A. Some Auxiliary Material

A.1. Review of the GP-UCB Algorithm

In this subsection we present a brief summary of the **GP-UCB** algorithm in (Srinivas et al., 2010). The algorithm is given in Algorithm 3.

The following theorem gives the rate of convergence for **GP-UCB**. Note that under an additive kernel, this is the same rate as Theorem 5 which uses a different acquisition function. Note the differences in the choice of β_t.

Theorem 6. *(Modification of Theorem 2 in (Srinivas et al., 2010)) Suppose f is constructed by sampling $f^{(j)} \sim \mathcal{GP}(\mathbf{0}, \kappa^{(j)})$ for $j = 1, \ldots, M$ and then adding them. Let all kernels $\kappa^{(j)}$ satisfy assumption 2 for some L, a, b. Further, we maximise the acquisition function $\widetilde{\varphi}_t$ to within $\zeta_0 t^{-1/2}$ accuracy at time step t. Pick $\delta \in (0,1)$ and choose*

$$\beta_t = 2\log\left(\frac{2t^2\pi^2}{\delta}\right) + 2D\log\left(Dt^3\right) \quad \in \quad \mathcal{O}\left(D\log t\right)$$

Then, **GP-UCB** *attains cumulative regret $R_T \in \mathcal{O}\left(\sqrt{D\gamma_T T \log T}\right)$ and hence simple regret $S_T \in \mathcal{O}\left(\sqrt{D\gamma_T \log T / T}\right)$. Precisely, with probability $> 1-\delta$,*

$$\forall T \geq 1, \quad R_T \leq \sqrt{8C_1\beta_T M T \gamma_t} + 2\zeta_0\sqrt{T} + C_2$$

where $C_1 = 1/\log(1+\eta^{-2})$ and C_2 is a constant depending on a, b, D, δ, L and η.

Proof. Srinivas et al. (2010) bound the regret for exact maximisation of the **GP-UCB** acquisition φ_t. By following an analysis similar to our proof of Theorem 5 the regret can be shown to be the same for an $\zeta_0 t^{-1/2}$- optimal maximisation. □

Algorithm 3 GP-UCB

Input: Kernel κ, Input Space $\mathcal{X}$.
For $t = 1, 2 \ldots$

- $\mathcal{D}_0 \leftarrow \varnothing$,
- $(\mu_0, \kappa_0) \leftarrow (\mathbf{0}, \kappa)$
- **for** $t = 1, 2, \ldots$
  1. $\mathbf{x}_t \leftarrow \operatorname{argmax}_{z \in \mathcal{X}} \mu_{t-1}(z) + \sqrt{\beta_t}\sigma_{t-1}(z)$
  2. $\mathbf{y}_t \leftarrow$ Query f at $\mathbf{x}_t$.
  3. $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \{(\mathbf{x}_t, \mathbf{y}_t)\}$.
  4. Perform Bayesian posterior updates to obtain μ_t, σ_t for $j = 1, \ldots, M$.

A.2. Sequential Optimisation Approaches

If the function is known to be additive, we could consider several other approaches for maximisation. We list two of them here and explain their deficiencies. We recommend that the reader read the main text before reading this section.

A.2.1. Optimise one group and proceed to the next

First, fix the coordinates of $x^{(j)}, j \neq 1$ and optimise w.r.t $x^{(1)}$ by querying the function for a pre-specified number of times. Then we proceed sequentially optimising with respect to $x^{(2)}, x^{(3)} \ldots$. We have outlined this algorithm in Algorithm 4. There are several reasons this approach is not desirable.

- First, it places too much faith on the additive assumption and requires that we know the decomposition at the start of the algorithm. Note that this strategy will only have searched the space in M d-dimensional subspaces. In our approach even if the function is not additive we can still hope to do well since we learn the best additive approximation to the true function. Further, if the decomposition is not known we could learn the decomposition "on the go" or at least find a reasonably good decomposition as we have explained in Section 4.4.

- Such a sequential approach is *not* an anytime algorithm. This in particular means that we need to predetermine the number of queries to be allocated to each group. After we proceed to a new group it is not straightforward to come back and improve on the solution obtained for an older group.
- This approach is not suitable for the bandits setting. We suffer large instantaneous regret up until we get to the last group. Further, after we proceed beyond a group since we cannot come back, we cannot improve on the best regret obtained in that group.

Our approach does not have any of these deficiencies.

Algorithm 4 Seq-Add-GP-UCB

Input: Kernels $\kappa^{(1)}, \ldots, \kappa^{(M)}$, Decomposition $(\mathcal{X}^{(j)})_{j=1}^{M}$, Query Budget T,

- $\mathbb{R}^D \ni \theta = \bigcup_{j=1}^{M} \theta^{(j)} = \text{rand}([0,1]^d)$
- **for** $j = 1, \ldots, M$
  1. $\mathcal{D}_0^{(j)} \leftarrow \varnothing$,
  2. $(\mu_0^{(j)}, \kappa_0^{(j)}) \leftarrow (\mathbf{0}, \kappa^{(j)})$.
  3. **for** $t = 1, 2, \ldots T/M$
     (a) $\mathbf{x}_t^{(j)} \leftarrow \text{argmax}_{z \in \mathcal{X}^{(j)}} \mu^{(j)}(z) + \sqrt{\beta_t}\sigma^{(j)}(z)$
     (b) $\mathbf{x}_t \leftarrow \mathbf{x}_t^{(j)} \bigcup_{k \neq j} \theta^{(k)}$.
     (c) $\mathbf{y}_t \leftarrow$ Query f at $\mathbf{x}_t$.
     (d) $\mathcal{D}_t^{(j)} = \mathcal{D}_{t-1}^{(j)} \cup \{(\mathbf{x}_t^{(j)}, \mathbf{y}_t)\}$.
     (e) Perform Bayesian posterior updates to obtain $\mu_t^{(j)}, \sigma_t^{(j)}$.
  4. $\theta^{(j)} \leftarrow \mathbf{x}_{T/M}^{(j)}$
- Return θ

A.2.2. Only change one Group per Query

In this strategy, the approach would be very similar to **Add-GP-UCB** except that at each query we will only update one group at time. If it is the k^{th} group the query point is determined by maximising $\widetilde{\varphi}_t^{(k)}$ for $\mathbf{x}_t^{(k)}$ and for all other groups we use values from the previous rotation. After M iterations we cycle through the groups. We have outlined this in Algorithm 5.

