Diversified Sampling for Batched Bayesian Optimization with Determinantal Point Processes

Elvis Nava
ETH Zurich
elvis.nava@ai.ethz.ch

Mojmír Mutný
ETH Zurich
mojmir.mutny@inf.ethz.ch

Andreas Krause
ETH Zurich
krausea@inf.ethz.ch

Abstract

In Bayesian Optimization (BO) we study black-box function optimization with noisy point evaluations and Bayesian priors. Convergence of BO can be greatly sped up by batching, where multiple evaluations of the black-box function are performed in a single round. The main difficulty in this setting is to propose at the same time diverse and informative batches of evaluation points. In this work, we introduce *DPP-Batch Bayesian Optimization (DPP-BBO)*, a universal framework for inducing batch diversity in sampling based BO by leveraging the repulsive properties of Determinantal Point Processes (DPP) to naturally diversify the batch sampling procedure. We illustrate this framework by formulating DPP-Thompson Sampling (DPP-TS) as a variant of the popular Thompson Sampling (TS) algorithm and introducing a Markov Chain Monte Carlo procedure to sample from it. We then prove novel Bayesian simple regret bounds for both classical batched TS as well as our counterpart DPP-TS, with the latter bound being tighter. Our real-world, as well as synthetic, experiments demonstrate improved performance of DPP-BBO over classical batching methods with Gaussian process and Cox process models.

Proceedings of the 25[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

1 INTRODUCTION

Gradient-free optimization of noisy black-box functions is a broadly relevant problem setting, with a multitude of applications such as de-novo molecule design (González et al., 2015), electron laser calibration (Kirschner et al., 2019a,b), and hyperparameter selection (Snoek et al., 2012) among many others. Several algorithms have been devised for such problems, some with theoretical guarantees, broadly referred to as Bayesian optimization (Mockus, 1982) or multi-armed bandits (Berry and Fristedt, 1985). Our work falls into Bayesian optimization as we assume a known prior for the unknown function, and use evaluated points to update our belief about the function.

In BO, the optimization procedure is performed sequentially by evaluating the noisy function on locations informed by past observations. In many real world applications, multiple evaluations (experiments) can be executed in parallel. We refer to this setting as *batched Bayesian optimization (Batch BO)*. This is a common situation when the experimental process is easily parallelizable, such as in high-throughput wet-lab experiments, or parallel training of multiple ML models on a cluster.

A main concern in the batched setting is *batch diversification*: guaranteeing that the selected experimental batch does not perform redundant evaluations. We tackle this problem via Determinantal Point Processes (DPP) (Kulesza and Taskar, 2012), a family of repulsive stochastic processes on sets of items. DPPs have already been successfully employed for Experimental Design (Derezinski et al., 2020), optimization (Mutný et al., 2020a), and in combination with a deterministic batched Bayesian optimization algorithm (Kathuria et al., 2016). In this work, we show how DPP-based diversification can naturally, and in a principled manner, be integrated into randomized algorithms for BO. Of special interest is the Thompson sampling BO algorithm, which is randomized but universally applicable (Thompson, 1933), often empirically outperforms

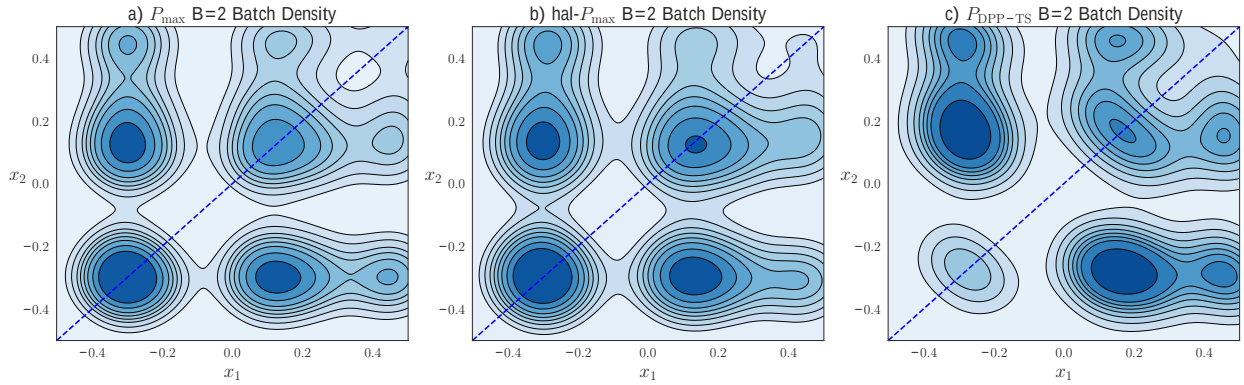

Figure 1: Diversification Demonstration. Given a Gaussian Process posterior on f defined over $\mathcal{X} = [\ 0.5, 0.5]$, we sample a batch of $B = 2$ evaluation points for our next optimization iteration using a randomized Batch BO algorithm. With Thompson Sampling (a), this corresponds to sampling from the symmetric 2d distribution $P_{\mathsf{max}}(x_1, x_2) = p_{\mathsf{max}}(x_1)p_{\mathsf{max}}(x_2)$. We wish to sample diverse batches, therefore we would like to reduce the probability mass near the diagonal (where $x_1 = x_2$). To do so, we can use hallucinated observations (b) or our DPP-TS sampling distribution (c), which exploits DPP repulsion properties. It is apparent how $P_{\mathsf{DPP\text{-}TS}}$, by assigning much less probability mass to locations near the diagonal, disfavors the selection of non-diverse batches.

UCB (Chapelle and Li, 2011), and in some cases has better computational properties (Mutný et al., 2020b).

1.1 Our Contribution

In this work we introduce a framework for randomized Batched Bayesian Optimization diversification through DPPs (DPP-BBO). Our main result is an algorithm called DPP-TS, which samples from a Regularized DPP, capturing both Thompson Sampling (posterior sampling) and information-theoretic batch diversity. We use a Markov Chain Monte Carlo (MCMC) approach adapted from the DPP literature (Anari et al., 2016) to sample batches for this new algorithm. We establish improved Bayesian Simple Regret bounds for DPP-TS compared to classical batching schemes for Thompson Sampling, and experimentally demonstrate its effectiveness w.r.t. BO baselines and existing techniques, both on synthetic and real-world data. Lastly, we demonstrate the generality of our diversification framework by applying it on an alternative randomized BO algorithm called *Perturbed History Exploration* (PHE) (Kveton et al., 2020); and extend it to cover Cox Process models in addition to classically assumed Gaussian Processes.

2 BACKGROUND

Bayesian Optimization The problem setting for Bayesian Optimization (BO) is as follows: we select a sequence of actions $x_t \in \mathcal{X}$, where t denotes the *iteration count* so that $t \in [1, T]$, and $\mathcal{X}$ is the action domain, which is either discrete or continuous. For each chosen action x_t, we observe a noisy reward $y_t = f(x_t) + \epsilon_t$ in sequence, where $f : \mathcal{X} \to \mathbb{R}$ is the unknown reward function, and ϵ_t are assumed to be i.i.d. Gaussian s.t. $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$ with known variance. Most BO algorithms select each point x_t through maximization of an *acquisition function* $x_t = \text{argmax}_{x \in \mathcal{X}}\, u_t(x | D_{t-1})$, determined by the state of an internal Bayesian model of f. We indicate with $D_{t-1} = \{(x_1, y_1), \ldots, (x_{t-1}, y_{t-1})\}$ the filtration consisting of the history of evaluation points and observations up to and including step $t-1$ on which the model is conditioned on. The main algorithmic design choices in BO are which acquisition function and which internal Bayesian model of f to use.

Gaussian Processes Obtaining any theoretical convergence guarantees in infinite or continuous domains is impossible without assumptions on the structure of f. A common assumption in Bayesian Optimization is that f is a sample from a Gaussian Process (GP) (Rasmussen and Williams, 2005) prior, which has the property of being versatile yet allowing for posterior updates to be obtained in closed form. Many BO algorithms make use of an internal GP model of f, which is initialized as a prior and then sequentially updated from feedback. This GP is parametrized by a kernel function $k(x, x')$ and a mean function $\mu(x)$. To denote that f is sampled from the GP, we write $f \sim GP(\mu, k)$.

Regret Minimization We quantify our progress towards maximizing the unknown f via the notion of *regret*. In particular, we define the *instantaneous regret* of action x_t as $r_t = f(x^\star) - f(x_t)$, with $x^\star = \text{argmax}_{x \in \mathcal{X}} f(x)$ being the optimal action.

A common objective for BO is that of minimizing *Bayesian Cumulative Regret* $\mathrm{BCR}_T = \mathsf{E}\left[\sum_{t=1}^{T} r_t\right] = \mathsf{E}\left[\sum_{t=1}^{T} (f(x^\star) \quad f(x_t))\right]$, where the expectation is over the prior of f, observation noise and algorithmic randomness. Obtaining bounds on the cumulative regret that scale sublinearly in T allows us to prove convergence of the *average regret* BCR_T/T, therefore also minimizing the *Bayesian Simple Regret* $\mathrm{BSR}_T = \mathsf{E}\left[\min_{t2[1,T]} r_t\right] = \mathsf{E}\left[\min_{t2[1,T]} f(x^\star) \quad f(x_t)\right]$ and guaranteeing convergence of our optimization of f.

Batch Bayesian Optimization We define *Batch Bayesian Optimization (BBO)* as the setting where, instead of sequentially proposing and evaluating points, our algorithms propose a batch of points of size B at every iteration t. Importantly, the batch must be finalized *before* obtaining any feedback for the elements within it. Batched Bayesian Optimization algorithms encounter two main challenges with respect to performance and theoretical guarantees: proposing diverse evaluation batches, and obtaining regret bounds competitive with full-feedback sequential algorithms, sublinear in the total number of experiments BT, where T denotes the iteration count T and B the batch size.

Determinantal Point Processes (DPPs) (Kulesza and Taskar, 2012) are a family of point processes characterized by the property of *repulsion*. We define a point process P over a set X as a probability measure over subsets of X. Given a similarity measure for pairs of points in the form of a kernel function, Determinantal Point Processes place high probability on subsets that are *diverse* according to the kernel.

We will now describe DPPs for finite domains due to their simplicity, however their definition can be extended to continuous X. For our purposes, we restrict our focus on L-ensemble DPPs: given a so-called L-ensemble kernel L defined as a matrix over the entire (finite) domain X, a Determinantal Point Process P_L is defined as the point process such that the probability of sampling the set $X \quad \mathsf{X}$ is proportional to the determinant of the kernel matrix L_X restricted to X

$$P_L(X) \;/\; \det(L_X). \tag{1}$$

Remarkably, the required normalizing constant can be obtained in closed form as $\sum_{X \quad \mathsf{X}} \det(L_X) = \det(L + I)$.

If the kernel L is such that $L_{ij} = l(x_i, x_j)$, for x_i, x_j 2 X, encodes the similarity between any pair of points x_i and x_j, then the determinant $\det(L_X)$ will be greater for diverse sets X. Intuitively, for the linear kernel, diversity can be measured by the area of the jXj-dimensional parallelepiped spanned by the vectors in X (see Section 2.2.1 from Kulesza and Taskar, 2012).

For our application, we require sampling of batches of points with a specific predetermined size. For this purpose, we focus on k-DPPs. A k-DPP P_L^k over X is a distribution over subsets of X with fixed cardinality k, such that the probability of sampling a specific subset X is proportional to that for the generic DPP case: $P_L^k(X) = \frac{\det(L_X)}{\sum_{X^0 \quad \mathsf{X}, jX^0 j=k} \det(L_{X^0})}$.

Sampling from DPPs and k-DPPs can be done with a number of efficient exact or approximate algorithms. The seminal exact sampling procedure for k-DPPs from Deshpande and Rademacher (2010) requires time $O(kN^{\omega+1} \log N)$ in the batch size k and the size of the domain N, with ω being the exponent of the arithmetic complexity of matrix multiplication. This does not scale well for large domains, nor does it work for the continuous case. Fortunately, an efficient MCMC sampling scheme with complexity of $O(Nk \log(\epsilon \;^{1}))$ introduced by Anari et al. (2016) works much better in practice. Variants of such MCMC schemes have been proven to also work for continuous domains (Rezaei and Gharan, 2019).

3 RELATED WORK

A number of different acquisition functions have been proposed for Bayesian Optimization, such as Probability of Improvement, Expected Improvement, Upper Confidence Bound (UCB) among many others (cf., Brochu et al., 2010). The Gaussian process version of UCB (GP-UCB, Srinivas et al., 2010) is a popular technique based on a deterministic acquisition function, with sublinear regret bounds for common kernels.

Thompson Sampling Thompson Sampling is an intuitive and theoretically sound BO algorithm using a randomized acquisition function (Thompson, 1933; Russo et al., 2020). When choosing the next evaluation point, we sample a realization from the current posterior modeling the objective function, and use this as the acquisition function to maximize $x_t = \mathrm{argmax}_{x2\mathsf{X}} \tilde{f}(x)$ where $\tilde{f}$ is the sample function, e.g. $\tilde{f} \quad \mathrm{GP}(\mu_t, K_t)$. Bayesian Cumulative Regret was first bounded as $O(\sqrt{T\gamma_T})$ by Russo and Van Roy (2014), where γ_T is the maximum mutual information obtainable from T observations (for more details on this well established quantity, see Appendix B). This bound matches lower bounds in T (Scarlett et al., 2017).

Batched UCB and Pure Exploration For Batched BO, heuristic algorithms such as Simulation Matching (Azimi et al., 2010) or Local Penalization (Gonzalez et al., 2016) attempt to solve the problem

of generating informative and diverse evaluation point batches, albeit without theoretical guarantees on regret. In particular, Local Penalization selects explicitly diversified batches by greedily penalizing already-sampled points with penalization factors in the acquisition function.

