

Think Global and Act Local: Bayesian Optimisation over High-Dimensional Categorical and Mixed Search Spaces

Xingchen Wan¹ Vu Nguyen² Huong Ha³ Binxin Ru¹ Cong Lu¹ Michael A. Osborne¹

Abstract

High-dimensional black-box optimisation remains an important yet notoriously challenging problem. Despite the success of Bayesian optimisation methods on continuous domains, domains that are categorical, or that mix continuous and categorical variables, remain challenging. We propose a novel solution – we combine local optimisation with a tailored kernel design, effectively handling high-dimensional categorical and mixed search spaces, whilst retaining sample efficiency. We further derive convergence guarantee for the proposed approach. Finally, we demonstrate empirically that our method outperforms the current baselines on a variety of synthetic and real-world tasks in terms of performance, computational costs, or both.

1. Introduction

Bayesian Optimisation (BO) (Jones et al., 1998; Brochu et al., 2010; Shahriari et al., 2016), which features expressive surrogate model(s) and sample efficiency, has found many applications in black-box optimisation, particularly when each evaluation is expensive. Such applications include but not limited to selection of chemical compounds (Hernández-Lobato et al., 2017), reinforcement learning (Parker-Holder et al., 2020), hyperparameter optimisation of machine learning algorithms (Snoek et al., 2012), and neural architecture search (Kandasamy et al., 2018; Nguyen et al., 2021; Ru et al., 2021)

Despite its impressive performance, various challenges still remain for BO. The popular surrogate choice, vanilla Gaussian Process (GP) models, is limited to problems of modest dimensionality defined in a continuous space. However, real-world optimisation problems are often neither

¹Machine Learning Research Group, University of Oxford, Oxford, UK ²Amazon, Adelaide, Australia ³RMIT University, Melbourne, Australia. Correspondence to: Xingchen Wan <xwan@robots.ox.ac.uk>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

low-dimensional nor continuous: many large-scale practical problems exhibit complex interactions among high-dimensional input variables, and are often *categorical* in nature or involve a mixture of both continuous and categorical input variables. An example of the former is the maximum satisfiability problem, whose exact solution is NP-hard (Creignou et al., 2001), and an example for the latter is the hyperparameter tuning for a deep neural network: the optimisation scope comprise both continuous hyperparameters, e.g., learning rate and momentum, and categorical ones, e.g., optimiser type {SGD, Adam, ...} and learning rate scheduler type {step decay, cosine annealing}.

These problems are challenging for a number of reasons: first, categorical variables do not have a natural ordering similar to continuous ones for which GPs are well-suited. Second, the search space grows exponentially with the dimension and the mixed spaces are usually high-dimensional, making the objective function highly multimodal, often heterogeneous, and thus difficult to be modelled by a good, global surrogate (Rana et al., 2017; Eriksson et al., 2019). Partially due to these difficulties, only very few prior works (Hutter et al., 2011; Gopakumar et al., 2018; Nguyen et al., 2020; Ru et al., 2020a) have focused on developing BO strategies for such problems, and, to the best of our knowledge, achieving promising performance, easy applicability for high-dimensional inputs and reasonable computing costs simultaneously is still an open question.

To tackle these challenging yet important problems, we propose a novel yet conceptually simple method. It not only fully preserves the merits of GP-based BO approaches, such as expressiveness and sample efficiency, but also demonstrates state-of-the-art performance in high-dimensional optimisation problems, involving categorical or mixed search spaces. Specifically, we make the following contributions:

- Propose a new GP-based BO approach which designs tailored GP kernels and harnesses the concept of local trust region to effectively handle high-dimensional optimisation over categorical and mixed search spaces.
- Derive convergence analysis to show that our proposed method converges to the global maximum of the objective function in both categorical and mixed space settings, under some assumptions.

- Empirically show that our method achieves superior performance, better sample efficiency, or both, over the existing approaches for a wide variety of tasks. The code implementation of our method is available at <https://github.com/xingchenwan/Casmopolitan>.

2. Related work

BO for high-dimensional problems A popular class of high-dimensional BO methods (Kandasamy et al., 2015; Rolland et al., 2018; Wang et al., 2017; 2018; Mutny and Krause, 2019) decompose the search space into multiple overlapping or disjoint low-dimensional subspaces and use an additive surrogate (e.g. additive GPs). However, accurately inferring the decomposition is often very expensive. Another group of BO methods (Binois et al., 2015; Wang et al., 2016; Binois et al., 2020) assume the objective function is mainly influenced by a small subset of effective dimensions and aims to learn such low-dimensional effective embedding (Wang et al., 2016; Nayebi et al., 2019; Letham et al., 2020). However, its effectiveness is conditional on the extent the assumption holds. A recent state-of-the-art approach is Trust-region Bayesian Optimisation (TURBO) (Eriksson et al., 2019), which constrains BO on local Trust Region (TR) centered around the best inputs so far. This circumvents the aforementioned issues such as the need for finding an accurate global surrogate and over-exploration due to large regions of high posterior variance. However, its convergence properties are not analysed, and it only works in continuous spaces.

BO for categorical search spaces The basic approach is to one-hot transform the categorical variables into continuous (Rasmussen, 2006; GPyOpt, 2016; Snoek et al., 2012). While simple in implementation, the drawbacks are equally obvious: first, for a d_h -dimensional problems with $\{n_1, \dots, n_{d_h}\}$ choices per input, the one-hot-transformed problem has $\sum_{i=1}^{d_h} n_i$ dimensions, further aggravating the curse of dimensionality. Second, categorical spaces differ fundamentally with the continuous in, for e.g., differentiability and continuity, with function values only defined in finite locations. These lead to difficulties in using gradient-based methods in acquisition function optimisation of the transformed problems.

To ameliorate these drawbacks, BOCS (Baptista and Poloczek, 2018) first tailors BO in categorical spaces: it uses a sparse monomial representation up to the second order and Bayesian linear regression as the surrogate, and is primarily used for boolean optimisation. Inevitably, its expressiveness is constrained by the quadratic model, while scaling beyond the second order and/or to high dimensionality is usually intractable due to the exponentially-increasing number of parameters that need be learnt explicitly. Combinatorial

Bayesian Optimisation (COMBO) (Oh et al., 2019) is a state-of-the-art method that instead uses a GP surrogate (which is capable of learning interactions of an arbitrary order), and is capable of dealing with multi-categorical problems via a combinatorial graph over all possible joint assignments of the variables and a diffusion graph kernel to model the interactions. Nonetheless, both methods deal with categorical optimisation only, which is an important problem in its own right, but does not extend to our setting of mixed-variable problems. They also suffer from poor scalability (e.g. to avoid overfitting COMBO approximately marginalises the posterior via Monte Carlo sampling instead of cheaper optimisation, and it needs to pre-compute the combinatorial graph beforehand). Other methods, such as COMEX and its inspired works (Dadkhahi et al., 2020; 2021) take a non-Bayesian black-box optimisation approach to improve computing efficiency, but they are typically less sample-efficient with respect to the number of function queries and are less suitable for problems where querying the objective functions is expensive. Finally, several recent works aim to improve BO on combinatorial structures by improving the effectiveness (Deshwal et al., 2020) or reducing the expenses (Swersky et al., 2020) of the *acquisition function*; these are largely orthogonal to our method, and we defer a thorough investigation on whether there are additional benefits by combining with these methods to a future work.

BO for mixed input types BO in mixed categorical-continuous search spaces is still rather under-explored, despite attempts in modelling less complicated spaces, such as mixed continuous-integer problems (Daxberger et al., 2019; Garrido-Merchán and Hernández-Lobato, 2020). In our specific setting, Categorical and Continuous Bayesian Optimisation (COCABO) (Ru et al., 2020a) first explicitly handles multiple categorical and continuous variables: it alternates between selecting the categorical inputs with a Multi-Armed Bandit (MAB) and the continuous inputs with GP-BO, and uses a tailored kernel to connect the two. However, COCABO requires optimising a MAB over a non-stationary reward (since the values of continuous variables improves over BO iterations and hence so does the function value). Furthermore, MAB requires pulling each arm at least once, and hence it is difficult to scale COCABO to high-dimensional problems, where the total number of possible arm combinations explode exponentially. Lastly, while the two sub-components are provably convergent, COCABO as a whole is not. Related works along this direction also include Gopakumar et al. (2018) and Nguyen et al. (2020), but the continuous inputs are constrained to be *specific* to the categorical choice, and being MAB-based, it also suffers from aforementioned limitations. Separately, Bliek et al. (2020) recently propose Mixed-Variate ReLU-based Surrogate Modelling (MVRSM), which the authors claim to be suitable for mixed-variable, high-dimensional problems. However,

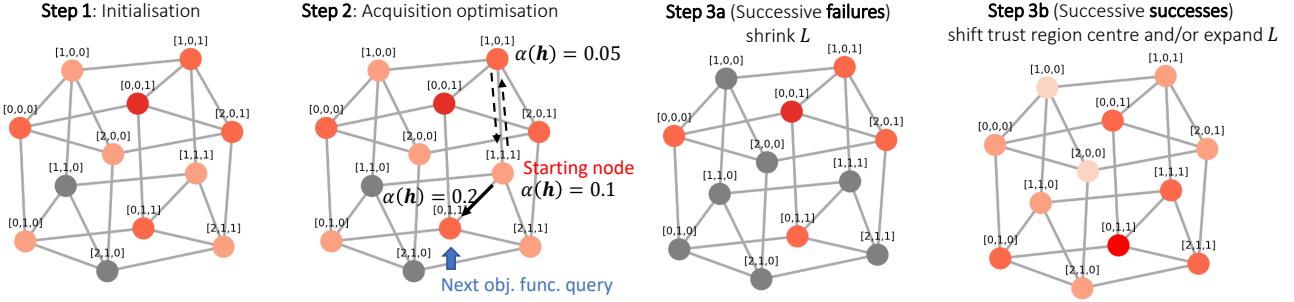


Figure 1. Illustration of CASMOPOLITAN in categorical space. Suppose we optimise over a 3-dimensional problem with $\{3, 2, 2\}$ choices for each input respectively. Initially (**Step 1**), the best location so far $\mathbf{h}_T^* = \arg \max_{\mathbf{h}} \{y_j\}_{j=1}^T$ (marked in red) is $[0, 0, 1]$ with TR radius $L = 2$ (the orange nodes, with different shades denoting their Hamming distances to \mathbf{h}_T^*). The gray nodes are outside the current TR. In optimisation of the acquisition function (**Step 2**), we conduct local search within the TR, moving to a neighbour only if it has a higher acquisition function value $\alpha(\cdot)$ and is still within the TR. In case of successive failures (**Step 3a**) in increasing \mathbf{h}_T^* , we shrink the TR down to length L_{\min}^h , below which we restart the optimisation, or in case of successive successes (**Step 3b**), we shift the TR centre to the new \mathbf{h}_T^* and/or expand TR up to length L_{\max}^h . Note that the combinatorial graph is shown here for illustration; it does not need to be computed explicitly or otherwise.

in trading for efficiency, the expressiveness is limited by the ReLU formulation and we compare against it in Sec. 4.

In addition to these more recent works explicitly handling the mixed spaces, earlier attempts such as SMAC with Random Forest (RF) (Breiman, 2001) surrogates (Hutter et al., 2011) are also compatible. However, the predictive distribution of the RF used to select new evaluation is less accurate due to reliance on randomness from bootstrap samples and the randomly chosen subset of variables to be tested at each node to split the data. Moreover, RFs easily suffer from overfitting and require careful hyperparameter choice.

3. CASMOPOLITAN: BO for Categorical and Mixed Search Spaces

Problem Statement We consider the problem of optimising an expensive black-box function, defined over a categorical domain or one with mixed continuous and categorical inputs. Formally, we consider a function in the mixed domain for generality: $f : [\mathcal{H}, \mathcal{X}] \rightarrow \mathbb{R}$ where \mathcal{H} and $\mathcal{X} \subset \mathbb{R}^{d_x}$ denote the categorical and continuous search spaces, respectively (for problems over categorical domains, we simply have $f : \mathcal{H} \rightarrow \mathbb{R}$ and the goal is to find $\mathbf{h}^* = \arg \max f(\mathbf{h})$). We further denote $\mathbf{z} = [\mathbf{h}, \mathbf{x}]$ to be an input in the mixed space where \mathbf{h} and \mathbf{x} are the categorical and continuous parts, d_h to be the number of categorical variables, i.e. $\mathbf{h} = [h_1, h_2, \dots, h_{d_h}]$, and the number of possible, distinct value that the j -th categorical variable may take to be n_j . Given f , at time t we observe the noisy perturbation of the form $y_t = f(\mathbf{z}_t) + \epsilon_t$ where $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$ and σ^2 is a noise variance which can be learned by maximizing the log-marginal likelihood (Rasmussen, 2006). We sequentially select inputs $\mathbf{z}_t \forall t = 1, \dots, T$ (or simply \mathbf{h}_t if the problem is purely categorical) to query f with the goal

Algorithm 1 CASMOPOLITAN.

```

1: Input: #init (the number of random initialing points at initialisation or restarts), #iter  $T$ , initial TR size for categorical  $L_0^h \in \mathbb{Z}^+$ , and continuous variables  $L_0^x \in \mathbb{R}^+$ .
2: Output: The best recommendation  $\mathbf{z}_T$ 
3:  $\text{restart} = \text{True} // Set \text{restart} \text{ to True initially}$ 
4: for  $t = 1, \dots, T$  do
5:   if  $\text{restart}$  then
6:     Reset TR  $L^h = L_0^h$  and  $L^x = L_0^x$  and reset GP. Randomly select #init points in the search space as  $\mathbf{z}_t$  (if at initialisation), or set the TR center as the point determined by Eq. (3) and randomly select #init points within the newly constructed TR as  $\mathbf{z}_t$  (if at subsequent restarts).
7:   else
8:     Construct a TR  $\text{TR}_h(\mathbf{h}_t^*)$  around the categorical dimensions of the best point  $\mathbf{h}_t^*$  using Eq. (2).
9:     Construct a hyper-rectangular TR of length  $L^x$ ,  $\text{TR}_x(\mathbf{x}_t^*)$  for the continuous variables.
10:    Select next query pt(s) within the TRs  $\mathbf{z}_t = \arg \max_{\mathbf{z}} \alpha(\mathbf{z})$  s.t.  $\mathbf{x} \in \text{TR}_x(\mathbf{x}_t^*), \mathbf{h} \in \text{TR}_h(\mathbf{h}_t^*)$ .
11:   end if
12:   Query at  $\mathbf{z}_t$  to obtain  $y_t$ ; fit/update the surrogate  $\mathcal{D}_t \leftarrow \mathcal{D}_{t-1} \cup (\mathbf{z}_t, y_t)$  and optimise GP hyperparameters.
13:   Update the TRs and decide whether to restart.
14: end for

```

of finding the maximiser the objective $\mathbf{z}^* = \arg \max f(\mathbf{z})$ with the fewest numbers of iterations. We further include a primer on GP and BO in App. A.