This is a reasonable approach and does not suffer from the same deficiencies as Algorithm 4. Maximising the acquisition function will also be slightly easier $\mathcal{O}(\zeta^{-d})$ since we need to optimise only one group at a time. However, the regret for this approach would be $\mathcal{O}(M\sqrt{D\gamma_T T \log T})$ which is a factor of M worse than the regret in our method (This can be show by following an analysis similar to the one in section B.2. This is not surprising, since at each iteration you are moving in d-coordinates of the space and you have to wait M iterations before the entire point is updated.

Algorithm 5 Add-GP-UCB-Buggy

Input: Kernels $\kappa^{(1)}, \ldots, \kappa^{(M)}$, Decomposition $(\mathcal{X}^{(j)})_{j=1}^{M}$

- $\mathcal{D}_0 \leftarrow \varnothing$,
- **for** $j = 1, \ldots, M$, $(\mu_0^{(j)}, \kappa_0^{(j)}) \leftarrow (\mathbf{0}, \kappa^{(j)})$.
- **for** $t = 1, 2, \ldots$
  1. $k = j \mod M$
  2. $\mathbf{x}_t^{(k)} \leftarrow \text{argmax}_{z \in \mathcal{X}^{(k)}} \mu^{(k)}(z) + \sqrt{\beta_t}\sigma^{(k)}(z)$
  3. **for** $j \neq k$, $\mathbf{x}_t^{(j)} \leftarrow \mathbf{x}_{t-1}^{(j)}$
  4. $\mathbf{x}_t \leftarrow \bigcup_{j=1}^{M} \mathbf{x}_t^{(j)}$.
  5. $\mathbf{y}_t \leftarrow$ Query f at $\mathbf{x}_t$.
  6. $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \{(\mathbf{x}_t, \mathbf{y}_t)\}$.
  7. Perform Bayesian posterior updates to obtain $\mu_t^{(j)}, \sigma_t^{(j)}$ for $j = 1, \ldots, M$.

B. Proofs of Results in Section 4.3

B.1. Bounding the Information Gain γ_T

For this we will use the following two results from Srinivas et al. (2010).

Lemma 7. *(Information Gain in GP, (Srinivas et al., 2010) Lemma 5.3) Using the basic properties of a GP, they show that*

$$I(y_A; f_A) = \frac{1}{2}\sum_{t=1}^{n} \log(1 + \eta^{-2}\sigma_{t-1}^2(x_t))$$

where σ_{t-1}^2 is the posterior variance after observing the first $t-1$ points.

Theorem 8. *(Bound on Information Gain, (Srinivas et al., 2010) Theorem 8) Suppose that $\mathcal{X}$ is compact and κ is a kernel on d dimensions satisfying Assumption 2. Let $n_T = C_9 T^\tau \log T$ where $C_9 = 4d + 2$. For any $T_* \in \{1, \ldots, \min(T, n_T)\}$, let $B_\kappa(T_*) = \sum_{s>T_*} \lambda_s$. Here $(\lambda_n)_{n\in\mathbb{N}}$ are the eigenvalues of κ w.r.t the uniform distribution over $\mathcal{X}$. Then,*

$$\gamma_T \le \inf_\tau \left(\frac{1/2}{1-e^{-1}} \max_{r\in\{1,\ldots,T\}} \left(T_* \log(rn_T/\eta^2) + C_9\eta^2(1-r/T)(T^{\tau+1}B_\kappa(T_*)+1)\log T\right) + \mathcal{O}(T^{1-\tau/d})\right)$$

B.1.1. Proof of Theorem 4-1

Proof. We will use some bounds on the eigenvalues for the simple squared exponential kernel given in (Seeger et al., 2008). It was shown that the eigenvalues $\{\lambda_s^{(i)}\}$ of $\kappa^{(i)}$ satisfied $\lambda_s^{(i)} \le c^d B^{s^{1/d_i}}$ where $B < 1$ (See Remark 9). Since the kernel is additive, and $x^{(i)} \cap x^{(j)} = \varnothing$ the eigenfunctions corresponding to $\kappa^{(i)}$ and $\kappa^{(j)}$ will be orthogonal. Hence the eigenvalues of κ will just be the union of the eigenvalues of the individual kernels – i.e. $\{\lambda_s\} = \bigcup_{j=1}^M \{\lambda_s^{(j)}\}$. As $B < 1$, $\lambda_s^{(i)} \le c^d B^{s^{1/d}}$. Let $T_+ = \lfloor T_*/M \rfloor$ and $\alpha = -\log B$. Then,

$$\begin{aligned}
B_\kappa(T_*) &= \sum_{s>T_*} \lambda_s \le Mc \sum_{s>T_+} B^{s^{1/d}} \\
&\le c^d M \left(B^{T_+^{1/d}} + \int_{T_+}^{\infty} \exp(-\alpha x^{1/d}) \right) \mathrm{d}x \\
&\le c^d M \left(B^{T_+^{1/d}} + d\alpha^{-d}\Gamma(d, \alpha T_+^{1/d}) \right) \\
&\le c^d M e^{-\alpha T_+^{1/d}} \left(1 + d!d\alpha^{-d}(\alpha T_+^{1/d})^{d-1} \right)
\end{aligned}$$

The last step holds true whenever $\alpha T_+^{1/d} \ge 1$. Here in the second step we bound the series by an integral and in the third step we used the substitution $y = \alpha x^{1/d}$ to simplify the integral. Here $\Gamma(s,x) = \int_x^\infty t^{s-1}e^{-t}\mathrm{d}t$ is the (upper) incomplete Gamma function. In the last step we have used the following identity and the bound for integral s and $x \ge 1$

$$\Gamma(s,x) = (s-1)!e^{-x}\sum_{k=0}^{s-1}\frac{x^k}{k!} \le s!e^{-x}x^{d-1}$$

By using $\tau = d$ and by using $T_* \le (M+1)T_+$, we use Theorem 8 to obtain the following bound on γ_T,

$$\begin{aligned}
\gamma_T \le \frac{1/2}{1-e^{-1}} \max_{r\in\{1,\ldots,T\}} \Bigg(&(M+1)T_+ \log(rn_T/\eta^2) + \\
&C_9\eta^2(1-r/T)\log T \left(1 + c^d M e^{-\alpha T_+^{1/d}} T^{d+1} \left(1 + d!d\alpha^{-d}(\alpha T_+^{1/d})^{d-1}\right)\right)\Bigg)
\end{aligned} \tag{7}$$