Desautels et al. (2014) are the first to provide a theoretically justified batched algorithm, introducing GP-BUCB, a batched variant of GP-UCB. To induce diversity within batches, they use *hallucinated observations*, so that $x_{\mathsf{GP\text{-}BUCB}\,t,b}$ is sampled by maximizing a UCB based on the hallucinated posterior $\tilde{D}_{t,b\ 1}$. The hallucinated history is constructed by using the posterior mean in place of the observed reward for points with delayed feedback. GP-BUCB attains a cumulative regret bound of $O\left(\sqrt{TB\beta_{TB}\gamma_{TB}}\right)$, which, however, requires an *initialization phase* before the deployment of the actual algorithm. For the first T_{init} iterations, the evaluations are chosen with Uncertainty Sampling, picking the point satisfying $x_t = \mathrm{argmax}_{x2X}\,\sigma_t(x)$, effectively exploring the whole domain, which limits the practicality of the method. To alleviate this, Contal et al. (2013) introduce the alternative GP-UCB Pure Exploration (GP-UCB-PE, Contal et al., 2013), which mixes the UCB acquisition function with a Pure Exploration strategy. Sampling a batch at timestamp t, GP-UCB-PE operates in two phases: the first point of each batch is sampled with standard GP-UCB, while the remaining $B\quad 1$ points are sampled by first defining a *high probability region* R^+ for the maximizer, and then performing Uncertainty Sampling $x_{\mathsf{UCB\text{-}PE}\,t,b} = \mathrm{argmax}_{x2\mathfrak{R}^+}\,\sigma_{t,b}(x)$. GP-UCB-PE's cumulative regret is bounded by $O\left(\sqrt{TB\beta_{TB}\gamma_{TB}}\right)$ without an initialization phase, as opposed to GP-BUCB.

Batched TS Kandasamy et al. (2018) are first to consider batching with Thompson sampling and GPs. They propose to simply resample multiple times from the posterior within each batch, effectively lifting the Thompson Sampling algorithm as-is to the batched case. By repeating TS sampling for each point within the batch, they bound the Bayesian cumulative regret by $O\left(\sqrt{TB\beta_{TB}\gamma_{TB}}\right)$. It is possible but not required to use hallucinated observations (hal-TS). However, an initialization phase identical to that of GP-BUCB is needed for the their proof on the bound to hold. A novel result from our work is an improved proof technique such that the initialization phase for Batched TS is not required for the Bayesian simple regret version of the bound to hold.

DPPs in Batched BO Kathuria et al. (2016) use Determinantal Point Process sampling to define a variation of GP-UCB-PE (Contal et al., 2013), called UCB-DPP-SAMPLE. They observe that the Uncertainty Sampling phase of GP-UCB-PE corresponds to greedy maximization of the posterior covariance matrix determinant $\det\left(K_{t,1\,X}\right)$ with respect to batches X of size $B\quad 1$ from R^+, with $K_{t,1\,X}$ being the covariance matrix produced by the posterior kernel of the GP after step $(t,1)$ and restricted to the set X. Finding the $(B\quad 1)$-sized submatrix of the maximum determinant is an NP-hard problem, and picking each element greedily so that it maximizes $\sigma^2_{t,b}(x) = k_{t,b}(x,x)$ fails to guarantee the best solution. Maximizing the above determinant is also equivalent to maximizing $\det\left(L_{t,1\,X}\right)$ for the DPP L-ensemble kernel defined as $L_{t,1} = I + \sigma^{\ 2}K_{t,1}$, called the *mutual information kernel* (Kathuria et al., 2016).

Instead of selecting the last $B\quad 1$ points of each batch with Uncertainty Sampling, UCB-DPP-SAMPLE samples them from a $(B\quad 1)$-DPP restricted to R^+ with the Mutual Information L-ensemble kernel $L_{t,1} = I + \sigma^{\ 2}K_{t,1}$. Kathuria et al. (2016) provide a bound for UCB-DPP-SAMPLE as a variation of the $O\left(\sqrt{TB\beta_{TB}\gamma_{TB}}\right)$ bound for GP-UCB-PE. However, as we illustrate in Appendix D, their bound is *necessarily worse* than the existing one for GP-UCB-PE.

The concurrent work of Nguyen et al. (2021) is another recent example of DPP usage in BBO diversification, proposing DPP sampling (with DPP kernel informed by a GP posterior) as a method of diverse batch selection, demonstrating good performance in experimental tasks, but no known theoretical regret guarantees.

4 THE DPP-BBO FRAMEWORK

A key insight our approach relies on is to view Thompson Sampling as a procedure that samples at each step from a *maximum distribution* p_{max} over $\mathcal{X}$, so that $x_t \quad p_{\mathsf{max},t}$ with

$$p_{\mathsf{max},t}(x) = \mathsf{E}_{f\ \ \mathsf{Post}_t}\left[\mathbb{1}[x = \operatorname*{argmax}_{x'2X} \tilde{f}(x')]\right]. \qquad (2)$$

A simple approach towards Batched Thompson Sampling is to obtain a batch X_t of evaluation points (with $|X_t| = B$) by sampling B times from the posterior in each round. This can again be interpreted as

$$X_t \quad P_{\mathsf{max},t} \text{ with } P_{\mathsf{max},t}(X) = \prod_{x_b 2 X} p_{\mathsf{max},t}(x_b). \quad (3)$$

This way, we can view Thompson Sampling or any other randomized Batch BO algorithm as iteratively sampling from a batch distribution over $\mathcal{X}^B$ dependent on t. The main downside of this simple approach is that *independently* obtaining multiple samples may lead to redundancy. As a remedy, in our DPP-BBO framework, we modify such sampling distributions by

reweighing them by a DPP likelihood. This technique is general, and allows us to apply DPP diversification to *any* randomized BBO algorithm with batch sampling likelihood $P_{\mathsf{A},t}(X)$.

Definition 1 (DPP-BBO Sampling Likelihood) *The batch sampling likelihood of generic DPP-BBO at step t is*

$$P_{\textsc{dpp-bbo},t}(X) \propto P_{\mathsf{A},t}(X) \det(L_{tX}) \tag{4}$$

with L_t being a DPP L-ensemble kernel defined over the domain $\mathcal{X}$.

Notice that the domain $\mathcal{X}$ does not need to be discrete, even though we introduced the approach on discrete ground sets in order to simplify notation. This is in contrast to the existing DPP-based BO algorithm from Kathuria et al. (2016), which requires the domain to be discrete in order to efficiently sample the DPP restricted to the arbitrary region $\mathbb{R}^+$ in the general case.

We now proceed to justify our formulation, defining the DPP-Thompson Sampling (DPP-TS) procedure in the process.

4.1 The DPP quality-diversity decomposition

DPPs capture element diversity but also take into account element *quality* independently of the similarity measure, as illustrated by Kulesza and Taskar (2012). Namely, L-ensemble DPPs can be decomposed into a quality-diversity representation, so that the entries of the L-ensemble kernel for the DPP are expressed as $L_{ij} = q_i \phi_i^\top \phi_j q_j$ with $q_i \in \mathbb{R}^+$ representing the *quality* of an item i, and $\phi_i \in \mathbb{R}^m$, $\|\phi_i\| = 1$ being normalized *diversity* features. We also define S with $S_{ij} = \phi_i^\top \phi_j$. This allows us to represent the DPP model as $P_L(X) \propto \left(\prod_{i \in X} q_i^2\right) \det(S_X)$.

We then consider a k-DPP with L-ensemble kernel L in its quality-diversity representation, and re-weigh the quality values of items by their likelihood under a Bayesian Optimization random sampling scheme $P_{A,t}(x)$ such as Thompson Sampling $P_{\mathsf{max},t}(x)$. Following this approach, we can obtain a new k-DPP likelihood by renormalizing the product of the Thompson Sampling likelihood of the batch P_{max} and an existing DPP likelihood P_L for $X = \{x_1, \dots, x_B\}$:

$$P_{\textsc{dpp-ts}}(X) \propto \left(\prod_{x_b \in X} p_{\mathsf{max}}(x_b)\right) P_L(X) \tag{5}$$

$$\propto \left(\prod_{x_b \in X} p_{\mathsf{max}}(x_b)\right) \det(L_X) \tag{6}$$

$$\propto \left(\prod_{x_b \in X} p_{\mathsf{max}}(x_b) \ q_{x_b}^2\right) \det(S_X) \tag{7}$$

The result is a k-DPP with L-ensemble kernel $\tilde{L}_{ij} = \sqrt{p_{\mathsf{max}}(x_i) p_{\mathsf{max}}(x_j)} L_{ij}$, generalizing the sampling distribution for batched TS as a stochastic process with repulsive properties. To recover original batched TS, we just need to set $L_t = I$.

4.2 The Mutual Information Kernel

For our choice of kernel, we follow the insight from Kathuria et al. (2016) and use $L_t = I + \sigma^{-2} K_t$, with K_t being the GP posterior kernel at step t. Consequently, the DPP loglikelihood of a set X at time t is proportional to the mutual information between the true function f and the observations obtained from X: $I(f_X; \mathbf{y}_X | \mathbf{y}_{1:t\ 1,1:B}) = \frac{1}{2} \log\det(I + \sigma^{-2} K_{tX})$ (see Appendix B). This is an example of a so-called *Regularized k-DPP*, a k-DPP such that a symmetric positive semidefinite regularization matrix A is added to an original unregularized L-ensemble DPP kernel, for the particular case of $A = \lambda I$. In such a setting, we allow for the same element to be selected multiple times and enforce that any set X must have nonzero probability of being selected. By tuning the strength of the regularization, we can tune how extreme we wish our similarity repulsion to be.

Definition 2 (DPP-TS Sampling Likelihood) *The batch sampling likelihood of DPP-TS at step t is*

$$P_{\textsc{dpp-ts},t}(X) \propto P_{\mathsf{max},t}(X) \det(I + \sigma^{-2} K_{tX}). \tag{8}$$

In Figure 1, we illustrate the $|X| = 2$ case to compare the original P_{max} TS distribution, a TS variant with hallucinated observations, and $P_{\textsc{dpp-ts}}$ with its repulsion properties.

4.3 Markov Chain Monte Carlo for DPP-BBO

Sampling from the mutual information DPP component $\det(I + \sigma^{-2} K_{tX})$ of DPP-TS on its own can be done easily and efficiently, as numerous algorithms exist for both exact and approximate sampling from k-DPPs (Kulesza and Taskar, 2012). Likewise, we assume we are in a setting in which Thompson Sampling on its own can be performed relatively efficiently, as sampling from $P_{\mathsf{max}\,t}$ reduces to sampling a function realization $\tilde{f}$ from the posterior, e.g. $\mathrm{GP}(\mu_t, K_t)$, and maximizing $\tilde{f}$ over $\mathcal{X}$.

However, when sampling from the product of the two distributions, we must resort to tools of approximate inference. The main issue with adopting standard approaches is that computation of the explicit likelihood $P_{\textsc{dpp-ts},t}$ is *doubly intractable*: computation of $P_{\mathsf{max},t}$ is intractable on its own, and it appears in the enumerator of $P_{\textsc{dpp-ts},t}$ before normalization.

Our approach for sampling from $P_{\text{DPP-TS},t}$ relies on a Markov Chain Monte Carlo (MCMC) sampler. We construct an ergodic Markov Chain over batches from $\Omega = \{X \mid X \subseteq \mathcal{X}, |X| = k\}$ with transition kernel $T(X'|X)$ such that the detailed balance equation $Q(X)T(X'|X) = Q(X')T(X|X')$ is satisfied almost surely with respect to $P_{\text{DPP-TS},t}$, with $Q(X) = P_{\text{max},t}(X)\det(L_{tX})$ being the unnormalized potential of $P_{\text{DPP-TS},t}$.

If $Q(X)$ were tractable, we could use the standard Metropolis-Hastings algorithm (Hastings, 1970), which satisfies the detailed balance equation. The problem with naively using Metropolis-Hasting MCMC sampling is that our $Q(X) = P_{\text{max},t}(X)\det(L_{tX})$ contains $P_{\text{max},t}(X) = \prod_{x_b \in X} p_{\text{max},t}(x_b)$, which is intractable and cannot be computed on the fly. As previously stated, the only thing we can easily do is sample from it by sampling $\tilde{f}$ and then maximizing it. However, if we modify the standard Metropolis-Hastings MCMC algorithm by using $p_{\text{max},t}$ proposals, we obtain Algorithm 1, which satisfies detailed balance. We refer to Appendix A for the proof. This algorithm can be interpreted as a variant of an existing k-DPP sampler proposed by Anari et al. (2016).

Algorithm 1 DPP-TS MCMC sampler

pick random initial batch X
repeat
  uniformly pick point $x_b \in X$ to replace
  sample candidate point $x_b' \sim p_{\text{max},t}(x_b')$
  define $X' = (X \setminus \{x_b\}) \cup \{x_b'\}$
  accept with probability $\alpha = \min\left\{1, \frac{\det(L_{tX'})}{\det(L_{tX})}\right\}$
  if accepted **then**
    $X = X'$
  end if
until converged

Algorithm 2 DPP-TS Algorithm

Input: Action space $\mathcal{X}$, GP prior μ_1, $k_1(.,.)$, history $D_0 = \{\}$
for $t = 1, \ldots, T$ **do**
  Sample $X_t \sim P_{\text{DPP-TS},t}(X_t)$ with Alg. 1
  Observe $y_{t,b} = f(x_{t,b}) + \epsilon_{t,b}$ for $b \in [1, B]$
  Add observations to history $D_t = D_{t-1} \cup \{(x_{t,1}, y_{t,1}), \ldots, (x_{t,B}, y_{t,B})\}$
  Update the GP with D_t to get μ_{t+1}, $k_{t+1}(.,.)$
end for

4.4 DPP-TS

Given the sampling distribution (Definition 2) and the above MCMC algorithm, we can now fully specify the overall procedure for our DPP-TS sampling algorithm summarized in Algorithm 2.