3.1. Categorical Search Space

Our first contribution is to propose a conceptually-simple yet effective BO strategy that preserves all of the advantages of GP modelling, but is specifically designed for the categorical search space (later extended to the mixed space in Sec. 3.2). We present an illustration in Fig. 1 and the pseudocode in Algorithm 1. We name our algorithm CASMOPOLITAN

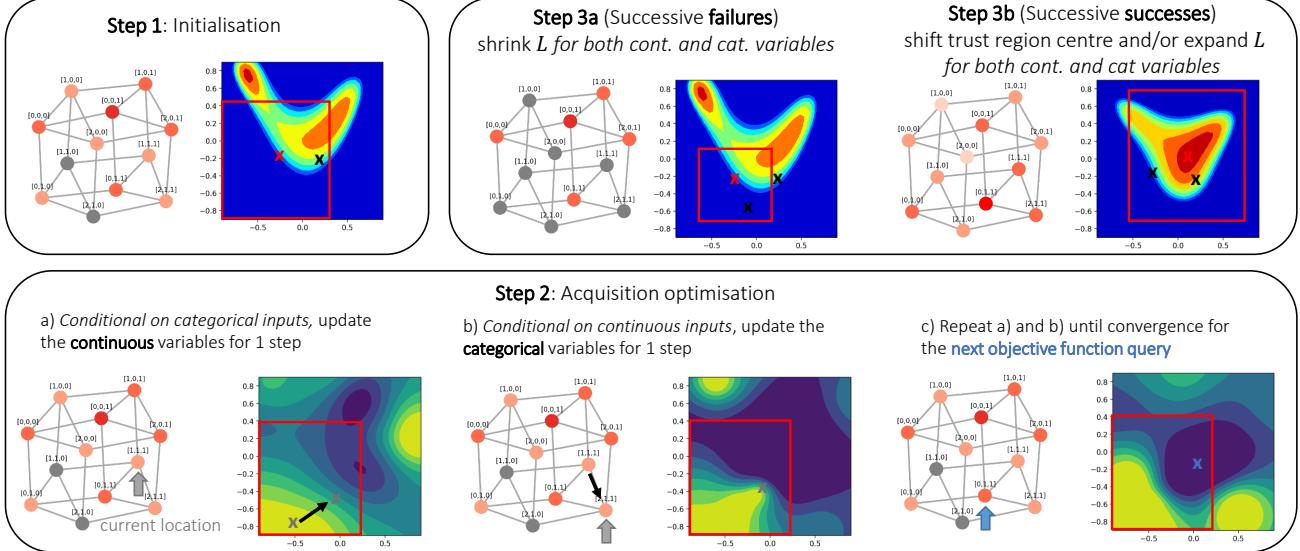


Figure 2. Illustration of CASMOPOLITAN in mixed space. Note that in **Steps 1 & 3** we show the GP posterior on \mathcal{X} conditioned on the incumbent \mathbf{h}_T^* , and in **Step 2** we show the acquisition function on \mathcal{X} conditioned on \mathbf{h} at various optimisation steps. Suppose we optimise over a 5-dimensional mixed problem with the categorical dimensions identical to that in Fig. 1 and 2 additional continuous dimensions. Initially (**Step 1**), the best location so far $\mathbf{z}_T^* = \arg \max_{\mathbf{z}} \{y_j\}_{j=1}^T = [\mathbf{h}_T^*, \mathbf{x}_T^*]$ (with the continuous TR and \mathbf{x}_T^* in red box and cross). In optimisation of acquisition function (**Step 2**), we interleave the local search on \mathcal{H} described in Sec. 3.1 with gradient-based optimisation on \mathcal{X} until convergence. In **Steps 3a/3b**, we adjust both the continuous and categorical TRs correspondingly and restart if/when either shrinks below its minimum length.

(CAtergical Spaces, or Mixed, OPTimisatiOn with Local-trust-regIons & TAilored Non-parametric), and we highlight the key design features in this section.

Kernel design In Line 12 of Algorithm 1, we impose GP on the categorical variables with a kernel defined directly on them (note that it does not increase the dimensions like one-hot transform). Specifically, we modify the overlap (or Hamming) kernel $k(\mathbf{h}, \mathbf{h}') = \frac{\sigma}{d_h} \sum_{i=1}^{d_h} \delta(h_i, h'_i)$, in Ru et al. (2020a) and Kondor and Lafferty (2002):

$$k_h(\mathbf{h}, \mathbf{h}') = \exp \left(\frac{1}{d_h} \sum_{i=1}^{d_h} \ell_i \delta(h_i, h'_i) \right), \quad (1)$$

where $\{\ell_i\}_{i=1}^{d_h}$ are the lengthscales(s)¹, and $\delta(\cdot, \cdot)$ is the Kronecker delta function. The modification affords additional expressiveness in modelling more complicated functions: for e.g., the kernel in Eq. (1) can discern the dimensions to which the objective function value is more sensitive via learning different lengthscales but the original categorical overlap kernel treats all dimensions equally. We empirically validate the performance gain of the exponentiated kernel in Sec. 4.4, and we prove this kernel is positive semi-definite (p.s.d) in App. D.1.

¹The lengthscales will be different for each dimension if we enable automatic relevance determination (ARD).

Trust region One key difficulty in applying GP-BO in high-dimensional problems is that the surrogate, by default, attempts to model the entire function landscape and over-explores. Optimisation over the categorical search space also suffer this problem. To effectively scale up the dimensions, we adapt the TR approach from Eriksson et al. (2019) in categorical search space (Line 8 in Algorithm 1). However, the challenge is that the Euclidean distance-based TR is no longer applicable; instead, we define TRs in terms of *Hamming distance*, i.e. a TR of radius L^h from the best location, \mathbf{h}^* , observed at iteration T includes all points that are up to L^h variables different from \mathbf{h}^* :

$$\text{TR}_h(\mathbf{h}^*)_{L^h} = \left\{ \mathbf{h} \mid \sum_{i=1}^{d_h} \delta(h_i, h_i^*) \leq L^h \right\}. \quad (2)$$

The TR radius is adjusted dynamically during optimisation, expanding on successive successes (if best function value f_T^* improves) and shrinking otherwise. Since Hamming distance is integer-valued bounded in $[0, d_h]$, we also set these two values as the minimum and maximum TR lengths L_{\min}^h and L_{\max}^h .

TRs in local optimisation are typically biased toward the starting points. Therefore, most local optimisation approaches rely on a restarting strategy to attain good performance (Shylo et al., 2011; Kim and Fessler, 2018). In our case, we restart the optimisation when the TR length L^h reaches the smallest possible value (Line 13 in Algorithm 1).

Rather than restarting randomly as in Eriksson et al. (2019), we propose to restart our method using GP-UCB principle (Srinivas et al., 2010), which as we will show in Section 3.3 is crucial for theoretical guarantee. Specifically, we introduce an auxiliary global GP model to achieve this. Suppose we are restarting the i -th time, we first fit the global GP model on a subset of data $D_{i-1}^* = \{\mathbf{h}_j^*, y_j^*\}_{j=1}^{i-1}$, where \mathbf{h}_j^* is the local maxima found after the j -th restart. Alternative, a random data point, if the found local maxima after the j -th restart is same as one of previous restart. Let us also denote $\mu_{gl}(\mathbf{h}; D_{i-1}^*)$ and $\sigma_{gl}^2(\mathbf{h}; D_{i-1}^*)$ as the posterior mean and variance of the global GP learned from D_{i-1}^* . Then, at the i -th restart, we select the following location $\mathbf{h}_i^{(0)}$ as the initial centre of the new TR:

$$\mathbf{h}_i^{(0)} = \arg \max_{\mathbf{h} \in \mathcal{H}} \mu_{gl}(\mathbf{h}; D_{i-1}^*) + \sqrt{\beta_i} \sigma_{gl}(\mathbf{h}; D_{i-1}^*), \quad (3)$$

where β_i is the trade-off parameter. As formally shown in Sec. 3.3, this strategy is optimal in deciding the next TR by balancing exploration against exploitation (Srinivas et al., 2010). Finally, while the use of UCB-restart is primarily theoretically driven, we show that it could offer empirical benefits over random restarts, and the readers are referred to App. B for details.

Optimisation of the acquisition function Since we preserve the discrete nature of the variables in our method, we cannot optimise the acquisition function via gradient-based methods. Instead, we use the simple strategy of local search within the TRs defined previously: at each BO iteration, we randomly sample an initial configuration $\mathbf{h}_0 \in \text{TR}_h(\mathbf{h}^*)$. We then randomly select a neighbour point of Hamming distance 1 to \mathbf{h}_0 , evaluate its acquisition function $\alpha(\cdot)$, and move from \mathbf{h}_0 if the neighbour has a higher acquisition function value and is still within the TR. We repeat this process until a pre-set budget of queries is exhausted and dispatch the best configurations for objective function evaluation (Line 10 in Algorithm 1).

3.2. Extension to Mixed Search Spaces

In addition to the purely categorical problems, our method naturally generalises to mixed, and potentially high-dimensional, categorical-continuous spaces, a setting frequently encountered in real life but hitherto under-explored in BO literature. To handle such an input $\mathbf{z} = [\mathbf{h}, \mathbf{x}]$ where \mathbf{x} is the continuous inputs, we first modify the GP kernel to the one proposed in Ru et al. (2020a):

$$k(\mathbf{z}, \mathbf{z}') = \lambda \left(k_x(\mathbf{x}, \mathbf{x}') k_h(\mathbf{h}, \mathbf{h}') \right) + (1 - \lambda) \left(k_h(\mathbf{h}, \mathbf{h}') + k_x(\mathbf{x}, \mathbf{x}') \right), \quad (4)$$

where $\lambda \in [0, 1]$ is a trade-off parameter, k_h is defined in Eq. (1) and k_x is a kernel over continuous variables (we use

the Matérn 5/2 kernel). While we use the same kernel as Ru et al. (2020a), we emphasise and formally show in Sec. 3.3 that, unlike COCABO, CASMOPOLITAN retains convergence guarantee even in the mixed space.

This formulation therefore allows us to use tailored kernels that are most appropriate for the different input types while still flexibly capturing the possible additive and multiplicative interactions between them. For the continuous inputs, we use a standard TURBO surrogate (Eriksson et al., 2019) by maintaining, and adjusting where necessary, separate standard hyper-rectangular TR(s) for them $\text{TR}_x(\mathbf{x}^*)_L = \{\mathbf{x} \mid \mathbf{x} \in \mathcal{X} \text{ and within the box centered around } \mathbf{x}^*\}$. We include an illustration in Fig. 2. We restart the continuous TR TR_x in similar manner as described in Eq. (3), if and when the either TR_h or TR_x length L reaches the smallest possible value.

Interleaved acquisition optimisation In Ru et al. (2020a), the categorical \mathbf{h} and continuous \mathbf{x} of the proposed points $\mathbf{z} = [\mathbf{h}, \mathbf{x}]$ are optimised separately similar to a *single* EM-style iteration: the categorical parts are first proposed by the multi-armed bandit; *conditioned on these*, the continuous parts are then suggested by optimising the acquisition function. In our approach, since both the categorical and continuous inputs are handled by a single, unified GP, we may propose points and optimise acquisition functions more naturally and effectively: at each optimisation step (Line 10 of Algorithm 1), we simply do one step of local search defined in Sec. 3.1 on the categorical variables, followed by one step of gradient-based optimisation of the acquisition function on the continuous variables. However, instead of doing this alternation once, we repeat until convergence or when a maximum number of steps is reached.

Other types of discrete input While we mainly focus on categorical-continuous problems, our method can be easily generalised to more complex settings by virtue of its highly flexible sub-components. For instance, we often encounter combinatorial variables with ordinal relations: for these, we treat them as categorical, but instead of using Kronecker delta function in Eq. (1) we encode the problem-specific distances. We defer a full investigation to a future work, but we include some preliminary studies in App. B.4.

3.3. Theoretical Analysis

We first provide upper bounds on the maximum information gains of our proposed categorical kernel in Eq. (1) and mixed kernel in Eq. (4) (Theorem 3.1). We then prove that after a restart, under Assumptions 3.1 and 3.2, CASMOPOLITAN converges to a local maxima after a finite number of iterations or converges to the global maximum (Theorem 3.2). Finally, we prove that with our UCB-restart strategy, under Assumptions 3.1, 3.2 and some assumptions described

in Srinivas et al. (2010), CASMOPOLITAN converges to the global maximum with a sublinear rate over the number of restarts in both categorical (Theorem 3.3) and mixed space settings (Theorem 3.4). We refer readers to App. D for the detailed proofs.

Theorem 3.1. *Let us define $\gamma(T; k; V) := \max_{A \subseteq V, |A| \leq T} \frac{1}{2} \log |I| + \sigma^{-2} [k(\mathbf{v}, \mathbf{v}')]_{\mathbf{v}, \mathbf{v}' \in A}$ as the maximum information gain achieved by sampling T points in a GP defined over a set V with a kernel k . Let us define $\tilde{N} := \prod_{j=1}^{d_h} n_j$, then we have,*

1. *For the categorical kernel k_h , $\gamma(T; k_h; \mathcal{H}) = \mathcal{O}(\tilde{N} \log T)$;*
2. *For the mixed kernel k , $\gamma(T; k; [\mathcal{H}, \mathcal{X}]) \leq \mathcal{O}((\lambda \tilde{N} + 1 - \lambda) \gamma(T; k_x; \mathcal{X}) + (\tilde{N} + 2 - 2\lambda) \log T)$.*

Using Theorem 3.1, the maximum information gain of the mixed kernel k can be upper bounded for some common continuous kernels k_x . For instance, when k_x is the Matérn kernel, the maximum information gain $\gamma(T; k; [\mathcal{H}, \mathcal{X}])$ of the mixed kernel is upper bounded by $\mathcal{O}((\lambda \tilde{N} + 1 - \lambda) T^{d_x(d_x+1)/(2v+d_x(d_x+1))} (\log T) + (\tilde{N} + 2 - 2\lambda) \log T)$ as $\gamma(T; k_{Mt}; \mathcal{X}) = \mathcal{O}(T^{d_x(d_x+1)/(2v+d_x(d_x+1))} (\log T))$ (Srinivas et al., 2010). Similar bounds can be established when k_x is the squared exponential or the linear kernel.

To analyse the convergence property of CASMOPOLITAN, similar to any TR-based algorithm (Yuan, 2000), we assume that (i) f is bounded in $[\mathcal{H}, \mathcal{X}]$ (Assumption 3.1), and (ii), given a small enough region, the surrogate model (i.e. GP) accurately approximates f with any data point belonging to this region (Assumption 3.2). We note that Assumption 3.1 is common as it is generally assumed in BO that f is Lipschitz continuous (Brochu et al., 2010), thus f is bounded given the search space is bounded. Assumption 3.2 considers the minimum TR lengths L_{\min}^x, L_{\min}^h are set to be small enough so that GP approximates f accurately in TRs specified in Assumption 3.2. We note that in practice, this assumption is only possible asymptotically, i.e. when the number of observed data in these TRs goes to infinity. In our implementation (see App. C), these TRs are always set to be very small so that Assumption 3.2 can be close to true.

Assumption 3.1. *The objective function $f(\mathbf{z})$ is bounded in $[\mathcal{H}, \mathcal{X}]$, i.e. $\exists F_l, F_u \in \mathbb{R} : \forall \mathbf{z} \in [\mathcal{H}, \mathcal{X}], F_l \leq f(\mathbf{z}) \leq F_u$.*

Assumption 3.2. *Let us denote L_{\min}^h, L_{\min}^x and L_0^h, L_0^x be the minimum and initial TR lengths for the categorical and continuous variables, respectively. Let us also denote α_s as the shrinking rate of the TRs. In the categorical setting, for any TR with length $\leq \lceil (L_{\min}^h + 1)/\alpha_s \rceil - 1$,² the corresponding local GP approximates f accurately. That is, the GP posterior mean approximates f accurately whilst the GP posterior variance is negligible within this TR. In the mixed space setting, the local*

GP approximates f accurately within any TR with length $L^x \leq \max(L_{\min}^x/\alpha_s, L_0^x(\lceil (L_{\min}^h + 1)/\alpha_s \rceil - 1)/L_0^h)$ and $L^h \leq \max(\lceil (L_{\min}^h + 1)/\alpha_s \rceil - 1, \lceil L_0^h L_{\min}^x / (\alpha_s L_0^x) \rceil)$.

Theorem 3.2. *Given Assumptions 3.1 & 3.2, after a restart, CASMOPOLITAN converges to a local maxima after a finite number of iterations or converges to the global maximum.*

Finally, we define the cumulative regret after I restarts, R_I , to be $\sum_{j=1}^I (f(\mathbf{z}^*) - f(\mathbf{z}_j^*))$ with \mathbf{z}_j^* being the local maxima found at the j -th restart and \mathbf{z}^* being the global maximum of f . We then provide the regret bounds of CASMOPOLITAN in both categorical (Theorem 3.3) and mixed space setting (Theorem 3.4). With these regret bounds, it can be seen that CASMOPOLITAN converges to the global maximum with a sublinear rate over the number of restarts (i.e. $R_I/I \xrightarrow{I \rightarrow \infty} 0$) in both categorical and mixed space settings.

