x^2 \in [0, 1]$. The objective functions are therefore very flat, and over-estimating the smoothness of the function is likely.

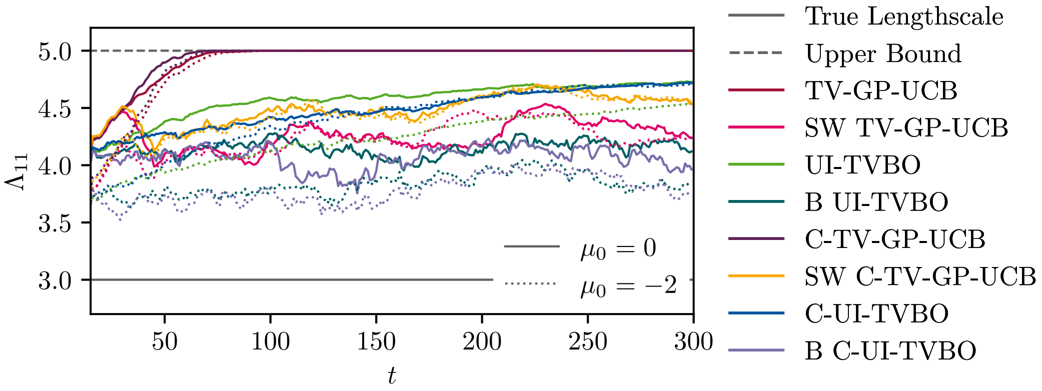

Figure 5.5.: Mean learned length scales of the out-of-model comparison. The dotted lines show the length scales learned with an optimistic prior mean whereas the solid lines show the length scales with a well-defined prior mean.

The length scales of the B2P forgetting variations without data selection strategy quickly reach the upper bound of the length scale, both for C-TVBO and standard

TVBO. This is a direct consequence of the forgetting strategy, as the expectation of a measurement in B2P forgetting propagates to the prior mean over time. This shows that learning the length scale can be more difficult by using B2P forgetting since it tends to overestimate the length scale. For very steep functions, this can result in increased regret. In contrast, the length scale for UI forgetting without data selection strategy only increases gradually.

For both, B2P and UI forgetting, using a data selection strategy results in a better estimate of the length scale as stale data is discarded and not considered in the hyperparameter optimization. However, this does not directly lead to a decrease in regret (Figure 5.4). The upper bound for the length scale could be increased to investigate this further.

The results of the two-dimensional within-model comparisons are shown in Figure 5.6. Again, the combination of both proposed variations C-UI-TVBO is the one with the lowest regret, even though the unconstrained B2P forgetting methods are very similar. Here, the trend that constraining the posterior for B2P forgetting can become problematic if the objective function is flat is even more evident.

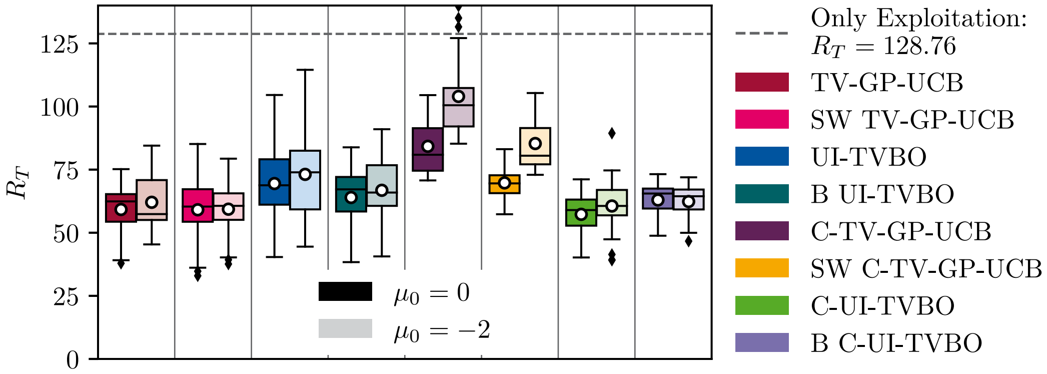

Figure 5.6.: Results for the two-dimensional out-of-model comparison. C-TVBO using B2P forgetting can result in higher regret for flat objective functions. The formatting is as in Figure 5.1.

The assumption of an increased sampling radius due to B2P forgetting and C-TVBO is confirmed when considering the distribution of the queries taken in the simulations in Figure 5.7. It shows that C-TVBO almost always prevents sampling at the bounds, which was one of the main motivations. Especially, it avoids sampling at the corners of the feasible set as observed for the unconstrained variations in the top row. For a convex function with the optimum within the feasible set, sampling at the corners will likely yield in high regret.

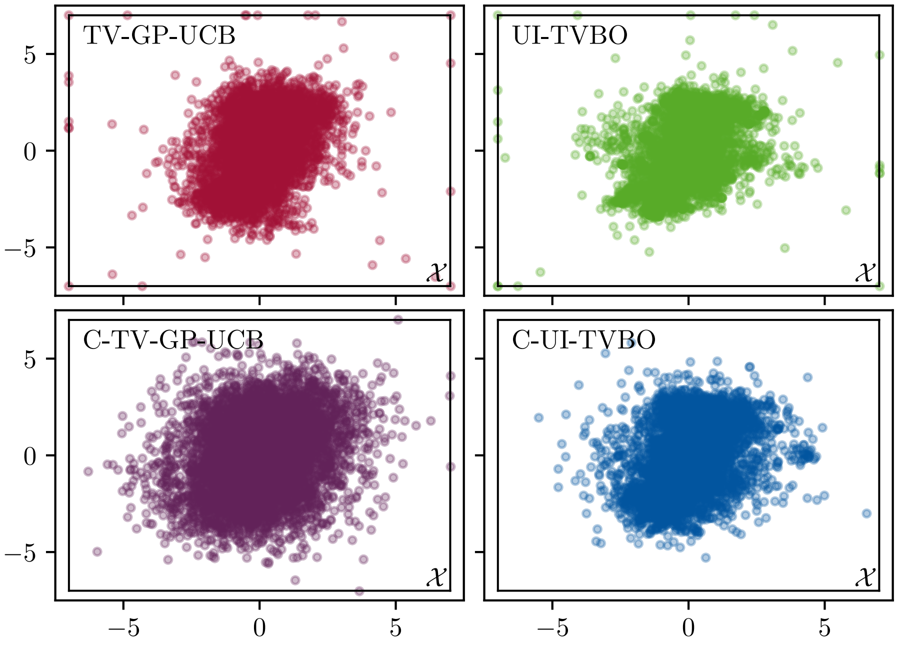

Figure 5.7.: Sample distribution of the two-dimensional out-of-model comparison for the optimistic mean $\mu_0 = -2$. C-TVBO reduces global exploration.

However, for B2P forgetting, constraining the posterior comes at the cost of an increased sampling radius around the predicted optimum. In contrast, unconstrained B2P forgetting samples more often at the boundaries of the feasible set. However, since the objective functions are very flat due to the bounding functions, this behavior does not increase the regret significantly.

In other scenarios, with a larger $b(\mathbf{X}_v)$, this frequent sampling at the bounds can lead to significant increases in the regret. Also, in practical applications, this explorative behavior to the bounds should be limited. Unconstrained UI forgetting also chooses queries at the bounds of the feasible set due to the ever-increasing variance. The constraints of C-TVBO limit the increase in variance and thus this explorative behavior. In contrast to B2P forgetting, the sampling radius is not significantly increased resulting in the low regret shown in Figure 5.6.

5.1.3. 1-D Moving Parabola

The objective functions for the within-model and out-of-model comparison were generated according to the model assumption of temporal change according to a Wiener process and allowed only for a qualitative comparison. For a quantitative comparison between the variations in Table 5.1, benchmarks with one and two dimensions inspired by the test functions in Renganathan *et al.* [37] where designed. They satisfy Assumption 1, and therefore, C-TVBO with convexity constraints can be applied.

The one dimensional benchmark is a moving parabola with the objective function as

$$f_t(x) = \begin{cases} g_{1\mathrm{D}}(x,t)\,, & t < 140 \\ g_{1\mathrm{D}}(x,t=50)\,, & 140 \leq t \leq 225 \\ g_{1\mathrm{D}}(x,t=-50)\,, & t > 225 \end{cases} \tag{5.4}$$

with

$$\begin{aligned} g_{1\mathrm{D}}(x,t) =& a_1(a_2 \cdot x + a_3 + a_4 \cdot t)^2 \\ &+ 2(a_2 \cdot x \sin(a_5 \cdot t)) - \cos(a_5 \cdot t)^2 + b. \end{aligned} \tag{5.5}$$

The coefficients a_1 to a_5 as well as b are displayed in Table 5.2.

Coefficient	a_1	a_2	a_3	a_4	a_5	b
Value	4	0.25	−0.5	−0.01	0.1	5

Table 5.2.: Coefficients for the 1-D moving parabola.

The feasible set for optimizing the acquisition function is defined as $\mathcal{X} = [-5, 9]$ and the time horizon is $T = 300$. The resulting objective function with the trajectory of the optimizer is displayed in Figure 5.8. It consists of a part with gradual change for $t < 140$ and two sudden changes at $t = 140$ and $t = 225$.

The variations in Table 5.1 were evaluated on five different runs with different, but for each variant consistent, initializations of $N = 15$ data points. The initial training data was normalized to zero mean and a standard deviation of one. Therefore, the prior mean was set to $\mu_0 = 0$, and the output variance was fixed to $\sigma_k^2 = 1$. Each subsequent data point was normalized using the mean and standard deviation of the initial data set. For the length scale a hyper prior of $\mathbf{\Lambda}_{11} \sim \mathcal{G}(15, {}^{10}\!/\!{}_{3})$ (5.3) was chosen. Furthermore, an interval for the length scale was set to $\mathbf{\Lambda}_{11} \in [2, 7]$. The bounding functions were defined as $a(\mathbf{X}_v) = 0$ and $b(\mathbf{X}_v) = 4$.