5 BAYESIAN REGRET BOUNDS

We now establish bounds on the Bayesian regret. Instead of assuming the existence of a fixed true f, we assume that the true function is sampled from a Gaussian Process prior $f \sim GP(0, K)$.

In particular, our regret bounds are obtained on a variant of Bayes regret called Bayes Batch Cumulative Regret $\text{BBCR}_{T,B} = \mathbb{E}\left[\sum_{t=1}^{T} \min_{b \in [1,B]} r_{t,b}\right] = \mathbb{E}\left[\sum_{t=1}^{T} \min_{b \in [1,B]} (f(x^\star) - f(x_{t,b}))\right]$ which only considers the best instantaneous regret within each batch, as we make use of proof techniques from Contal et al. (2013) involving such a formulation. It is straightforward to see that by bounding BBCR we at the same time bound the Bayes Simple Regret (introduced in Section 2), as $\text{BSR}_{T,B} \leq \text{BBCR}_{T,B}/T$, similarly to how $\text{BSR}_{T,B} \leq \text{BCR}_{T,B}/TB$.

5.1 Improved bound on BBCR for Batched Thompson Sampling

Our first theoretical contribution is an improved version of the bound on Bayesian Simple Regret from Kandasamy et al. (2018). Our version of the algorithm requires no initialization procedure to guarantee sublinear regret in contrast to prior work.

Unlike the original Gaussian TS Bayesian bounds from Russo and Van Roy (2014), Kandasamy et al. (2018) analyze the problem over a continuous domain. Therefore, it requires an additional assumption previously used in the Bayesian continuous-domain GP-UCB bound (Srinivas et al., 2010).

Assumption 3 (Gradients of GP Sample Paths) *Let $\mathcal{X} \subset [0, l]^d$ compact and convex with $d \in \mathbb{N}$ and $l > 0$, $f \sim GP(0, K)$ where k is a stationary kernel. Moreover, there exist constants $a, b > 0$ such that $P\left(\sup_{x \in \mathcal{X}} \left|\frac{\partial f(x)}{\partial x_i}\right| > L\right) \leq ae^{-(L/b)^2}$ $\forall L > 0, \forall i \in \{1, \ldots, d\}$.*

Using the above assumption, we can show the following theorem.

Theorem 4 (BBCR Bound for Batched TS) *If $f \sim GP(0, K)$ with covariance kernel bounded by 1 and noise model $\mathcal{N}(0, \sigma^2)$, and either*

- *Case 1: finite $\mathcal{X}$ and $\beta_t = 2\ln\left(\frac{B(t^2+1)|\mathcal{X}|}{p\sqrt{2\pi}}\right)$;*
- *Case 2: compact and convex $\mathcal{X} \subset [0, l]^d$, with Assumption 3 satisfied and $\beta_t = 4(d+1)\log(Bt) + 2d\log(dab\sqrt{\log(4da/p)})$.*

Then Batched Thompson Sampling attains Bayes Batch Cumulative Regret of

$$\mathrm{BBCR}_{\mathsf{TS}T,B} \quad \frac{C_1}{B} + \sqrt{C_2 \frac{T}{B} \beta_T \gamma_{TB}} \tag{9}$$

with $C_1 = 1$ *for Case 1,* $C_1 = \frac{\pi^2}{6} + \frac{\sqrt{2\pi}}{12}$ *for Case 2, and* $C_2 = \frac{2}{\log(1+\sigma^{-2})}$.

Therefore, $\mathrm{BSR}_{\mathsf{TS}T,B} \quad \frac{C_1}{TB} + \sqrt{C_2 \frac{1}{TB} \beta_T \gamma_{TB}}$. We point to Appendix C.1 for proof. The bound from Kandasamy et al. (2018) (without an initialization phase) is similar, except for the presence of an $\exp(C)$ factor in the square root term, which scales linearly with B. Our version of the bound does not contain $\exp(C)$, allowing thus for sublinear regret in B.

5.2 BBCR bound for DPP-TS

We now shift the focus to our novel DPP-TS algorithm and obtain an equivalent bound. To do so, we modify the algorithm we developed and introduce DPP-TS-alt, so that for every batch:

a) For the first sample in the batch $x_{\mathsf{DPP\text{-}TS\text{-}alt}\ t,1}$, we sample from $p_{\mathsf{max},t}$ as in standard Thompson Sampling; b) For all the other samples $x_{\mathsf{DPP\text{-}TS\text{-}alt}\ t,b}$ with $b \in [2, B]$, we sample from joint $P_{\mathsf{DPP\text{-}TS}\ t}$, using the most updated posterior variance matrix $K_{t,1}$ to define the DPP kernel.

The reason why we introduced DPP-TS as such and not DPP-TS-alt in the first place is both for simplicity and because in practice their performance is virtually identical (see Appendix E). We have the following

Theorem 5 (BBCR Bound for DPP-TS) *Consider the same assumptions as for Theorem 4. Then DPP-TS (in its DPP-TS-alt variant) attains Bayes Batch Cumulative Regret of*

$$\mathrm{BBCR}_{\mathsf{DPP\text{-}TS}T,B} \quad \frac{C_1}{B} + \sqrt{C_2 \frac{T}{B} \beta_T \gamma_{TB}} \quad C_3 \tag{10}$$

with $C_1 = 1$ *for Case 1,* $C_1 = \frac{\pi^2}{6} + \frac{\sqrt{2\pi}}{12}$ *for Case 2,* $C_2 = \frac{2}{\log(1+\sigma^{-2})}$, *and* $C_3 < 0$ *(defined in Appendix C.2).*

We can thus obtain $\mathrm{BSR}_{\mathsf{DPP\text{-}TS}T,B} \quad \frac{C_1}{TB} \quad \frac{C_3}{T} + \sqrt{C_2 \frac{1}{TB} \beta_T \gamma_{TB}}$. Moreover, this bound is necessarily tighter than that for standard TS: $\frac{C_1}{TB} \quad \frac{C_3}{T} + \sqrt{C_2 \frac{1}{TB} \beta_T \gamma_{TB}} \quad \frac{C_1}{TB} + \sqrt{C_2 \frac{1}{TB} \beta_T \gamma_{TB}}$. We point to Appendix C.2 for the proof.

6 EXPERIMENTS AND COMPARISONS

To make the case for our algorithmic framework's effectiveness in practice, we perform a series of benchmark tests on synthetic and real world optimization problems, comparing DPP-BBO against classic BBO algorithms on Simple Regret metrics. (Cumulative Regret comparisons feature in Appendix E.)

6.1 DPP-TS Comparisons on Synthetic Data

We first compare DPP-TS on synthetic benchmarks against regular batched TS, GP-BUCB, hallucinated TS (Batched Thompson Sampling with hallucinations as in GP-BUCB), Pure DPP Exploration (DPP sampling from the DPP component of DPP-TS) and Uniform Exploration (uniform random sampling over the domain). We exclude algorithms that are not applicable to continuous domains.

Figure 2 details a number of such comparisons on synthetic benchmark functions under different settings, averaged over 15 experimental runs. For 2.a and 2.b we optimize over a discrete finite domain $\mathcal{X}$, using an exact Gaussian Process prior with a squared exponential kernel. The acquisition function is maximized by calculation of the explicit maximum over the discretized domain.

For 2.c and 2.d, we optimize over a continuous domain $\mathcal{X} = [0, l]^d$, using an approximate Gaussian Process prior specified with Quadrature Fourier Features (Mutný and Krause, 2018). These functions are additive and, hence, the optimization can be done dimension-wise. When optimizing the one-dimensional projection of the acquisition function we use first order gradient descent with restarts.

Specific benchmarks we use are the Rosenbrock function $f(x) = 100(x_2 \quad x_1^2)^2 + (x_1 \quad 1)^2$; the Stiblinski-Tang function $f(x) = \frac{1}{2}\sum_{i=i}^{d}\left(x_i^4 \quad 16x_i^2 + 5x_i\right)$; and the Michalewicz function $f(x) = \sum_{i=i}^{d} \sin(x_i) \sin^{2d}\left(ix_i^2/\pi\right)$.

Overall, DPP-TS converges very quickly to sampling good maximizers, almost always beating or at least equaling the Simple Regret performance of the other algorithms, while exhibiting low-variance behavior. The added diversity from the DPP sampling procedure appears to favor quickly finding better maxima while not getting stuck in suboptimal but high-confidence regions, as seems to often happen to GP-UCB. A series of additional experiments is discussed in Appendix E, including experiments on Cumulative Regret, DPP-TS with parametrized DPP kernels, and a comparison between DPP-TS and DPP-TS-alt which shows them

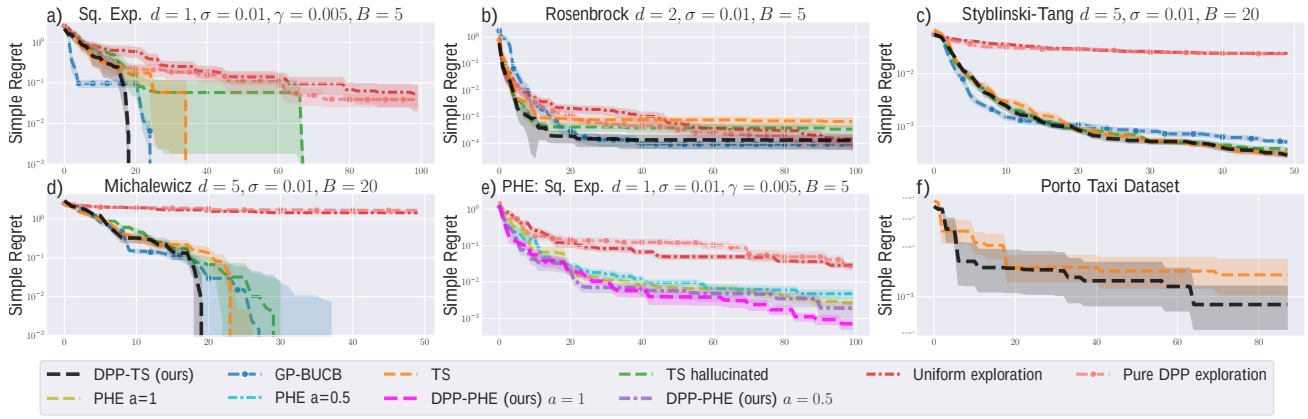

Figure 2: Comprehensive experimental comparisons between DPP-TS and classic BBO techniques for Simple Regret (log scale): **a)** f sampled from Squared Exponential GP; **b)** Rosenbrock; **c)** Styblinski-Tang; **d)** Michalewicz; **e)** PHE experiment with f sampled from QFF Squared Exponential GP; **f)** Cox process sensing experiment on the Porto taxi dataset. The named functions are defined in Section 6.1. Overall, DPP-TS outperforms or equals the other algorithms, quickly sampling good maximizers thanks to improved batch diversification.

to be of equivalent performance in practice.

6.2 DPP-Perturbed History Exploration

To further demonstrate the effectiveness and versatility of the DPP-BBO framework, we apply it to the recently introduced Perturbed History Exploration (PHE) algorithm (Kveton et al., 2020). PHE is a BO algorithm which is agnostic of the specific model f_θ chosen for modeling f. Assuming that rewards are bounded, and given a parameter a, the algorithm introduces *pseudo-rewards* a for each observation in its global history, and at each step maximizes its learned perturbed f_θ to propose a new evaluation point. We can interpret this procedure as sampling from $p_{\mathsf{PHE},t}(x)$, with the stochastic component stemming from the pseudo-reward generation. Given this, we can define DPP-PHE as $P_{\mathsf{DPP\text{-}PHE},t}(X) \propto \left(\prod_{x \in X} p_{\mathsf{PHE},t}(x)\right) \det(I + \sigma^{-2} K_{tX})$ where K_t is an approximation of the Bayesian posterior covariance for the f_θ model.

Figure 2.e experimentally compares PHE and DPP-PHE for $a = 0.5$ and $a = 1$ on a synthetic function (over a continuous X) sampled from a 1-d squared exponential GP prior, while using as internal model a QFF GP regression. We can see that DPP-PHE improves on the Simple Regret when compared to regular PHE for the same a.

6.3 DPP-TS and Cox Process Sensing

To benchmark our DPP-TS algorithm on a real world setting and demonstrate the versatility of the modeling choice, we turn to a Cox Process Sensing problem in the form of taxi routing on a 2-dimensional city grid, as considered by Mutný and Krause (2021). Given a dataset of geo-localized taxi cab hails in Porto and a subdivision of the city into an 8x8 grid, we aim to learn the best locations where to schedule a fleet of taxis while, at beginning of each day - corresponding to a single iteration, we only observe the taxi hailing events in the grid cells which had vehicles scheduled to them.

We put a Gaussian process prior on the unknown rate function of a Poisson process, yielding a Cox Process with Poisson Process likelihood. The likelihood of observing a realization $D = \{x_n\}_{n=1}^N$ over the domain X for a Poisson Process with rate function $\lambda(.)$ is $p(D \mid \lambda(.)) = \exp\left(\int_X \lambda(x)\,dx\right) \prod_n \lambda(x_n)$. This Poisson process specification is used in the construction of a Cox process model, which is $p(D, \lambda(.), \Theta) = p(D \mid \lambda(.))\ p(\lambda(.) \mid \Theta)\ p(\Theta)$, with $\lambda(.)$ being a Gaussian Process conditioned on being positive-valued over the domain. We adopt the inference scheme along with the approximation scheme to maintain positivity of the rate function from Mutný and Krause (2022). The samples from the posterior are obtained via Langevin dynamics.