Theorem 3.3. *Let us consider the categorical setting, $f : \mathcal{H} \rightarrow \mathbb{R}$. Let $\zeta \in (0, 1)$ and $\beta_i = 2 \log(|\mathcal{H}|^2 \pi^2 / 6\zeta)$ at the i -th restart. Suppose the objective function f satisfies that: there exists a class of functions which pass through all the local maxima of f ,³ share the same global maximum with f , and is sampled from the auxiliary global GP $GP(0, k_h)$. Then given Assumptions 3.1 & 3.2, CASMOPOLITAN obtains a regret bound of $\mathcal{O}^*(\sqrt{I \gamma(I; k_h, \mathcal{H}) \log |\mathcal{H}|})$ w.h.p. Formally,*

$$\Pr\left\{R_I \leq \sqrt{C_1 I \beta_I \gamma(I; k_h, \mathcal{H})} \quad \forall I \geq 1\right\} \geq 1 - \zeta,$$

with $C_1 = 8/\log(1 + \sigma^{-2})$, $\gamma(I; k_h, \mathcal{H}) = \mathcal{O}(\tilde{N} \log(\tilde{N}) \log(I))$ and $\tilde{N} = \prod_{j=1}^{d_h} n_j$.

Theorem 3.4. *Let us consider the mixed space setting, $f : [\mathcal{H}, \mathcal{X}] \rightarrow \mathbb{R}$. Let $\zeta \in (0, 1)$. Suppose the objective function f satisfies that: there exists a class of functions g which pass through all the local maxima of f , share the same global maximum with f and lies in the RKHS $\mathcal{G}_k([\mathcal{H}, \mathcal{X}])$ corresponding to the kernel k of the auxiliary global GP model. Suppose that the noise ϵ_i has zero mean conditioned on the history and is bounded by σ almost surely. Assume $\|g\|_k^2 \leq B$, and let $\beta_i = 2B + 300\gamma_i \log(i/\zeta)^3$, then given Assumptions 3.1 & 3.2, CASMOPOLITAN obtains a regret bound of $\mathcal{O}^*(\sqrt{I \gamma(I; k, [\mathcal{H}, \mathcal{X}]) \beta_I})$ w.h.p. Specifically,*

$$\Pr\left\{R_I \leq \sqrt{C_1 I \beta_I \gamma(I; k, [\mathcal{H}, \mathcal{X}])} \quad \forall I \geq 1\right\} \geq 1 - \zeta,$$

with $C_1 = 8/\log(1 + \sigma^{-2})$, $\gamma(I; k, [\mathcal{H}, \mathcal{X}]) = \mathcal{O}((\lambda \tilde{N} + 1 - \lambda) \gamma(T; k_x, \mathcal{X}) + (\tilde{N} + 2 - 2\lambda) \log T)$ and $\tilde{N} = \prod_{j=1}^{d_h} n_j$.

Discussion We show in Theorem 3.2 that our TR-based algorithm with BO converges to a local maxima or global maximum after a restart. We note that similar convergence can

³This means for every function g belonging to this class of functions, $g(\mathbf{h}_j^*) = f(\mathbf{h}_j^*)$ where \mathbf{h}_j^* is a local maxima of f .

²The operator $\lceil \cdot \rceil$ denotes the ceiling function.

be found in the original TR-based algorithms using gradient-descent (Yuan, 2000). However, our proof technique is very different from Yuan (2000). In addition, in Theorems 3.3 & 3.4, the fact that CASMOPOLITAN converges to the global maximum with a sublinear rate over the number of restarts - not over the number of iterations as in Srinivas et al. (2010) - can be considered as the price paid for a more relaxed assumption. In particular, Srinivas et al. (2010) assume that it is possible to model the objective function f with a GP with kernel k on the whole search space. On the other hand, we relax this assumption in Theorems 3.3 & 3.4 by assuming that there is a class of functions, which pass through the local maxima and share the same global maximum with f , that we can model with a GP with kernel k . Further details on this class of functions can be found in Apps. D.4 & D.5.

Despite the aforementioned strengths, there are some limitations with our theoretical analysis. First, the maximum information gains $\gamma(T; k_h; \mathcal{H})$ and $\gamma(T; k; [\mathcal{H}, \mathcal{X}])$ derived in Theorem 3.1 increase exponentially with the dimension of the categorical input (d_h). Thus, these terms can be large when the categorical dimension is high. As we are solving a noisy NP-hard combinatorial problem, it might not be possible to get away these exponential terms without a strict assumption. Second, as briefly discussed above, Assumption 3.2 is true asymptotically, resulting Theorems 3.2, 3.3 and 3.4 to hold asymptotically. One way to eliminate this assumption is to instead prove CASMOPOLITAN achieves ϵ -accuracy, that is, CASMOPOLITAN can find a point whose function value is within ϵ of the objective function global maximum, where ϵ is a small positive value depending on the minimum TR lengths L_{\min}^x, L_{\min}^h . We consider these directions for future work.

4. Experiments

4.1. Categorical Problems

We first evaluate our proposed method on a number of optimisation problems in the categorical search space against a number of competitive baselines, including TPE (Bergstra et al., 2011), SMAC (Hutter et al., 2011), BOCS (Baptista and Poloczek, 2018)⁴ and COMBO (Oh et al., 2019) which claims the state-of-the-art performance amongst comparable algorithms. We also include two additional baselines: BO, which performs the naïve BO approach after converting the categorical variables into one-hot representations, and TURBO, which is identical to BO except that we additionally incorporate the TR approach in Eriksson et al. (2019). We experiment on following real-life problems (for detailed implementation and descriptions for the setup of

⁴BOCS is only run in Contamination, as it by default does not support multi-categorical optimisation and on Weighted Maximum Satisfiability (MAXSAT), a single trial takes more than 100 hours, rendering comparison infeasible within our computing constraints.

these problems and those in Sec. 4.2, see App. C).

- Contamination control over 25 binary variables (3.35×10^7 configurations). This problem and the Pest control problem below simulate the dynamics of real-life problems whose evaluations are extremely expensive (Hu et al., 2010).
- Pest control over 25 variables, with 5 possible options for each (2.98×10^{17} configurations) (Oh et al., 2019).
- Weighted maximum satisfiability (MAXSAT) problem over 60 binary variables (1.15×10^{18} configurations).

In all experiments in this section and Sec. 4.2, we report the sequential version (denoted as `CASMOPOLITAN-1` as batch size $b = 1$) of our method as all baselines we consider are also sequential. We investigate the parallel version of varying batch sizes of our method in Sec. 4.3.

The results are shown in Fig. 3: our method achieves the best convergence speed and sample efficiency in general, and in terms of the performance at termination, our method again outperforms the rest except in Contamination and MAXSAT where it performs on par with COMBO. However, it is worth noting that in terms of wall-clock speed, our method is 2 – 3 times faster than COMBO in the problems considered (See App. B).

4.2. Mixed Problems

We then consider the optimisation problems involving a mix of continuous and categorical input variables. In these experiments, in addition to SMAC, TPE, BO and TURBO described in Section 4.1, we also include a number of recent advancements in this setup including COCABO (Ru et al., 2020a) and MVRSM (Bliiek et al., 2020). Additionally, we run a small comparison against several other high-dimensional BO methods such as ALEBO (Letham et al., 2020) and REMBO (Wang et al., 2016), and the readers are referred to details in App. B. Note that we do not compare against BOCS and COMBO since they are suitable for purely categorical spaces only. Under this setup, we consider the following synthetic and real-life problems of increasing dimensionality and complexity:

- Func2C with $d_h = 2$ and $d_x = 2$, and Func3C with $d_h = 3$ and $d_x = 3$, respectively (Ru et al., 2020a).
- Hyperparameter tuning of the XGBoost model (Chen and Guestrin, 2016) on the MNIST dataset (LeCun, 1998), with $d_x = 5$ and $d_h = 3$ with 2 choices for each.
- 53-dimensional Ackley function (Ackley-53) (Bliiek et al., 2020) with $d_h = 50$ where $\mathbf{h} \in \{0, 1\}^{50}$ and $d_x = 3$ where $\mathbf{x} \in [-1, 1]^3$.
- Black-box adversarial attack on a CNN trained on CIFAR-10 inspired by Ru et al. (2020b), but with adapted sparse setups where we perturb a small number of pixels only. The task is an optimisation problem with $d_h = 43$ (42 pixel locations being attacked with $n_{1:42} = 14$ choices each and the image upsampling technique which has $n_{43} = 3$

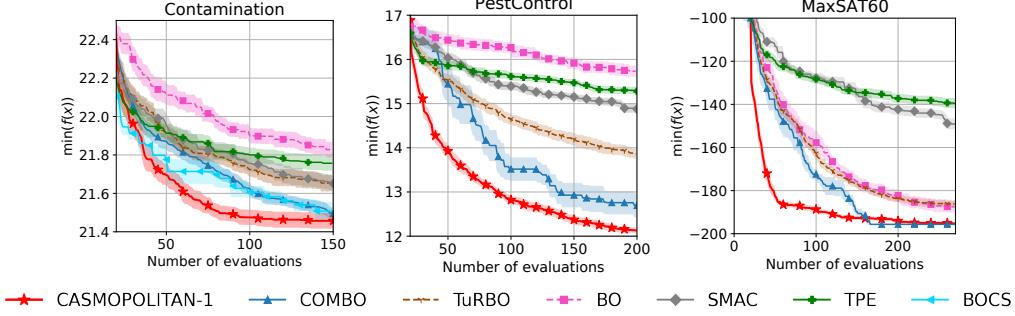


Figure 3. Results on various categorical optimisation problems. Lines and shaded area denote mean ± 1 standard error.

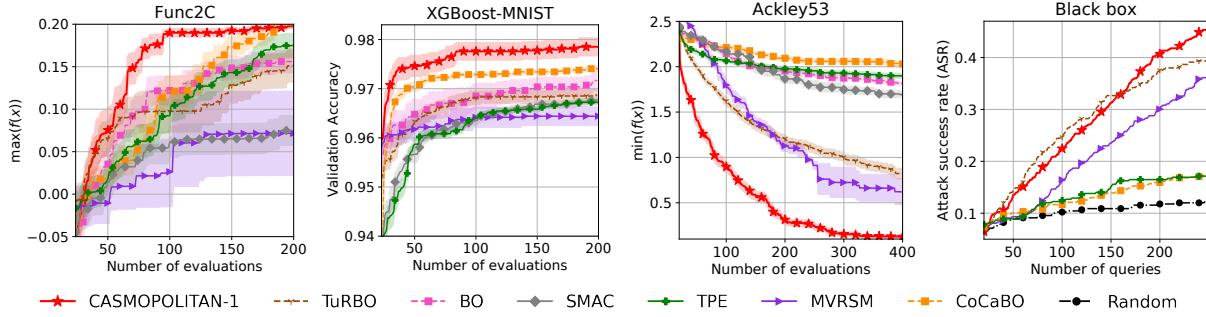


Figure 4. Results on various mixed optimisation problems. Lines and shaded area denote mean ± 1 standard error (except for Black-box where we show the ASR against number of queries). Additional experiment results in App. B.

choices) and $d_x = 42$ for continuous perturbation added to each pixel under attack. We perform a total of 450 targeted attack instances and limit the maximum budget to be 250 queries for each attack to simulate a highly constrained attack setup.

We report the results on the objective function values in Fig. 4 except for the black-box attack, where we instead report the attack success rate ASR against the number of queries following Ru et al. (2020b) (Additional attack results are shown in App. B). In this problem we also compare against random search, as it has been shown to be a strong baseline both in adversarial attack (Croce et al., 2020) and high-dimensional black-box optimisation (Rana et al., 2017) literature. Overall, it is evident that CASMOPOLITAN performs the best, but it is also interesting to observe that in lower dimensions (the first 2 problems), COCABO featuring tailored categorical kernels performs well, while MVRSM and categorical variable-agnostic TURBO, both focusing on high dimensions, under-perform. However, in high-dimensional problems (last 2 problems), the relative performance switches completely, suggesting that the focus on dimensionality now outweighs the importance of treating different input types differently. Nonetheless, with both tailored kernels and focus on scaling to high dimensions, CASMOPOLITAN consistently out-performs by a comfortable margin, further demonstrating its versatility.

4.3. Parallel Setting

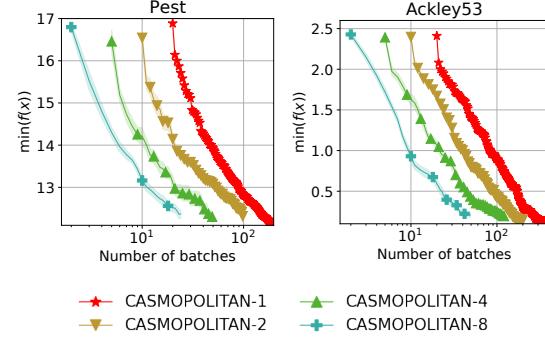


Figure 5. Parallel CASMOPOLITAN on representative categorical and mixed problems by *number of batches* excluding the initially randomly-sampled batches. Note the x-axis is in log-scale for better presentation. We show the comparison by *number of function queries* in App. B.

We would often like to exploit parallelism in computing where we dispatch different queries to the black-box objective function for independent evaluations. This setting necessitates the development of *batch* methods to propose a batch of b points for simultaneous evaluation at each BO iteration. However, this often involves trade-off between wall-clock time efficiency against performance, because surrogates in batch methods are updated only once per b

objective function evaluations. Here we investigate the performance of CASMOPOLITAN under different batch sizes where $b = 1$ (sequential setting) 2, 4 & 8 in Pest Control and Ackley-53 problems previously considered; where $b > 1$, we use the Kriging believer strategy (Ginsbourger et al., 2010) during acquisition optimisation to deliver b proposals simultaneously. In both experiments, we keep the budget of the objective function queries to be identical to that in Sec. 4.1 & 4.2 but scale the number of batches accordingly, and the results are shown in Fig. 5: it is evident that larger batch sizes, while leading to almost linear reduction in wall-clock time, do not lead to significant performance deterioration, except some minor under-performance at the end which seems to scale with b . However, in both problems, CASMOPOLITAN even with the largest batch size investigated still outperforms *sequential* baselines in Figs. 3 & 4.

4.4. Ablation Studies

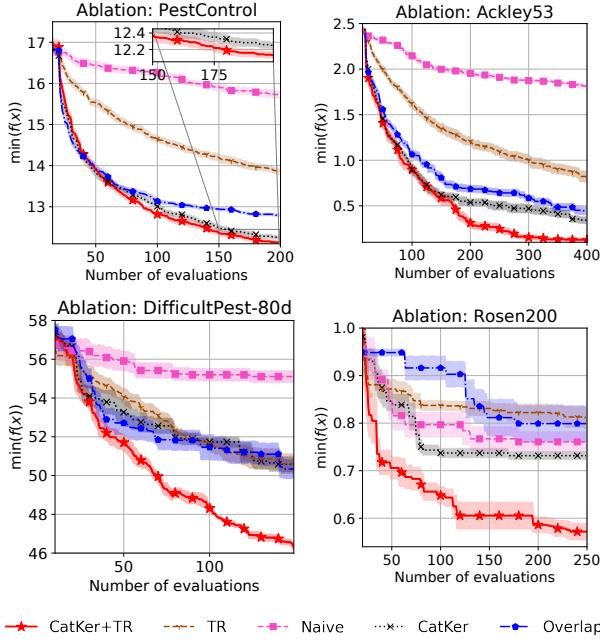


Figure 6. Ablation studies of our method in categorical (left) and mixed (right) optimisation problems. First row: Pest (left), Ackley-53 (right); Second row: DifficultPest (left), Rosenbrock-200 (right).