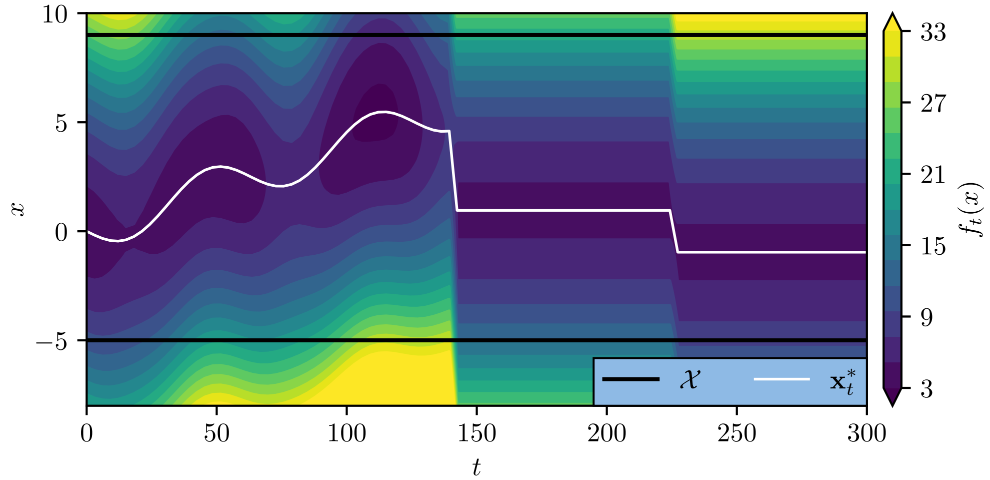

Figure 5.8.: Objective function in (5.4) of the one-dimensional moving parabola.

For the variations without data selection strategy, a sensitivity analysis regarding the forgetting factor was performed and the results are shown in Figure 5.9.

The forgetting factor for B2P forgetting is only defined as $\epsilon \in (0, 1)$, whereas the UI forgetting factor is defined as $\hat{\sigma}_w^2 \in (0, \infty)$. Therefore, the significantly higher regret for B2P forgetting at high forgetting factors compared to UI forgetting is expected. Figure 5.9 shows that for B2P forgetting, constraining the GP posterior by using the proposed method C-TVBO reduces the regret for forgetting factors up to $\epsilon = 0.0215$. However, for higher forgetting factors, constraining the posterior increases the regret. This is a result of B2P forgetting. As the constraints limit the exploration of the acquisition function, queries around the predicted optimum are chosen. However, since the optimum is never chosen directly due to the LCB acquisition function, the expectation of the optimal function value $\mathbb{E}[f_t(\hat{\mathbf{x}}_t^*)]$ propagates towards the prior mean. $\mu_0 > \mathbb{E}[f_t(\hat{\mathbf{x}}_t^*)]$ suggests that the resulting posterior becomes more level over time. Therefore, the sampling radius around the predicted optimum of the acquisition function is increased due to the constant β_{t+1}, resulting in increased regret compared to the unconstrained case as in Section 5.1.2. The exception for $\epsilon = 0.3$ seems to be an outlier, where the objective function is constructed in such a way that the regret with B2P forgetting and C-TVBO is significantly reduced for this specific forgetting factor.

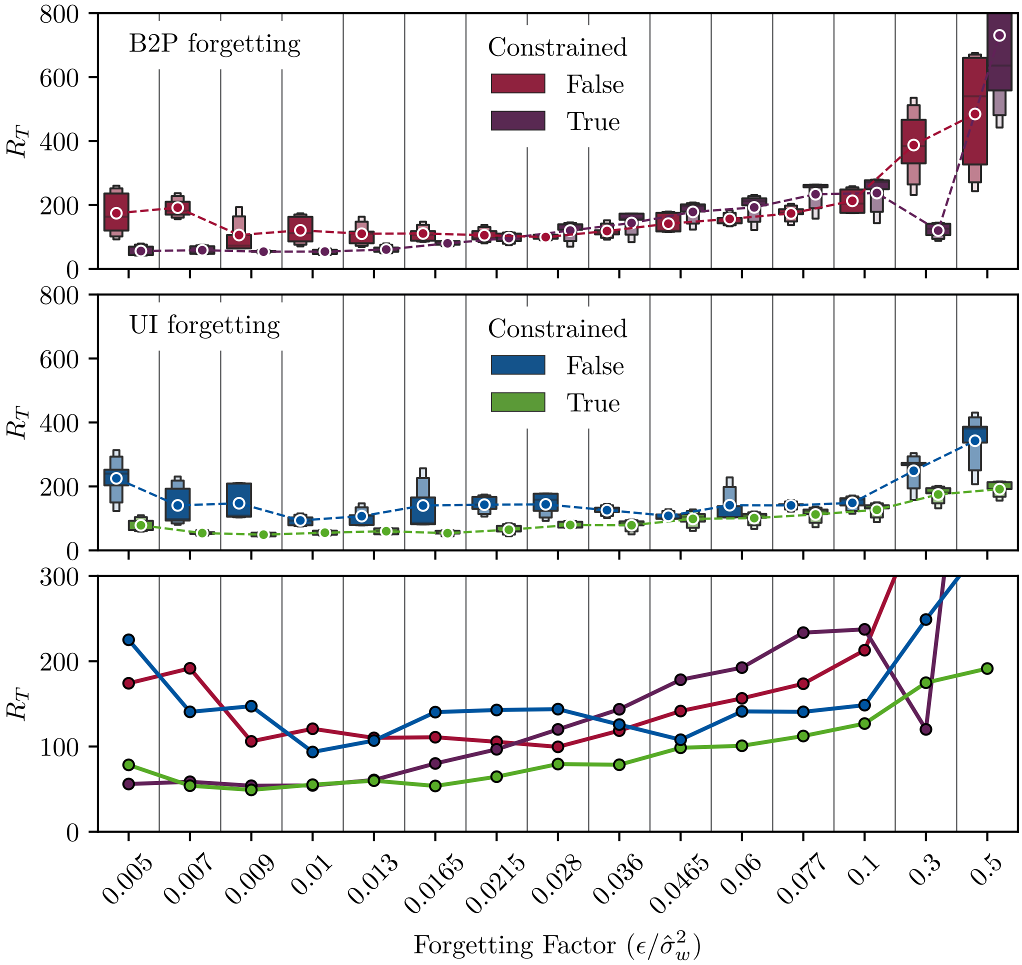

Figure 5.9.: Sensitivity analysis of the UI and B2P forgetting factors on the dynamic cumulative regret with the example of the moving parabola. The means (white markers) are compared in the graph at the bottom. C-TVBO with UI forgetting results in the lowest regret for almost all forgetting factors.

For UI forgetting, applying C-TVBO reduces the regret for all forgetting factors as $\mathbb{E}[f_t(\hat{\mathbf{x}}_t^*]$ does not propagate towards the prior mean supporting Hypothesis 3. Furthermore, C-TVBO reduces the regret's variance. Comparing the mean regret of UI forgetting and B2P forgetting, the bottom graph of Figure 5.9 shows that the combination of the proposed methods results in the lowest regret, expect for $\epsilon = \hat{\sigma}_w^2 = 0.005$ and $\epsilon = \hat{\sigma}_w^2 = 0.3$. Furthermore, the regret of UI forgetting and B2P forgetting at low forgetting factors is very similar, supporting Hypothesis 2.

To further test Hypothesis 1, the forgetting factors with the lowest mean regret in the sensitivity analysis in Figure 5.9 are listed in Table 5.3 and further compared by also considering an optimistic prior mean of $\mu_0 = -1$. These trajectories are shown in Appendix A.5.

Variation	**Forgetting Factor** ϵ / $\hat{\sigma}_w^2$
UI-TVBO, B UI-TVBO	0.01
TV-GP-UCB, SW TV-GP-UCB	0.028
C-UI-TVBO, B C-UI-TVBO	0.009
C-TV-GP-UCB, SW C-TV-GP-UCB	0.009

Table 5.3.: Forgetting factors with the lowest regret in the sensitivity analysis.

For the variations with data selection strategy, the same forgetting factors are chosen as without data selection strategy. The results of the direct comparison the well-defined prior mean $\mu_0 = 0$ and the optimistic prior mean $\mu_0 = -1$ are shown in Figure 5.10. Again, it is apparent that taking into account the temporal change in the objective function is advisable in terms of regret as all variations outperform exploiting the posterior mean after the initialization.

As in previous experiments, B2P forgetting shows a higher sensitivity to the optimistic mean with an increased mean regret supporting Hypothesis 1. The variation with the lowest mean regret independent of the prior mean is the variation combining both proposed methods, UI-TVBO and C-TVBO.

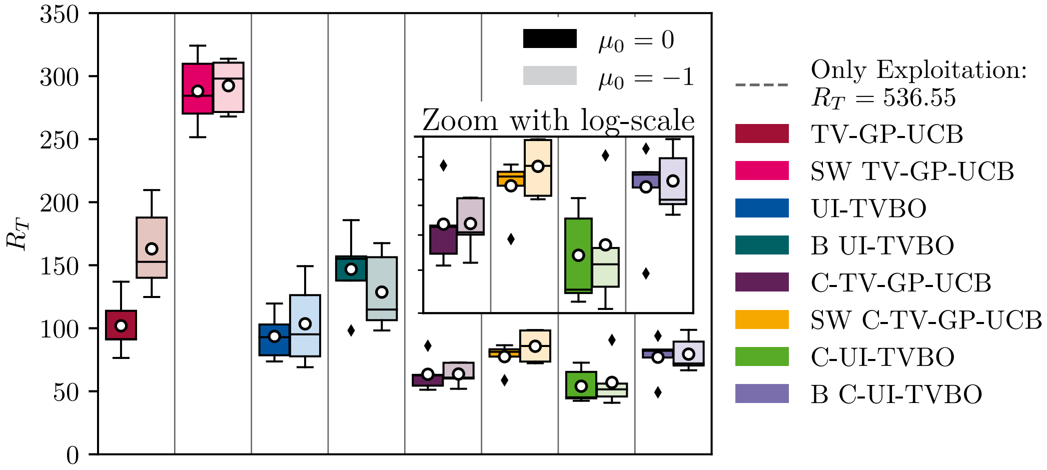

Figure 5.10.: Results of the one-dimensional moving parabola. C-TVBO results in lower regret and UI forgetting shows lower sensitivity to a shifted mean. The formatting is as in Figure 5.1.