In our experiment, we compare TS for Cox Process Sensing from Mutný and Krause (2021) with our DPP-TS approach, leveraging our diversifying process to improve city coverage by our scheduled taxi fleets. As DPP kernel, we use the mutual information kernel that is obtained when the posterior for the rate function is approximated with a Gaussian distribution, known as the Laplace Approximation.

In Figure 2.f we depict allocation of 5 taxis to city blocks and report the simple regret. DPP-TS reliably achieves lower simple regret than standard Thompson Sampling sensing with resampling.

7 CONCLUSIONS

In this work we introduced DPP-BBO, a natural and easily applicable framework for enhancing batch diversity in BBO algorithms which works in more settings than previous diversification strategies: it is directly applicable to the continuous domain case, when due to approximation and non-standard models we are unable to compute hallucinations or confidence intervals (as in the Cox process example), or more generally when used in combination with any randomized BBO sampling scheme or arbitrary diversity kernel. Moreover, for DPP-TS we show improved theoretical guarantees and strong practical performance on simple regret.

Acknowledgements

This research was supported by the ETH AI Center and the SNSF grant 407540 167212 through the NRP 75 Big Data program. This publication was created as part of NCCR Catalysis (grant number 180544), a National Centre of Competence in Research funded by the Swiss National Science Foundation.

References

Anari, N., Gharan, S. O., and Rezaei, A. (2016). Monte Carlo Markov Chain Algorithms for Sampling Strongly Rayleigh Distributions and Determinantal Point Processes. *arXiv:1602.05242 [cs, math]*. arXiv: 1602.05242.

Azimi, J., Fern, A., and Fern, X. (2010). Batch bayesian optimization via simulation matching. In Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R., and Culotta, A., editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc.

Berry, D. A. and Fristedt, B. (1985). *Bandit problems: Sequential Allocation of Experiments*. Monographs on Statistics and Applied Probability. Springer Netherlands.

Brochu, E., Cora, V. M., and de Freitas, N. (2010). A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. *arXiv:1012.2599 [cs]*. arXiv: 1012.2599.

Chapelle, O. and Li, L. (2011). An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257.

Contal, E., Buffoni, D., Robicquet, A., and Vayatis, N. (2013). Parallel Gaussian Process Optimization with Upper Confidence Bound and Pure Exploration. *arXiv:1304.5350 [cs, stat]*, 7908:225–240. arXiv: 1304.5350.

Derezinski, M., Liang, F., and Mahoney, M. (2020). Bayesian experimental design using regularized determinantal point processes. In *International Conference on Artificial Intelligence and Statistics*, pages 3197–3207. PMLR. ISSN: 2640-3498.

Desautels, T., Krause, A., and Burdick, J. W. (2014). Parallelizing Exploration-Exploitation Tradeoffs in Gaussian Process Bandit Optimization. *Journal of Machine Learning Research*, 15(119):4053–4103.

Deshpande, A. and Rademacher, L. (2010). Efficient volume sampling for row/column subset selection. *arXiv:1004.4057 [cs]*. arXiv: 1004.4057.

Gonzalez, J., Dai, Z., Hennig, P., and Lawrence, N. (2016). Batch bayesian optimization via local penalization. In Gretton, A. and Robert, C. C., editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 648–657, Cadiz, Spain. PMLR.

González, J., Longworth, J., James, D. C., and Lawrence, N. D. (2015). Bayesian Optimization for Synthetic Gene Design. *arXiv:1505.01627 [stat]*. arXiv: 1505.01627.

Hastings, W. K. (1970). Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57(1):97–109. Publisher: [Oxford University Press, Biometrika Trust].

Kandasamy, K., Krishnamurthy, A., Schneider, J., and Poczos, B. (2018). Parallelised Bayesian Optimisation via Thompson Sampling. In *International Conference on Artificial Intelligence and Statistics*, pages 133–142. PMLR. ISSN: 2640-3498.

Kathuria, T., Deshpande, A., and Kohli, P. (2016). Batched Gaussian Process Bandit Optimization via Determinantal Point Processes. *arXiv:1611.04088 [cs]*. arXiv: 1611.04088.

Kirschner, J., Mutný, M., Hiller, N., Ischebeck, R., and Krause, A. (2019a). Adaptive and safe bayesian optimization in high dimensions via one-dimensional subspaces. *ICML 2019*.

Kirschner, J., Nonnenmacher, M., Mutnỳ, M., Krause, A., Hiller, N., Ischebeck, R., and Adelmann, A. (2019b). Bayesian optimisation for fast and safe parameter tuning of swissfel. In *FEL2019, Proceedings of the 39th International Free-Electron Laser Conference*, pages 707–710. JACoW Publishing.

Kulesza, A. and Taskar, B. (2012). Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2-3):123–286. arXiv: 1207.6083.

Kveton, B., Szepesvári, C., Ghavamzadeh, M., and Boutilier, C. (2020). Perturbed-History Exploration

in Stochastic Linear Bandits. In *Uncertainty in Artificial Intelligence*, pages 530–540. PMLR. ISSN: 2640-3498.

Li, C., Jegelka, S., and Sra, S. (2016). Fast DPP Sampling for Nyström with Application to Kernel Methods. *arXiv:1603.06052 [cs]*. arXiv: 1603.06052.

Mockus, J. (1982). The bayesian approach to global optimization. *System Modeling and Optimization*, pages 473–481.

Mutný, M., Derezisnki, M., and Krause, A. (2020a). Convergence analysis of block coordinate algorithms with determinantal sampling. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*. AISTATS.

Mutný, M., Johannes, K., and Krause, A. (2020b). Experimental design for orthogonal projection pursuit regression. *AAAI2020*.

Mutný, M. and Krause, A. (2018). Efficient High Dimensional Bayesian Optimization with Additivity and Quadrature Fourier Features. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Mutný, M. and Krause, A. (2021). No-regret algorithms for capturing events in poisson point processes. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7894–7904. PMLR.

Mutný, M. and Krause, A. (2022). Sensing cox processes via posterior sampling and positive bases. *AISTATS 2022*.

Nguyen, V., Le, T., Yamada, M., and Osborne, M. A. (2021). Optimal Transport Kernels for Sequential and Parallel Neural Architecture Search. *arXiv:2006.07593 [cs, stat]*. arXiv: 2006.07593.

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning series. MIT Press, Cambridge, MA, USA.

Rezaei, A. and Gharan, S. O. (2019). A Polynomial Time MCMC Method for Sampling from Continuous Determinantal Point Processes. In *International Conference on Machine Learning*, pages 5438–5447. PMLR. ISSN: 2640-3498.

Russo, D. and Van Roy, B. (2014). Learning to Optimize via Posterior Sampling. *Mathematics of Operations Research*, 39(4):1221–1243. Publisher: INFORMS.

Russo, D., Van Roy, B., Kazerouni, A., Osband, I., and Wen, Z. (2020). A Tutorial on Thompson Sampling. *arXiv:1707.02038 [cs]*. arXiv: 1707.02038.

Scarlett, J., Bogunovic, I., and Cevher, V. (2017). Lower bounds on regret for noisy gaussian process bandit optimization. *arXiv preprint arXiv:1706.00090*.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical Bayesian Optimization of Machine Learning Algorithms. *arXiv:1206.2944 [cs, stat]*. arXiv: 1206.2944.

Srinivas, N., Krause, A., Kakade, S., and Seeger, M. (2010). Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In *ICML*.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294.

Supplementary Material: Diversified Sampling for Batched Bayesian Optimization with Determinantal Point Processes

Appendix A MARKOV CHAIN MONTE CARLO SAMPLING FOR DPP-BBO

Our approach for sampling from $P_{\mathsf{DPP\text{-}TS}\,t}$ leverages a Markov Chain Monte Carlo (MCMC) sampler. We construct an ergodic Markov Chain over batches from $\Omega = \{X \mid X \subset \mathcal{X}, |X| = k\}$ with transition kernel $T(X'|X)$ such that the Detailed Balance equation

$$Q(X)T(X'|X) = Q(X')T(X|X') \tag{11}$$

is satisfied almost surely with respect to $P_{\mathsf{DPP\text{-}TS}\,t}$, with $Q(X) = P_{\mathsf{max}\,t}(X)\det(L_X)$ being the unnormalized potential of $P_{\mathsf{DPP\text{-}TS}\,t}$.

Running the Markov chain will at the limit produce a limiting distribution $\pi(X)$ independent of the initial distribution $\pi_0(X)$. If the above mentioned property of Detailed Balance is satisfied, $\pi(X)$ will be equivalent to the true distribution $P_{\mathsf{DPP\text{-}TS}\,t}$, meaning we can use the Markov Chain to approximately sample from $P_{\mathsf{DPP\text{-}TS}\,t}$ provided we run it long enough.

If $Q(X)$ were tractable, we could use the standard Metropolis-Hastings algorithm (Hastings, 1970): at each step sampling a candidate batch X' from a proposal distribution $R(X'|X)$, then accepting the candidate with probability $\alpha = \min\left\{1, \frac{Q(X')R(X|X')}{Q(X)R(X'|X)}\right\}$.

Algorithm 3 Metropolis-Hastings MCMC

sample initial X at random
repeat
  sample candidate $X' \sim R(X'|X)$
  accept with probability $\alpha = \min\left\{1, \frac{Q(X')R(X|X')}{Q(X)R(X'|X)}\right\}$
  if accepted **then**
    $X = X'$
  end if
until converged

Theorem 6 (Metropolis-Hastings (Hastings, 1970)) *The Markov Chain obtained from the Metropolis-Hastings Algorithm satisfies the Detailed Balance equation $Q(X)T(X'|X) = Q(X')T(X|X')$ over the support of the proposal distribution $R(X'|X)$.*

Proof. — We assume that $R(X'|X) > 0\ \forall X, X'$, and we analyze the two cases for the Detailed Balance equation.

- Case $X = X'$: The equivalence is trivial for any $T(X|X)$.
- Case $X \neq X'$:

  We can express the transition kernel as $T(X'|X) = \alpha R(X'|X)$. Assume that for the transition $X \to X'$ we have $\frac{Q(X')R(X|X')}{Q(X)R(X'|X)} < 1$, and therefore $T(X'|X) = \frac{Q(X')R(X|X')}{Q(X)R(X'|X)}R(X'|X) = \frac{Q(X')R(X|X')}{Q(X)}$. Then, for the inverse transition $X' \to X$ we necessarily have $\frac{Q(X')R(X|X')}{Q(X)R(X'|X)} \geq 1$ and $T(X|X') = R(X|X')$.

  The resulting Detailed Balance equation is

$$Q(X)\frac{Q(X')R(X|X')}{Q(X)} = Q(X')R(X|X') \tag{12}$$

and we have equality.

If the proposal distribution has the same support of the true distribution, Metropolis-Hastings allows us to approximately sample from it.

A.1 Metropolis-Hastings with $p_{\max}$ proposals

The problem with naively using Metropolis-Hasting MCMC sampling is that our $Q(X) = P_{\max\, t}(X)\det(L_X)$ contains $P_{\max}(X) = \prod_{x_b 2 X} p_{\max}(x_b)$, which is intractable and cannot be computed on the fly. As previously stated, the only thing we can easily do is sample from it by sampling $\tilde{f}$ and then maximizing it. In order to obtain a suitable MCMC sampler, we need to subtly alter existing samplers.

We first propose an MCMC algorithm which samples whole batches at every step:

Algorithm 4 Full batch MCMC sampler

pick random initial batch X
repeat
  sample candidate batch $X^0 \quad P_{\max}(X^0)$
  accept with probability $\alpha = \min\left\{1, \frac{\det(L_{X^0})}{\det(L_X)}\right\}$
  if accepted **then**
    $X = X^0$
  end if
until converged

This algorithm is equivalent to Metropolis-Hastings: if in MH we chose $R(X^0 jX) = P_{\max\, t}(X^0)$, the fraction in the definition of the acceptance probability α would in fact reduce to

$$\frac{Q(X^0)R(X jX^0)}{Q(X)R(X^0 jX)} = \frac{P_{\max}(X^0)\det(L_{X^0})P_{\max}(X)}{P_{\max}(X)\det(L_X)P_{\max}(X^0)} = \frac{\det(L_{X^0})}{\det(L_X)}. \tag{13}$$

By virtue of this equivalence, Theorem 6 applies to our procedure as well and our sampler approximately samples from the true $P_{\text{DPP-TS}}$ distribution.

In a similar fashion, it's possible to define a more efficient MCMC sampler which only changes one point from the batch at every step. Since we're in the k-DPP setting, it's possible for us to consider the distribution over a batch X as a k-dimensional multivariate distribution over the x_b 2 X. Then, the obtained sampler can be seen as akin to a Gibbs sampler, and is the one showed in the main paper as Algorithm 1.

This again reduces to Metropolis-Hastings, with proposal

$$R(X^0 jX) = \begin{cases} 0 & \text{if } 9i, j : x_i^0 \; 6 = \; x_i \; ^ \; x_j^0 \; 6 = \; x_j \\ \frac{1}{k} p_{\max}(x_i^0) & \text{if } !9i : x_i^0 \; 6 = \; x_i \\ \sum_{x_i^0 2 X^0} \frac{1}{k} p_{\max}(x_i^0) & \text{if } X = X^0 \end{cases} \tag{14}$$

When sampling a proposal point with $X \; 6 = \; X^0$, the fraction in the definition of the acceptance probability α then becomes

$$\frac{Q(X^0)R(X jX^0)}{Q(X)R(X^0 jX)} = \frac{\left(\prod_{x_j^0 2 X^0} p_{\max}(x_j^0)\right)\det(L_{X^0})\frac{1}{k}p_{\max}(x_i)}{\left(\prod_{x_j 2 X} p_{\max}(x_j)\right)\det(L_X)\frac{1}{k}p_{\max}(x_i^0)} \tag{15}$$

$$= \frac{\left(\prod_{x_j^0 2 X^0 n f x_i^0 g} p_{\max}(x_j^0)\right)\det(L_{X^0})}{\left(\prod_{x_j 2 X n f x_i g} p_{\max}(x_j)\right)\det(L_X)} = \frac{\det(L_{X^0})}{\det(L_X)} \tag{16}$$

the last simplification being allowed because $X \; n \; f x_i g = X^0 \; n \; f x_i^0 g$.