Now we need to pick T_+ so as to balance these two terms. We will choose $T_+ = \left(\frac{\log(Tn_T)}{\alpha}\right)^d$ which is less than $\min(T, n_T)/M$ for sufficiently large T. Then $e^{-\alpha T_+^{1/d}} = 1/Tn_T$. Then the first term S_1 inside the paranthesis is,

$$S_1 = (M+1)\log^d\left(\frac{Tn_T}{\alpha}\right)\log\left(\frac{rn_T}{\eta^2}\right) \in \mathcal{O}\left(M\left(\log(Tn_T)\right)^d \log(rn_T)\right)$$

$$\in \mathcal{O}\left(M\left(\log(T^{d+1}\log T)\right)^d \log(rT^d \log T)\right)$$
$$\in \mathcal{O}\left(Md^{d+1}(\log T)^{d+1} + Md^d(\log T)^d \log(r)\right)$$

Note that the constant in front has exponential dependence on d but we ignore it since we already have d^d, $(\log T)^d$ terms. The second term S_2 becomes,

$$\begin{aligned}
S_2 &= C_9\eta^2(1-r/T)\log T\left(1+\frac{c^d M}{Tn_T}T^{d+1}\left(1+d!d\alpha^{-d}(\log(Tn_T)^{d-1})\right)\right)\\
&\leq C_9\eta^2(1-r/T)\left(\log T+\frac{c^d M}{C_9}\left(1+d!d\alpha^{-d}(\log(Tn_T)^{d-1})\right)\right)\\
&\leq C_9\eta^2(1-r/T)\left(\mathcal{O}(\log T)+\mathcal{O}(1)+\mathcal{O}(d!d^d(\log T)^{d-1})\right)\Big)\\
&\in \mathcal{O}\left((1-r/T)d!d^d(\log T)^{d-1}\right)
\end{aligned}$$

Since S_1 dominates S_2, we should choose $r = T$ to maximise the RHS in (7). This gives us,

$$\gamma_T \;\in\; \mathcal{O}\left(Md^{d+1}(\log T)^{d+1}\right) \;\in\; \mathcal{O}\left(Dd^d(\log T)^{d+1}\right)$$

□

B.1.2. PROOF OF THEOREM 4-2

Proof. Once again, we use bounds given in (Seeger et al., 2008). It was shown that the eigenvalues $\{\lambda_s^{(i)}\}$ for $\kappa^{(i)}$ satisfied $\lambda_s^{(i)} \leq c^d s^{-\frac{2\nu+d_j}{d_j}}$ (See Remark 9). By following a similar argument to above we have $\{\lambda_s\} = \bigcup_{j=1}^M \{\lambda_s^{(j)}\}$ and $\lambda_s^{(i)} \leq c^d s^{-\frac{2\nu+d}{d}}$. Let $T_+ = \lfloor T_*/M \rfloor$. Then,

$$B_\kappa(T_*) = \sum_{s>T_*}\lambda_s \leq Mc^d \sum_{s>T_+} s^{-\frac{2\nu+d}{d}} \leq Mc^d\left(T_+^{-\frac{2\nu+d}{d}} + \int_{T_+}^{\infty} s^{-\frac{2\nu+d}{d}}\right) \leq C_8 2^d M T_+^{1-\frac{2\nu+d}{d}}$$

where C_8 is an appropriate constant. We set $T_+ = (Tn_T)^{\frac{d}{2\nu+d}}(\log(Tn_T))^{-\frac{d}{2\nu+d}}$ and accordingly we have the following bound on γ_T as a function of $T_+ \in \{1,\ldots,\min(T,n_T)/M\}$,

$$\gamma_T \leq \inf_\tau\left(\frac{1/2}{1-e^{-1}}\max_{r\in\{1,\ldots,T\}}\left((M+1)T_+\log(rn_T/\eta^2)+C_9\eta^2(1-r/T)\left(\log T + C_8 2^d M T_+\log(Tn_T)\right)\right)+\mathcal{O}(T^{1-\tau/d})\right) \tag{8}$$

Since this is a concave function on r we can find the optimum by setting the derivative w.r.t r to be zero. We get $r \in \mathcal{O}(T/2^d\log(Tn_T))$ and hence,

$$\begin{aligned}
\gamma_T &\in \inf_\tau\left(\mathcal{O}\left(MT_+\log\left(\frac{Tn_T}{2^d\log(Tn_T)}\right)\right)+\mathcal{O}\left(M2^dT_+\log(Tn_T)\right)+\mathcal{O}(T^{1-\tau/d})\right)\\
&\in \inf_\tau\left(\mathcal{O}\left(M2^d\log(Tn_T)\left(\frac{T^{\tau+1}\log(T)}{(\tau+1)\log(T)+\log\log T}\right)^{\frac{d}{2\nu+d}}\right)+\mathcal{O}(T^{1-\tau/d})\right)\\
&\in \inf_\tau\left(\mathcal{O}\left(M2^d\log(Tn_T)T^{\frac{(\tau+1)d}{2\nu+d}}\right)+\mathcal{O}(T^{1-\tau/d})\right)\\
&\in \mathcal{O}\left(M2^dT^{\frac{d(d+1)}{2\nu+d(d+1)}}\log(T)\right)
\end{aligned}$$

Here in the second step we have substituted the values for T_+ first and then n_T. In the last step we have balanced the polynomial dependence on T in both terms by setting $\tau = \frac{2\nu d}{2\nu+d(d+1)}$.

□

Remark 9. *The eigenvalues and eigenfunctions for the kernel are defined with respect to a base distribution on $\mathcal{X}$. In the development of Theorem 8, Srinivas et al. (2010) draw n_T samples from the uniform distribution on $\mathcal{X}$. Hence, the eigenvalues/eigenfunctions should be w.r.t the uniform distribution. The bounds given in Seeger et al. (2008) are for the uniform distribution for the Matérn kernel and a Gaussian Distribution for the Squared Exponential Kernel. For the latter case, Srinivas et al. (2010) argue that the uniform distribution still satisfies the required tail constraints and therefore the bounds would only differ up to constants.*

B.2. Rates on Add-GP-UCB

Our analysis in this section draws ideas from Srinivas et al. (2010). We will try our best to stick to their same notation. However, unlike them we also handle the case where the acquisition function is optimised within some error. In the ensuing discussion, we will use $\tilde{\mathbf{x}}_t = \bigcup_j \tilde{\mathbf{x}}_t^{(j)}$ to denote the true maximiser of $\widetilde{\varphi}_t$ – i.e. $\tilde{\mathbf{x}}_t^{(j)} = \operatorname{argmax}_{z \in \mathcal{X}^{(j)}} \widetilde{\varphi}_t^{(j)}(z)$. $\mathbf{x}_t = \bigcup_j \mathbf{x}_t^{(j)}$ denotes the point chosen by **Add-GP-UCB** at the t^{th} iteration. Recall that $\mathbf{x}_t$ is $\zeta_0 t^{-1/2}$–optimal; I.e. $\widetilde{\varphi}_t(\tilde{\mathbf{x}}_t) - \widetilde{\varphi}_t(\mathbf{x}_t) \leq \zeta_0 t^{-1/2}$.