5.1.4. 2-D Moving Parabola

Based on similar concepts, the one-dimensional moving parabola is extended to two dimensions. The objective function for the two-dimensional moving parabola is

$$f_t(\mathbf{x} = \{x_1, x_2\}) = \begin{cases} g_{2\text{D}}(\mathbf{x}, t)\,, & t < 140 \\ a_1\left((a_2 \cdot x_1 - 2a_2)^2 + (a_2 \cdot x_2 - 2a_2)^2\right) + b\,, & t \geq 140 \end{cases} \tag{5.6}$$

with

$$\begin{aligned} g_{2\text{D}}(\mathbf{x} = \{x_1, x_2\}, t) =& a_1\left((a_2 \cdot x_1)^2 + (a_2 \cdot x_2 - 0.5\sin(a_5 \cdot t))^2\right) \\ &+ 2(a_2 \cdot x_1 \sin(a_5 \cdot t)) - \cos(a_5 \cdot t)^2 + b. \end{aligned} \tag{5.7}$$

Again, it consists of a part with gradual change for $t < 140$ and a subsequent sudden change at $t = 140$. The coefficients a_1 to a_5 and b are the same as for the one-dimensional moving parabola in Section 5.1.3 and are displayed in Table 5.2. Since the factors of the objective function are identical, the temporal change is very similar. Therefore, the forgetting factors in Table 5.3 are used to evaluate the variations. The feasible set for the two-dimensional moving parabola was $\mathcal{X} = [-7, 7]^2$, Gamma hyperpriors were used for the length scales as $\mathbf{\Lambda}_{11}, \mathbf{\Lambda}_{22} \sim \mathcal{G}(15, {}^{10}\!/_{3})$ (5.3) with bounds $\mathbf{\Lambda}_{11}, \mathbf{\Lambda}_{22} \in [2, 7]$, and the bounding functions were defined as $a(\mathbf{X}_v) = 0$ and $b(\mathbf{X}_v) = 4$. As in the one-dimensional moving parabola benchmark, scaling based on the initial set was applied. The results with firstly a well-defined prior mean and secondly an optimistic prior mean are shown in Figure 5.11.

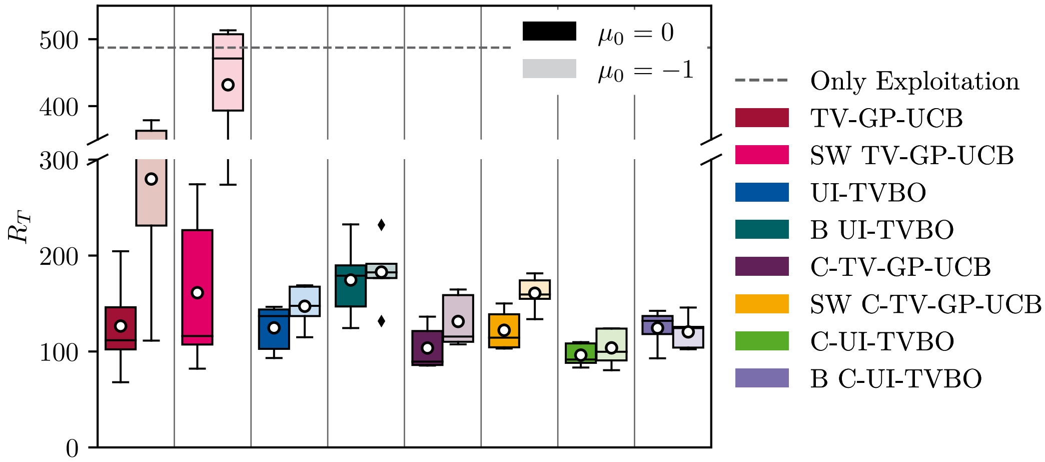

Figure 5.11.: Results of the two-dimensional moving parabola. The white circles represent the mean of each variation. The darker shades show the performance with a well-defined mean, while the lighter shades show the performance with an optimistic mean.

It shows the same trends as in the one-dimensional moving parabola: B2P shows a higher sensitivity to the shifted mean (Hypothesis 1), B2P and UI forgetting have similar performance in terms of regret for with a well-defined prior mean (Hypothesis 2), and C-TVBO reduces the regret as well as its variance compared to standard TVBO (Hypothesis 3). The best performing variation in terms of cumulative regret is again the combination of the proposed methods UI-TVBO and C-TVBO.

However, it is interesting to note that similar to the two-dimensional simulations of within and out-of-model comparison, the differences in regret between standard TVBO and C-TVBO are smaller than in the one-dimensional examples. One reason for this could be the smaller number of VOPs per dimension $N_{v/D}$ chosen for computational feasibility.

5.2. LQR Problem of an Inverted Pendulum

As the last experiment, the variations in Table 5.1 are applied to a real-world problem – the LQR problem of an inverted pendulum. The LQR problem is a classic problem in fundamental control theory controlling a linear dynamical system by minimizing a quadratic cost function J. It therefore satisfies Assumption 1 with the objective function as $f_t \coloneqq J$ making it a suitable application to benchmark the proposed methods. In the following, t denotes the time step of the TVBO algorithm, whereas $\hat{t}$ denotes the time steps of the linear system. For an infinite horizon and discrete time steps the LQR problem is formalized as the optimization problem

$$\min_{\mathbf{u}_{\hat{t}}(\cdot)} J = \lim_{\hat{T}\to\infty} \mathbb{E}\left[\sum_{\hat{t}=0}^{\hat{T}-1} \mathbf{x}_{\hat{t}}^T \mathbf{Q} \mathbf{x}_{\hat{t}} + \mathbf{u}_{\hat{t}}^T \mathbf{R} \mathbf{u}_{\hat{t}}\right] \tag{5.8}$$

$$\text{s.t. } \mathbf{x}_{\hat{t}+1} = \mathbf{A}_k \mathbf{x}_{\hat{t}} + \mathbf{B}_k \mathbf{u}_{\hat{t}} + \mathbf{w}_{\hat{t}} \tag{5.9}$$

with $\mathbf{x}_{\hat{t}} \in \mathrm{I\!R}^D$ as the state, $\mathbf{u}_{\hat{t}} \in \mathrm{I\!R}^P$ as the input, and $\mathbf{w}_{\hat{t}} \sim \mathcal{N}(\mathbf{0}, \bar{\sigma}_n^2 \mathbf{I})$ as iid Gaussian noise at each time step $\hat{t}$. $\mathbf{Q}$ and $\mathbf{R}$ in (5.8) are positive-definite weighting matrices. Furthermore, (5.9) describes a time-discrete linear model with $\mathbf{A}_k$ as the state matrix and $\mathbf{B}_k$ as the input matrix. If the linear model is known, the optimal feedback controller as

$$\mathbf{u}_{\hat{t}} = \mathbf{K}^* \mathbf{x}_{\hat{t}} \tag{5.10}$$

with the optimal controller gain $\mathbf{K}^* \in \mathrm{I\!R}^{P \times D}$ can be calculated by solving the discrete algebraic Ricatti equation

$$\mathbf{P} = \mathbf{A}_k^T \mathbf{P} \mathbf{A} - \mathbf{A}_k^T \mathbf{P} \mathbf{B}_k \left(\mathbf{R} + \mathbf{B}_k^T \mathbf{P} \mathbf{B}_k\right)^{-1} \mathbf{B}_k^T \mathbf{P} \mathbf{A}_k + \mathbf{Q} \tag{5.11}$$

and setting

$$\mathbf{K}^* = -\left(\mathbf{R} + \mathbf{B}_k^T \mathbf{P} \mathbf{B}_k\right)^{-1} \mathbf{B}_k^T \mathbf{P} \mathbf{A}_k. \tag{5.12}$$

As mentioned, the system at consideration is the inverted pendulum. A pendulum is attached to a horizontally moving cart as shown in Figure 5.12.

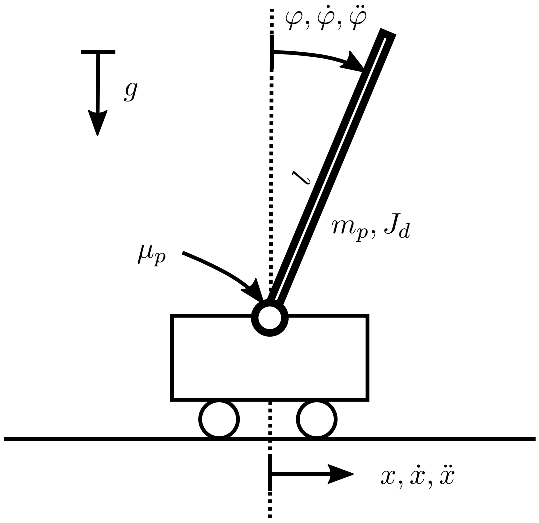

Figure 5.12.: Inverted pendulum with parameters and state variables.

The goal of the LQR problem is to find the optimal controller stabilizing the unstable upper equilibrium point

$$x = 0,\, \dot{x} = 0,\, \varphi = 0,\, \dot{\varphi} = 0 \tag{5.13}$$

and minimizing the costs in (5.8).

Since the pendulum introduces trigonometric functions into the force balance equations, the inverted pendulum is non-linear. The non-linear system equation for the angular acceleration $\ddot{\varphi}$ is

$$\ddot{\varphi} = \frac{1}{2}\frac{m_p g l}{J_d} \cdot \sin(\varphi) - \frac{1}{2}\frac{m_p l}{J_d} \cdot \cos(\varphi) \cdot \ddot{x} - \frac{\mu_p}{J_d} \cdot \dot{\varphi} \tag{5.14}$$

with m_p as the mass, J_d as the moment of inertia, and l as the length of the pendulum. Furthermore, μ_p denotes the fiction in the bearing as shown in Figure 5.12. The control variable is the velocity of the cart as $u \coloneqq \dot{x}_{sp}$ modeled as a first-order lag transfer function resulting in the differential equation for the acceleration as

$$\ddot{x} = \frac{1}{T_1}(K_u \cdot \dot{x}_{sp} - \dot{x}) = \frac{1}{T_1}(K_u \cdot u - \dot{x}). \tag{5.15}$$

Substituting (5.15) into (5.14) yields

$$\ddot{\varphi} = \frac{1}{2}\frac{m_p g l}{J_d} \cdot \sin(\varphi) - \frac{1}{2}\frac{m_p l}{J_d} \cdot \cos(\varphi) \cdot \frac{1}{T_1}(K_u \cdot u - \dot{x}) - \frac{\mu_p}{J_d} \cdot \dot{\varphi}. \tag{5.16}$$

The equations (5.15) and (5.16) build the non-linear state space of the system. As the LQR problem requires a linear model, the non-linear system is linearized around the upper equilibrium points in (5.13) yielding in a linear continuous state space model as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \tag{5.17}$$

with $\mathbf{x} = [x, \dot{x}, \varphi, \dot{\varphi}]^T$ as the state vector. The state matrix and input matrix are

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{1}{T_1} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{1}{2}\frac{m_p l}{J_d T_1} & \frac{1}{2}\frac{m_p l g}{J_d} & -\frac{\mu_p}{J_d} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ \frac{K_u}{T_1} \\ 0 \\ -\frac{1}{2}\frac{m_p l}{J_d}\frac{K_u}{T_1} \end{bmatrix}. \tag{5.18}$$

The continuous state space model in (5.17) is discretized with zero-order hold and a sampling interval of $T_s = 0.02\text{s}$ yielding in a time-invariant discrete state space model as in (5.9). The parameter of the system are shown in Table 5.4.