The only difference from the MH formulation of Theorem 6 is that the support of $R(X^0|X)$ is not the same of $P_{\text{DPP-TS}}$, as we disallow sampling of batches X^0 with more than one different element to X. However, since $R(X^0|X) = 0 \;\;,\;\; R(X|X^0) = 0$, we still satisfy detailed balance in all points. We can then see that k transitions are sufficient to obtain any X^0 from an existing X when $|X| = k$, and therefore our Markov Chain remains ergodic. With all conditions satisfited, even Algorithm 1 allows us to approximately sample from the true $P_{\text{DPP-TS}}$.

Algorithm 1 is a simple modification of an existing k-DPP sampler proposed by Anari et al. (2016). Furthermore, Rezaei and Gharan (2019) introduces a similar MCMC algorithm for continuous domain DPPs that performs efficiently under certain conditions. Because of its simplicity and effectiveness, Algorithm 1 is the one we use in all our experiments.

A.2 Additional Gibbs samplers

To further demonstrate the simplicity of converting existing k-DPP samplers to $P_{\text{DPP-TS}}$ samplers, we modify Li et al. (2016)'s algorithm to sample from our $P_{\text{DPP-TS}}$.

Algorithm 5 Modified Gibbs sampler from Li et al. (2016)

pick random initial batch X
repeat
  Sample b from uniform Bernoulli distribution
  if $b = 1$ **then**
    uniformly pick point $x_i \in X$ to replace
    sample candidate point $x_i^0 \quad p_{\max}(x_i^0)$
    define $X^0 = X \setminus \{x_i\} \cup \{x_i^0\}$
    accept with probability $\alpha = \frac{\det(L_{X^0})}{\det(L_{X^0}) + \det(L_X)}$
    if accepted **then**
      $X = X^0$
    end if
  end if
until converged

Repeating the same steps used for the other Single-point proposal MCMC sampler, we can check that this also satisfies detailed balance. Overall, the procedure is very similar to our preferred Algorithm 1.

Appendix B MUTUAL INFORMATION AND EXPERIMENTAL DESIGN

Modern theoretical analyses of Bayesian Optimization algorithms such as that of Srinivas et al. (2010) make use of Mutual Information and other information-theoretic quantities related to f. A comprehensive definition of such concepts is also required for our regret analysis.

The main quantity of interest in the aforementioned analysis is indeed the *Mutual Information* $I(f; \mathbf{y}_{1:T})$ between f and a set of observations $\mathbf{y}_{1:T}$ from points $X_{1:T} = \{x_1, \dots, x_T\}$, sometimes referred to as *Information Gain*. This measures the amount of information learned about the function f by observing $\mathbf{y}_{1:T}$, and for a GP it can be written as

$$I(f; \mathbf{y}_{1:T}) = H(\mathbf{y}_{1:T}) \quad H(\mathbf{y}_{1:T}|f) = \frac{1}{2} \sum_{t=1}^{T} \log\left(1 + \sigma^{-2} \sigma_t^2(x_t)\right) \tag{17}$$

$$= \frac{1}{2} \log\det\left(I + \sigma^{-2} K_{X_{1:T}}\right) \tag{18}$$

where $H(\mathbf{y}_{1:T})$ is the differential entropy of the distribution over observations $\mathbf{y}_{1:T}$, $H(\mathbf{y}_{1:T}|f)$ is the differential entropy of the observations conditioned on f, $\sigma_t^2(x_t)$ is the posterior variance over $f(x_t)$ conditioned on the partial observations $\mathbf{y}_{1:t-1}$, and $K_{X_{1:T}} = K(X_{1:T}, X_{1:T})$ is the kernel matrix for the prior GP.

The *Conditional Mutual Information* between f and observations $\mathbf{y}_{t:T}$ given previous observations $\mathbf{y}_{1:t\ \ 1}$ is then

$$I\left(f;\mathbf{y}_{t:T}j\mathbf{y}_{1:t\ \ 1}\right) = H\left(\mathbf{y}_{t:T}j\mathbf{y}_{1:t\ \ 1}\right) \quad H\left(\mathbf{y}_{t:T}jf,\mathbf{y}_{1:t\ \ 1}\right) \tag{19}$$

$$= H\left(\mathbf{y}_{t:T}j\mathbf{y}_{1:t\ \ 1}\right) \quad H\left(\mathbf{y}_{t:T}jf\right) \tag{20}$$

$$= \frac{1}{2}\sum_{t^0=t}^{T}\log\left(1+\sigma^{\ \ 2}\sigma^2_{t^0}\left(x_{t^0}\right)\right) = \frac{1}{2}\log\det\left(I+\sigma^{\ \ 2}K_{tX_{t:T}}\right) \tag{21}$$

with K_t corresponding to the kernel matrix for the posterior kernel k_t of the GP conditioned on the observations $\mathbf{y}_{1:t\ \ 1}$.

Mutual Information satisfies the property of *submodularity*, meaning that the information gain $I(f;\mathbf{y}_X j\mathbf{y}_{1:t})$ over f of observations $\mathbf{y}_X$ incurs diminishing returns when conditioned on more and more samples $\mathbf{y}_{1:t}$. Essentially, this means that $I(f;\mathbf{y}_X j\mathbf{y}_{1:t}) \quad I(f;\mathbf{y}_X j\mathbf{y}_{1:t^0})$ for any $t^0 > t$. The most information any set of observations $\mathbf{y}_X$ is able to obtain on f would be at the very beginning $I(f;\mathbf{y}_X)$, not conditioned on any previous examples. Likewise, observing specific additional data will never increase the information any future samples will obtain.

Bayesian Optimization bounds often employ the *Maximum Information Gain* γ_T with respect to f obtainable from any observation set $\mathbf{y}_X$ of size at most T:

$$\gamma_T = \max_{X2\mathcal{X},jXj\ \ T} I(f;\mathbf{y}_X). \tag{22}$$

This quantity also bounds any conditional information gain, as by submodularity $I(f;\mathbf{y}_X j\mathbf{y}_{1:t}) \quad I(f;\mathbf{y}_X)$, as previously discussed.

Appendix C NOVEL REGRET BOUNDS PROOFS

We recall the definition of Bayesian Batch Cumulative Regret for a generic Bayesian Optimization algorithm:

Definition 7 (Bayes Batch Cumulative Regret)

$$\mathrm{BBCR}_{\mathsf{algo}_{T,B}} = \mathsf{E}\left[\sum_{t=1}^{T}\min_{b2[1,B]} r_{\mathsf{algo}_{t,b}}\right] = \mathsf{E}\left[\sum_{t=1}^{T}\min_{b2[1,B]}\left(f(x^\star) \quad f(x_{\mathsf{algo}\ t,b})\right)\right]. \tag{23}$$

We make use of proof techniques from Contal et al. (2013) and Russo and Van Roy (2014) to prove a bound on Bayesian Batch Cumulative Regret for TS and DPP-TS equivalent to one obtainable by sequential full-feedback (non batched) TS, and by consequence a bound on Bayesian Simple Regret.

C.1 The BBCR Bound Proof for TS

Before proving the bound for the novel DPP-TS, we do so for regular Batched Thompson Sampling.

We first recall a few statements from Russo and Van Roy (2014), necessary to justify subsequent steps in our proof. If given the current posterior $\mathrm{GP}(\mu_t, K_t)$, we define the Upper Confidence Bound $U_t(x) = \mu_t(x) + \sqrt{\beta_t}\sigma_t(x)$ for any β_t exactly as in GP-UCB, we can show the following:

Proposition 8 (Russo and Van Roy (2014)) *For any U_t sequence defined by some β_t sequence*

$$\mathrm{BCR}_{\mathsf{TS}\ T} = \mathsf{E}\left[\sum_{t=1}^{T}\left(U_t(x_{\mathsf{TS}_t}) \quad f(x_{\mathsf{TS}_t})\right)\right] + \mathsf{E}\left[\sum_{t=1}^{T}\left(f(x^\star) \quad U_t(x^\star)\right)\right]. \tag{24}$$

This can be shown by first rewriting

$$\mathsf{E}\left[\sum_{t=1}^{T}\left(f(x^\star) \quad f(x_{\mathsf{TS}_t})\right)\right] = \mathsf{E}\left[\sum_{t=1}^{T}\left(f(x^\star) \quad U_t(x_{\mathsf{TS}_t}) + U_t(x_{\mathsf{TS}_t}) \quad f(x_{\mathsf{TS}_t})\right)\right] \tag{25}$$

$$= \mathsf{E}\left[\sum_{t=1}^{T}\left(U_t(x_{\mathsf{TS}_t}) \quad f(x_{\mathsf{TS}_t})\right)\right] + \mathsf{E}\left[\sum_{t=1}^{T}\left(f(x^\star) \quad U_t(x_{\mathsf{TS}_t})\right)\right] \tag{26}$$

and noticing that, conditioned on the history $D_{t\ 1}$, $x^\star$ and $x_{\mathsf{TS}t}$ are identically distributed and $U_t(.)$ is a deterministic function. Therefore $\mathsf{E}\left[U_t(x^\star)|D_{t\ 1}\right] = \mathsf{E}\left[U_t(x_{\mathsf{TS}t})|D_{t\ 1}\right]$, and the overall expectation over these terms maintains equality.

Russo and Van Roy (2014) the proceed to bound both components. Assuming a finite domain and using $\beta_t = 2\ln\left(\frac{B(t^2+1)|\mathcal{X}|}{\sqrt{2\pi}}\right)$, following them we obtain

$$\mathsf{E}\left[f(x^\star) \quad U_t(x^\star)\right] \quad \frac{1}{B(t^2+1)}, \qquad \mathsf{E}\left[\sum_{t=1}^{T}(f(x^\star) \quad U_t(x^\star))\right] \quad \frac{1}{B}\sum_{t=1}^{T}\left(\frac{1}{t^2+1}\right) \quad \frac{1}{B} \tag{27}$$

and

$$\mathsf{E}\left[U_t(x_{\mathsf{TS}t}) \quad f(x_{\mathsf{TS}t})\right] = \mathsf{E}\left[\sqrt{\beta_t}\sigma_t(x_{\mathsf{TS}t})\right]. \tag{28}$$

With this established, as a first step in our proof we modify Lemma 1 from Contal et al. (2013) and introduce

Lemma 9 *For finite $\mathcal{X}$ and $\beta_t = 2\ln\left(\frac{B(t^2+1)|\mathcal{X}|}{\sqrt{2\pi}}\right)$, we have*

$$\mathsf{E}\left[\min_{b\in[1,B]} r_{\mathsf{TS}t,b}\right] \quad \mathsf{E}\left[r_{\mathsf{TS}t,1}\right] \quad \frac{1}{B(t^2+1)} + \mathsf{E}\left[\sqrt{\beta_t}\sigma_{t,1}(x_{\mathsf{TS}\,t,1})\right]. \tag{29}$$

Proof. —

$$\mathsf{E}\left[\min_{b\in[1,B]} r_{\mathsf{TS}t,b}\right] \quad \mathsf{E}\left[r_{\mathsf{TS}t,1}\right] = \mathsf{E}\left[f(x^\star) \quad f(x_{\mathsf{TS}\,t,1})\right] \tag{30}$$

$$= \mathsf{E}\left[f(x^\star) \quad U_{t,1}(x^\star) + U_{t,1}(x_{\mathsf{TS}\,t,1}) \quad f(x_{\mathsf{TS}\,t,1})\right] \tag{31}$$

$$\frac{1}{B(t^2+1)} + \mathsf{E}\left[\sqrt{\beta_t}\sigma_{t,1}(x_{\mathsf{TS}\,t,1})\right]. \tag{32}$$

Step (31) can be performed due to Russo and Van Roy's Proposition 8, by adding and subtracting the UCB $U_{t,1}(x_{\mathsf{TS}\,t,1}) = \mu_{t,1}(x_{\mathsf{TS}\,t,1}) + \sqrt{\beta_t}\sigma_{t,1}(x_{\mathsf{TS}\,t,1})$ and noticing that, conditioned on the history $D_{t\ 1}$, $x^\star$ and $x_{\mathsf{TS}t,1}$ are identically distributed and $U_{t,1}(.)$ is a deterministic function. Therefore $\mathsf{E}\left[U_{t,1}(x^\star)|D_{t\ 1,B}\right] = \mathsf{E}\left[U_{t,1}(x_{\mathsf{TS}t,1})|D_{t\ 1,B}\right]$.

To obtain step (32), we then separate the terms from Equation (31) into $\mathsf{E}\left[f(x^\star) \quad U_{t,1}(x^\star)\right]$ and $\mathsf{E}\left[U_{t,1}(x_{\mathsf{TS}\,t,1}) \quad f(x_{\mathsf{TS}\,t,1})\right]$, bounding the first with Equation (27) and the second with Equation (28).

After proving this essential Lemma, we proceed with adapting Lemma 2 from Contal et al. (2013). Unlike them, we need not bother with guarantees about a maximizer high probability region R^+ as the one defined for GP-UCB-PE, as we are operating in expectation.