Denote $p = \sum_j d_j$. π_t denotes a sequence such that $\sum_t \pi_t^{-1} = 1$. For e.g. when we use $\pi_t = \pi^2 t^2/6$ below, we obtain the rates in Theorem 5.

In what follows, we will construct discretisations $\Omega^{(j)}$ on each group $\mathcal{X}^{(j)}$ for the sake of analysis. Let $\omega_j = |\Omega^{(j)}|$ and $\omega_m = \max_j \omega_j$. The discretisation of the individual groups induces a discretisation Ω on $\mathcal{X}$ itself, $\Omega = \{\mathbf{x} = \bigcup_j \mathbf{x}^{(j)} : \mathbf{x}^{(j)} \in \Omega^{(j)}, j = 1, \ldots, M\}$. Let $\omega = |\Omega| = \prod_j \omega_j$. We first establish the following two lemmas before we prove Theorem 5.

Lemma 10. *Pick $\delta \in (0,1)$ and set $\beta_t = 2\log(\omega_m M \pi_t/\delta)$. Then with probability $> 1-\delta$,*

$$\forall t \geq 1, \forall \mathbf{x} \in \Omega, \qquad |f(\mathbf{x}) - \mu_{t-1}(\mathbf{x})| \leq \beta_t^{1/2} \sum_{j=1}^{M} \sigma_{t-1}^{(j)}(\mathbf{x}^{(j)})$$

Proof. Conditioned on $\mathcal{D}_{t-1}$, at any given $\mathbf{x}$ and t we have $f(\mathbf{x}^{(j)}) \sim \mathcal{N}(\mu_{t-1}^{(j)}(\mathbf{x}^{(j)}), \sigma_{t-1}^{(j)} j),\ \forall j = 1, \ldots M$. Using the tail bound, $\mathbb{P}(z > M) \leq \frac{1}{2} e^{-M^2/2}$ for $z \sim \mathcal{N}(0,1)$ we have with probability $> 1 - \delta/\omega M \pi_t$,

$$\frac{|f^{(j)}(\mathbf{x}^{(j)}) - \mu_{t-1}^{(j)}(\mathbf{x}^{(j)})|}{\sigma_{t-1}^{(j)}(\mathbf{x}^{(j)})} > \beta_t^{1/2} \leq e^{-\beta_t/2} = \frac{\delta}{\omega_m M \pi_t}$$

By using a union bound $\omega_j \leq \omega_m$ times over all $\mathbf{x}^{(j)} \in \Omega^{(j)}$ and then M times over all discretisations the above holds with probability $> 1 - \delta/\pi_t$ for all $j = 1, \ldots, M$ and $\mathbf{x}^{(j)} \in \Omega^{(j)}$. Therefore, we have $|f(\mathbf{x}) - \mu_{t-1}(\mathbf{x})| \leq |f(\mathbf{x}^{(j)}) - \mu_{t-1}^{(j)}(\mathbf{x}^{(j)})| \leq \beta_t^{1/2} \sum_j \sigma_{t-1}^{(j)}(\mathbf{x}^{(j)})$ for all $\mathbf{x} \in \Omega$. Now using the union bound on all t yields the result. □

Lemma 11. *The posterior mean μ_{t-1} for a GP whose kernel $\kappa(\cdot, x)$ is L-Lipschitz satisfies,*

$$\mathbb{P}\left(\forall t \geq 1 \quad |\mu_{t-1}(x) - \mu_{t-1}(x')| \leq \left(f(\mathbf{x}_*) + \eta\sqrt{2\log(\pi_t/2\delta)}\right) L\eta^{-2} t \|x - x'\|_2\right) \geq 1 - \delta$$

Proof. Note that for given t,

$$\mathbb{P}\left(y_t < f(\mathbf{x}_*) + \eta\sqrt{2\log(\pi_t/2\delta)}\right) \leq \mathbb{P}\left(\epsilon_t/\eta < \sqrt{2\log(\pi_t/2\delta)}\right) \leq \delta/\pi_t$$

Therefore the statement is true with probability $> 1 - \delta$ for all t. Further, $\Delta \succ \eta^2 I$ implies $\|\Delta^{-1}\|_{op} \leq \eta^{-2}$ and $|k(x,z) - k(x',z)| \leq L\|x - x'\|$. Therefore

$$\begin{aligned} |\mu_{t-1}(x) - \mu_{t-1}(x')| &= |Y_{t-1}^\top \Delta^{-1}(k(x, X_T) - k(x', X_T)| \leq \|Y_{t-1}\|_2 \|\Delta^{-1}\|_{op} \|k(x, X_{t-1}) - k(x', X_{t-1})\|_2 \\ &\leq \left(f(\mathbf{x}_*) + \eta\sqrt{2\log(\pi_t/2\delta)}\right) L\eta^{-2}(t-1)\|x - x'\|_2 \end{aligned}$$

□

B.2.1. PROOF OF THEOREM 5

Proof. First note that by Assumption 2 and the union bound we have, $\mathbb{P}(\forall i \ \sup_{x^{(j)} \in \mathcal{X}^{(j)}} |\partial f^{(j)}(x^{(j)})/\partial x_i^{(j)}| > J) \leq d_i a e^{-(J/b)^2}$. Since, $\partial f(x)/\partial x_i^{(j)} = \partial f^{(j)}(x^{(j)})/\partial x_i^{(j)}$, we have,

$$\mathbb{P}\left(\forall i = 1, \ldots, D \ \sup_{x \in \mathcal{X}} \left|\frac{\partial f(x)}{\partial x_i}\right| > J\right) \leq p a e^{-(J/b)^2}$$

By setting $\delta/3 = pae^{-J^2/b^2}$ we have with probability $> 1 - \delta/3$,

$$\forall x, x' \in \mathcal{X}, \ |f(x) - f(x')| \leq b\sqrt{\log(3ap/\delta)}\|x - x'\|_1 \tag{9}$$