Param.	m_p	J_d	l	$\mu_p \mid \mu_{p,0}$	K_u	T_1
Value	0.0804kg	$0,5813 \cdot 10^{-3}\text{kg}\,\text{m}^2$	0.147m	$2.2 \cdot 10^{-3}\text{N}\,\text{m}\,\text{s}$	1	1s

Table 5.4.: Parameters of the inverted pendulum.

The algorithms considered in this thesis are designed for time-varying objective functions. Therefore, the friction in the bearing μ_p is assumed to be time-varying as

$$\mu_p(t) = \begin{cases} \mu_{p,0}\,, & t < 50 \\ \mu_{p,0} + \mu_{p,0} \cdot \left(-1.5 \cdot \cos\left(\frac{\pi}{50}(t-50)\right) + 1.5\right)\,, & 50 \leq t \leq 100 \\ 3\mu_{p,0} + \frac{1}{2}\mu_{p,0} \cdot \sin\left(-\frac{\pi}{100}t\right)\,, & t > 100 \end{cases} \tag{5.19}$$

yielding a time-varying LQR cost function J_t with $\mathbf{K}_t^*$ as the optimal controller and J_t^* as the optimal cost at time step t. Figure 5.13 shows the optimal cost over time normalized by the initial optimal cost J_0^* with a standard deviation of the noise in the system as $\bar{\sigma}_n = 6 \cdot 10^{-4}$, a time horizon for the LQR cost function of $\hat{T} = 20$, and the initial conditions as

$$x_0 = 4\mathrm{m},\ \dot{x}_0 = 0\frac{\mathrm{m}}{\mathrm{s}},\ \varphi_0 = 0.1\mathrm{rad},\ \dot{\varphi}_0 = 0.1\frac{\mathrm{rad}}{\mathrm{s}}. \tag{5.20}$$

Furthermore, the dashed line in Figure 5.13 shows the cost over time if the controller gain is kept constant over time as $\mathbf{K}_0^*$, not adjusting to the changing system dynamics. Although the system remains stable, the costs are significantly higher compared to the optimal cost trajectory.

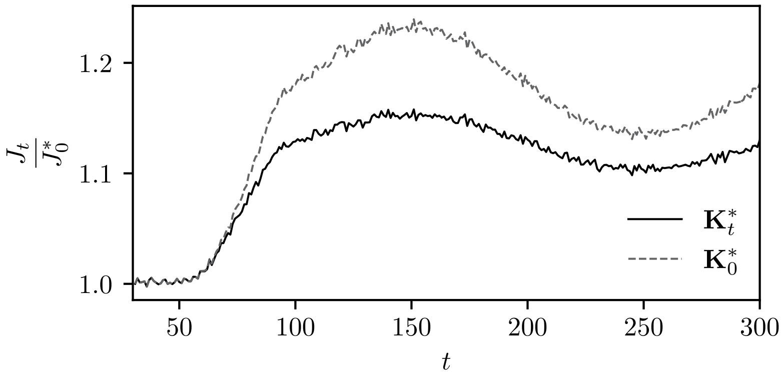

Figure 5.13.: Normalized optimal costs of the LQR problem of an inverted pendulum with the optimal controller at each time step (solid black line) and the optimal controller from the time step $t = 0$ (dashed gray line).

In the following, the variations in Table 5.1 are benchmarked on this time-varying cost function without prior knowledge of the system dynamics and the controller gain $\mathbf{K}$ as the decision variable for the optimization. The controller gain $\mathbf{K}$ of the feedback controller has the dimensions 1×4 as

$$\mathbf{K} = [\theta_1, \theta_2, \theta_3, \theta_4]. \tag{5.21}$$

Since C-TVBO does not scale well in the dimensions (see Figure 2.1), the first two entries are always set to the optimal values θ_1^*, θ_2^* calculated according to (5.12) and the black-box optimization is performed using θ_3 and θ_4 as degrees of freedom. The weighting matrices are set to $\mathbf{Q} = 10 \cdot \mathrm{eye}(4)$ and $\mathbf{R} = 1$. To have accurate feedback about the cost, the simulations are performed using the linearized system, not the non-linear system.

The feasible set is $\mathcal{X} = [-50, -25] \times [-4, -2]$ considering only stable controllers and avoiding numerical issues. Furthermore, the feasible set is scaled using $[3, 1/4]$ to have similar spatial intervals in each dimension. Gamma hyperpriors are used for the length scales as $\mathbf{\Lambda}_{11}, \mathbf{\Lambda}_{22} \sim \mathcal{G}(6, 10/3)$ (5.3) with bounds $\mathbf{\Lambda}_{11}, \mathbf{\Lambda}_{22} \in [0.5, 6]$. As the LQR cost function is flat around the optimum, the bounding functions for C-TVBO are defined as $a(\mathbf{X}_v) = 0$ and $b(\mathbf{X}_v) = 2$. Furthermore, the number of VOPs per dimension is set as $N_{v/D} = 4$ and $\delta = 1.2$ (4.21). As in the moving parabola experiments, the initial training data of $N = 30$ data points is normalized to zero mean and a standard deviation of one. The forgetting factors were chosen as $\hat{\sigma}_w^2 = \epsilon = 0.03$. As the optimal cost J_t^* increases after $t = 50$ (Figure 5.13), the initial prior mean will result in an optimistic prior mean over time. Therefore, it is expected that the variations using the proposed modeling approach with UI forgetting will outperform the variations using B2P forgetting as stated in Hypothesis 1.

Figure 5.14 shows the results of five different but for each variant consistent initializations. All variations outperform the initial optimal controller $\mathbf{K}_0^*$.

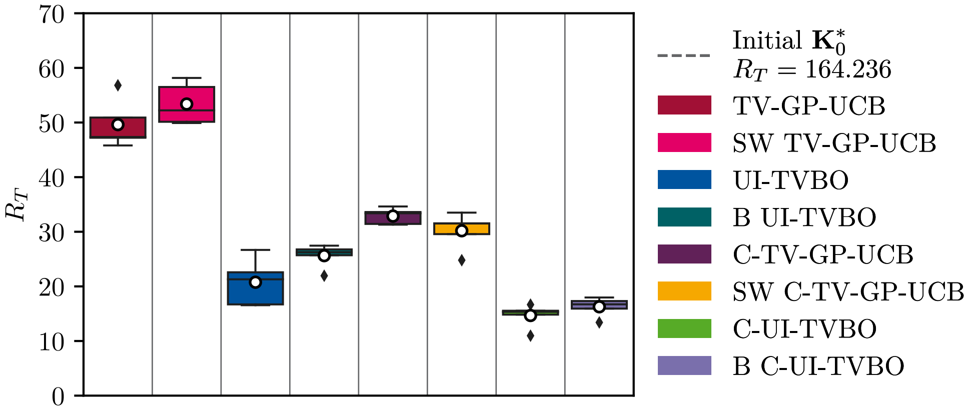

Figure 5.14.: Results of the LQR problem of an inverted pendulum. UI forgetting is less sensitive to the increase in cost over time. C-TVBO further reduces the regret as well as its variance.

Furthermore, using the proposed method C-TVBO reduces the mean regret compared to standard TVBO (Hypothesis 3). As expected, the variations using B2P forgetting are more sensitive to the increase in cost over time compared to UI forgetting resulting in a higher regret. To further investigate the exploration behavior of the variations, the regret is split into an exploration regret and an exploitation regret as

$$R_T = \sum_{t=1}^{T} \left(f_t(\mathbf{x}_t) - f_t(\mathbf{x}_t^*)\right) = \underbrace{\sum_{t=1}^{T} \left(f_t(\mathbf{x}_t) - f_t(\hat{\mathbf{x}}_t)\right)}_{:=\hat{R}_T \text{ (Exploration)}} + \underbrace{\sum_{t=1}^{T} \left(f_t(\hat{\mathbf{x}}_t) - f_t(\mathbf{x}_t^*)\right)}_{:=R_T^* \text{ (Exploitation)}} . \quad (5.22)$$

The exploitation regret R_T^* represents the cost of the predicted optimum $\hat{\mathbf{x}}_t$ deviating from the true optimum $\mathbf{x}_t^*$. In contrast, the exploration regret $\hat{R}_T$ captures the cost of choosing a query deviating from the predicted optimum. The split regret for the LQR problem is displayed in Figure 5.15. It shows that the constrained models deviate more from the optimum initially than is the case for the unconstrained models. A reason for this is that the objective function is very flat in the beginning. However, after the increase in cost, it becomes apparent that regardless of the algorithm, the optimum can be tracked equally well in comparison since the lines of R_T^* of unconstrained and constrained variation run roughly parallel for $t > 100$.

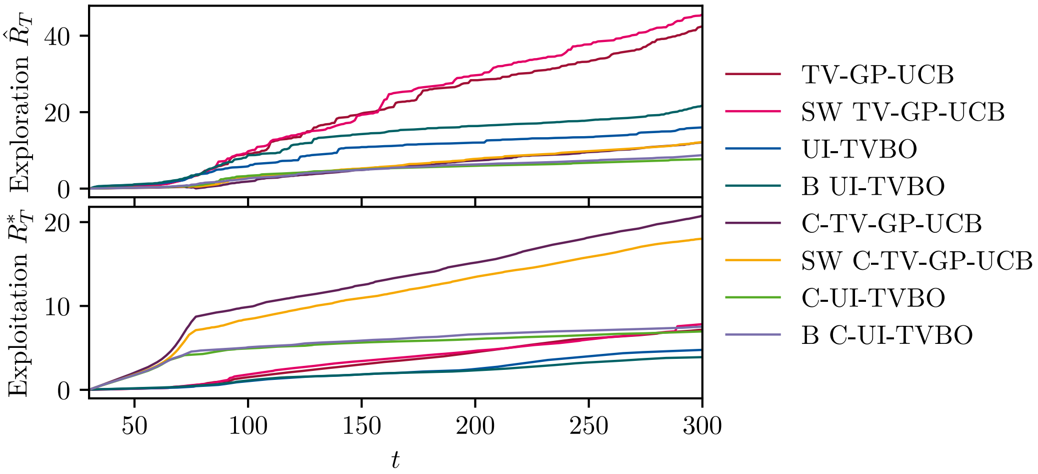

Figure 5.15.: Exploration and exploitation regret of the LQR problem of an inverted pendulum. The regret is split according to (5.22).