Our method introduces a number of modifications over the naïve BO approach. To understand the benefits of these, we conduct ablation studies in both the categorical and mixed problems. Specifically, we include the following setups.

- The naïve BO approach with global GP surrogate and one-hot transformation on the categorical variables (Naive);
- One-hot transformed BO, but with *local* TRs i.e. TURBO (TR);
- GP with *global* surrogates, but with the categorical over-

lapping kernel in Ru et al. (2020a) where applicable (Overlap);

- BO with *global* GP surrogate, but with the kernel defined in Eq. (1) or (4), where appropriate (CatKer);
- Our approach that incorporates both local modelling and the kernel in Eq. (1) or (4) (CatKer+TR).

We firstly include Pest Control and the Ackley-53 problems as representative problems for the categorical and mixed setups for the ablation studies. To further understand the relative importance of the various features of CASMOPOLITAN especially as the dimensionality of the problems changes, we also include two even higher-dimensional problems, namely 1) Pest control with number of stages expanded to 80, which we term DifficultPest (the number of possible configurations is more than 8.27×10^{55}), and 2) 200-d Rosenbrock with 100 binary dimensions and 100 continuous dimensions (detailed in App. C).

We show the results in Fig. 6: in most problems, the usage of the categorical kernel leads to improvements over baselines, with kernels used in our method generally outperforming the overlap kernel. Unsurprisingly, the additional benefits of local optimisation and the use of trust regions increase with increasing dimensionality and complexity of the problems, with largest benefits coming from the two high-dimensional problems of the second row. Nonetheless, it is worth noting that even in the relatively modestly-dimensioned Pest Control problem where the difference between CatKer+TR and CatKer seems small, the outperformance is still statistically significant (Two-sample Student's t-test yields $p = 0.043 < 0.05$ at the final iteration). Finally, our method, similar to TURBO, introduces a number of additional hyperparameters related to the TR; we examine the sensitivity of performance towards these extra hyperparameters in App. B.

5. Conclusion and Future Work

We propose CASMOPOLITAN, a novel GP-BO approach using ideas of tailored kernels and trust regions to tackle the challenging high-dimensional optimisation problem over categorical and mixed search spaces. We both analyse our method theoretically and empirically demonstrate its effectiveness over a wide range of problems. Possible future directions may extend our model to even more diverse search spaces, such as problems on graphs, trees, and/or in conditional spaces.

Acknowledgements

The authors would like to thank the Oxford-Man Institute of Quantitative Finance for providing computing resources in this project. The authors also thank the anonymous ICML reviewers and the area chair for the constructive feedback which helped to improve the paper.

References

- Moustafa Alzantot, Yash Sharma, Supriyo Chakraborty, Huan Zhang, Cho-Jui Hsieh, and Mani B Srivastava. Genattack: Practical black-box attacks with gradient-free optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1111–1119, 2019. (Cited on page 19)
- Ricardo Baptista and Matthias Poloczek. Bayesian optimisation of combinatorial structures. In *International Conference on Machine Learning*, pages 462–471. PMLR, 2018. (Cited on pages 2, 7, 13, and 21)
- James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, pages 2546–2554, 2011. (Cited on page 7)
- Felix Berkenkamp, Angela P. Schoellig, and Andreas Krause. No-regret bayesian optimization with unknown hyperparameters. *Journal of Machine Learning Research*, 20(50):1–24, 2019. (Cited on page 20)
- John Bibby. Axiomatisations of the average and a further generalisation of monotonic sequences. *Glasgow Mathematical Journal*, 15(1):63–65, 1974. (Cited on page 23)
- Mickaël Binois, David Ginsbourger, and Olivier Roustant. A warped kernel improving robustness in Bayesian optimization via random embeddings. In *International Conference on Learning and Intelligent Optimization*, pages 281–286. Springer, 2015. (Cited on page 2)
- Mickaël Binois, David Ginsbourger, and Olivier Roustant. On the choice of the low-dimensional domain for global optimization via random embeddings. *Journal of global optimization*, 76(1):69–90, 2020. (Cited on page 2)
- Laurens Bliek, Sicco Verwer, and Mathijs de Weerdt. Black-box mixed-variable optimisation using a surrogate model that satisfies integer constraints. *arXiv preprint arXiv:2006.04508*, 2020. (Cited on pages 2, 7, 19, 21, and 26)
- Leo Breiman. Random forests. *Machine learning*, 45(1): 5–32, 2001. (Cited on page 3)
- Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010. (Cited on pages 1, 6, and 13)
- Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SigKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2016. (Cited on page 7)
- Nadia Creignou, Sanjeev Khanna, and Madhu Sudan. *Complexity classifications of boolean constraint satisfaction problems*. SIAM, 2001. (Cited on page 1)
- Francesco Croce, Maksym Andriushchenko, Naman D Singh, Nicolas Flammarion, and Matthias Hein. Sparsers: a versatile framework for query-efficient sparse black-box adversarial attacks. *arXiv preprint arXiv:2006.12834*, 2020. (Cited on page 8)
- Hamid Dadkhahi, Karthikeyan Shanmugam, Jesus Rios, Payel Das, Samuel C Hoffman, Troy David Loeffler, and Subramanian Sankaranarayanan. Combinatorial black-box optimization with expert advice. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1918–1927, 2020. (Cited on page 2)
- Hamid Dadkhahi, Jesus Rios, Karthikeyan Shanmugam, and Payel Das. Fourier representations for black-box optimization over categorical variables. 2021. (Cited on page 2)
- Erik Daxberger, Anastasia Makarova, Matteo Turchetta, and Andreas Krause. Mixed-variable Bayesian optimization. *arXiv preprint arXiv:1907.01329*, 2019. (Cited on page 2)
- Aryan Deshwal, Syrine Belakaria, Janardhan Rao Doppa, and Alan Fern. Optimizing discrete spaces via expensive evaluations: A learning to search framework. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3773–3780, Apr. 2020. (Cited on page 2)
- David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local Bayesian optimization. In *Advances in Neural Information Processing Systems*, pages 5496–5507, 2019. (Cited on pages 1, 2, 4, 5, 7, 20, and 21)
- Peter I Frazier. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018. (Cited on page 13)
- Jacob R Gardner, Geoff Pleiss, David Bindel, Kilian Q Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *arXiv preprint arXiv:1809.11165*, 2018. (Cited on page 13)
- Eduardo C Garrido-Merchán and Daniel Hernández-Lobato. Dealing with categorical and integer-valued variables in Bayesian optimization with Gaussian processes. *Neurocomputing*, 380:20–35, 2020. (Cited on page 2)
- David Ginsbourger, Rodolphe Le Riche, and Laurent Carraro. Kriging is well-suited to parallelize optimization. In *Computational intelligence in expensive optimization problems*, pages 131–162. Springer, 2010. (Cited on pages 9 and 20)

- Shivapratap Gopakumar, Sunil Gupta, Santu Rana, Vu Nguyen, and Svetha Venkatesh. Algorithmic assurance: An active approach to algorithmic testing using Bayesian optimisation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5465–5473, 2018. (Cited on pages 1 and 2)
- GPyOpt. GPyOpt: A Bayesian optimization framework in python. <http://github.com/SheffieldML/GPyOpt>, 2016. (Cited on page 2)
- José Miguel Hernández-Lobato, James Requeima, Edward O Pyzer-Knapp, and Alán Aspuru-Guzik. Parallel and distributed Thompson sampling for large-scale accelerated exploration of chemical space. In *International Conference on Machine Learning*, pages 1470–1479, 2017. (Cited on page 1)
- Yingjie Hu, JianQiang Hu, Yifan Xu, Fengchun Wang, and Rong Zeng Cao. Contamination control in food supply chain. In *Proceedings of the 2010 Winter Simulation Conference*, pages 2678–2681. IEEE, 2010. (Cited on pages 7, 17, and 26)
- Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization*, pages 507–523. Springer, 2011. (Cited on pages 1, 3, and 7)
- Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998. (Cited on page 1)
- Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. High dimensional Bayesian optimisation and bandits via additive models. In *International Conference on Machine Learning*, pages 295–304, 2015. (Cited on page 2)
- Kirthevasan Kandasamy, Willie Neiswanger, Jeff Schneider, Barnabas Poczos, and Eric P Xing. Neural architecture search with bayesian optimisation and optimal transport. In *Advances in Neural Information Processing Systems*, pages 2016–2025, 2018. (Cited on page 1)
- Donghwan Kim and Jeffrey A Fessler. Adaptive restart of the optimized gradient method for convex optimization. *Journal of Optimization Theory and Applications*, 178(1): 240–263, 2018. (Cited on page 4)
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015. (Cited on page 20)
- Risi Kondor and John D. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *International Conference on Machine Learning*, pages 315–322, 2002. (Cited on page 4)
- Andreas Krause and Cheng S Ong. Contextual Gaussian process bandit optimization. In *Advances in Neural Information Processing Systems*, pages 2447–2455, 2011. (Cited on page 23)
- Yann LeCun. The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>, 1998. (Cited on page 7)
- Ben Letham, Roberto Calandra, Akshara Rai, and Eytan Bakshy. Re-examining linear embeddings for high-dimensional bayesian optimization. *Advances in Neural Information Processing Systems*, 33, 2020. (Cited on pages 2, 7, and 16)
- Vladimir Mazya and Tatyana Shaposhnikova. *Jacques Hadamard: A Universal Mathematician*. 1st edition, 1999. (Cited on page 23)
- Mojmír Mutný and Andreas Krause. Efficient high dimensional Bayesian optimization with additivity and quadrature fourier features. *Advances in Neural Information Processing Systems 31*, pages 9005–9016, 2019. (Cited on page 2)
- Amin Nayebi, Alexander Munteanu, and Matthias Poloczek. A framework for Bayesian optimization in embedded subspaces. In *International Conference on Machine Learning*, pages 4752–4761. PMLR, 2019. (Cited on page 2)
- Dang Nguyen, Sunil Gupta, Santu Rana, Alistair Shilton, and Svetha Venkatesh. Bayesian optimization for categorical and category-specific continuous inputs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5256–5263, 2020. (Cited on pages 1 and 2)
- Vu Nguyen, Tam Le, Makoto Yamada, and Michael A Osborne. Optimal transport kernels for sequential and parallel neural architecture search. In *International Conference on Machine Learning*, 2021. (Cited on page 1)
- Changyong Oh, Jakub Tomczak, Efstratios Gavves, and Max Welling. Combinatorial Bayesian optimization using the graph cartesian product. In *Advances in Neural Information Processing Systems*, pages 2914–2924, 2019. (Cited on pages 2, 7, 13, 15, 18, 21, and 26)
- Jack Parker-Holder, Vu Nguyen, and Stephen J Roberts. Provably efficient online hyperparameter optimization with population-based bandits. *Advances in Neural Information Processing Systems*, 33, 2020. (Cited on page 1)
- Santu Rana, Cheng Li, Sunil Gupta, Vu Nguyen, and Svetha Venkatesh. High dimensional Bayesian optimization with elastic gaussian process. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 2883–2891, 2017. (Cited on pages 1 and 8)

- Carl Edward Rasmussen. Gaussian processes for machine learning. 2006. (Cited on pages 2, 3, and 13)
- Paul Rolland, Jonathan Scarlett, Ilija Bogunovic, and Volkan Cevher. High-dimensional Bayesian optimization via additive models with overlapping groups. In *International conference on artificial intelligence and statistics*, pages 298–307. PMLR, 2018. (Cited on page 2)
- Binxin Ru, Ahsan Alvi, Vu Nguyen, Michael A Osborne, and Stephen Roberts. Bayesian optimisation over multiple continuous and categorical inputs. In *International Conference on Machine Learning*, pages 8276–8285. PMLR, 2020a. (Cited on pages 1, 2, 4, 5, 7, 9, 18, 19, 20, 21, and 26)
- Binxin Ru, Adam Cobb, Arno Blaas, and Yarin Gal. Bayesopt adversarial attack. In *International Conference on Learning Representations*, 2020b. (Cited on pages 7, 8, and 19)
- Binxin Ru, Xingchen Wan, Xiaowen Dong, and Michael Osborne. Interpretable neural architecture search via Bayesian optimisation with weisfeiler-lehman kernels. *International Conference on Learning Representations*, 2021. (Cited on page 1)
- Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016. (Cited on pages 1 and 13)
- Oleg V Shylo, Timothy Middelkoop, and Panos M Pardalos. Restart strategies in optimization: parallel and serial cases. *Parallel Computing*, 37(1):60–68, 2011. (Cited on page 4)
- Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, pages 2951–2959, 2012. (Cited on pages 1 and 2)
- Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning*, pages 1015–1022, 2010. (Cited on pages 5, 6, 7, 24, and 25)
- Kevin Swersky, Yulia Rubanova, David Dohan, and Kevin Murphy. Amortized bayesian optimization over discrete spaces. In *Conference on Uncertainty in Artificial Intelligence*, pages 769–778. PMLR, 2020. (Cited on page 22)
- James Joseph Sylvester. Xxxvii. on the relation between the minor determinants of linearly equivalent quadratic functions. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 1(4):295–305, 1851. doi: 10.1080/14786445108646735. (Cited on page 22)
- Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 742–749, 2019. (Cited on page 19)
- Zi Wang, Chengtao Li, Stefanie Jegelka, and Pushmeet Kohli. Batched high-dimensional Bayesian optimization via structural kernel learning. In *International Conference on Machine Learning*, pages 3656–3664. PMLR, 2017. (Cited on page 2)
- Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. Batched large-scale Bayesian optimization in high-dimensional spaces. In *International Conference on Artificial Intelligence and Statistics*, pages 745–754. PMLR, 2018. (Cited on page 2)
- Ziyu Wang, Masrour Zoghi, Frank Hutter, David Matheson, N Freitas, et al. Bayesian optimization in high dimensions via random embeddings. AAAI Press/International Joint Conferences on Artificial Intelligence, 2013. (Cited on page 16)
- Ziyu Wang, Frank Hutter, Masrour Zoghi, David Matheson, and Nando de Feitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016. (Cited on pages 2 and 7)
- Ya-xiang Yuan. A review of trust region algorithms for optimization. In *Iciam*, volume 99, pages 271–282. Citeseer, 2000. (Cited on pages 6 and 7)

Appendix

A. Primer on GP and BO

Gaussian processes We consider a GP surrogate model for a black-box function f which takes an input $\mathbf{z} = [\mathbf{h}, \mathbf{x}]$ and returns an output $y = f(\mathbf{z}) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Here, the input includes a continuous variable \mathbf{x} and a categorical variable \mathbf{h} . A GP defines a probability distribution over functions f under the assumption that any finite subset $\{(\mathbf{z}_i, f(\mathbf{z}_i))\}$ follows a normal distribution (Rasmussen, 2006). Formally, a GP is denoted as $f(\mathbf{z}) \sim \text{GP}(m(\mathbf{z}), k(\mathbf{z}, \mathbf{z}'))$, where $m(\mathbf{z})$ and $k(\mathbf{z}, \mathbf{z}')$ are called the mean and covariance functions respectively, i.e. $m(\mathbf{z}) = \mathbb{E}[f(\mathbf{z})]$ and $k(\mathbf{z}, \mathbf{z}') = \mathbb{E}[(f(\mathbf{z}) - m(\mathbf{z}))(f(\mathbf{z}') - m(\mathbf{z}'))^T]$. The covariance function (kernel) $k(\mathbf{z}, \mathbf{z}')$ can be thought of as a similarity measure relating $f(\mathbf{z})$ and $f(\mathbf{z}')$. There have been various proposed kernels which encode different prior beliefs about the function $f(\mathbf{z})$, typically in the continuous space. Popular choices include the Square Exponential kernel, the Matérn kernel (Rasmussen, 2006).