Now, we construct a sequence of discretisations $\Omega_t^{(j)}$ satisfying $\|x^{(j)} - [x^{(j)}]_t\|_1 \leq d_j/\tau_t \ \ \forall x^{(j)} \in \Omega_t^{(j)}$. Here, $[x^{(j)}]_t$ is the closest point to $x^{(j)}$ in $\Omega_t^{(j)}$ in an L_2 sense. A sufficient discretisation is a grid with τ_t uniformly spaced points. Then it follows that for all $x \in \Omega_t$, $\|x - [x]_t\|_1 \leq p/\tau_t$. Here Ω_t is the discretisation induced on $\mathcal{X}$ by the $\Omega_t^{(j)}$'s and $[x]_t$ is the closest point to x in Ω_t. Note that $\|x^{(j)} - [x^{(j)}]_t\|_2 \leq \sqrt{d_j}/\tau_t \ \forall x^{(j)} \in \Omega^{(j)}$ and $\|x - [x]_t\|_2 \leq \sqrt{p}/\tau_t$. We will set $\tau_t = pt^3$–therefore, $\omega_{tj} \leq (pt^3)^d \triangleq \omega_{mt}$. When combining this with (9), we get that with probability $> 1 - \delta/3$, $|f(x) - f([x])| \leq b\sqrt{\log(3ap/\delta)}/t^3$. By our choice of β_t and using Lemma 10 the following is true for all $t \geq 1$ and for all $x \in \mathcal{X}$ with probability $> 1 - 2\delta/3$,

$$|f(x) - \mu_{t-1}([x]_t)| \leq |f(x) - f([x]_t)| + |f([x]_t) - \mu_{t-1}([x]_t)| \leq \frac{b\sqrt{\log(3ap/\delta)}}{t^2} + \beta_t^{1/2} \sum_{j=1}^{M} \sigma_{t-1}^{(j)}([x^{(j)}]_t) \tag{10}$$

By Lemma 11 with probability $> 1 - \delta/3$ we have,

$$\forall x \in \mathcal{X}, \quad |\mu_{t-1}(x) - \mu_{t-1}([x]_t)| \leq \frac{L\left(f(\mathbf{x}_*) + \eta\sqrt{2\log(3\pi_t/2\delta)}\right)}{\sqrt{p}\eta^2 t^2} \tag{11}$$

We use the above results to obtain the following bound on the instantaneous regret r_t which holds with probability $> 1 - \delta$ for all $t \geq 1$,

$$\begin{aligned}
r_t &= f(\mathbf{x}_*) - f(\mathbf{x}_t) \\
&\leq \mu_{t-1}([\mathbf{x}_*]_t) + \beta_t^{1/2} \sum_{j=1}^{M} \sigma_{t-1}^{(j)}([\mathbf{x}_*^{(j)}]_t) - \mu_{t-1}([\mathbf{x}_t]_t) + \beta_t^{1/2} \sum_{j=1}^{M} \sigma_{t-1}^{(j)}([\mathbf{x}_t^{(j)}]_t) + \frac{2b\sqrt{\log(3ap/\delta)}}{t^3} \\
&\leq \frac{2b\sqrt{\log(3ap/\delta)}}{t^3} + \frac{\zeta_0}{\sqrt{t}} + \beta_t^{1/2}\left(\sum_{j=1}^{M} \sigma_{t-1}^{(j)}(\mathbf{x}_t^{(j)}) + \sum_{j=1}^{M} \sigma_{t-1}^{(j)}([\mathbf{x}_t^{(j)}]_t)\right) + \mu_{t-1}(\mathbf{x}_t) - \mu_{t-1}([\mathbf{x}_t]_t) \\
&\leq \frac{2b\sqrt{\log(3ap/\delta)}}{t^3} + \frac{L\left(f(\mathbf{x}_*) + \eta\sqrt{2\log(\pi_t/2\delta)}\right)}{\sqrt{p}\eta^2 t^2} + \frac{\zeta_0}{\sqrt{t}} + \beta_t^{1/2}\left(\sum_{j=1}^{M} \sigma_{t-1}^{(j)}(\mathbf{x}_t^{(j)}) + \sum_{j=1}^{M} \sigma_{t-1}^{(j)}([\mathbf{x}_t^{(j)}]_t)\right)
\end{aligned} \tag{12}$$

In the first step we have applied Equation (10) at $\mathbf{x}_*$ and $\mathbf{x}_t$. In the second step we have used the fact that $\widetilde{\varphi}_t([\mathbf{x}_*]_t) \leq \widetilde{\varphi}_t(\tilde{\mathbf{x}}_t) \leq \widetilde{\varphi}_t(\mathbf{x}_t) + \zeta_0 t^{-1/2}$. In the third step we have used Equation (11).

For any $x \in \mathcal{X}$ we can bound $\sigma_t(x)^2$ as follows,

$$\sigma_t(x)^2 = \eta^2\eta^{-2}\sigma_t(x)^2 \leq \frac{1}{\log(1 + \eta^{-2})} \log\left(1 + \eta^{-2}\sigma_t(x)^2\right)$$

Here we have used the fact that $u^2 \leq v^2 \log(1 + u^2)/\log(1 + v^2)$ for $u \leq v$ and $\sigma_t(\mathbf{x})^2 \leq \kappa(x, x) = 1$. Write $C_1 = \log^{-1}(1 + \eta^{-2})$. By using Jensen's inequality and Definition 3 for any set of T points $\{x_1, x_2, \ldots x_T\} \subset \mathcal{X}$,

$$\left(\sum_{t=1}^{T}\sum_{j=1}^{M} \sigma_t^{(j)}(x^{(j)})\right)^2 \leq MT \sum_{t=1}^{T}\sum_{j=1}^{M} \sigma_t^{(j)}(x^{(j)})^2 \leq C_1 MT \sum_{t=1}^{T} \log\left(1 + \eta^{-2}\sigma_t(x)^2\right) \leq 2C_1 MT\gamma_T \tag{13}$$

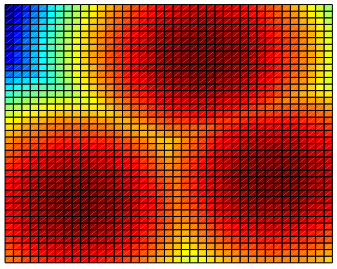


Figure 4. Illustration of the trimodal function $f_{d'}$ in $d' = 2$.