The upper subplot in Figure 5.15, however, shows that the cost caused by exploration is significantly higher for the unconstrained variations as the constraints in C-TVBO limit the exploration as discussed in Section 4.2. Especially for the variations that use B2P forgetting, the cost caused by excessive exploration is high compared to UI forgetting. Furthermore, the tracking of the optimum is better with UI forgetting, despite the lower exploration cost.

5.3. Discussion

Different synthetic experiments (Section 5.1) and an application example (Section 5.2) were conducted to compare the variations, evaluate the proposed methods UI-TVBO and C-TVBO presented in Chapter 4, and test Hypotheses 1 though 3. The hypotheses are individually revisited and discussed, and afterwards, further particularities that emerged in the results are discussed.

Regarding Hypothesis 1

Hypothesis 1 stated that B2P forgetting would be more sensitive to an optimistic prior mean compared to UI forgetting, which would be reflected in higher regret. All synthetic experiments confirmed the hypothesis. Only the two-dimensional within- and out-of-model comparisons could not directly confirm the hypothesis. However, further investigations of the within-model comparison also showed higher sensitivity. Furthermore, it could be concluded that a pessimistic prior mean can also lead to an increased regret since it can restrict the exploratory behavior of B2P forgetting too much and thus aggravating the learning of the temporal change. The LQR application example showed that B2P forgetting was very sensitive to the increase in cost over time reflected in increased exploration behavior as shown in Figure 5.15.

Regarding Hypothesis 2

Hypothesis 2 stated, that given a well-defined prior mean, the regret of B2P forgetting and UI forgetting is comparable. This was also confirmed by the synthetic experiments of the moving parabola (Sections 5.1.3 and 5.1.4), which were designed for a quantitative comparison. Here, the optimal forgetting factors for B2P and UI forgetting were selected on the basis of a sensitivity analysis and both, B2P and UI forgetting, showed very similar performance in terms of regret for optimal forgetting parameters.

Regarding Hypothesis 3

The final hypothesis, Hypothesis 3, stated that C-TVBO would result in lower regret compared to standard TVBO, regardless of the forgetting strategy, because it incorporates prior knowledge. This could not be confirmed, since for very flat objective functions the combination of B2P forgetting and C-TVBO resulted in higher regret compared to standard TVBO. The constraints and the flat posterior due to B2P forgetting increased the sampling radius around the optimum such that the regret was higher than the one caused by occasional sampling at the bounds by unconstrained B2P forgetting. However, for UI forgetting it was demonstrated that the use of C-TVBO always results in lower regret compared to standard TVBO. In contrast to B2P forgetting, the posterior around the expected optimum does not propagate to the prior mean as described in Section 5.1.3. Thus, the sampling radius is not significantly increased.

Further Observations

It could be observed that the differences between standard TVBO and C-TVBO were smaller in the two-dimensional examples than in the one-dimensional examples. One reason for this may be the smaller number of VOPs per dimension $N_{v/D}$, which enforce convexity. Further investigations would be necessary, e.g. reducing $N_{v/D}$ in the one-dimensional examples, in order to be able to confirm this assumption. Nevertheless, this also directly points out a deficiency of C-TVBO. If the sensitivity to $N_{v/D}$ is very high, methods other than proposed by Agrell [1] are needed to enforce the convexity of the posterior. Also, to scale C-TVBO to higher dimensions, other approaches would be needed.

While the main focus of the comparisons was on the forgetting strategies, the data selection strategies were also used for the individual variations. Here, it was shown that the size of a sliding window of $W = 30$ is too small for B2P forgetting in the one-dimensional experiments. This was expected since it was not chosen according to (4.23) but to have a comparable number of training points as the binning approach. In the two-dimensional experiments, a sliding window size of $W = 80$ already showed to be competitive. The chosen number of bins per dimension of 20 for UI forgetting turned out to be sufficient since, especially for C-TVBO, the regret was very similar to that of UI forgetting without data selection strategy.

Lastly, it should be noted that learning hyperparameters in TVBO is difficult because the samples are strongly correlated through the acquisition function and not iid, which is required for maximum likelihood approaches. Thus, bounds for the length scale had to be used to avoid length scales that were either too small or too large. Furthermore, the noise was fixed to capture the temporal change and not disregard them as noise. The out-of-model comparisons in Section 5.1.2 also showed that learning hyperparameters is even more challenging if B2P forgetting is used due to the propagation of the expected value to the prior mean. The discussed data selection strategies improved learning the hyperparameters since they reduced the data set and neglected irrelevant data.

6. Conclusion and Outlook

In this thesis, modeling the temporal dimension in TVBO was categorized into B2P and UI forgetting. Previous methods for GP-based TVBO model have exclusively used B2P forgetting. In B2P forgetting, the information about previous measurements is lost over time. In contrast, a modeling approach for TVBO using UI forgetting based on a Wiener process was proposed – UI-TVBO. The performance of UI-TVBO with a well-defined prior mean is similar to the performance of the state-of-the-art modeling approach of B2P forgetting. However, UI-TVBO shows significantly higher robustness for non-mean reverting objective functions, which was demonstrated in an application example, the LQR problem of an inverted pendulum, and other synthetic experiments. To limit the increasing variance of UI-TVBO and additionally incorporate prior knowledge about the shape of the objective function into TVBO, the method C-TVBO was developed and proposed. C-TVBO constrains the posterior of the GP at each time step using VOPs, preventing undesirable exploration. It showed for almost all simulations an improvement regarding the regret and the tendency to a reduced variance. Only the combination of B2P forgetting and C-TVBO with very flat objective functions showed an increase in regret compared to standard TVBO due to the forgetting. This was not observed for methods using UI forgetting and the combination of both proposed methods, UI-TVBO and C-TVBO, showed the lowest regret outperforming the current state-of-the-art method.

While working on this thesis, some ideas have emerged on how to further develop the proposed concepts, as well as other challenges that arise in TVBO that would be interesting future research directions. It would be interesting to use iterative Kalman filter updates similar to Carron *et al.* [12] instead of modeling the temporal dimension with the GP for TVBO to design other implementations of UI forgetting. This would also require different data selection strategies. Here, the use of approaches as in Titsias [46] would be interesting, using inducing points, which only approximate the posterior at the current time step.

To make C-TVBO applicable to higher dimensions, new methods not based on VOPs for constraining the posterior are needed. An interesting basis for such a method for GPs could be Aubin-Frankowski and Szabo [2], which does not require VOPs and guarantees satisfying shape constraints not only at finite points but uniformly.

Furthermore, research on novel acquisition functions for TVBO is necessary. On the one hand, non-myopic acquisition functions would be conceivable as in Renganathan *et al.* [37]. On the other hand, an online optimization of the exploration-exploitation trade-off parameter in UCB based on the change in the objective function would be an interesting research direction. This change could be estimated online based on the difference between the expected measured value at the query location and the actual measured value.

A. Appendix

A.1. Numerically Stable Calculation of the Constrained Gaussian Process Posterior Distribution

A more stable calculation of the factors A_i, B_i and Σ is provided in [1, Lemma 2] using Cholesky factorization instead of calculating inverses. In the following, $\mathrm{chol}(P)$ is defined as the lower triangular Cholesky factor of P and $X = (A \setminus B)$ as the solution to the linear system $AX = B$, which can be computed very efficiently if A is rectangular.

Following now [1, Lemma 2], let $L = \mathrm{chol}(K_{\mathbf{X},\mathbf{X}} + \sigma_v^2 \mathbf{I})$, $v_1 = L \setminus \mathcal{L} K_{\mathbf{X}_v,\mathbf{X}}$ and $v_2 = L \setminus K_{\mathbf{X},\mathbf{X}_*}$. Then the factors (2.16) to (2.20) can be computed as

$$A_1 = \left(L^T \setminus v_1\right)^T \tag{A.1}$$

$$A_2 = \left(L^T \setminus v_2\right)^T \tag{A.2}$$

$$B_1 = \mathcal{L} K_{\mathbf{X}_v,\mathbf{X}_v} \mathcal{L}^T + \sigma_v^2 \mathbf{I} - v_1^T v_1 \tag{A.3}$$

$$B_2 = K_{\mathbf{X}_*,\mathbf{X}_*} - v_2^T v_2 \tag{A.4}$$

$$B_3 = K_{\mathbf{X}_*,\mathbf{X}_v} \mathcal{L}^T - v_2^T v_1. \tag{A.5}$$

Let $L_1 = \mathrm{chol}(B_1)$ and let $v_3 = L_1^T \setminus B_3^T$, than the final factors for the posterior distribution (2.26) can be computed as

$$A = \left(L_1^T \setminus v_3\right)^T, \quad B = A_2 - AA_1, \quad \Sigma = B_2 - v_3^T v_3. \tag{A.6}$$

For the derivation and proof it is referred to [1, Appendix B].