Assume the zero mean prior $m(\mathbf{z}) = 0$, to predict $f_* = f(\mathbf{z}_*)$ at a new data point \mathbf{z}_* , we have,

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \mathbf{K} & \mathbf{k}_*^T \\ \mathbf{k}_* & k_{**} \end{bmatrix}\right), \quad (5)$$

where $k_{**} = k(\mathbf{z}_*, \mathbf{z}_*)$, $\mathbf{k}_* = [k(\mathbf{z}_*, \mathbf{z}_i)]_{\forall i \leq N}$ and $\mathbf{K} = [k(\mathbf{z}_i, \mathbf{z}_j)]_{\forall i, j \leq N}$. Combining Eq. (5) with the fact that $p(f_* | f)$ follows a univariate Gaussian distribution $\mathcal{N}(\mu(\mathbf{z}_*), \sigma^2(\mathbf{z}_*))$, the GP posterior mean and variance can be computed as,

$$\begin{aligned} \mu(\mathbf{z}_*) &= \mathbf{k}_* [\mathbf{K} + \sigma^2 \mathbf{I}]^{-1} \mathbf{y}, \\ \sigma^2(\mathbf{z}_*) &= k_{**} - \mathbf{k}_* [\mathbf{K} + \sigma^2 \mathbf{I}]^{-1} \mathbf{k}_*^T. \end{aligned}$$

As GPs give full uncertainty information with any prediction, they provide a flexible nonparametric prior for Bayesian optimisation. We refer the interested readers to Rasmussen (2006) for further details on GPs.

Bayesian optimisation Bayesian optimisation is a powerful sequential approach to find the global optimum of an expensive black-box function $f(\mathbf{z})$ without making use of derivatives. First, a surrogate model is learned from all the current observed data $\mathcal{D}_t = \{\mathbf{z}_i, y_i\}_{i=1}^t$ to approximate the behavior of $f(\mathbf{z})$. Second, an acquisition function is derived from the surrogate model to select new data points that mostly inform about the global optimum. The process is conducted iteratively until the evaluation budget is depleted, and the global optimum is estimated based on all the sampled data. In-depth discussions about Bayesian optimisation beyond this brief overview can be found in recent surveys (Brochu et al., 2010; Shahriari et al., 2016; Frazier, 2018).

B. Additional Experimental Results

B.1. Running Time Comparison

In this section, we provide comparison of CASMOPOLITAN against some baselines in terms of wall-clock running time on a number of problems considered. However, since we conduct our experiments on a shared server, inevitably there are fluctuations in wall clock time depending on the server load, leading to (perhaps rather significant) amount of uncertainty over the computing time reported here and thus, the figures here are for ballpark reference only. CASMOPOLITAN scales $\mathcal{O}(N^3)$, where N here refers to the number of training data (note that in CASMOPOLITAN, this is not necessarily the total number of observations, but only the number of training samples of the GP surrogate of the current restart), which is the time complexity of any GP-BO method where the computational bottleneck is the inversion of the covariance matrix (Shahriari et al., 2016). Practically, due to the implementation in GPytorch which utilises Blackbox Matrix-matrix (BBMM) multiplication which reduces the cost of exact GP inference to $\mathcal{O}(N^2)$ (Gardner et al., 2018). Overall, the computing cost of CASMOPOLITAN is generally comparable to TURBO. On the other hand, previous methods generally scale worse. For example, in addition to the inherent $\mathcal{O}(N^3)$ complexity (or $\mathcal{O}(N^2)$ if BBMM is similarly exploited), COMBO additionally incurs the cost in the graph Fourier transform of $\mathcal{O}(\sum_{i=1}^{d_h} n_h^3)$ using the notations of our paper (Proposition 2.3.1 in Oh et al. (2019)). Furthermore, it also uses slice sampling for the approximate marginalisation of the posterior predictive distribution, which is arguably more expensive than simple optimisation of the log marginal-likelihood. A single iteration of BOCS incurs complexity of $\mathcal{O}(N^2 d_h^2)$ (Baptista and Poloczek, 2018), suggesting that the runtime of BOCS quadratically also with respect to the *dimensionality* of the problem. Furthermore, it is worth noting that the quadratic dependence of d_h stems from the second order *approximation* of their sparse Bayesian linear regression model. This term will become much more expensive if a higher order approximation is used, e.g. it becomes d_h^m if an m -th order approximation is used.

For the categorical problems, COMBO achieves comparable performance in terms of the function value at termination in 2 out of 3 problems in Fig. 3 and thus the analysis of computing cost against it is of our prime interest (other methods are either not competitive in terms of performance (e.g. TPE), or are *much* more expensive and/or more constrained in applicability (e.g. BOCS). The comparison against COMBO (and BOCS where applicable) is shown in Table 1, where it is evident that our method offers around 2-3 times speedup compared to COMBO, whereas BOCS is orders-of-magnitude more expensive.

For the mixed problems, we analyse the black-box attack

Table 1. Wall-clock time comparison of a single trial (mean \pm standard deviation across 20 trials for COMBO and CASMOPOLITAN) on categorical problems on a shared Intel Xeon server.

Problem	Ours	COMBO	BOCS
Pest	1130 ± 80 s	3330 ± 50 s	~ 1 d
Contamination	2350 ± 180 s	6630 ± 400 s	n.s.
MAXSAT-60	12100 ± 3000 s	34300 ± 2000 s	o.o.t.

n.s.: setup not supported

o.o.t.: run out-of-time (> 100 hours) and did not finish.

problem. On average, the time taken to attack an image (successful or not) is around 45 mins with MVRSM, which uses ReLU surrogate instead of GP, whereas TURBO and our CASMOPOLITAN run roughly $1.6\times$ and $2.0\times$ more expensive – TURBO run faster likely due to its more frequent restarts. However, in realistic setups suitable for BO where either the objective function evaluation time completely eclipses the algorithm running time (e.g. tuning of large-scale machine learning system) or where *sample efficiency*, as contrasted *wall-clock efficiency*, is otherwise more valued (e.g. the black-box attack setup discussed here), the larger cost of CASMOPOLITAN is likely justifies by its better performance. Overall, we believe that CASMOPOLITAN offers a sound balance between good performance and reasonable computing cost.

B.2. Additional Problems

Func3C The results are shown in Fig. 7. The results are broadly comparable to that of Func2C in Fig. 4(a), although in this case COCABO and vanilla-BO perform more strongly near the end.

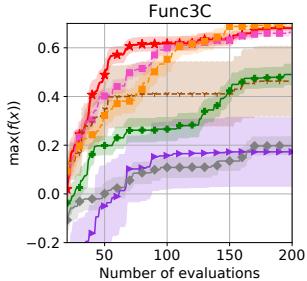


Figure 7. Results on Func3C

Noisy Contamination We conduct a further experiment on the Contamination problem but with an additional noise variance of 1×10^{-2} , and the results are shown in Fig. 8. In this case, we again see CASMOPOLITAN and COMBO outperforming the rest and CASMOPOLITAN again enjoys a faster convergence than the other methods. In this particular case, COMBO outperforms CASMOPOLITAN, albeit marginally, at the end.

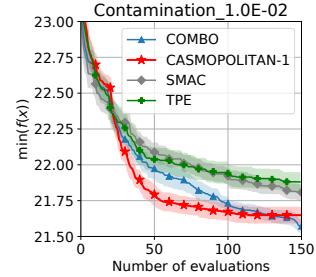


Figure 8. Results on Contamination problem with noise variance of 0.01.

Ordinal problems Sometimes we encounter ordinal problems, which are discrete variables that are similar to the categorical problems, but unlike categorical, there exists some kind of ordering between the different values that a variable can take. For example, in deep learning we often choose batch size as powers of 2, where possible batch sizes are $\{64, 128, 256, 512\}$. While current methods and popular packages (e.g. COCABO and the Bayesmark⁵ package) often treat these as ordinary categorical variables by ignoring such ordinal structure, this practice might not be optimal. In this section, we describe an exemplary adaptation of CASMOPOLITAN in the ordinal setting that recognises and leverage such relations, and conduct a preliminary experiment to validate it as a demonstration of the versatility of our approach.

In our tailored kernel for the categorical variables (Eq. (1)), we use Kronecker delta function which only has two possible outcomes: 0 if the two values are different or 1 if the two values are the same. This is appropriate in the categorical setting because there exists no ordering amongst different choices a variable may take (e.g. consider choosing from $\{\text{SGD, Adam, RMSprop}\}$: to SGD, Adam can be considered “as different as” RMSprop. However, in ordinal-structured problems such as the batch size example above, the choice of 128 is certainly “more similar” to 64 than 256. To recognise this, we modify Eq. (1), reproduced below for convenience:

$$k_h(\mathbf{h}, \mathbf{h}') = \exp\left(\frac{1}{d_n} \sum_{i=1}^{d_h} \ell_i \delta(h_i, h'_i)\right).$$

For ordinal variables, we use the *ordinal kernel* k_o by replacing the Kronecker delta function with an appropriate distance metric. One possible formulation is:

$$k_o(\mathbf{h}, \mathbf{h}') = \exp\left(\frac{1}{d_n} \sum_{i=1}^{d_h} \ell_i \left(1 - \frac{|h_i - h'_i|}{|h_i - h'_i|_{\max}}\right)\right), \quad (6)$$

where $|h_i - h'_i|$ is the distance that is dependent on the problem-specific metric and $|h_i - h'_i|_{\max}$ is the maximum possible distance (in the context of the batch size problem,

⁵<https://github.com/uber/bayesmark>

this is $512 - 64$). Note that when no ordinal structure exists, $|h_i - h'_i|$ is either 0 or $|h_i - h'_i|_{\max}$ and Eq. (6) reduces to the categorical kernel in Eq. (1).

We further include a preliminary empirical validation on the 2D discretised Branin problem introduced in Oh et al. (2019), where each dimension of the Branin function in $[-1, 1]^2$ is discretised into 51 equally spaced points – as such, the problem has 2 ordinal dimensions with 51 choices for each. Note that this is a rather extreme example due to the large number of choices relative to the number of variables, and the fact that the function landscape resembles much more to a continuous problem instead of a typical ordinal one, but we include it for the sake of illustration. We also do not use trust region for this example due to the low dimensionality and the fact that the point of this experiment is to compare categorical and ordinal kernels. We show the results in Fig. 9, where we also include the results for COMBO which is the strongest baseline shown to outperform other methods such as SMAC and TPE in Oh et al. (2019). It is worth emphasising that COMBO also explicitly accounts for the ordinal relations, so the comparison of it against CASMOPOLITAN with ordinal kernel is fair.

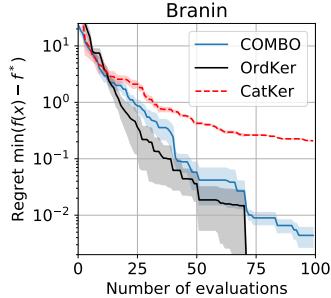


Figure 9. Results on Discretised Branin. Lines and shades denote mean and standard error across 20 trials. Note that since the optimum is known in this case ($f^* = 0.404$), in the y-axis we show the regret in log-scale.

It is clear that CASMOPOLITAN with ordinal kernel (OrdKer) outperforms both the ordinal-agnostic CASMOPOLITAN (CatKer) and the ordinal-aware COMBO in both convergence speed and final performance (OrdKer converges to f^* every single trial). To show why it is the case, we plot the GP posterior variance of CASMOPOLITAN with each kernel in Fig. 10: categorical kernel measures similarity via the Hamming distances only, and thus each observation \mathbf{h}_i only reduces posterior variance on the points sharing at least one common dimension as \mathbf{h}_i . On the other hand, ordinal kernel further accounts for the similarity amongst different values an input may take, and thus each evaluation also reduces the variance in the vicinity of \mathbf{h}_i .

While we only consider a toy problem here, the fact that k_o is a simple modification over k_h means it is trivial to scale the

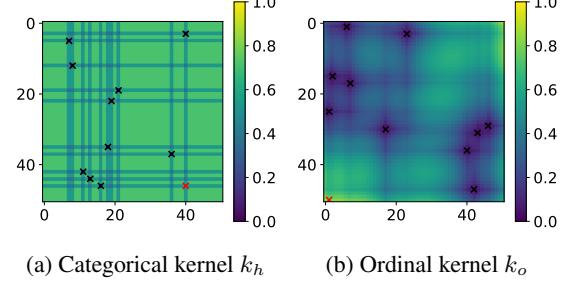


Figure 10. Posterior variance of GP with categorical and ordinal kernels after 10 random initial points on the discretised Branin problem. Black markers are the GP observations; red markers are the proposed locations for the next evaluations.

approach to high dimensions with the local TR approaches described in the main text and/or to the mixed inputs, such as ordinal-continuous or even ordinal-categorical-continuous search space. We defer a thorough investigation to this even richer class of problems to a future work, which we believe would be an exciting extension to the present work.

B.3. Parallel CASMOPOLITAN by number of objective function queries

Supplementary to Fig. 5 which shows the comparison of performances of CASMOPOLITAN of varying batch sizes by *number of batches*, here we compare the performance by *number of objective function queries* in Fig. 11. It is evident that increasing the number of batches, at least in the experiments we consider, does not lead to significant performance deterioration even though we may achieve near-linear reduction in wall-clock time if we have sufficient parallel computing resources.

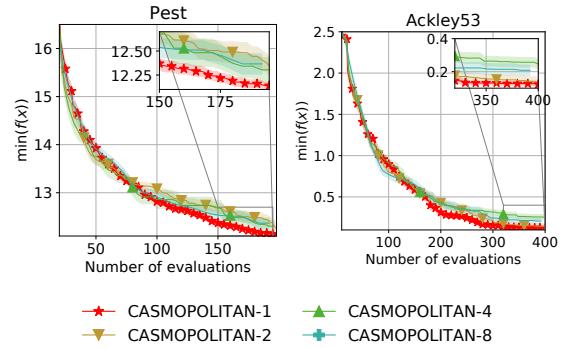


Figure 11. Parallel CASMOPOLITAN on representative categorical and mixed problems by *number of function queries*.

B.4. Additional Results on the Black-Box Attack Task

Supplementary to the main text, in Fig. 12 we show more examples of the adversarial examples generated by our method, where the diagonal images are the original, unperturbed images in the CIFAR-10 validation dataset that the CNN initially

classifies correctly while the off-diagonal entries are the adversarial examples. From the 50 images attacked by us, we select an image and we compare the objective function value against number of queries in the 9 attack instances in Fig. 13. It is clear that our method achieves higher success rate within the highly limited budget (successful in 6/9 instances, as opposed to 3 and 1 in TURBO and MVRSM), and even in cases where attack is unsuccessful within the budget, our method still increases the loss more and pushes it closer to the success boundary.

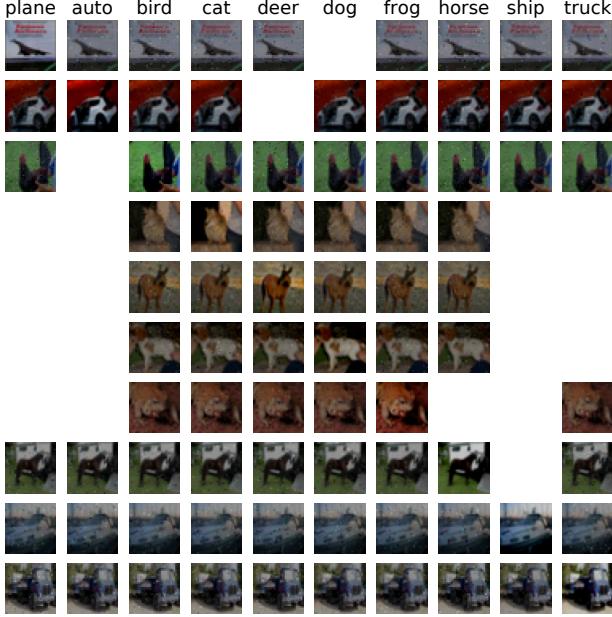


Figure 12. Some adversarial examples generated by our method.