Finally we can bound the cumulative regret with probability $> 1 - \delta$ for all $T \geq 1$ by,

$$R_T = \sum_{t=1}^{T} r_t \leq C_2(a, b, D, L, \delta) + \zeta_0 \sum_{t=1}^{T} t^{-1/2} + \beta_T^{1/2} \left(\sum_{t=1}^{T} \sum_{j=1}^{M} \sigma_{t-1}^{(j)}(\mathbf{x}_t^{(j)}) + \sum_{t=1}^{T} \sum_{j=1}^{M} \sigma_{t-1}^{(j)}([\mathbf{x}_t^{(j)}]_t) \right)$$
$$\leq C_2(a, b, D, L, \delta) + 2\zeta_0 \sqrt{T} + \sqrt{8 C_1 \beta_T M T \gamma_T}$$

where we have used the summability of the first two terms in Equation (12). Here, for $\delta < 0.8$, the constant C_2 is given by,

$$C_2 \geq b\sqrt{\log(3ap/\delta)} + \frac{\pi^2 L f(\mathbf{x}_*)}{6\sqrt{p}\eta^2} + \frac{L\pi^{3/2}}{\sqrt{12p\delta}\eta}$$

□

C. Experiments

To demonstrate the efficacy of **Add-GP-UCB** over **GP-UCB** we optimise the acquisition function under a constrained budget. Following, Brochu et al. (2010) we use DiRect to maximise $\varphi_t, \widetilde{\varphi}_t$. To demonstrate the efficacy of **Add-GP-UCB** we optimise the acquisition function under a constrained budget. We compare **Add-GP-UCB** against **GP-UCB**, random querying (RAND) and DiRect[1]. On the real datasets we also compare it to the Expected Improvement (GP-EI) acquisition function which is popular in BO applications and the method of Wang et al. (2013) which uses a random projection before applying BO (REMBO). We have multiple instantiations of **Add-GP-UCB** for different values for (d, M). For optimisation, we perform comparisons based on the simple regret S_T and for bandits we use the time averaged cumulative regret R_T/T.

For all GPB/ BOmethods we set $N_{init} = 10$, $N_{cyc} = 25$ in all experiments. Further, for the first 25 iterations we set the bandwidth to a small value (10^{-5}) to encourage an explorative strategy. We use SE kernels for each additive kernels and use the same scale σ and bandwidth h hyperparameters for all the kernels. Every 25 iterations we maximise the marginal likelihood with respect to these 2 hyperparameters in addition to the decomposition.

In contrast to existing literature in the BO community, we found that the UCB acquisitions outperformed GP-EI. One possible reason may be that under a constrained budget, UCB is robust to imperfect maximisation (Theorem 5) whereas GP-EI may not be. Another reason may be our choice of constants in UCB (Section 4.4).

C.1. Simulations on Synthetic Data

First we demonstrate our technique on a series of synthetic examples. For this we construct additive functions for different values for the maximum group size d' and the number of groups M'. We use the prime to distinguish it from **Add-GP-**

[1]There are several global optimisation methods based on simulated annealing, cross entropy methods, genetic algorithms etc. We choose DiRect since its easy to configure and known to work well in practice.