A.2. Sampling from the Constrained Gaussian Process Prior Distribution

Sampling from the GP prior distribution is similar to sampling from the posterior distribution as described in section 2.1.2. To sample from the constrained prior, the joint distribution of $\mathbf{y}$, $\mathbf{f}_*$ and $\tilde{C}$ in (2.14) has to be first conditioned on $\tilde{C}$. Afterwards, the prior distribution $\mathbf{f}_* \sim \mathcal{N}(\mu_{\mathbf{X}_*} + \hat{A}(\hat{\mathbf{C}} - \mathcal{L}\mu_{\mathbf{X}_v}), \hat{\Sigma})$ can be obtained through marginalizing out $\mathbf{f}_*$ from the conditioned joint distribution. The resulting multivariate normal distribution is compound Gaussian distribution with a truncated mean with the following factors

$$\begin{aligned} \hat{B}_1 &= \mathcal{L}K_{\mathbf{X}_v,\mathbf{X}_v}\mathcal{L}^T + \sigma_v^2\mathbf{I} & \hat{v}_3 &= \hat{L}_1^T \setminus \hat{B}_3^T \\ \hat{B}_2 &= K_{\mathbf{X}_*,\mathbf{X}_*} & \hat{A} &= \left(\hat{L}_1^T \setminus \hat{v}_3\right)^T \\ \hat{B}_3 &= K_{\mathbf{X}_*,\mathbf{X}_v}\mathcal{L}^T & \hat{\Sigma} &= \hat{B}_2 - \hat{v}_3^T\hat{v}_3, \\ \hat{L}_1 &= \text{chol}(\hat{B}_1) \end{aligned} \tag{A.7}$$

and the truncated multivariate normal distribution

$$\hat{\mathbf{C}} = \hat{C}|C \sim \mathcal{TN}\left(\mathbf{0}, \hat{B}_1, a(\mathbf{X}_v), b(\mathbf{X}_v)\right). \tag{A.8}$$

The algorithm for sampling from the prior is displayed in Algorithm 7 below.

Algorithm 7 Sampling form the constrained prior distribution

```
Initialize: Calculate factors $\hat{A}$, $\hat{\Sigma}$, $\hat{B}_1$
Find a matrix $\mathbf{Q}$ s.t. $\mathbf{Q}^T\mathbf{Q} = \Sigma \in \mathbb{R}^{M \times M}$ using Cholesky decomposition.
Generate $\hat{\mathbf{C}}_k$, a $N_v \times k$ matrix where each column is a sample of $\hat{C}|C$ from the
    truncated multivariate normal distribution (A.8).
Generate $\mathbf{U}_k$, a $M \times k$ matrix with k samples of the multivariate standard normal
    distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_M)$ with $\mathbf{I}_M \in \mathbb{R}^{M \times M}$.
Calculate the $M \times k$ matrix where each column is a sample from the distribution
    $\mathbf{f}_*|C$ as
        $\mu_{\mathbf{X}_*} \oplus \left[A(-\mathcal{L}\mu_{\mathbf{X}_v} \oplus \tilde{\mathbf{C}}_k) + \mathbf{Q}\mathbf{U}_k\right]$    (A.9)
    with $\oplus$ representing the operation of adding the $M \times 1$ vector on the left hand
    side to each column of the $M \times k$ matrix on the right hand side.
```

A.3. Derivatives of the Squared-Exponential Kernel

To constrain the GP posterior, the partial derivatives of the spatial kernel are needed. Following are the partial derivatives of the SE kernel

$$k(\mathbf{x}, \mathbf{x}') = \sigma_k^2 \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{x}')^T \boldsymbol{\Lambda}^{-1}(\mathbf{x}-\mathbf{x}')\right), \quad \boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\Lambda}_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \boldsymbol{\Lambda}_{DD} \end{bmatrix}. \tag{A.10}$$

Second derivative w.r.t. x'_j:

$$\frac{\partial^2 k(\mathbf{x}, \mathbf{x}')}{\partial x'^2_j} = \Lambda_{jj}^{-1}\Big(\hat{\mathrm{d}}_j^2(\mathbf{x}, \mathbf{x}') - 1\Big)k(\mathbf{x}, \mathbf{x}') \tag{A.11}$$

Second derivative w.r.t. x_j and x'_j (diagonal elements of $\mathcal{L}K_{*,*}\mathcal{L}^T$):

$$\frac{\partial^4 k(\mathbf{x}, \mathbf{x}')}{\partial x_j^2 \partial x'^2_j} = \Lambda_{jj}^{-2}\Big(\hat{\mathrm{d}}_j^2(\mathbf{x}, \mathbf{x}')\,\hat{\mathrm{d}}_j^2(\mathbf{x}, \mathbf{x}') - 6\,\hat{\mathrm{d}}_j^2(\mathbf{x}, \mathbf{x}') + 3\Big)k(\mathbf{x}, \mathbf{x}') \tag{A.12}$$

Second derivative w.r.t. x_i and x'_j (off-diagonal elements of $\mathcal{L}K_{*,*}\mathcal{L}^T$):

$$\frac{\partial^4 k(\mathbf{x}, \mathbf{x}')}{\partial x_i^2 \partial x'^2_j} = \Lambda_{ii}^{-1}\Lambda_{jj}^{-1}\Big(\hat{\mathrm{d}}_i^2(\mathbf{x}, \mathbf{x}')\,\hat{\mathrm{d}}_j^2(\mathbf{x}, \mathbf{x}') - \hat{\mathrm{d}}_i^2(\mathbf{x}, \mathbf{x}') - \hat{\mathrm{d}}_j^2(\mathbf{x}, \mathbf{x}') + 1\Big)k(\mathbf{x}, \mathbf{x}') \tag{A.13}$$

with the squared distance in dimension k normalized by the corresponding length-scale $\hat{\mathrm{d}}_k^2(\mathbf{x}, \mathbf{x}') = \Lambda_{kk}^{-1}(x_k - x'_k)^2$.

A.4. Correlation between Forgetting Factors

The forgetting factors of B2P forgetting as in $k_{T,tv}$ and UI forgetting as in $k_{T,wp}$ both imply the variance for $\tau = 0$ after one time step after observing a measurement. This is shown below for one training point x at time step t_1 and a test points x_* at time step t_2 with $\tau = x - x_* = 0$.

Back-2-Prior Forgetting

Posterior covariance using the temporal kernel $k_{T,tv}$, $\tau = 0$, $t_2 > t_1$, and $\Delta t = 1$:

$$\sigma_k^2 \cdot (1-\epsilon)^{\frac{|t_2-t_2|}{2}} - \sigma_k^2 \cdot (1-\epsilon)^{\frac{|t_2-t_1|}{2}} \left[\sigma_k^2 \cdot (1-\epsilon)^{\frac{|t_1-t_1|}{2}}\right]^{-1} \sigma_k^2 \cdot (1-\epsilon)^{\frac{|t_1-t_2|}{2}} \tag{A.14}$$

$$= \sigma_k^2 - \sigma_k^2 \cdot (1-\epsilon)^{\frac{|t_2-t_1|}{2}} (1-\epsilon)^{\frac{|t_1-t_2|}{2}} \tag{A.15}$$

$$= \sigma_k^2 - \sigma_k^2 \cdot (1-\epsilon)^{|t_2-t_1|} \tag{A.16}$$

$$= \sigma_k^2 \cdot \epsilon \quad \text{with } \Delta t = 1 \tag{A.17}$$

For $\sigma_k^2 = 1$ it can bee seen, that the variance after one time step is ϵ.

Uncertainty-Injection Forgetting

Posterior covariance using the temporal kernel $k_{T,wp}$, $\tau = 0$, $t_2 > t_1$, and $\Delta t = 1$::

$$\sigma_k^2 \cdot \sigma_w^2(\min(t_2,t_2) - c_0) - \frac{\sigma_k^2 \cdot \sigma_w^2(\min(t_2,t_1) - c_0) \cdot \sigma_k^2 \cdot \sigma_w^2(\min(t_2,t_1) - c_0)}{\sigma_k^2 \cdot \sigma_w^2(\min(t_1,t_1) - c_0)} \tag{A.18}$$

$$= \sigma_k^2 \cdot \sigma_w^2(\min(t_2,t_2) - c_0) - \sigma_k^2 \cdot \sigma_w^2(\min(t_2,t_1) - c_0) \tag{A.19}$$

$$= \sigma_k^2 \cdot \sigma_w^2(t_2 - c_0) - \sigma_k^2 \cdot \sigma_w^2(t_1 - c_0) \tag{A.20}$$

$$= \sigma_k^2 \cdot \sigma_w^2(t_2 - t_1) \tag{A.21}$$

$$= \sigma_k^2 \cdot \sigma_w^2 = \hat{\sigma}_w^2 \quad \text{with } \Delta t = 1 \tag{A.22}$$

For $\sigma_k^2 = 1$ it can bee seen, that the variance after one time step is $\sigma_w^2 = \hat{\sigma}_w^2$.

A.5. Trajectories of the 1-D Moving Parabola

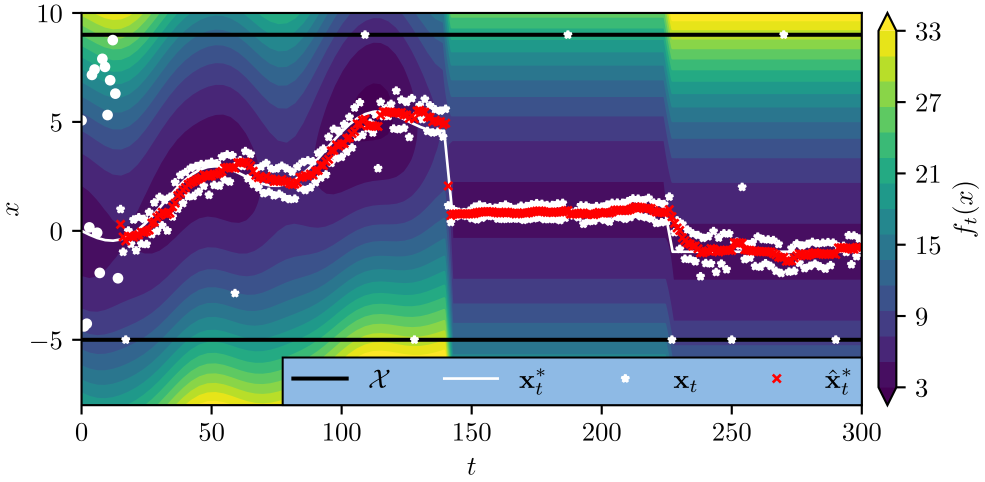

Figure A.1.: Trajectory of unconstrained B2P forgetting ($\epsilon = 0.028$) for the one-dimensional moving parabola. The white circles denote the initial training data.