Lemma 10 *In expectation, the deviation of the first point within a batch selected by TS is bounded by the one for any point within the previous batch selected by TS, thus*

$$\mathsf{E}\left[\sigma_{t+1,1}(x_{\mathsf{TS}\ t+1,1})\right] \quad \mathsf{E}\left[\sigma_{t,b}(x_{\mathsf{TS}\ t,b})\right] \qquad \forall t \in [1, T \quad 1],\ \forall b \in [1, B]. \tag{33}$$

Proof. — For any time t, for every step (t, b) within the batch, the points $x_{\mathsf{TS}\ t,b}$ for TS are independently sampled from $P_{\mathsf{max}\ t,1}$, which depends on history up to $D_{t\ 1,B}$. Therefore, given $D_{t,b\ 1}$, $\sigma_{t,b}(x)$ is a deterministic function, and $x_{\mathsf{TS}\ t,b}$ and the true $x^\star$ have the same distribution. We thus have that $\forall t \in [1,T]$, $\forall b \in [1,B]$:

$$\mathsf{E}\left[\sigma_{t,b}(x_{\mathsf{TS}\ t,b})\right] = \mathsf{E}\left[\mathsf{E}\left[\sigma_{t,b}(x_{\mathsf{TS}\ t,b})\middle| D_{t,b\ 1}\right]\right] \tag{34}$$

$$= \mathsf{E}\left[\mathsf{E}\left[\sigma_{t,b}(x^\star)\middle| D_{t,b\ 1}\right]\right] = \mathsf{E}\left[\sigma_{t,b}(x^\star)\right] \tag{35}$$

Because of the law of non-increasing variance (Rasmussen and Williams, 2005), we have that

$$\mathsf{E}\left[\sigma_{t+1,1}(x^\star)\right] \quad \mathsf{E}\left[\sigma_{t,b}(x^\star)\right] \qquad \forall t \in [1, T \quad 1],\ \forall b \in [1, B] \tag{36}$$

and therefore:

$$\mathsf{E}\left[\sigma_{t+1,1}(x_{\mathsf{TS}\ t+1,1})\right] \quad \mathsf{E}\left[\sigma_{t,b}(x_{\mathsf{TS}\ t,b})\right] \qquad \forall t \in [1, T \quad 1],\ \forall b \in [1, B]. \tag{37}$$

We can then introduce

Lemma 11 *In expectation, the sum of deviations for the first points of all batches selected by TS is bounded by the sum of deviations for all points selected by TS, divided by B.*

$$\mathsf{E}\left[\sum_{t=1}^{T}\sigma_{t,1}(x_{\mathsf{TS}\ t,1})\right] \quad \mathsf{E}\left[\frac{1}{B}\sum_{t=1}^{T}\sum_{b=1}^{B}\sigma_{t,b}(x_{\mathsf{TS}\ t,b})\right] \tag{38}$$

Proof. — For all t, using Lemma 10 and summing over b, we can get

$$\mathsf{E}\left[\sigma_{t,1}(x_{\mathsf{TS}\ t,1}) + (B \quad 1)\sigma_{t+1,1}(x_{\mathsf{TS}\ t+1,1})\right] \quad \mathsf{E}\left[\sigma_{t,1}(x_{\mathsf{TS}\ t,1}) + \sum_{b=2}^{B}\sigma_{t,b}(x_{\mathsf{TS}\ t,b})\right] \tag{39}$$

Summing both sides over t and dividing by B, we obtain the desired result.

Lemma 12 *Assuming without loss of generality that, for all t and b, $\left(\sigma_{t,b}(x_{\mathsf{TS}\ t,b})\right)^2 \quad 1$, the sum of variances of the points selected by TS is bounded by a constant factor times γ_{TB}:*

$$\sum_{t=1}^{T}\sum_{b=1}^{B}\left(\sigma_{t,b}(x_{\mathsf{TS}\ t,b})\right)^2 \quad C_2\gamma_{TB} \tag{40}$$

with $C_2 = 2/\log(1+\sigma^{\ 2})$ and γ_{TB} being the maximum information gain on f from TB observations as defined in Appendix B.

Proof. — The information gain on f from a sequence of TB observations can be expressed in terms of the posterior variances

$$I\left(f(x_{1:T,1:B}); \mathbf{y}_{1:T,1:B}\right) = \frac{1}{2}\sum_{t=1}^{T}\sum_{b=1}^{B}\log\left(1+\sigma^{\ 2}\left(\sigma_{t,b}(x_{t,b})\right)^2\right) \tag{41}$$

as seen in Appendix B, and is bounded by γ_{TB} by definition. We can then obtain, thanks to the bounded variance assumption:

$$\sum_{t=1}^{T}\sum_{b=1}^{B}\left(\sigma_{t,b}(x_{\mathsf{TS}\ t,b})\right)^2 \quad \sum_{t=1}^{T}\sum_{b=1}^{B}\frac{1}{\log(1+\sigma^{\ 2})}\log\left(1+\sigma^{\ 2}\left(\sigma_{t,b}(x_{\mathsf{TS}\ t,b})\right)^2\right) \tag{42}$$

$$= \frac{2}{\log(1+\sigma^{\ 2})}\, I\left(f(x_{\mathsf{TS}\ 1:T,1:B}); \mathbf{y}_{1:T,1:B}\right). \tag{43}$$

Finally, we can conclude by introducing our Bayesian Batch Cumulative Regret bound.

Theorem 13 (Bayes Batch Cumulative Regret Bound for Batched Thompson Sampling) *If f $GP(0, K)$ with covariance kernel bounded by 1 and noise model $\mathsf{N}(0, \sigma^2)$, and either*

- *Case 1: finite X and $\beta_t = 2\ln\left(\frac{B(t^2+1)j\mathsf{X}j}{\sqrt{2\pi}}\right)$;*

- *Case 2: compact and convex* $X \subset [0,l]^d$*, with Assumption 3 satisfied and* $\beta_t = 4(d+1)\log(Bt) + 2d\log(dab\sqrt{\pi})$.

Then Batched Thompson Sampling attains Bayes Batch Cumulative Regret of

$$\mathrm{BBCR}_{\mathrm{TS}T,B} \quad \frac{C_1}{B} + \sqrt{C_2 \frac{T}{B} \beta_T \gamma_{TB}} \tag{44}$$

with $C_1 = 1$ *for Case 1,* $C_1 = \frac{\pi^2}{6} + \frac{\sqrt{2\pi}}{12}$ *for Case 2, and* $C_2 = \frac{2}{\log(1+\sigma^{-2})}$.

Proof. — Using the previous lemmas together with Russo and Van Roy inequalities, we can show for Case 1:

$$\mathrm{BBCR}_{\mathrm{TS}T,B} = \mathbb{E}\Big[\sum_{t=1}^{T} \min_{b \in [1,B]} r_{\mathrm{TS}t,b}\Big] \quad \mathbb{E}\Big[\sum_{t=1}^{T} r_{\mathrm{TS}t,1}\Big] \tag{45}$$

$$\sum_{t=1}^{T} \frac{1}{B(t^2+1)} + \mathbb{E}\Big[\sum_{t=1}^{T} \sqrt{\beta_t}\sigma_{t,1}(x_{\mathrm{TS}t,1})\Big] \quad \text{by Lemma 9} \tag{46}$$

$$\frac{C_1}{B} + \mathbb{E}\Big[\sqrt{\beta_T}\frac{1}{B}\sum_{t=1}^{T}\sum_{b=1}^{B}\sigma_{t,b}(x_{t,b})\Big] \quad \text{by Eq. (27) and Lemma 11} \tag{47}$$

$$\frac{C_1}{B} + \mathbb{E}\Big[\sqrt{\beta_T}\frac{1}{B}\sqrt{TB\sum_{t=1}^{T}\sum_{b=1}^{B}(\sigma_{t,b}(x_{t,b}))^2}\Big] \quad \text{by Cauchy-Schwartz} \tag{48}$$

$$\frac{C_1}{B} + \sqrt{C_2 \frac{T}{B} \beta_T \gamma_{TB}} \quad \text{by Lemma 12} \tag{49}$$

For Case 2, we simply modify the steps of Lemma 9 with the corresponding inequalities used by Kandasamy et al. (2018) for their continuous-domain version of the bound.

The bound we just derived scales equivalently to the bound obtainable by standard sequential TS with full feedback. This is in contrast to the bound previously obtained by Kandasamy et al. (2018) for Batched TS without initialization:

$$\mathrm{BBCR}_{\mathrm{TS}T,B} \quad \frac{C_1}{B} + \sqrt{C_2 \frac{T}{B} \exp(C) \beta_T \gamma_{TB}} \tag{50}$$

which depends on an additional factor $\exp(C)$ dependent on B, rendering the bound not convergent in B unless a wasteful initialization procedure is performed before TS.

C.2 The BBCR Bound Proof for DPP-TS

We now consider DPP-TS again, and strive to obtain an equivalent bound.

In order to do so, we must modify the algorithm we developed and introduce DPP-TS-alt, so that for every batch:

- For the first sample in the batch $x_{\text{DPP-TS-alt}\,t,1}$, we sample from $p_{\max t}$ as in standard Thompson Sampling;
- For all the other samples $x_{\text{DPP-TS-alt}\,t,b}$ with $b \in [2,B]$, we sample from joint $P_{\text{DPP-TS}\,t}$, using the most updated posterior variance matrix $K_{t,1}$ to define the DPP kernel.

We begin by noting that Lemma 9 is applicable to DPP-TS-alt as well, as $x_{\text{DPP-TS-alt}\,t,1}$ behaves exactly in the same way as $x_{\text{TS}\,t,1}$, since DPP-TS-alt has been explicitly defined as using standard Thompson Sampling for the first sampled point of every batch.

Then, we must translate Lemma 10 to DPP-TS-alt as well, which requires a more involved proof. In fact, we must split the undertaking in three preliminary lemmas.

First, when considering the batch sampled at time t, we introduce for sake of argument an *alternative point* $\tilde{x}_{t,B}$ to take the place of the last sampled point of the batch $x_{\text{DPP-TS-alt } t,B}$. The original point $x_{\text{DPP-TS-alt } t,B}$ is sampled from the DPP, and is distributed as $x_{\text{DPP-TS-alt } t,B} \quad P_{\text{DPP-TS } t}(x_{t,B} | x_{t,1}, \dots, x_{t,B\ 1}) = p_{\text{DPP-TS } t,B}(x_{t,B})$ when conditioned on the previous points of the batch. Instead, we define the replacement as $\tilde{x}_{t,B} \quad p_{\max\ t,1}$, sampled from standard Thompson Sampling with the posterior available from observations up to $(t, 1)$.

Lemma 14 *At time t, let $x_{\text{DPP-TS-alt } t,B}$ be the last point of the batch chosen by DPP-TS-alt. Let us in its place define an alternative element $\tilde{x}_{t,B}$ obtained by following the DPP-TS-alt procedure up to step $(t, B \quad 1)$ and then sampling using regular TS from the available maximizer posterior distribution $p_{\max\ t,1}$ instead of from conditioned $p_{\text{DPP-TS } t,B}$.*

We can then show that

$$\mathsf{E}\left[\sigma_{t+1,1}(x_{\text{DPP-TS-alt } t+1,1})\right] \quad \mathsf{E}\left[\sigma_{t,B}(\tilde{x}_{t,B})\right]. \tag{51}$$

Proof. — Equations (34) and (36) are still valid for $\tilde{x}_{t,B}$ (being sampled from TS), as conditioned on history $D_{t,B\ 1}$, $\tilde{x}_{t,B}$ has the same distribution of $x^\star$, and so we obtain that

$$\mathsf{E}\left[\sigma_{t+1,1}(x_{\text{DPP-TS-alt } t+1,1})\right] = \mathsf{E}\left[\sigma_{t+1,1}(x^\star)\right] \tag{52}$$

$$\mathsf{E}\left[\sigma_{t,B}(x^\star)\right] = \mathsf{E}\left[\mathsf{E}\left[\sigma_{t,B}(\tilde{x}_{t,B}) \middle| D_{t,B\ 1}\right]\right] \tag{53}$$

Lemma 15 *Given $\tilde{x}_{t,B}$ defined as in Lemma 14, we have that*

$$\mathsf{E}\left[\sigma_{t,B}(\tilde{x}_{t,B})\right] \quad \mathsf{E}\left[\sigma_{t,B}(x_{\text{DPP-TS-alt } t,B})\right]. \tag{54}$$

Proof. — To prove the lemma, we first observe that (from Appendix B)

$$\det\left(I + \sigma^{\ 2} K_{t X_{1:B}}\right) = \prod_{b=1}^{B} \left(1 + \sigma^{\ 2} \sigma_b^2(x_b)\right). \tag{55}$$

We then obtain the marginal distribution of the last point of a DPP-TS batch by summing over the domain:

$$p_{\text{DPP-TS } t,B}(x_{t,B}) = \sum_{(x_{t,1},\dots,x_{t,B\ 1}) 2 \mathcal{X}^{B\ 1}} P_{\text{DPP-TS } t}(x_{t,1}, \dots, x_{t,B\ 1}, x_{t,B}) \tag{56}$$

$$/ \sum_{(x_{t,1},\dots,x_{t,B\ 1}) 2 \mathcal{X}^{B\ 1}} \left(\left(\prod_{b=1,\dots,B} p_{\max t,1}(x_{t,b}) \right) \det\left(I + \sigma^{\ 2} K_{t X_{1:B}}\right) \right) \tag{57}$$

$$/ \sum_{(x_{t,1},\dots,x_{t,B\ 1}) 2 \mathcal{X}^{B\ 1}} \left(\prod_{b=1,\dots,B} p_{\max t,1}(x_{t,b}) \left(1 + \sigma^{\ 2} \sigma_b^2(x_{t,b})\right) \right) \tag{58}$$

$$= p_{\max t,1}(x_{t,B}) \tag{59}$$

$$\left(1 + \sigma^{\ 2} \sigma_B^2(x_{t,B})\right) \sum_{(x_{t,1},\dots,x_{t,B\ 1}) 2 \mathcal{X}^{B\ 1}} \left(\prod_{b=1,\dots,B\ 1} p_{\max t,1}(x_{t,b}) \left(1 + \sigma^{\ 2} \sigma_b^2(x_{t,b})\right) \right) \tag{60}$$

$$/\ p_{\max t,1}(x_{t,B}) \left(1 + \sigma^{\ 2} \sigma_B^2(x_{t,B})\right). \tag{61}$$

We can then show that

$$p_{\text{DPP-TS } t,B}(x) = \frac{1 + \sigma^{\ 2} \sigma_{t,B}^2(x)}{\sum_{x^0 2 \mathcal{X}} p_{\max t,1}(x^0)(1 + \sigma^{\ 2} \sigma_{t,B}^2(x^0))} p_{\max t,1}(x) \tag{62}$$

$$= \frac{1 + \sigma^{\ 2} \sigma_{t,B}^2(x)}{\mathsf{E}_{p_{\max t,1}}\left[1 + \sigma^{\ 2} \sigma_{t,B}^2(x^0)\right]} p_{\max t,1}(x) \tag{63}$$

$$= (1 + \delta(x)) p_{\max t,1}(x) \tag{64}$$

with $\delta(x) = \frac{\sigma^{-2}\sigma^2_{t,B}(x) \quad \mathbb{E}_{P_{\max t,1}}\left[\sigma^{-2}\sigma^2_{t,B}(x^0)\right]}{\mathbb{E}_{P_{\max t,1}}\left[1+\sigma^{-2}\sigma^2_{t,B}(x^0)\right]}$.