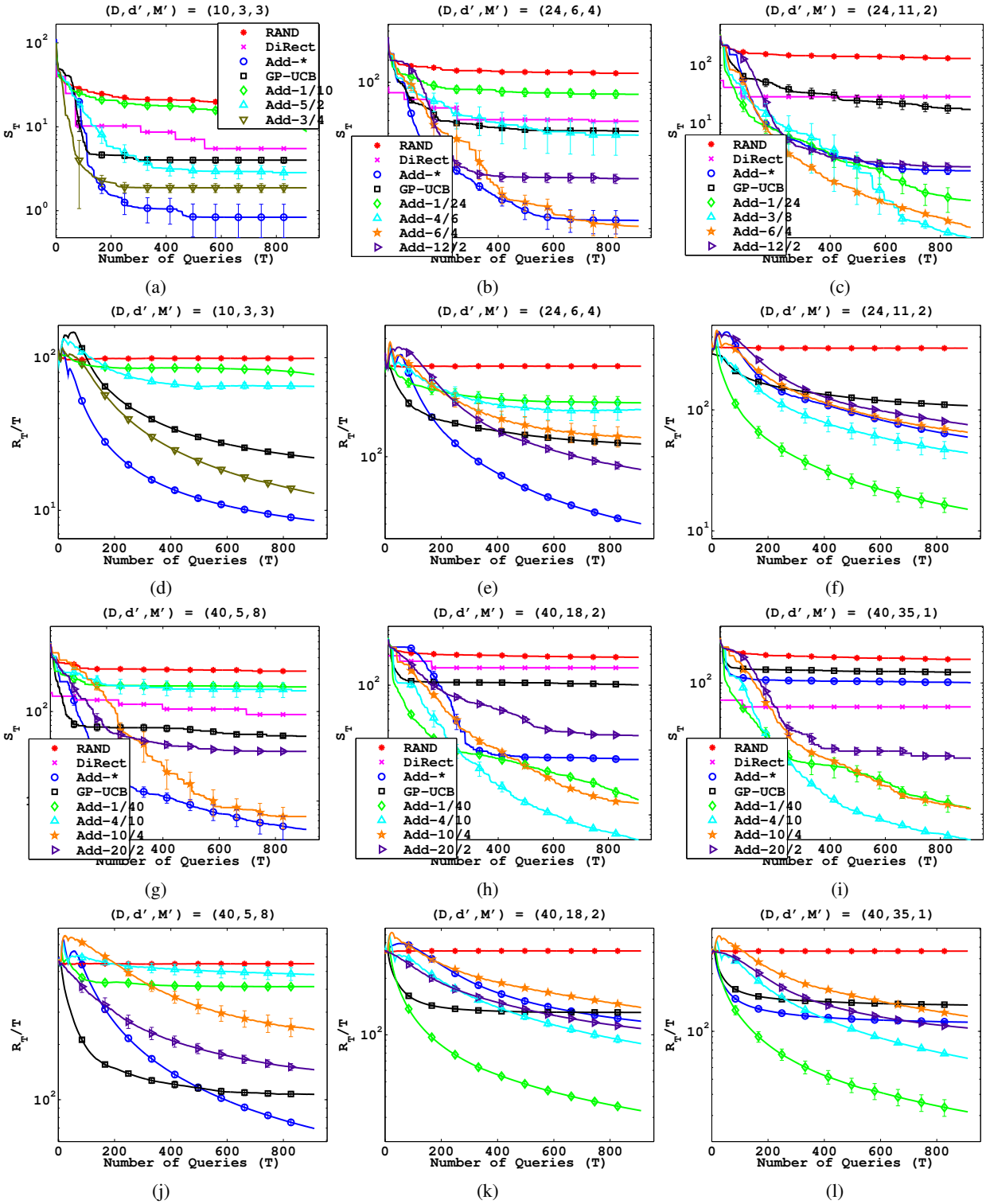


Figure 5. Results on the synthetic datasets. In all images the x-axis is the number of queries and the y-axis is the regret in log scale. We have indexed each experiment by their (D, d', M') values. The first row is S_T for the experiments with (D, d', M') set to $(10, 3, 3), (24, 6, 4), (24, 11, 2)$ and the second row is R_T/T for the same experiments. The third row is S_T for $(40, 5, 8), (40, 18, 2), (40, 35, 1)$ and the fourth row is the corresponding R_T. In some figures, the error bars are not visible since they are small and hidden by the bullets. All figures were produced by averaging over 20 runs.

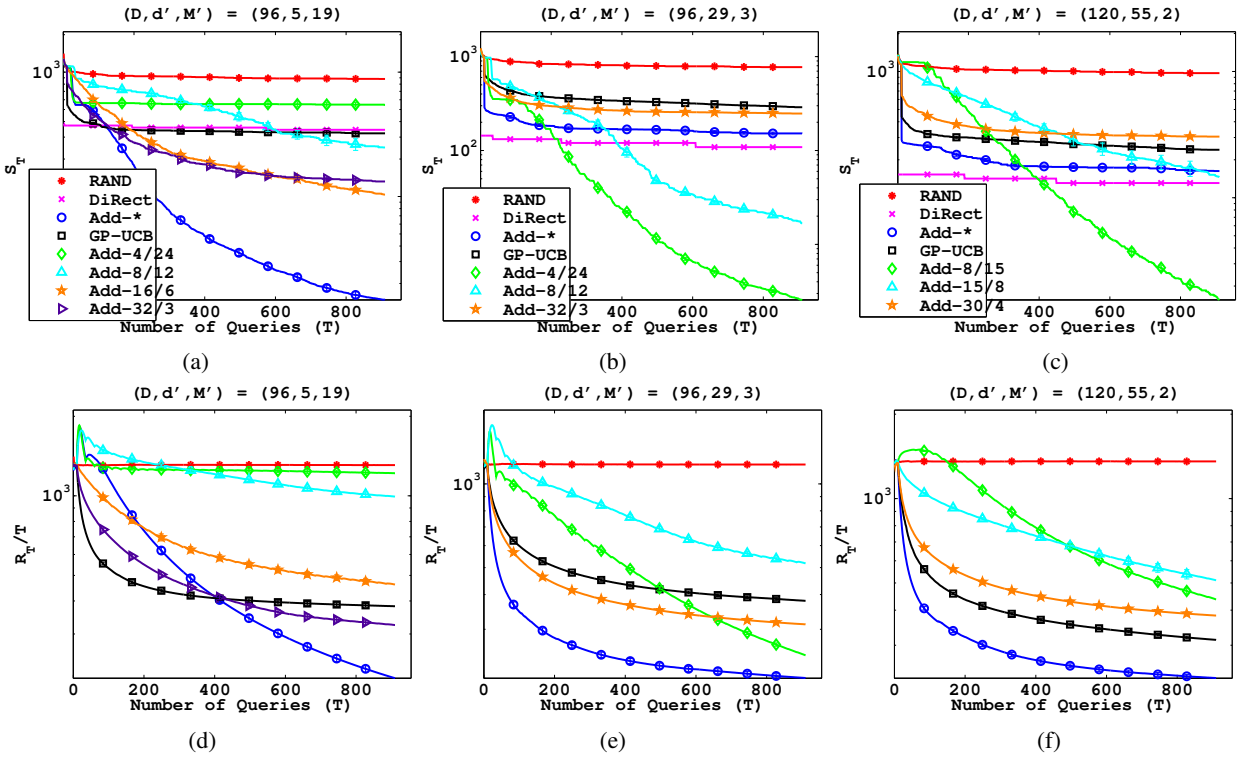


Figure 6. More results on synthetic experiments. The simple regret S_T (first row) and cumulative regret R_T/T (second row) for functions with (D, d', M') set to $(96, 5, 19), (96, 29, 3), (120, 55, 2)$ respectively. Read the caption under Figure 5 for more details.

UCB instantiations with different combinations of (d, M) values. The d' dimensional function $f_{d'}$ is,

$$f_d(x) = \log\left(0.1\frac{1}{h_{d'}^{d'}}\exp\left(\frac{\|x-v_1\|^2}{2h_{d'}^2}\right) + \quad 0.1\frac{1}{h_{d'}^{d'}}\exp\left(\frac{\|x-v_2\|^2}{2h_{d'}^2}\right) + 0.8\frac{1}{h_{d'}^{d'}}\exp\left(\frac{\|x-v_3\|^2}{2h_{d'}^2}\right)\right) \tag{14}$$

where v_1, v_2, v_3 are fixed d' dimensional vectors and $h_{d'} = 0.01d'^{0.1}$. Then we create M' groups of coordinates by randomly adding d' coordinates into each group. On each such group we use $f_{d'}$ and then add them up to obtain the composite function f. Precisely,

$$f(x) = f_{d'}(x^{(1)}) + \cdots + f_{d'}(x^{(M)})$$

The remaining $D - d'M'$ coordinates do not contribute to the function. Since $f_{d'}$ has 3 modes, f will have $3^{M'}$ modes. We have illustrated $f_{d'}$ for $d' = 2$ in Figure 4. In the synthetic experiments we use an instantiation of **Add-GP-UCB** that knows the decomposition–i.e. $(d, M) = (d', M')$ and the grouping of coordinates. We refer to this as **Add-**$\star$. For the rest we use a (d, M) decomposition by creating M groups of size at most d and find a good grouping by partially maximising the marginal likelihood (Section 4.4). We refer to them as **Add-**d/M.

For **GP-UCB** we allocate a budget of $\min(5000, 100D)$ DiRect function evaluations to optimise the acquisition function. For all **Add-**d/M methods we set it to 90% of this amount[2] to account for the additional overhead in posterior inference for each $f^{(j)}$. Therefore, in our $10D$ problem we maximise φ_t with $\beta_t = 2\log(2t)$ with 1000 DiRect evaluations whereas for **Add-**$2/5$ we maximise each $\widetilde{\varphi}_t^{(j)}$ with $\beta_t = 0.4\log(2t)$ with 180 evaluations.