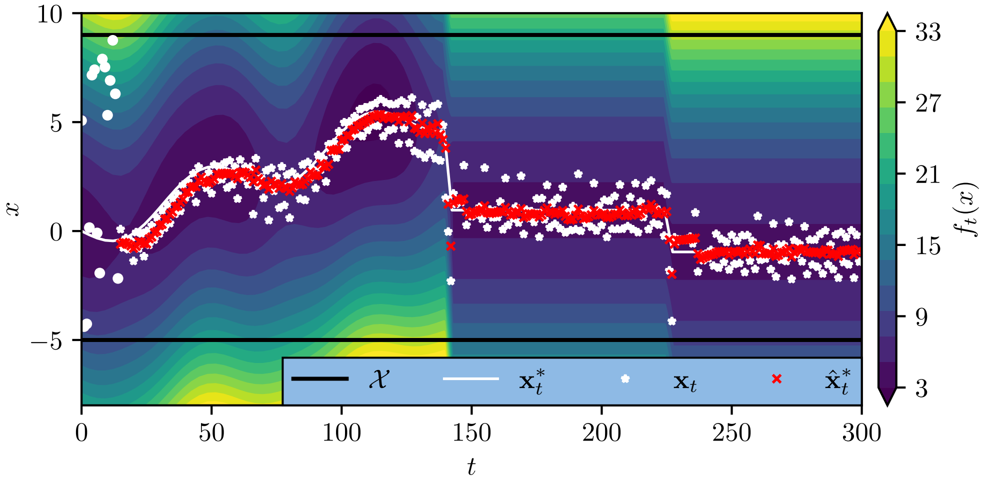

Figure A.2.: Trajectory of constrained B2P forgetting ($\epsilon = 0.009$) for the one-dimensional moving parabola. The white circles denote the initial training data.

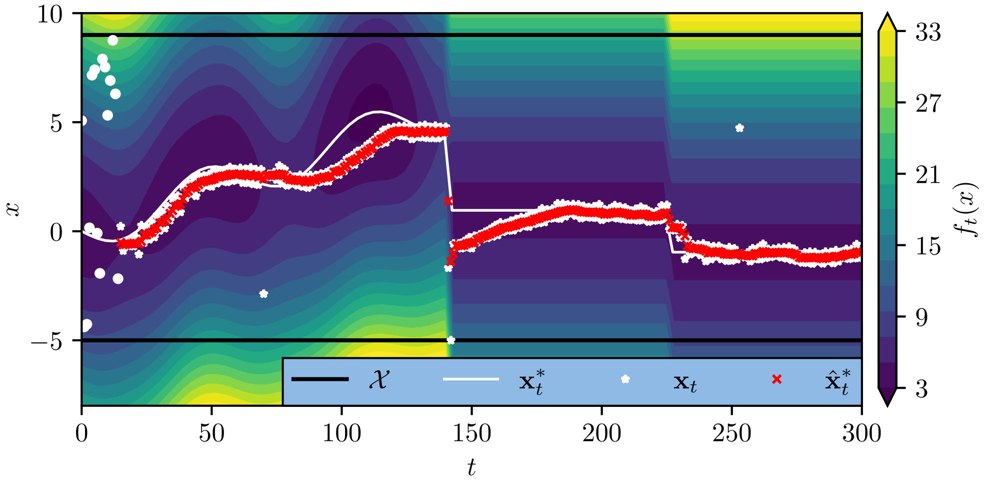

Figure A.3.: Trajectory of unconstrained UI forgetting ($\hat{\sigma}_w^2 = 0.01$) for the one-dimensional moving parabola. The white circles denote the initial training data.

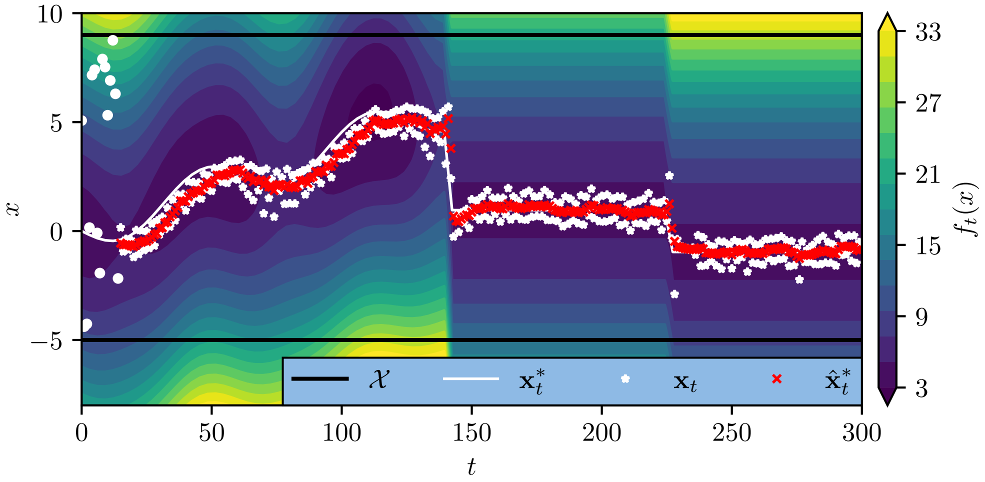

Figure A.4.: Trajectory of constrained UI forgetting ($\hat{\sigma}_w^2 = 0.009$) for the one-dimensional moving parabola. The white circles denote the initial training data.

B. Hyperparameters

Within-Model Comparison (1D)

Hyperparameter	**Value**
UI forgetting factor $\hat{\sigma}_w^2$	0.03
B2P forgetting factor ϵ	0.03
Feasible set $\mathcal{X}$	$[-5, 9]$
Number of VOPs per dimension $N_{v/D}$	10
Local approximation factor δ	1.5
Bounding functions $a(\mathbf{X}_v), b(\mathbf{X}_v)$	$a(\mathbf{X}_v) = 0, b(\mathbf{X}_v) = 1$
Output variance σ_k^2	1
Length scale $\mathbf{\Lambda}_{11}$	3
Noise σ_n^2	0.02
Number of bins per dimensions	20
Sliding window size W	30

Table B.1.: Hyperparameters of the one-dimensional within-model comparison.

Within-Model Comparison (2D)

Hyperparameter	**Value**
UI forgetting factor $\hat{\sigma}_w^2$	0.03
B2P forgetting factor ϵ	0.03
Feasible set $\mathcal{X}$	$[-7, 7]^2$
Number of VOPs per dimension $N_{v/D}$	5
Local approximation factor δ	1.5
Bounding functions $a(\mathbf{X}_v), b(\mathbf{X}_v)$	$a(\mathbf{X}_v) = 0, b(\mathbf{X}_v) = 1$
Output variance σ_k^2	1
Length scales $\mathbf{\Lambda}_{11}, \mathbf{\Lambda}_{22}$	3
Noise σ_n^2	0.02
Number of bins per dimensions	20
Sliding window size W	80

Table B.2.: Hyperparameters of the two-dimensional within-model comparison.

Out-Of-Model Comparison (1D)

Hyperparameter	**Value**
UI forgetting factor $\hat{\sigma}_w^2$	0.03
B2P forgetting factor ϵ	0.03
Feasible set $\mathcal{X}$	$[-5, 9]$
Number of VOPs per dimension $N_{v/D}$	10
Local approximation factor δ	1.5
Bounding functions $a(\mathbf{X}_v), b(\mathbf{X}_v)$	$a(\mathbf{X}_v) = 0, b(\mathbf{X}_v) = 1$
Output variance σ_k^2	1
Length scale $\mathbf{\Lambda}_{11}$	$\mathcal{G}(11, 10/3)$, bounds $[2, 5]$
Noise σ_n^2	0.02
Number of bins per dimensions	20
Sliding window size W	30

Table B.3.: Hyperparameters of the one-dimensional out-of-model comparison.

Out-Of-Model Comparison (2D)

Hyperparameter	**Value**
UI forgetting factor $\hat{\sigma}_w^2$	0.03
B2P forgetting factor ϵ	0.03
Feasible set $\mathcal{X}$	$[-7, 7]^2$
Number of VOPs per dimension $N_{v/D}$	5
Local approximation factor δ	1.5
Bounding functions $a(\mathbf{X}_v), b(\mathbf{X}_v)$	$a(\mathbf{X}_v) = 0, b(\mathbf{X}_v) = 1$
Output variance σ_k^2	1
Length scales $\mathbf{\Lambda}_{11}, \mathbf{\Lambda}_{22}$	$\mathcal{G}(11, 10/3)$, bounds $[2, 5]$
Noise σ_n^2	0.02
Number of bins per dimensions	20
Sliding window size W	80

Table B.4.: Hyperparameters of the two-dimensional out-of-model comparison.

1-D Moving Parabola

Hyperparameter	**Value**
UI forgetting factor $\hat{\sigma}_w^2$	varying
B2P forgetting factor ϵ	varying
Feasible set $\mathcal{X}$	$[-5, 9]$
Number of VOPs per dimension $N_{v/D}$	10
Local approximation factor δ	1.5
Bounding functions $a(\mathbf{X}_v), b(\mathbf{X}_v)$	$a(\mathbf{X}_v) = 0, b(\mathbf{X}_v) = 4$
Output variance σ_k^2	1
Length scale $\mathbf{\Lambda}_{11}$	$\mathcal{G}(15, {}^{10}\!/_{3})$, bounds $[2, 7]$
Noise σ_n^2	0.02
Number of bins per dimensions	20
Sliding window size W	30

Table B.5.: Hyperparameters of the one-dimensional moving parabola.

2-D Moving Parabola

Hyperparameter	**Value**
UI forgetting factor $\hat{\sigma}_w^2$	C-TVBO: 0.009, standard: 0.01
B2P forgetting factor ϵ	C-TVBO: 0.009, standard: 0.028
Feasible set $\mathcal{X}$	$[-7, 7]^2$
Number of VOPs per dimension $N_{v/D}$	5
Local approximation factor δ	1.5
Bounding functions $a(\mathbf{X}_v), b(\mathbf{X}_v)$	$a(\mathbf{X}_v) = 0, b(\mathbf{X}_v) = 4$
Output variance σ_k^2	1
Length scales $\mathbf{\Lambda}_{11}, \mathbf{\Lambda}_{22}$	$\mathcal{G}(15, {}^{10}\!/_{3})$, bounds $[2, 7]$
Noise σ_n^2	0.02
Number of bins per dimensions	20
Sliding window size W	80

Table B.6.: Hyperparameters of the two-dimensional moving parabola.