As both $p_{\mathsf{max}\,t,1}$ and $p_{\mathsf{DPP\text{-}TS}\,t,B}$ are distributions, we have that

$$\sum_{x \in X} p_{\mathsf{max}\,t,1}(x) = 1 \tag{65}$$

$$\sum_{x \in X} p_{\mathsf{DPP\text{-}TS}\,t,B}(x) = \sum_{x \in X}(1+\delta(x))p_{\mathsf{max}\,t,1}(x) = 1 \tag{66}$$

and therefore

$$\sum_{x \in X} \delta(x) p_{\mathsf{max}\,t,1}(x) = 0 \tag{67}$$

We can rewrite $\mathsf{E}_{p_{\mathrm{DPP\text{-}TS}\,t,B}}[\sigma_{t,B}(x)]$ as

$$\mathsf{E}_{p_{\mathrm{DPP\text{-}TS}\,t,B}}\left[\sigma_{t,B}(x)\right] = \sum_{x \in X}(1+\delta(x))p_{\mathsf{max}\,t,1}(x)\sigma_{t,B}(x) \tag{68}$$

$$= \mathsf{E}_{p_{\max t,1}}\left[\sigma_{t,B}(x)\right] + \sum_{x \in X}\delta(x)p_{\mathsf{max}\,t,1}(x)\sigma_{t,B}(x) \tag{69}$$

and knowing that by definition of $\delta(x)$

$$\delta(x) \quad 0 \quad , \quad \sigma^{-2}\sigma^2_{t,B}(x) \quad \mathsf{E}_{p_{\max t,1}}\left[\sigma^{-2}\sigma^2_{t,B}(x^0)\right] \tag{70}$$

$$\sigma_{t,B}(x) \quad \sqrt{\mathsf{E}_{p_{\max t,1}}\left[\sigma^2_{t,B}(x^0)\right]} \tag{71}$$

we can see that

$$\begin{aligned}
&\sum_{x \in X}\delta(x)p_{\mathsf{max}\,t,1}(x)\sigma_{t,B}(x) \\
&= \sum_{\substack{x \in X \\ \delta(x) \quad 0}}\delta(x)p_{\mathsf{max}\,t,1}(x)\sigma_{t,B}(x) + \sum_{\substack{x \in X \\ \delta(x)<0}}\delta(x)p_{\mathsf{max}\,t,1}(x)\sigma_{t,B}(x) \\
&\left(\sum_{x \in X}\delta(x)p_{\mathsf{max}\,t,1}(x)\right)\sqrt{\mathsf{E}_{p_{\max t,1}}\left[\sigma^2_{t,B}(x^0)\right]} = 0.
\end{aligned} \tag{72}$$

We combine this with Equation (69) to prove that

$$\mathsf{E}\left[\mathsf{E}\left[\sigma_{t,B}(\tilde{x}_{t,B})\middle| D_{t,B\ \ 1}\right]\right] \quad \mathsf{E}\left[\mathsf{E}\left[\sigma_{t,B}(x_{\mathsf{DPP\text{-}TS\text{-}alt}\,t,B})\middle| D_{t,B\ \ 1}\right]\right], \tag{73}$$

hence $p_{\mathsf{DPP\text{-}TS}\,t,B}$, the conditional distribution of the last point of the batch, is a reweigthing of $p_{\mathsf{max}\,t,1}$ in which more probability mass is put on points with higher posterior variance $\sigma^2_{t,B}(x)$.

Lemma 16 *For any timestep t and for all $b \in [3,B]$, we have that*

$$\mathsf{E}\left[\sigma_{t,b}(x_{\mathsf{DPP\text{-}TS\text{-}alt}\,t,b})\right] \quad \mathsf{E}\left[\sigma_{t,b\ \ 1}(x_{\mathsf{DPP\text{-}TS\text{-}alt}\,t,b\ \ 1})\right]. \tag{74}$$

Proof. — Given any timestep t and any $b \in [3,B]$, we first condition on history $D_{t,b\ \ 2}$, and take $P_{\mathsf{DPP\text{-}TS}\,t,b\ \ 1}$ as the conditional joint distribution for selecting the remaining points of the batch with $b^0 \in [b \quad 1, B]$ for DPP-TS-alt. Because of the conditioning, the distribution is defined on deterministic quantities only.

We can see that $P_{\mathsf{DPP\text{-}TS}\,t,b\ \ 1}(X) \ / \ P_{\mathsf{max}\,t,1}(X)\det(L_{t,b\ \ 1\,X})$ (defined following a similar argument as that at the beginning of the proof for Lemma 15), rewritten as a joint distribution $P_{\mathsf{DPP\text{-}TS}\,t,b\ \ 1}(x_{b\ \ 1},\ldots,x_B)$, has the property of exchangeability with respect to points of the batch $X = (x_{b\ \ 1},\ldots,x_B)$, meaning that

changing the order of the points within the batch will not affect the probability $P_{\text{DPP-TS}\ t,b\ \ 1}(X)$. This is true as $P_{\max t,1}(X) = \prod_{b'=b\ \ 1}^{B} p_{\max t,1}(x_b')$ depends on each x_b' independently, and the DPP term $\det(L_{t,b\ \ 1X})$ is clearly exchangeable for any valid kernel $L_{t,b\ \ 1}$, regardless of regularization by I.

Therefore, any two points $x_{\text{DPP-TS-alt}\ t,b}$ and $x_{\text{DPP-TS-alt}\ t,b\ \ 1}$ within the batch have the same marginal distribution. This is true because of exchangeability, as integrating all other points must lead to the same result regardless of the position of the marginalized point within the batch. Furthermore, $\sigma_{t,b\ \ 1}$ is a deterministic function given $D_{t,b\ \ 2}$, and so:

$$\mathsf{E}\left[\sigma_{t,b\ \ 1}(x_{\text{DPP-TS-alt}\ t,b})\middle| D_{t,b\ \ 2}\right] = \mathsf{E}\left[\sigma_{t,b\ \ 1}(x_{\text{DPP-TS-alt}\ t,b\ \ 1})\middle| D_{t,b\ \ 2}\right]. \tag{75}$$

By the law of non-decreasing variance (Rasmussen and Williams, 2005), we also have that

$$\mathsf{E}\left[\sigma_{t,b}(x_{\text{DPP-TS-alt}\ t,b})\middle| D_{t,b\ \ 2}\right] \quad \mathsf{E}\left[\sigma_{t,b\ \ 1}(x_{\text{DPP-TS-alt}\ t,b})\middle| D_{t,b\ \ 2}\right]. \tag{76}$$

Combining Equations (75) and (76), we finally obtain

$$\mathsf{E}\left[\sigma_{t,b}(x_{\text{DPP-TS-alt}\ t,b})\middle| D_{t,b\ \ 2}\right] \quad \mathsf{E}\left[\sigma_{t,b\ \ 1}(x_{\text{DPP-TS-alt}\ t,b\ \ 1})\middle| D_{t,b\ \ 2}\right], \tag{77}$$

and taking the overall expectation over both sides yields the lemma.

Finally, we can now obtain the promised lemma:

Lemma 17 *In expectation, when using DPP-TS-alt, the deviation of the first point of a batch, selected by standard TS, is bounded by the one for any point within the previous batch, selected by DPP-TS, thus*

$$\mathsf{E}\left[\sigma_{t+1,1}(x_{\text{DPP-TS-alt}\ t+1,1})\right] \quad \mathsf{E}\left[\sigma_{t,b}(x_{\text{DPP-TS-alt}\ t,b})\right] \quad \forall t \in [1, T \quad 1],\ \forall b \in [2, B]. \tag{78}$$

Proof. — Combine Lemmas 14, 15 and 16 to obtain the inequality.

Lemma 11 from the TS proof follows from lemma 17, and lemma 12 applies to the DPP-TS-alt as well without any further adjustments. It's then a matter of combining the new results to get the final bound.

Theorem 18 (Bayes Batch Cumulative Regret Bound for DPP-TS) *If $f \quad GP(0, K)$ with covariance kernel bounded by 1 and noise model $\mathcal{N}(0, \sigma^2)$, and either*

- *Case 1: finite $\mathcal{X}$ and $\beta_t = 2\ln\left(\frac{B(t^2+1)j\mathcal{X}j}{\sqrt{2\pi}}\right)$;*
- *Case 2: compact and convex $\mathcal{X} \subset [0, l]^d$, with Assumption 3 satisfied and $\beta_t = 4(d+1)\log(Bt) + 2d\log(dab\sqrt{\pi})$.*

Then DPP-TS (in its DPP-TS-alt variant) attains Bayes Batch Cumulative Regret of

$$\text{BBCR}_{\text{DPP-TS}T,B} \quad \frac{C_1}{B} + \sqrt{C_2 \frac{T}{B} \beta_T \gamma_{TB}} \quad C_3 \tag{79}$$

with $C_1 = 1$ for Case 1, $C_1 = \frac{\pi^2}{6} + \frac{\sqrt{2\pi}}{12}$ for Case 2, $C_2 = \frac{2}{\log(1+\sigma^{\ 2})}$ and $C_3 \quad 0$.

Proof. — Using the previous lemmas together with Russo and Van Roy inequalities, we can show for Case 1:

$$\mathrm{BBCR}_{\text{DPP-TS}T,B} = \mathsf{E}\left[\sum_{t=1}^{T} \min_{b2[1,B]} r_{\text{DPP-TS}t,b}\right] \quad \mathsf{E}\left[\sum_{t=1}^{T} r_{\text{DPP-TS}t,1}\right] \tag{80}$$

$$\sum_{t=1}^{T} \frac{1}{B(t^2+1)} + \mathsf{E}\left[\sum_{t=1}^{T} \sqrt{\beta_t}\sigma_{t,1}(x_{\text{DPP-TS}t,1})\right] \quad \text{by Lemma 9} \tag{81}$$

$$\frac{C_1}{B} + \mathsf{E}\left[\sqrt{\beta_T}\frac{1}{B}\sum_{t=1}^{T}\sum_{b=1}^{B}\sigma_{t,b}(x_{t,b})\right] \quad \text{by Eq. (27) and Lemma 11} \tag{82}$$

$$\frac{C_1}{B} + \mathsf{E}\left[\sqrt{\beta_T}\frac{1}{B}\sqrt{TB\sum_{t=1}^{T}\sum_{b=1}^{B}(\sigma_{t,b}(x_{t,b}))^2}\right] \quad \text{by Cauchy-Schwartz} \tag{83}$$

$$\frac{C_1}{B} + \sqrt{C_2\frac{T}{B}\beta_T\gamma_{TB}} \quad \text{by Lemma 12} \tag{84}$$

For Case 2, we again simply modify the steps of Lemma 9 with the corresponding inequalities used by Kandasamy et al. (2018) for their continuous-domain version of the bound.

We have shown that the Bayesian bound for DPP-TS is at least as good as that of standard Batched TS. In fact, the bound is even better than for that for TS, as we can add negative factors $C_{3t} = \mathsf{E}\left[\delta(x)\sigma_{t,B}(x)\right]$ at every iteration t, leftover from Equation 69 in lemma 15. From this we obtain the negative factor C_3.