B.5. Sensitivity Studies on the Additional Hyperparameters

Similar to TURBO, our method introduces some additional hyperparameters related to the trust regions. In this section we examine the sensitivity of the performance of CASMOPOLITAN towards these hyperparameters on Pest and Ackley53 problems. Specifically, we test the sensitivity towards:

- Initial trust region length: unlike the hyperrectangular TRs for the continuous space where L_{\min}^x and L_{\max}^x are additional hyperparameters, the Hamming distance-based TRs are constrained to be positive integers in $(0, d_h]$, relieving us from the need to tune L_{\min}^h and L_{\max}^h . Nonetheless, the trust region length at the beginning of each restarts is still a free hyperparameter.
- Failure tolerance (`fail_tol`): the number of successive failures to shrink the trust region. An aggressive `fail_tol` setting (i.e. one that is very small) could

lead to rapid trust region shrinking and possibly more frequent restarts. Note that it is generally rare to have a large number of consecutive successes in increasing the function value, and therefore we fix the success tolerance (`succ_tol`) to be 2.

- Shrinking rate of TRs (α_s): the multiplier when TR shrinking is triggered $L \leftarrow \alpha_s L$; a more aggressive value of α_s leads to more rapid shrinking and restart upon stagnation in improving $f(\mathbf{z})$. Note that α_s is always coupled with the expansion rate $\alpha_e = \frac{1}{\alpha_s}$ when TR expansion is triggered, and hence we do not further test the sensitivity to α_e . Also, we use the same expansion and shrinking rates in both continuous and categorical TRs.

We show the results in Fig. 14 where the default hyperparameter values are `fail_tol` = 40, initial trust region length 20 (for Pest Control with $d_h = 25$) or 40 (for Ackley-53 with $d_h = 50$) and $\alpha_s = 0.667$ (and thus $\alpha_e = 1.5$). In each of the experiments presented in Fig. 14, we only tune the hyperparameter in interest, and leave all others at their default values. For the mixed problems, we do not tune the hyperparameters specific to the continuous TRs (e.g. the initial, min and max continuous TR lengths) and instead leave them at their default values in the official TURBO implementation. Furthermore, due to the large number of hyperparameter configurations, we only run each configuration once. The results show that the performance of CASMOPOLITAN is generally insensitive to the hyperparameter choice, as the vast majority of the results fall within 2 standard deviations of results in the main text *running the exactly the same configurations*, suggesting that, as a whole, the impact on performance due to different hyperparameter choices might not be more significant compared to the inherent randomness in initialisation in different trials. It is further worth noting that in all cases CASMOPOLITAN still outperforms the corresponding next best baseline – this suggests that the performance difference is mainly driven by the choice of *different algorithms*, as opposed to *different hyperparameters of the same algorithm*.

B.6. Comparison against ALEBO and REMBO

In this section we run a small comparison of our method against REMBO (Wang et al., 2013) and ALEBO (Letham et al., 2020), the representative methods of the class of high-dimensional BO methods. We compare against them in the Ackley-53 problem with setups identical to the description in Sec. 4 in the main text, and we show the results in Fig. 15 where for both algorithms, we run under their respective default hyperparameter settings. We observe that while both outperform COCABO, they are outperformed by CASMOPOLITAN and TURBO by a large margin.

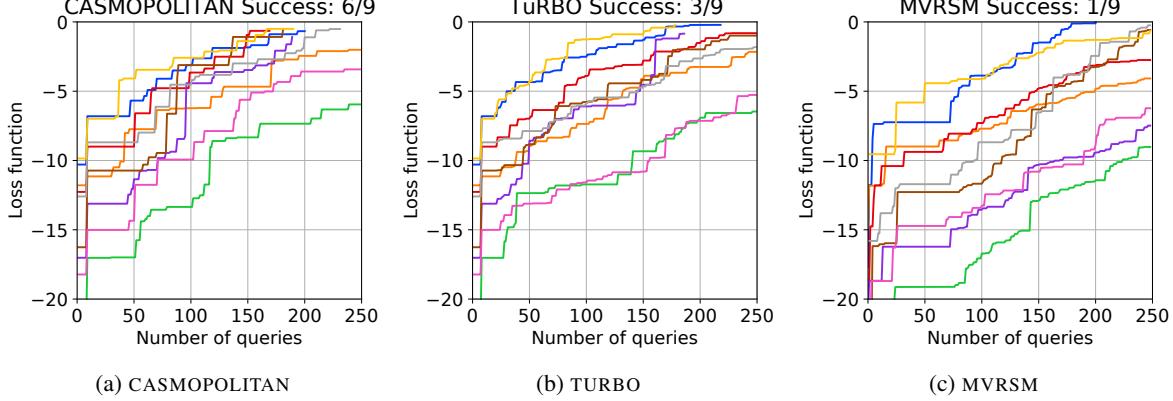


Figure 13. Attack loss function (described in Eq. (11) against the number of queries of an attacked image in CASMOPOLITAN, TURBO and MVRSM, three most competitive methods of the black-box adversarial attack problem. Lines on each image denote the targeted attack to the 9 target classes. Some lines terminate earlier than the full budget because the attack is successful before using up all the query budget.

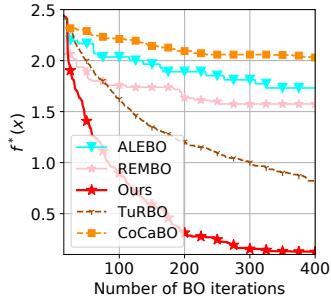


Figure 15. Comparison against REMBO and ALEBO on the Ackley-53 problem. The lines denote the mean performance across 10 random trials.

B.7. Empirical Comparison of UCB-based and Random Restarts

As discussed, the primary motivation of using UCB-based restarts of the trust regions is to theoretically driven, but in this section we investigate whether there exists any practical, finite-time benefits of using the UCB-based restarts.

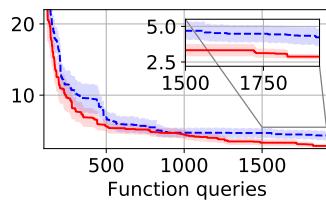


Figure 16. UCB vs random restarts in 20d Levy over 10 trials. t-test p-value between the two method is 0.048 at the final iteration. Shaded denote ± 1 standard error.

Practically, optimising the UCB on the auxiliary GP exactly is difficult. Instead, at each restart of the trust region, we simply sample a large number of points, compute their UCB

scores based on the auxiliary GP, and select the top ones as the initialising points for the next restart. We emphasise that the auxiliary GP is not meant to fit well to the objective function, as otherwise we do not need trust regions to constrain the surrogate, but is to instead generate better initialising points than random selection. Based on this described procedure, we conduct an experiment comparing UCB vs random restarts on 20-dimensional Levy function, and we show the results in Fig. 16. It can be seen that using UCB-based restarts leads to small but statistically significant improvements over the randomly initialising baseline. Furthermore, in terms of running time, since the auxiliary GPs scale with the number of *restarts* instead of number of observations, we find the UCB variant to be only 1.9% slower in terms of running time. With these results, we expect that the proposed UCB criterion to lead to practical benefits even in modestly higher-dimensional problems given an extended query budget (such that we would typically observe a number of TR restarts for the effect of initialisation at the start of each restart to be significant).

C. Implementation Details

C.1. Description of the categorical problems

A table containing the details and other characteristic details of all the test problems are shown in Table 2.

Contamination Control Contamination Control is a binary optimisation problem in food supply chain (Hu et al., 2010): at each stage, we have the choice of whether to introduce contamination control, but early use of contamination control could inevitably lead to increase in cost and as such our objective is to minimise food contamination with the smallest monetary cost (hence a minimisation problem). It is worth noting that in this problem and the Pest Control

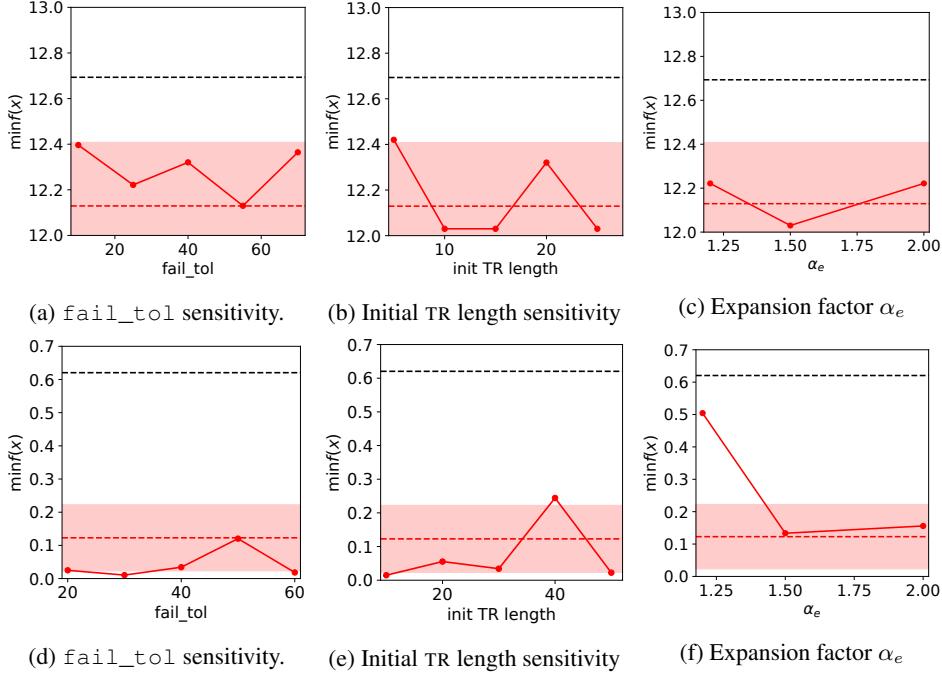


Figure 14. Sensitivity of CASMOPOLITAN performance towards various hyperparameters in Pest control (top row) and Ackley-53 (bottom row). The red lines/shades denote the mean ± 2 standard deviation of the baseline results in the main text, whereas the dotted black line is the performance of the next best-performing baseline (COMBO and MVRSM respectively).

problem described below, the actions taken by the previous stage have implications on the following stages, thus leading to highly complicated interactions amongst the different variables. In this problem, we use the implementation used in Oh et al. (2019). However, it is worth noting that while Oh et al. (2019) consider a 21-stage (with a total of $2^{21} \approx 2.1 \times 10^6$ configurations) problem, we increase the total number of stages to 25 (with a total of $2^{25} \approx 3.4 \times 10^7$ configurations). In this experiment we limit the maximum number of evaluations of 150, as the running time of BOCS quickly increases beyond our computing budget if we set the it to a significantly higher value.

Pest Control We use the problem proposed in Oh et al. (2019) which expands the contamination control problem into a multi-categorical optimisation problem: at each stage, we now need not only to determine whether to take an action (to use pesticide or not), but also the type of the pesticide (4 choices in total). This thus gives rise to 5 potential choices for each stage. Similar to Contamination control, we again increase the total number of stages to 25 (as opposed to 21 in Oh et al. (2019)) to give an expanded and more complicated search space. In Ablation Studies of Sec. 4, we also include a variant named DifficultPest, where the total number of stages is further increased to 80.

Weighted Maximum Satisfiability Maximum satisfiability problem is a classical combinatorial optimisation

problem that aims to determine the maximum number of clauses of a given Boolean formula in conjunctive normal form (CNF) that can be made true by an assignment of truth values to the variables. Similar to Oh et al. (2019), we take the same 60-variable benchmark from Maximum Satisfiability Competition 2018⁶ (`frb-frb10-6-4.wcnf` problem from <https://maxsat-evaluations.github.io/2018/benchmarks.html>)

C.2. Description of the mixed problems

Func2C and Func3C These synthetic problems were first proposed in Ru et al. (2020a). In Func2C ($d_x = 2, d_h = 2$), the value of h determines the objective function value that is a linear combination of three benchmark functions, namely Beale, Six-Hump Camel and Rosenbrook (abbreviated as `bea`, `cam` and `ros` in Table 2); the function form of these 3 functions are:

$$\begin{aligned} \text{bea}(\mathbf{x}) &= (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 + \\ &\quad (2.625 - x_1 + x_1 x_2^3)^2. \\ \text{cam}(\mathbf{x}) &= (5 - 2.1x_1^2 + \frac{x_1^4}{3})x_1^2 + x_1 x_2 + (-4 + 4x_2^2)x_2^2. \\ \text{ros}(\mathbf{x}) &= (1 - x_1)^2 + 100(x_2 - x_1^2)^2. \end{aligned} \quad (7)$$

⁶<http://sat2018.azurewebsites.net/competitions/>

Func3C ($d_x = 2, d_h = 3$) is similar but has one extra categorical dimension to enable more complicated interactions.

XG-MNIST ($d_x = 5, d_h = 3$) This is a real hyperparameter tuning task of a machine learning model (XGBoost). The tunable continuous hyperparameters are maximum depth, minimum split loss, subsample, learning rate of the optimiser and the amount of regularisation. The categorical variables are the booster type, grow policies and training objective. We use the `xgboost` python package and adopt a train-test split of 7 : 3 on the MNIST data. Note that this setup is identical to that used in Ru et al. (2020a).

Ackley-53 is a stylised version of the original 53-dimensional Ackley function, whose original form is given by:

$$f(\mathbf{z}) = -a \exp \left(-b \sqrt{\frac{1}{d} \sum_{i=1}^d z_i^2} \right) - \exp \left(\frac{1}{d} \sum_{i=1}^d \cos(cz_i) \right) + a + \exp(1), \quad (8)$$

where in this case $a = 20, b = 0.2, c = 2\pi$ and $d = 53$ and we define $\mathbf{z} \in [-1, 1]^{53}$. From this continuous form, the first 50 dimensions are modified to be *binary* variables that take the value of either 0 and 1, and the final 3 variables are continuous and limited in the range of $[-1, 1]^3$. This adaptation is first proposed in Bliek et al. (2020). This function has a known global minimiser of $\mathbf{h}^* = [0, \dots, 0]$ and $\mathbf{x}^* = [0, 0, 0]$ with $f^*(\mathbf{z}) = 0$.

Rosenbrock-200 is a stylised and scaled version of the classical Rosenbrock function. The Rosenbrock function is given by:

$$f(\mathbf{z}) = \frac{1}{50000} \left(\sum_{i=1}^{d-1} (100(z_{i+1} - z_i^2)^2 + (z_i - 1)^2) \right), \quad (9)$$

where in this case $d = 200$. The first 100 dimensions are then converted to binary variables, while the final 100 dimensions are continuous in the range of $[-2, 2]^{100}$.

Black-box adversarial attack We adapt black-box setup from Ru et al. (2020b), one of the first works that introduce BO in the image adversarial attack setting. Specifically, denoting \mathcal{M} as the target model (or the victim model, in this case a CNN image classifier) from which we may query an image input I , the BO agent can only observe the prediction scores on all C classes (for CIFAR-10, $C = 10$): $\mathcal{M}(I) : \mathbb{R}_+^d \rightarrow [0, 1]^C$ (thus a “black-box”, since gradients, architecture and other information of the classifier itself are never revealed to the attack agent). Therefore, denoting I as the original, unperturbed image that \mathcal{M} correctly gives

its prediction as c , the targeted adversarial attack objective is to find some perturbation $\delta \in \mathbb{R}^d$ to be superposed on the original image such that \mathcal{M} now mis-classify the perturbed image to another target class t . In this work, we use the identical CNN models to the previous works (Ru et al., 2020b; Tu et al., 2019; Alzantot et al., 2019), which approximately gives 80% validation accuracy on the CIFAR-10 dataset. Ru et al. (2020b) further claim that the query efficiency of the BayesOpt attack strategy can be enhanced by searching the perturbation over a latent space $\tilde{\delta} \in \mathbb{R}^{d_r}$ with reduced dimension $d_r \ll d$ and upsampling it back to the original high-resolution image space \mathbb{R}^d . This leads to a categorical variable which is the downsampling/upsampling technique, and in this work we have 3 options: bilinear, nearest and bicubic interpolations. In our attack on CIFAR10 images, we set $d = 32 \times 32 \times 3$ and $d_r = 14 \times 14 \times 3$ following Ru et al. (2020b).