The results are given in Figures 5 and 6. We refer to each example by the configuration of the additive function–its (D, d', M') values. In the $(10, 3, 3)$ example **Add-**$\star$ does best since it knows the correct model and the acquisition function can be maximised within the budget. However **Add-**$3/4$ and **Add-**$5/2$ models do well too and outperform **GP-UCB**. **Add-**$1/10$ performs poorly since it is statistically not expressive enough to capture the true function. In the $(24, 11, 2)$, $(40, 18, 2)$, $(40, 35, 1)$, $(96, 29, 3)$ and $(120, 55, 2)$ examples **Add-**$\star$ outperforms **GP-UCB**. However, it is not competitive

[2]While the 90% seems arbitrary, in our experiments this was hardly a factor as the cost was dominated by the inversion of Δ.

with the **Add**-d/M for small d. Even though **Add**-$\star$ knew the correct decomposition, there are two possible failure modes since d' is large. The kernel is complex and the estimation error is very high in the absence of sufficient data points. In addition, optimising the acquisition is also difficult. This illustrates our previous argument that using an additive kernel can be advantageous even if the function is not additive or the decomposition is not known. In the $(24, 6, 4)$, $(40, 5, 8)$ and $(96, 5, 19)$ examples **Add**-$\star$ performs best as d' is small enough. But again, almost all **Add**-d/M instantiations outperform **GP-UCB**. In contrast to the small D examples, for large D, **GP-UCB** and **Add**-d/M with large d perform worse than DiRect. This is probably because our budget for maximising φ_t is inadequate to optimise the acquisition function to sufficient accuracy. For some of the large D examples the cumulative regret is low for **Add-GP-UCB** and **Add**-d/M with large d. This is probably since they have already started exploiting where as the **Add**-d/M with small d methods are still exploring. We posit that if we run for more iterations we will be able to see the improvements.

C.2. SDSS Astrophysical Dataset

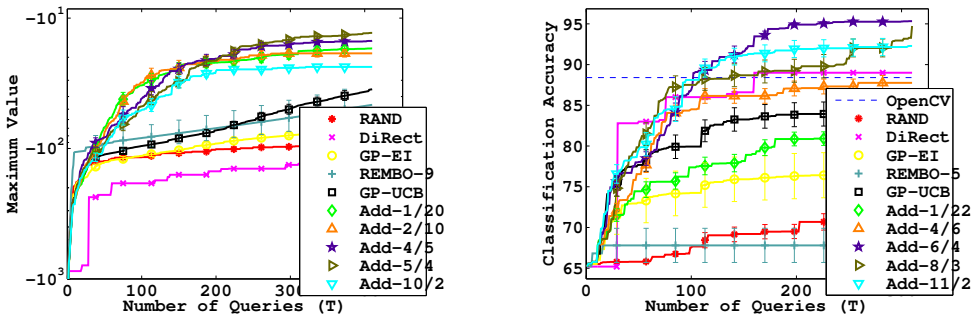
Here we used Galaxy data from the Sloan Digital Sky Survey (SDSS). The task is to find the maximum likelihood estimators for a simulation based astrophysical likelihood model. Data and software for computing the likelihood are taken from Tegmark et al (2006). The software itself takes in only 9 parameters but we augment this to 20 dimensions to emulate the fact that in practical astrophysical problems we may not know the true parameters on which the problem is dependent. This also allows us to effectively demonstrate the superiority of our methods over alternatives. Each query to this likelihood function takes about 2-5 seconds. In order to be wall clock time competitive with RAND and DiRectwe use only 500 evaluations for **GP-UCB**, GP-EI and REMBO and 450 for **Add**-d/M to maximise the acquisition function.

We have shown the Maximum value obtained over 400 iterations of each algorithm in Figure 3(a). Note that RAND outperforms DiRect here since a random query strategy is effectively searching in 9 dimensions. Despite this advantage to RAND all BO methods do better. Moreover, despite the fact that the function may not be additive, all **Add**-d/M methods outperform **GP-UCB**. Since the function only depends on 9 parameters we use REMBO with a 9 dimensional projection. Despite this advantage to REMBO it is not competitive with the **Add**-d/M methods. Possible reasons for this may include the scaling of the parameter space by $\sqrt{d}$ in REMBO and the imperfect optimisation of the acquisition function. Here **Add**-$5/4$ performs slightly better than the rest since it seems to have the best tradeoff between being statistically expressive enough to capture the function while at the same time be easy enough to optimise the acquisition function within the allocated budget.

C.3. Viola & Jones Face Detection

The Viola & Jones (VJ) Cascade Classifier (Viola & Jones, 2001) is a popular method for face detection in computer vision based on the Adaboost algorithm. The K-cascade has K weak classifiers which outputs a score for any given image. When we wish to classify an image we pass that image through each classifier. If at any point the score falls below a certain threshold the image is classified as negative. If the image passes through all classifiers then it is classified as positive. The threshold values at each stage are usually pre-set based on prior knowledge. There is no reason to believe that these threshold values are optimal. In this experiment we wish to find an optimal set of values for these thresholds by optimising the classification accuracy over a training set.

For this task, we use 1000 images from the Viola & Jones face dataset containing both face and non-face images. We use the implementation of the VJ classifier that comes with OpenCV (Bradski & Kaehler, 2008) which uses a 22-stage cascade and modify it to take in the threshold values as a parameter. As our domain $\mathcal{X}$ we choose a neighbourhood around the configuration given in OpenCV. Each function call takes about 30-40 seconds and is therefore the dominant cost in this experiment. We use 1000 DiRect evaluations to optimise the acquisition function for **GP-UCB**, GP-EI and REMBO and 900 for the **Add**-d/M instantiations. Since we do not know the structure of the function we use REMBO with a 5 dimensional projection. The results are given in Figure 3(b). Not surprisingly, REMBO performs worst as it is only searching on a 5 dimensional space. Barring **Add**-$1/22$ all other instantiations perform better than **GP-UCB** with **Add**-$6/4$ performing the best. Interestingly, we also find a value for the thresholds that outperform the configuration used in the OpenCV implementation.



(a) (b)

Figure 7. Results on the Astrophysical experiment (a) and the Viola and Jones dataset (b). The x-axis is the number of queries and the y-axis is the maximum value. (a) was produced by averaging over 20 runs and (b) over 11 runs.