Appendix D DISCUSSION OF KATHURIA ET AL. (2016)'S BOUND

Despite their meaningful insight and better practical performance compared to Contal et al. (2013)'s original GP-UCB-PE, Kathuria et al. (2016)'s proposed regret bound for UCB-DPP-SAMPLE does not improve on existing bounds, as it is founded on bounding the expected information gain from the last $k = B$ 1 points of every batch:

$$\mathsf{E}_{S \ k\text{-DPP}(L_{t,1})}\left[\log\det((L_{t,1})_S)\right] \tag{85}$$

$$= \sum_{jSj=k} \frac{\det((L_{t,1})_S)\log(\det((L_{t,1})_S))}{\sum_{jSj=k}\det((L_{t,1})_S)} \tag{86}$$

$$= \sum_{jSj=k} \frac{\det((L_{t,1})_S)\log(\frac{\det((L_{t,1})_S)}{\sum_{jSj=k}\det((L_{t,1})_S)})}{\sum_{jSj=k}\det((L_{t,1})_S)} + \sum_{jSj=k} \frac{\det((L_{t,1})_S)\log(\sum_{jSj=k}\det((L_{t,1})_S))}{\sum_{jSj=k}\det((L_{t,1})_S)} \tag{87}$$

$$= H(k\text{-DPP}(L_{t,1})) + \log\left(\sum_{jSj=k}\det((L_{t,1})_S)\right) \tag{88}$$

$$H(k\text{-DPP}(L_{t,1})) + \log\left(jXj^k \max\det((L_{t,1})_S)\right) \tag{89}$$

$$H(k\text{-DPP}(L_{t,1})) + k\log\left(jXj\right) + \log\left(\max\det((L_{t,1})_S)\right) \tag{90}$$

When summing this over all iterations T, the last term is bounded by some $C^0\gamma_{TB}$ with C^0 1, while the first two terms summed together are trivially positive, as $H(k\text{-DPP}(L_{t,1}))$ $k\log\left(jXj\right)$. We then have that $\sum_{t=1}^{T}($ $H(k\text{-DPP}(L_{t,1})) + k\log\left(jXj\right)) + C^0\gamma_{TB}$ γ_{TB}, therefore we gain nothing as opposed to just bounding with the maximum information gain γ_{TB}. For this reason, the bound is always looser than the original GP-UCB-PE bound.

Appendix E ADDITIONAL EXPERIMENTS

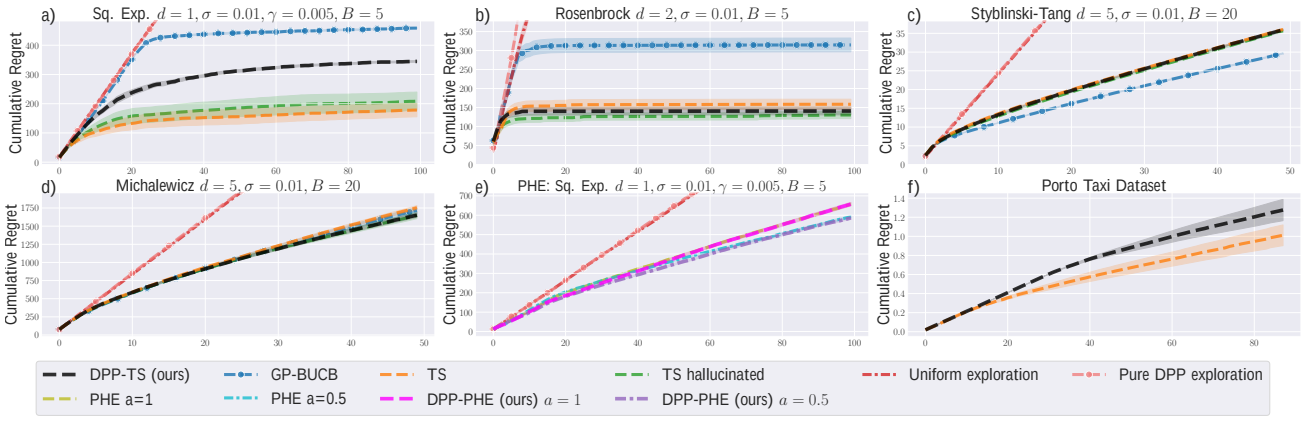

Figure 3: Comprehensive experimental comparisons between DPP-TS and classic BBO techniques for Cumulative Regret, featuring the same experimental settings as those shown in Figure 2. DPP-TS is no longer the best performing algorithm according to this metric, as the DPP component favors additional and potentially suboptimal exploration through batch diversification. However, even when it is not better, DPP-TS still quickly converges to asymptotic behavior identical to that of TS. In Appendix E.2 we illustrate a method for overcoming DPP-TS's limitations on Cumulative Regret.

E.1 Cumulative Regret

When performing the experiments from Section 6 we mainly track Simple Regret, our target metric of choice on which we prove our theoretical bounds. Optimizing for Simple Regret corresponds to searching for good maximizers and heavily favors exploration, therefore we heavily benefit from DPP-BBO's batch diversification properties. However, we still wish to track the classic Cumulative Regret performance of our algorithms, in order to gain insight into whether DPP-BBO can be used to also optimize for such a metric.

Overall, when compared to classic TS and hallucinated TS on Cumulative Regret as seen in Figure 3, DPP-TS is no longer the best performing algorithm, often over-exploring at the beginning, but still quickly converging to sublinearity. In other cases, its performance is virtually identical. For those situations in which the DPP component causes excessive exploration, we propose a solution (in Appendix E.2) involving limiting the use of DPP-TS to an *initialization phase*.

E.2 λ-parametrized DPP kernel

In order to explicitly control the degree of exploration induced by the DPP reweighting of P_{max}, we can parametrize our sampling distribution $P_{\mathsf{DPP\text{-}TS}}$ with a λ exploration parameter.

Using $\lambda \in [0, 1]$, we would like a parametrization such that:

- For $\lambda = 1$ we recover the original formulation from Definition 2: $P_{\mathsf{DPP\text{-}TS}}(X) \propto P_{\mathsf{max}\,t}(X) \det(I + \sigma^{-2} K_{tX})$;
- For $\lambda = 0$ we obtain regular Thompson Sampling P_{max};

A proposal for such a $P_{\mathsf{DPP\text{-}TS}}$ parametrization is to use a multiplicative λ, such as

$$P_{\mathsf{DPP\text{-}TS}}(X) \propto P_{\mathsf{max}\,t}(X) \det(I + \lambda\sigma^{-2} K_{tX}) \tag{91}$$

In Figure 4 we illustrate an experiment comparing TS and parametrized DPP-TS with different values for λ. It is straightforward to observe that by interpolating between TS and DPP-TS we observe a tradeoff in Simple Regret against Cumulative Regret performance. Smaller values of λ correspond to slower convergence in Simple Regret but overall lower Cumulative Regret.

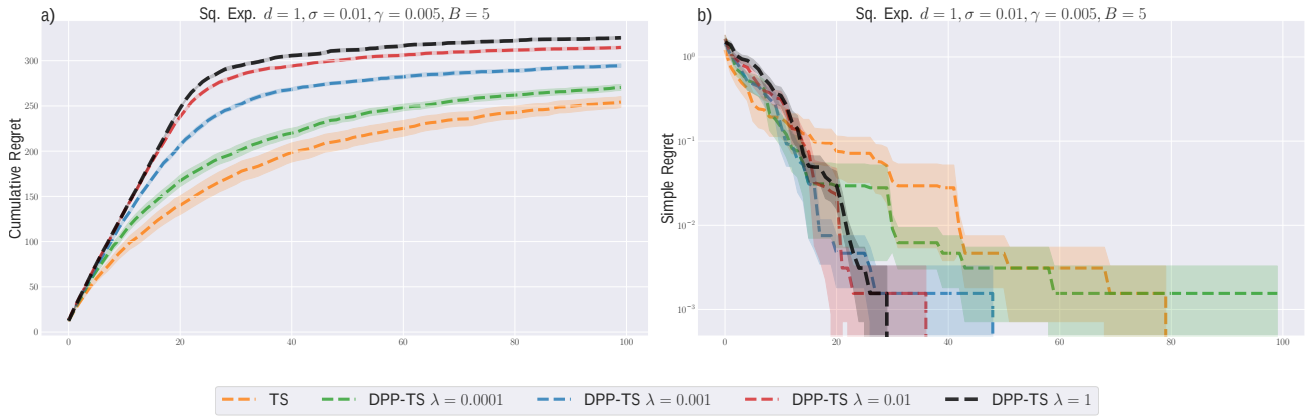

Figure 4: Experimental comparison of DPP-TS with different λ parameter in the parametrized mutual information DPP kernel, for both Cumulative and Simple Regret. Changing λ corresponds to interpolating between the behaviors of TS and DPP-TS, with TS favoring Cumulative Regret and DPP-TS favoring Simple Regret.

Moreover, we investigate the use of time-varying λ_t for the purpose of using DPP-TS as an *initialization* phase for regular TS. Figure 5 illustrates a successful example of such a procedure: λ_t is set to be equal to 1 (equivalent to the original DPP-TS formulation) up until iteration T_{init}, then equal to 0 (equivalent to TS). By doing this, we can constrain the DPP-TS over-exploration behavior to the very first batches we evaluate, and obtain both lower/equivalent Cumulative Regret than regular TS and lower Simple Regret, as seen in the cases for $T_{\text{init}} = 15$ and $T_{\text{init}} = 24$.

The experiments for this Section all involve 1-d true functions sampled from a Gaussian Process with Squared Exponential kernel and $\gamma = 0.005$, evaluated on a discrete grid of 1024 points, with observational noise of $\sigma = 0.01$. The algorithms use a correctly-specified internal GP model with the same parameters of the true GP prior, and batch size is $B = 5$.

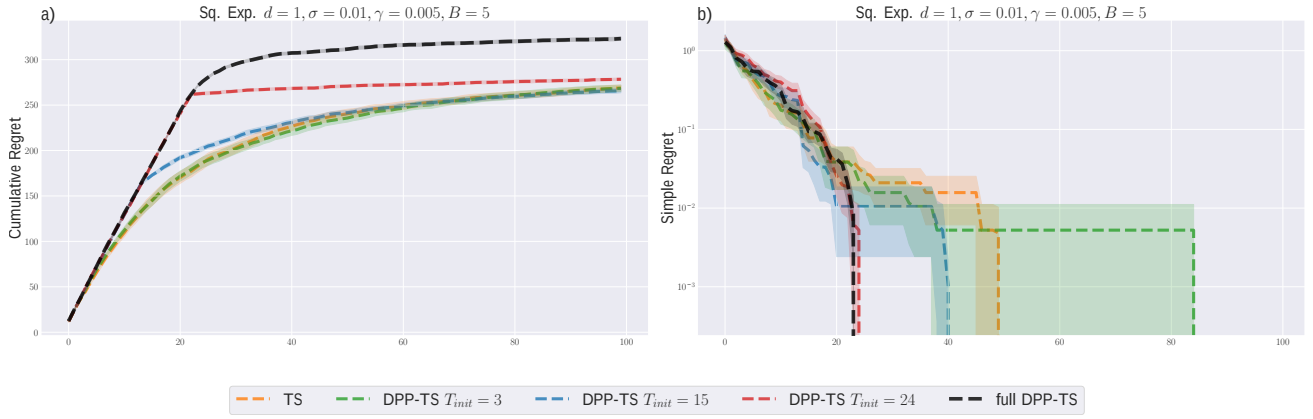

Figure 5: Experimental comparison of DPP-TS when used as an initialization scheme for the first T_{init} iterations, for both Cumulative and Simple Regret. It is apparent how, with properly tuned T_{init}, it is possible to maintain the fast Simple Regret convergence properties of DPP-TS while not overshooting classic TS in Cumulative Regret.

E.3 DPP-TS and DPP-TS-alt comparison

Throughout this work, we refer to DPP-TS as the procedure formalized in Algorithm 2, which is a simple and natural diversifying extension of classic TS. However, the algorithm we prove our theoretical Bayes Simple Regret bound on with Theorem 5 is a slightly different procedure which we name DPP-TS-alt, that differs from DPP-TS in that it selects the first point of every batch with standard TS, a technicality required for the proof technique to work. As mentioned in the main text, the reason why we introduced DPP-TS as such and not DPP-TS-alt in the first place is both for simplicity and because in practice their performance is virtually identical.

In Figure 6 we illustrate an experiment comparing DPP-TS and DPP-TS-alt (with classic TS for reference),

showing performance of DPP-TS and DPP-TS-alt to be equivalent in both Cumulative and Simple Regret.

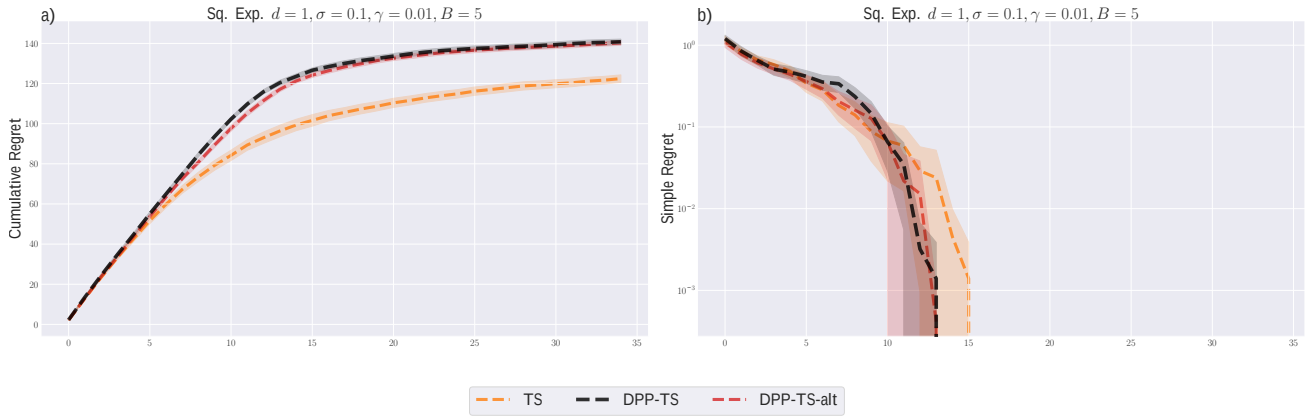

Figure 6: Experimental comparison of DPP-TS and DPP-TS-alt, for both Cumulative and Simple Regret.