In our work, we adopt a *sparse* setup where instead of perturbing all the pixels, we only perturb one pixel per row per colour in the latent space, allowing a total of $s = 14 \times 3 = 42$ pixels in the reduced space to take non-zero values. Such setup corresponds to add perturbation to some pixels of the original image only, which is more actionable in real life (for e.g., to evade real-life image classifiers this only requires one to carefully manipulate some parts of a printed image; this is contrasted to L_2 attack, another often studied setup where we perturb a small amount on *every* pixel of the image which is less feasible in real life). We additionally impose a constraint on the pixels ϵ to limit the maximum amount of perturbations. Mathematically, the goal is formulated as:

$$\begin{aligned} & \arg \max_{j \in \{1, \dots, C\}} \mathcal{M}(I + \text{Upsample}(\tilde{\delta}))_j = t. \\ & \text{s.t. } \|\{\tilde{\delta}_i \mid \tilde{\delta}_i \neq 0\}\| \leq s \text{ and } \|\tilde{\delta}\|_\infty \leq \epsilon, \end{aligned} \quad (10)$$

where the first $\|\cdot\|$ denote the cardinality of the set of non-zero elements of $\tilde{\delta}$ and the second $\|\cdot\|$ is the L_∞ norm.

In summary, the variables $\mathbf{z} \in \mathbb{R}^{85}$ that we need to search over include 42 categorical variables deciding the positions of the pixels to be perturbed at each row (thus 14 choices for each variable), 1 categorical variable on the type of upsampling technique chosen (3 choices) and 42 continuous variables defining the amount of perturbation to be added to each chosen pixel. These setups conveniently cast the problem of finding adversarial perturbation as a mixed continuous-categorical optimisation problem for which CAS-MOPOLITAN is suitable. In this case, we follow Ru et al. (2020b) and select the following as the objective function f we aim to maximise:

$$f(\mathbf{z}) = \left[\log \mathcal{M}(I + \boldsymbol{\delta}(\mathbf{z}))_t - \log \mathcal{M}(I + \boldsymbol{\delta}(\mathbf{z}))_c \right], \quad (11)$$

where $\boldsymbol{\delta}(\mathbf{z})$ is the image perturbation $\boldsymbol{\delta}$ induced by our combined choices of the pixel locations and the corresponding

amount of perturbations. Essentially, in Eq. (11), we aim to maximise the difference between the logit value of the target class t and the true class c , and trivially the attack succeeds if and when $f(\mathbf{z}) > 0$. Thus, we terminate each experiment either the attack succeeds or the maximum budget (250) is reached. It is finally worth noting that the combined dimension \mathbf{z} is 85-dimensional whose one-hot transformed dimension amounts to 633, which is clearly beyond the common scope of usage of vanilla GP-BO that neither gives special treatments to the categorical dimensions nor is tailored for high-dimensional optimisation.

C.3. Experimental setup

We run all experiments on a shared Intel Xeon server with 256GB of RAM. For all categorical problems, we run 20 random trials with the exception of BOCS on Contamination Control, where we only run 5 trials due to the very long running time of BOCS and our computing constraints (reported in App. B.1). For the mixed problems, we follow Ru et al. (2020a), where we run 20 trials for the synthetic problems and 10 trials for the real-life problems. For black-box attack, we run attack once on all 450 attack instances on 50 images. We report mean and standard error in all cases.

CASMOPOLITAN Our algorithm introduces a number of additional hyperparameters relating to the initialisation, adjustment and restarting of the trust regions. In the categorical space, the distances (Hamming distance) are always integers, and the minimum (L_{\min}^h) and maximum (L_{\max}^h) trust region sizes are always set to 0 and the dimensionality of the problem (i.e. the diameter of the combinatorial graph), respectively. The failure tolerance, which is the number of successive failures in increasing the best objective function value before shrinking the trust region size (`fail_tol`), is set to 40 unless otherwise specified; the success tolerance (`succ_tol`), which is the number of successive successes in increasing the best function value before expanding the trust region, is set to 2. We investigate the sensitivity in performance of our algorithm to these hyperparameters in App. B3. The only other GP hyperparameter is the amount of noise (or jitter), where we constrain the noise variance in the interval of $[10^{-5}, 0.1]$ and this value is learnt as a hyperparameter during the log-marginal likelihood optimisation. We always start the experiments with 20 initial randomly sampled points.

In the mixed setting that involves continuous variables, unless specified otherwise we always use the Matern 5/2 kernel. In these continuous problems, we bound the lengthscale in the range of $[0.01, 0.5]$ and outputscale in $[0.5, 5]$ and in all cases, we normalise the continuous inputs \mathbf{x} into hypercubes $[0, 1]^{d_x}$ and standardise the targets by their mean and standard deviation from the initially randomly sampled locations \mathbf{y} : $\tilde{\mathbf{y}} = \frac{\mathbf{y} - \bar{\mathbf{y}}}{\sigma(\mathbf{y})}$. When we compute the mixed kernel in

Eq. (4), we set $\lambda = 0.5$ as it is empirically shown to perform the best in Ru et al. (2020a) that initially propose this kernel. On the hyperparameters specific to the continuous trust regions, since they are identical to those introduced in TURBO (Eriksson et al., 2019), we do not change these settings from their default values ($L_{\min}^x = 0.5^7$, $L_{\max}^x = 1.6$, $L_0^x = 0.8$) with the exceptions of α_s (and hence α_e), `succ_tol` and `fail_tol` which all follow the settings of the categorical trust regions described above, instead of being independent hyperparameters. For the trade-off parameter β_i at each restart, we follow the common practice of setting β_i to a constant value (Berkenkamp et al., 2019). In our case, we set $\sqrt{\beta_i} = 1.96$ as it has been shown this value of β_i performs well for a variety of BO tasks (Berkenkamp et al., 2019).

During optimisation of the acquisition function, we use the local search strategy (for categorical optimisation; in Sec. 3.1) or the interleaved strategy (for mixed optimisation; in Sec. 3.2). In all cases, we initialise the search at the best location found so far, and we set the maximum number of local/interleaved search to be 100; for interleaved search in mixed space, one local search move + one gradient-based optimisation step count as one interleaved search step, we use Adam (Kingma and Ba, 2015) as the default optimiser for the log-marginal likelihood with learning rate 0.1 and maximum step 100, although we find the performance to be relatively stable at least for maximum step in the range of [100, 300] and learning rate in the range of [0.03, 0.3]. By default, we optimise the log-marginal likelihood 3 times and select the point(s) with the largest acquisition function across the 3 runs, although we do not find optimising with just 1 restart to be significantly worse. In this work we use expected improvement (EI) as the acquisition function, although our work is compatible with any other common choice such as GP-UCB or Thompson sampling. When CASMOPOLITAN is run in the batch setting, we use the Kriging Believer strategy (Ginsbourger et al., 2010) to select b points simultaneously: specifically, given observation data $D_t = \{\mathbf{z}_i, y_i\}_{i=0}^t$ and a GP model, we first optimise the acquisition function as usual to propose the first out of the b points required:

$$\mathbf{z}_t^{(1)} = \arg \max \alpha(\mathbf{z} \mid D_t) \quad (12)$$

We then fully trust $\mu(\mathbf{z}_t^{(1)})$, the predictive mean at $\mathbf{z}_t^{(1)}$, as a perfect proxy of the true objective function value $f(\mathbf{z}_t^{(1)})$, and use this ‘‘hallucinated’’ input-output tuple $(\mathbf{z}_t^{(1)}, \mu(\mathbf{z}_t^{(1)}))$ to update the GP $D_{t-1} \leftarrow D_{t-1} \cup (\mathbf{z}_t^{(1)}, \mu(\mathbf{z}_t^{(1)}))$. Conditioned on this GP with ‘‘hallucinated’’ inputs and observations, we then optimise the acquisition function value again to propose the second point $\mathbf{z}_t^{(2)}$ and this process is repeated until all b proposals are selected.

Other baselines Where possible and open-sourced, for the other baselines we use the implementation from their respective original authors:

- **TURBO** We use the official implementations made available by Eriksson et al. (2019) at <https://github.com/uber-research/TuRBO>. For the implementation of TURBO in categorical space or where categorical variables are involved, we use the modified implementation supplied by the organisers of the 2020 NeurIPS Black-box Optimisation Challenge which includes TURBO as a baseline that additionally supports one-hot transformation on the categorical variables (https://github.com/rdtturnerm1/bbo_challenge_starter_kit). Note that for the vanilla one-hot GP-BO, we also adapt from the TURBO implementation but remove the additional features such as trust regions and restarting.
- **COCABO** We use the official implementation by Ru et al. (2020a) at https://github.com/rubinxin/COCABO_code. Note that in original COCABO, there is an option for the value of λ in Eq. (4) to be optimised as a hyperparameter within bounds of $[0, 1]$; in our work for fairness of comparison, we fix λ to 0.5 since it is the value used in our method. It is worth noting that $\lambda = 0.5$ is also shown to be performing overall the best in COCABO from the results reported in Ru et al. (2020a).
- **MVRSM** We use the official implementation by Bliek et al. (2020) at <https://github.com/lbliek/MVRSM>.
- **COMBO** We use the official implementation by Oh et al. (2019) at <https://github.com/QUVA-Lab/COMBO>.
- **BOCS** We use the official implementation by Baptista and Poloczek (2018) at <https://github.com/baptistar/BOCS>.
- **TPE** TPE (Tree Parzans Estimator) is implemented by the Hyperopt python package, available at <http://hyperopt.github.io/hyperopt/>.
- **SMAC** We use the implementation at <https://github.com/automl/SMAC3>.

D. Proofs and Further Theoretical Analysis

D.1. Lemma D.1

Lemma D.1. *The proposed categorical kernel in Eq. (1) and mixed kernel in Eq. (4) are valid kernels (i.e. positive semi-definite kernels).*

Proof. For the categorical kernel in Eq. (1), we have that exponential of a kernel is also a kernel, and since the categorical overlap kernel is a valid kernel (Ru et al., 2020a), its

exponentiated version is also a valid kernel. For the mixed kernel in Eq. (4), since addition and multiplication between kernels result in valid kernels, and since both $k_x(., .)$ and $k_h(., .)$ are valid kernels, therefore, the mixed kernel in Eq. (4) is also a valid kernel. \square

D.2. Proof of Theorem 3.1

In this section, we derive the maximum information gain of the categorical kernel k_h (Section D.2.1) and the mixed kernel k (Section D.2.2).

D.2.1. MAXIMUM INFORMATION GAIN OF THE CATEGORICAL KERNEL

We derive the maximum information gain of the categorical kernel k_h proposed in Eq. (1) by bounding $\gamma(T; k_h; \mathcal{H})$ directly. Let us first consider the case when the objective function f has only one categorical variable h with n distinct values (i.e. $h \in \{A_1, A_2, \dots, A_n\}$ where A_i is a categorical value and $A_i \neq A_j$ when $i \neq j$). Let us consider T data points h_1, h_2, \dots, h_T , then its corresponding covariance matrix K_T is $[k_h(h_i, h_j)]_{i,j=1}^T$. As the maximum information gain $\gamma(T; k_h; \mathcal{H})$ is equal to $\log |I_T + \sigma^{-2} K_T|$ where I_T is the identity matrix of size T ⁷, thus, we will bound $\gamma(T; k_h; \mathcal{H})$ by bounding $\log |I_T + \sigma^{-2} K_T|$. Our general idea is to perform a decomposition of K_T , i.e. expressing $K_T = \Phi E \Psi^T$ where $\Phi \in \mathbb{R}^{T \times n}$, $\Psi \in \mathbb{R}^{T \times n}$, and $E \in \mathbb{R}^{n \times n}$, and then apply the Sylvester’s determinant theory and the Hadamard’s inequality to derive an upper bound for $\log |I_T + \sigma^{-2} K_T|$.

In the sequel, for ease of notation, we define the function q as a mapping from A_i to i . In particular, $q(A_i) = i$, $\forall i = 1, \dots, n$. With the categorical kernel $k_h(h, h') = \exp(l\delta(h, h'))$, in the following, we will prove that K_T can be decomposed as,⁸

$$K_T = \Phi E \Psi^T, \quad (13)$$

where $\Phi, \Psi \in \mathbb{R}^{T \times n}$, $E \in \mathbb{R}^{n \times n}$, and

$$\Phi = \begin{bmatrix} \phi(h_1) \\ \phi(h_2) \\ \vdots \\ \phi(h_T) \end{bmatrix}, \Psi = \begin{bmatrix} \psi(h_1) \\ \psi(h_2) \\ \vdots \\ \psi(h_T) \end{bmatrix},$$

$$E = \text{diag}(\exp(l) + n - 1, \exp(l) - 1, \dots, \exp(l) - 1),$$

with $\phi(h_i)$ being an n -dimensional row vector with 1 at the 1st column, (-1) at the $q(h_i)$ -th column, and 1 at the

⁷ $|S|$ denotes the determinant of matrix S .

⁸When $T = n$, this decomposition is equivalent to the eigen-decomposition. That is, the diagonal of matrix E consists of the eigenvalues of K_T and each column of Φ is an eigenvector of K_T .

$(q(h_i) + 1)$ -th column, i.e.,

$$\phi(h_i) = \begin{cases} [1 1 0 \dots 0 0], & \text{if } q(h_i) = 1 \\ [1 0 0 \dots 0 (-1) 1 0 \dots 0], & \text{if } 1 < q(h_i) < n \\ [1 0 0 \dots 0 (-1)], & \text{if } q(h_i) = n \end{cases}$$

and $\psi(h_i)$ being an n -dimensional row vector with the following formula,

$$\psi(h_i) = \begin{cases} \frac{1}{n}[1 (n-1) (n-2) \dots 1], & \text{if } q(h_i) = 1 \\ \frac{1}{n}[1 (-1) \dots -(q(h_i)-1) (n-q(h_i)) \dots 1], & \text{if } 1 < q(h_i) < n \\ \frac{1}{n}[1 (-1) \dots -(n-1)], & \text{if } q(h_i) = n. \end{cases}$$

To prove the decomposition in Eq. (13), we compute the element at the i -th row and j -th column of $\Phi E \Psi^T$, i.e. $[\Phi E \Psi^T]_{ij}$, and then prove that $[\Phi E \Psi^T]_{ij}$ is equal to $[K_T]_{ij}$. To compute $[\Phi E \Psi^T]_{ij}$, it can be directly seen that,

$$[\Phi E \Psi^T]_{ij} = \sum_{r=1}^n \phi_r(h_i) E_r \psi_r(h_j),$$

where $\phi_r(h_i)$ denotes the r -th element of $\phi(h_i)$, $\psi_r(h_j)$ denotes the r -th element of $\psi(h_j)$ and E_r denotes the r -th element on the diagonal of matrix E . We then consider the following three cases:

Case 1: $q(h_j) = q(h_i)$. First, let us consider $1 < q(h_i) < n$, then we have,

$$\begin{aligned} [\Phi E \Psi^T]_{ij} &= 1 \times (\exp(l) + n - 1) \times \frac{1}{n} \\ &\quad + (-1) \times (\exp(l) - 1) \times \frac{(-q(h_i) + 1)}{n} \\ &\quad + 1 \times (\exp(l) - 1) \times \frac{(n - q(h_i))}{n} \\ &= \exp(l). \end{aligned}$$

Note that when $q(h_j) = q(h_i)$, we will have $h_j = h_i$, thus, the element $[K_T]_{ij}$ is equal to $\exp(l)$. Similar arguments can be made when $q(h_i) = 1$ or $q(h_i) = n$, that is, $[K_T]_{ij}$ is equal to $\exp(l)$. Therefore, $[\Phi E \Psi^T]_{ij} = [K_T]_{ij} = \exp(l)$.