LQR Problem

Hyperparameter	**Value**
UI forgetting factor $\hat{\sigma}_w^2$	0.03
B2P forgetting factor ϵ	0.03
Feasible set $\mathcal{X}$	$[-50, -25] \times [-4, -2]$
Scaling	$[3, 1/4]$
Number of VOPs per dimension $N_{v/D}$	4
Local approximation factor δ	1.2
Bounding functions $a(\mathbf{X}_v), b(\mathbf{X}_v)$	$a(\mathbf{X}_v) = 0, b(\mathbf{X}_v) = 2$
Output variance σ_k^2	1
Length scales $\mathbf{\Lambda}_{11}, \mathbf{\Lambda}_{22}$	$\mathcal{G}(6, 10/3)$, bounds $[0.5, 6]$
Noise σ_n^2	0.02
Number of bins per dimensions	20
Sliding window size W	80

Table B.7.: Hyperparameters of the LQR problem.

Acronyms

B2P	Back-To-Prior
BM	Brownian motion
BO	Bayesian optimization
C-TVBO	Constrained-TVBO
EI	expected improvement
GP	Gaussian process
KL	Kullback-Leibler
LCB	lower-confidence-bound
LQR	linear-quadratic regulator
MAB	multi-armed bandit
MCMC	Markov-Chain-Monte-Carlo
PI	probability of improvement
RKHS	reproducing kernel Hilbert space
SE	squared-exponential
TVBO	time-varying Bayesian optimization
UCB	upper-confidence-bound
UI	Uncertainty-Injection
UI-TVBO	Uncertainty-Injection-in-TVBO
VOP	virtual observation point

List of Figures

List of Tables

Bibliography

[1] C. Agrell, "Gaussian processes with linear operator inequality constraints," *Journal of Machine Learning Research*, vol. 20, no. 135, pp. 1–36, 2019.

[2] P.-C. Aubin-Frankowski and Z. Szabo, "Hard shape-constrained kernel machines," in *Advances in Neural Information Processing Systems*, vol. 33, Curran Associates, Inc., 2020, pp. 384–395.

[3] P. Auer, "Using confidence bounds for exploitation-exploration trade-offs," *Journal of Machine Learning Research*, vol. 3, no. Nov, pp. 397–422, 2002.

[4] F. Bachoc, A. Lagnoux, and A. F. López-Lopera, "Maximum likelihood estimation for gaussian processes under inequality constraints," *Electronic Journal of Statistics*, vol. 13, no. 2, pp. 2921–2969, 2019.

[5] A. Baheri and C. Vermillion, "Altitude optimization of airborne wind energy systems: A bayesian optimization approach," in *American Control Conference*, IEEE, 2017, pp. 1365–1370.

[6] M. Balandat, B. Karrer, D. R. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy, "BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization," in *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[7] O. Besbes, Y. Gur, and A. Zeevi, "Stochastic multi-armed-bandit problem with non-stationary rewards," in *Advances in Neural Information Processing Systems*, vol. 27, Curran Associates, Inc., 2014.

[8] O. Besbes, Y. Gur, and A. Zeevi, "Non-stationary stochastic optimization," *Operations Research*, vol. 63, no. 5, pp. 1227–1244, Oct. 2015, ISSN: 1526-5463.

[9] I. Bogunovic, J. Scarlett, and V. Cevher, “Time-varying gaussian process bandit optimization,” in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 51, PMLR, May 2016, pp. 314–323.

[10] Z. I. Botev, “The normal law under linear restrictions: Simulation and estimation via minimax tilting,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 79, no. 1, pp. 125–148, Feb. 2016.

[11] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge university press, 2004.

[12] A. Carron, M. Todescato, R. Carli, L. Schenato, and G. Pillonetto, “Machine learning meets kalman filtering,” in *IEEE 55th Conference on Decision and Control*, 2016, pp. 4594–4599.

[13] Q. Chen, N. Golrezaei, and D. Bouneffouf, “Dynamic bandits with temporal structure,” *SSRN 3887608*, 2021.

[14] Y. Deng, X. Zhou, B. Kim, A. Tewari, A. Gupta, and N. Shroff, *Weighted gaussian process bandits for non-stationary environments*, 2021.

[15] D. Duvenaud, “Automatic model construction with gaussian processes,” Ph.D. dissertation, 2014.

[16] J. Gardner, G. Pleiss, K. Q. Weinberger, D. Bindel, and A. G. Wilson, “Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration,” in *Advances in Neural Information Processing Systems*, vol. 31, Curran Associates, Inc., 2018.

[17] A. R. Geist and S. Trimpe, “Learning constrained dynamics with gauss principle adhering gaussian processes,” in *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, ser. Proceedings of Machine Learning Research (PMLR), vol. 120, PMLR, Jun. 2020, pp. 225–234.

[18] P. Hennig and C. Schuler, “Entropy search for information-efficient global optimization,” *Journal of Machine Learning Research*, vol. 13, pp. 1809–1837, Jun. 2012.

[19] H. Imamura, N. Charoenphakdee, F. Futami, I. Sato, J. Honda, and M. Sugiyama, *Time-varying gaussian process bandit optimization with non-constant evaluation time*, 2020.

[20] M. Jauch and V. Peña, *Bayesian optimization with shape constraints*, 2016.

[21] T. Jeong and H. Kim, "Objective bound conditional gaussian process for bayesian optimization," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 139, PMLR, Jul. 2021, pp. 4819–4828.

[22] C. Jidling, N. Wahlström, A. Wills, and T. B. Schön, "Linearly constrained gaussian processes," in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.

[23] C. König, M. Turchetta, J. Lygeros, A. Rupenyan, and A. Krause, "Safe and efficient model-free adaptive control via bayesian optimization," *arXiv*, 2021.

[24] A. Krause and C. Ong, "Contextual gaussian process bandit optimization," in *Advances in Neural Information Processing Systems*, vol. 24, Curran Associates, Inc., 2011.

[25] H. J. Kushner, "A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise," *Journal of Basic Engineering*, vol. 86, no. 1, pp. 97–106, Mar. 1964.

[26] A. Marco, P. Hennig, J. Bohg, S. Schaal, and S. Trimpe, "Automatic lqr tuning based on gaussian process global optimization," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 270–277.

[27] A. Marco, P. Hennig, S. Schaal, and S. Trimpe, "On the design of lqr kernels for efficient controller learning," in *IEEE 56th Annual Conference on Decision and Control*, 2017, pp. 5193–5200.

[28] F. Meier and S. Schaal, "Drifting gaussian processes with varying neighborhood sizes for online model learning," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 264–269.

[29] F. M. Nyikosa, M. A. Osborne, and S. J. Roberts, *Bayesian optimization for dynamic problems*, 2018.

[30] A. M. Ospina, A. Simonetto, and E. Dall'Anese, "Time-varying optimization of networked systems with human preferences," *arXiv*, 2021.

[31] L. Owen, J. Browder, B. Letham, G. Stocek, C. Tymms, and M. Shvartsman, *Adaptive nonparametric psychophysics*, 2021.

[32] J. Parker-Holder, V. Nguyen, S. Desai, and S. Roberts, *Tuning mixed input hyperparameters on the fly for efficient population based autorl*, 2021.

[33] J. Parker-Holder, V. Nguyen, and S. J. Roberts, "Provably efficient online hyperparameter optimization with population-based bandits," in *Advances in Neural Information Processing Systems*, vol. 33, Curran Associates, Inc., 2020, pp. 17 200–17 211.

[34] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035.

[35] H. Raj, S. Dey, H. Gupta, and P. K. Srijith, "Improving adaptive bayesian optimization with spectral mixture kernel," in *Neural Information Processing - 27th International Conference*, ser. Communications in Computer and Information Science, vol. 1333, Springer, 2020, pp. 370–377.

[36] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, ser. Adaptive Computation and Machine Learning. Cambridge, MA, USA: MIT Press, Jan. 2006, p. 248.

[37] S. A. Renganathan, J. Larson, and S. Wild, *Recursive two-step lookahead expected payoff for time-dependent bayesian optimization*, 2020.

[38] J. Riihimäki and A. Vehtari, "Gaussian processes with monotonicity information," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 9, PMLR, May 2010, pp. 645–652.

[39] S. Sarkka, A. Solin, and J. Hartikainen, “Spatiotemporal learning via infinite-dimensional bayesian filtering and smoothing: A look at gaussian process regression through kalman filtering,” *IEEE Signal Processing Magazine*, vol. 30, no. 4, pp. 51–61, 2013.

[40] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, “Taking the human out of the loop: A review of bayesian optimization,” *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2016.

[41] A. Slivkins and E. Upfal, “Adapting to a changing environment: The brownian restless bandits,” in *21st Conference on Learning Theory*, Jul. 2008, pp. 343–354.

[42] J. Snoek, H. Larochelle, and R. P. Adams, “Practical bayesian optimization of machine learning algorithms,” in *Advances in Neural Information Processing Systems*, vol. 25, Curran Associates, Inc., 2012.

[43] F. Solowjow and S. Trimpe, “Event-triggered learning,” *Automatica*, vol. 117, p. 109 009, Jul. 2020.

[44] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, “Gaussian process optimization in the bandit setting: No regret and experimental design,” in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML’10, Madison, WI, USA: Omnipress, 2010, pp. 1015–1022.

[45] J. Su, J. Wu, P. Cheng, and J. Chen, “Autonomous vehicle control through the dynamics and controller learning,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 5650–5657, 2018.

[46] M. Titsias, “Variational learning of inducing variables in sparse gaussian processes,” in *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 5, PMLR, Apr. 2009, pp. 567–574.

[47] S. Van Vaerenbergh, M. Lazaro-Gredilla, and I. Santamaria, “Kernel recursive least-squares tracker for time-varying regression,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1313–1326, 2012.

[48] S. Van Vaerenbergh, I. Santamaría, and M. Lázaro-Gredilla, “Estimation of the forgetting factor in kernel recursive least squares,” in *IEEE International Workshop on Machine Learning for Signal Processing*, 2012, pp. 1–6.

[49] J. Wang, M. Ren, I. Bogunovic, Y. Xiong, and R. Urtasun, *Cost-efficient online hyperparameter optimization*, 2021.

[50] X. Wang and J. O. Berger, “Estimating shape constrained functions using gaussian processes,” *SIAM/ASA Journal on Uncertainty Quantification*, vol. 4, no. 1, pp. 1–25, 2016.

[51] X. Zhou and N. Shroff, “No-regret algorithms for time-varying bayesian optimization,” in *55th Annual Conference on Information Sciences and Systems*, IEEE, 2021, pp. 1–6.