Case 2: $q(h_j) \geq q(h_i) + 1$. Let us first consider $1 < q(h_i)$, then,

$$\begin{aligned} [\Phi E \Psi^T]_{ij} &= 1 \times (\exp(l) + n - 1) \times \frac{1}{n} \\ &\quad + (-1) \times (\exp(l) - 1) \times \frac{(-q(h_i) + 1)}{n} \\ &\quad + 1 \times (\exp(l) - 1) \times \frac{(-q(h_i))}{n} \\ &= 1. \end{aligned}$$

In this case, with $q(h_j) \geq q(h_i) + 1$, we will have $h_i \neq h_j$, hence, the element $[K_T]_{ij}$ is equal to 1. Similar arguments can be made when $q(h_i) = 1$, that is, in this case, $[K_T]_{ij}$ is also equal to 1. Therefore, $[\Phi E \Psi^T]_{ij} = [K_T]_{ij} = 1$.

Case 3: $q(h_j) \leq q(h_i) - 1$. Let us first consider $q(h_i) < n$, then,

$$\begin{aligned} [\Phi E \Psi^T]_{ij} &= 1 \times (\exp(l) + n - 1) \times \frac{1}{n} \\ &\quad + (-1) \times (\exp(l) - 1) \times \frac{(n - q(h_i))}{n} \\ &\quad + 1 \times (\exp(l) - 1) \times \frac{(n - q(h_i) - 1)}{n} \\ &= 1. \end{aligned}$$

Similar to *Case 2*, we also have $h_i \neq h_j$. Similar arguments can be made when $q(h_i) = n$, $[K_T]_{ij}$ is also equal to 1. Hence, $[\Phi E \Psi^T]_{ij} = [K_T]_{ij} = 1$.

Combining *Cases 1, 2, 3*, we proved the decomposition in Eq. (13). Now using this decomposition, we have,

$$\gamma(T; k_h; \mathcal{H}) = \log |I_T + \sigma^{-2} K_T| = \log |I_T + \sigma^{-2} \Phi E \Psi^T|.$$

By Sylvester's determinant theorem (Sylvester, 1851),

$$\gamma(T; k_h; \mathcal{H}) = \log |I_n + \sigma^{-2} \Psi^T \Phi E|. \quad (14)$$

Next, we prove the matrix $\Psi^T \Phi E$ is a positive semi-definite (p.s.d.) matrix, and the maximum element on the diagonal of $\Psi^T \Phi E$ is equal or less than $T(\exp(l) + n - 1)$. Let us denote m_i as the number of times the categorical value A_i appears in T data points. It can be directly seen that,

$$[\Psi^T \Phi]_{ij} = \sum_{r=1}^T \psi_i(h_r) \phi_j(h_r) = \sum_{r=1}^n m_r \psi_i(A_r) \phi_j(A_r).$$

Hence, the matrix $\Psi^T \Phi$ can be written as,

$$\Psi^T \Phi = \Psi_A^T F \Phi_A, \quad \Phi_A, \Psi_A, E \in \mathbb{R}^{n \times n}, \quad (15)$$

where

$$\Phi_A = \begin{bmatrix} \phi(A_1) \\ \phi(A_2) \\ \vdots \\ \phi(A_n) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & -1 & 1 & 0 & \dots & 0 & 0 \\ 1 & 0 & -1 & 1 & \dots & 0 & 0 \\ \vdots & & & & & \ddots & \\ 1 & 0 & 0 & 0 & \dots & -1 & 1 \\ 1 & 0 & 0 & 0 & \dots & 0 & -1 \end{bmatrix},$$

$$\begin{aligned}\Psi_A &= \begin{bmatrix} \psi(A_1) \\ \psi(A_2) \\ \dots \\ \psi(A_n) \end{bmatrix}, \\ &= \frac{1}{n} \begin{bmatrix} 1 & n-1 & n-2 & \dots & 2 & 1 \\ 1 & -1 & n-2 & \dots & 2 & 1 \\ 1 & -1 & -2 & \dots & 2 & 1 \\ & & & & \dots & \\ 1 & -1 & -2 & \dots & -(n-2) & 1 \\ 1 & -1 & -2 & \dots & -(n-2) & -(n-1) \end{bmatrix}, \\ F &= \text{diag}(m_1, m_2, \dots, m_n).\end{aligned}$$

It is straightforward that $\Psi_A^T \Phi_A = I_n$, thus, from Eq. (15), we can see that $\Psi_A^T F \Phi_A$ is an eigendecomposition of $\Psi^T \Phi$, and hence, the eigenvalues of $\Psi^T \Phi$ are m_1, m_2, \dots, m_n . As $m_i \geq 0, \forall i = 1, \dots, n$, so $\Psi^T \Phi$ is a p.s.d. matrix, and therefore, $\Psi^T \Phi E$ is also a p.s.d. matrix. Besides, note that the r -th element on the diagonal of $\Psi^T \Phi E$ can be computed as $[\Psi^T \Phi]_{rr} E_r$ where $[\Psi^T \Phi]_{rr}$ and E_r are the r -th elements on the diagonal of $\Psi^T \Phi$ and E , respectively. Since $[\Psi^T \Phi]_{rr} \leq \sum_{i=1}^T (n-1)/n \times 1 \leq T$, and $E_r \leq (\exp(l) + n - 1)$, hence, $[\Psi^T \Phi]_{rr} E_r \leq T(\exp(l) + n - 1)$. This results that the maximum element on the diagonal of $\Psi^T \Phi E$ is equal or smaller than $T(\exp(l) + n - 1)$.

Combining Eq. (14) and the Hadamard's inequality (Mazya and Shaposhnikova, 1999) on the positive semi-definite matrix $\Psi^T \Phi E$, we have,

$$\gamma(T; k_h; \mathcal{H}) \leq \log |I_n + \sigma^{-2} W|,$$

where $W = \text{diag}(\text{diag}^{-1}(\Psi^T \Phi E))$. Since the maximum element on the diagonal of $\Psi^T \Phi E$ is equal or smaller than $T(\exp(l) + n - 1)$. Therefore, $\gamma(T; k_h; \mathcal{H}) = \mathcal{O}(n \log(1 + \sigma^{-2} T(\exp(l) + n - 1))) = \mathcal{O}(n \log T)$. \square

Now let consider the case when the objective function f has d_h categorical variables where each variable has n_j distinct values. This can be considered to be equivalent to the case when f has one variable with $\prod_{j=1}^{d_h} n_j$ distinct values. Thus, the same proof can be used, and we have $\gamma(T; k_h; \mathcal{H}) = \mathcal{O}((\prod_{j=1}^{d_h} n_j) \log T)$. \square

D.2.2. MAXIMUM INFORMATION GAIN OF THE MIXED KERNEL

We make use of Theorems 2 and 3 in Krause and Ong (2011) to bound the maximum information gain of the mixed kernel k . In particular, Theorem 2 states that given two kernels: k_h on \mathcal{H} and k_x on \mathcal{X} , and if k_h is a kernel on \mathcal{H} with rank at most m , then $\gamma(T; k_h k_x; [\mathcal{H}, \mathcal{X}]) \leq m\gamma(T; k_x; \mathcal{X}) + m \log T$. On the other hand, Theorem 3 states that for any two kernels k_h on \mathcal{H} and k_x on \mathcal{X} , then $\gamma(T; k_h + k_x; [\mathcal{H}, \mathcal{X}]) \leq \gamma(T; k_x; \mathcal{X}) + \gamma(T; k_h; \mathcal{X}) + 2 \log T$.

As proven in Section D.2.1, the kernel k_h has at most rank $\tilde{N} = \prod_{j=1}^{d_h} n_j$ (based on the eigendecomposition). Thus, using Theorem 2 in Krause and Ong (2011), we have

$$\gamma(T; k_h k_x; [\mathcal{H}, \mathcal{X}]) \leq \tilde{N} \gamma(T; k_x; \mathcal{X}) + \tilde{N} \log T. \quad (16)$$

Similarly, using Theorem 3 in Krause and Ong (2011), we obtain

$$\gamma(T; k_h + k_x; [\mathcal{H}, \mathcal{X}]) \leq \mathcal{O}(\gamma(T; k_x; \mathcal{X}) + (\tilde{N} + 2) \log T). \quad (17)$$

We have the mixed kernel k defined as $\lambda(k_x k_h) + (1 - \lambda)(k_h + k_x)$ where $\lambda \in [0, 1]$ is a trade-off parameter. By combining Eqs. (16) and (17), we have,

$$\begin{aligned}\gamma(T; k; [\mathcal{H}, \mathcal{X}]) &\leq \lambda \mathcal{O}(\tilde{N} \gamma(T; k_x; \mathcal{X}) + \tilde{N} \log T) \\ &\quad + (1 - \lambda)(\gamma(T; k_x; \mathcal{X}) + (\tilde{N} + 2) \log T) \\ &\leq \mathcal{O}((\tilde{N} \lambda + 1 - \lambda) \gamma(T; k_x; \mathcal{X}) \\ &\quad + (\tilde{N} + 2 - 2\lambda) \log T). \quad \square\end{aligned}$$

D.3. Proof of Theorem 3.2

We prove that under Assumptions 3.1 & 3.2, after a restart, (1) if CASMOPOLITAN terminates after a finite number of iterations, then it converges to a local maxima of f , or, (2) if CASMOPOLITAN does not terminate after a finite number of iterations, then it converges to the global maximum of f . We prove this property by contradiction.

First, let us assume after a restart, case (2) occurs, i.e. CASMOPOLITAN does not terminate after a finite number of iterations. This means when the iteration t goes to infinity, the TR length L^h is not shrunk below L_{\min}^h in the categorical setting, or, both L^h and L^x are not shrunk below L_{\min}^h and L_{\min}^x in the mixed space setting. From the algorithm description, the TR is shrunk after `fail_tol` consecutive failures. Thus, if after $N_{\min} = \text{fail_tol} \times m$ iterations where $m = \lceil \log_{\alpha_e} (L_0^h / L_{\min}^h) \rceil$ ⁹ in the categorical setting and $m = \max(\lceil \log_{\alpha_e} (L_0^h / L_{\min}^h) \rceil, \lceil \log_{\alpha_e} (L_0^x / L_{\min}^x) \rceil)$ in the mixed space setting, there is no success, CASMOPOLITAN terminates. This means, in order for case (2) to occur, CASMOPOLITAN needs to have at least one improvement per N_{\min} iterations. Let consider the series $\{f(\mathbf{z}^k)\}_{k=1}^{\infty}$ where $f(\mathbf{z}^k) = \max_{i=(k-1)N_{\min}+1, \dots, kN_{\min}} \{f(\mathbf{z}_i)\}$ and $f(\mathbf{z}_i)$ is the function value at iteration i . This series is strictly increasing and the objective function $f(\mathbf{z})$ is bounded (Assumption 3.1). Thus, using the monotone convergence theorem (Bibby, 1974), this series converges to the global maximum of the objective function f .

Second, let consider case (1) occurs, i.e. CASMOPOLITAN terminates after a finite number of iterations. We will prove that in this case, CASMOPOLITAN converges to a local maxima of $f(\mathbf{z})$ given Assumption 3.2. For simplicity,

⁹The operator $\lceil \cdot \rceil$ denotes the ceiling function

let us consider the categorical setting first. Let us denote L_s as the largest TR length that after being shrunk, the algorithm terminates. By the definition of L_s , we have $\lfloor \alpha_s L_s \rfloor \leq L_{\min}^h$.¹⁰ Due to $\lfloor \alpha_s L_s \rfloor \leq \alpha_s L_s < \lfloor \alpha_s L_s \rfloor + 1$, we have $L_s < (L_{\min}^h + 1)/\alpha_s$. And because L_s is an integer, we finally have $L_s \leq \lceil (L_{\min}^h + 1)/\alpha_s \rceil - 1$. By choosing $L_s = \lceil (L_{\min}^h + 1)/\alpha_s \rceil - 1$, we have that $\forall L > L_s, \alpha_s L \geq \alpha_s \lceil (L_{\min}^h + 1)/\alpha_s \rceil > L_{\min}^h$. This says that for all TR with length $L > L_s$, after being shrunk one time, the algorithm doesn't terminate yet. Therefore, $L_s = \lceil (L_{\min}^h + 1)/\alpha_s \rceil - 1$ is the largest TR length that after being shrunk, the algorithm terminates. This tells us when the TR length first becomes smaller or equal than L_s , CASMOPOLITAN does not terminate yet (*Conclusion 1*). In addition, since GP can fit f accurately within a TR with length L_s (*Assumption 3.2*), for any TR with length $L \leq L_s$, the solution of BO is a success (*Conclusion 2*). Combining *Conclusions 1 & 2*, we have that when TR length first becomes smaller or equal than L_s , if the current TR center is not a local maxima, CASMOPOLITAN can find a new data point whose function value larger than the function value of current TR center. Thus, in the next iteration, the TR still keeps the same length whilst having center as the new found data point. This process occurs iteratively until a local maxima is reached (i.e. when CASMOPOLITAN fails to improve from the current center), and CASMOPOLITAN terminates.

Similar arguments can be made for the mixed space setting. Let us remind that for the mixed space setting, CASMOPOLITAN terminates when either the continuous TR length $\leq L_{\min}^x$ or the categorical TR length $\leq L_{\min}^h$. Now let us consider two cases. Case (i): when the continuous TR reaches L_{\min}^x/α_s , the corresponding length of the categorical TR is $\lceil L_0^h L_{\min}^x / (\alpha_s L_0^x) \rceil$. Case (ii): when the categorical TR length reaches $\lceil (L_{\min}^h + 1)/\alpha_s \rceil - 1$, the corresponding length of the continuous TR is $L_0^x(\lceil (L_{\min}^h + 1)/\alpha_s \rceil - 1)/L_0^h$. Based on *Assumption 3.2*, GP can fit accurately a TR with continuous length $L^x \leq \max(L_{\min}^x/\alpha_s, L_0^x(\lceil (L_{\min}^h + 1)/\alpha_s \rceil - 1)/L_0^h)$ and $L^h \leq \max(\lceil (L_{\min}^h + 1)/\alpha_s \rceil - 1, \lceil L_0^h L_{\min}^x / (\alpha_s L_0^x) \rceil)$, then when Case (i) or Case (ii) occurs, the GP approximates accurately the objective function f within the corresponding TR, and thus similar argument as in the categorical setting can be made. That is, if the current TR center is not a local maxima, then CASMOPOLITAN can find a new data point whose function value larger than the function value of current TR center. And this process occurs iteratively until a local maxima is reached, and CASMOPOLITAN terminates. \square

D.4. Proof of Theorem 3.3

Let us first remind our restart strategy in the categorical setting. At the i -th restart, we first fit an auxiliary global GP model $GP(0, k_h)$ on a subset of data $D_{i-1}^* =$

$\{\mathbf{h}_j^*, f(\mathbf{h}_j^*)\}_{j=1}^{i-1}$, where \mathbf{h}_j^* is the local maxima found after the j -th restart, or, a random data point, if the found local maxima after the j -th restart is same as one of previous restart. Let us also denote $\mu_{gl}(\mathbf{h}; D_{i-1}^*)$ and $\sigma_{gl}^2(\mathbf{h}; D_{i-1}^*)$ as the posterior mean and variance of the global GP learned from D_{i-1}^* . Then, at the i -th restart, we select the following location $\mathbf{h}_i^{(0)}$ as the initial centre of the new TR:

$$\mathbf{h}_i^{(0)} = \arg \max_{\mathbf{h} \in \mathcal{H}} \mu_{gl}(\mathbf{h}; D_{i-1}^*) + \sqrt{\beta_i} \sigma_{gl}(\mathbf{h}; D_{i-1}^*),$$

where β_i is the trade-off parameter in GP-UCB ([Srinivas et al., 2010](#)).

To prove the convergence property of CASMOPOLITAN, apart from *Assumptions 3.1 & 3.2*, let us also assume that at the i -th restart, there exists a function $g_i(\mathbf{h})$ that: (a) is a sample from the global $GP(0, k_h)$, (b) shares the same global maximum \mathbf{h}^* with f , and, (c) passes through the all the local maxima of f and any data point \mathbf{h}' in $D_{i-1}^* \cup \{\mathbf{h}_i^{(0)}\}$ that are not local maxima (i.e. $g_i(\mathbf{h}') = f(\mathbf{h}') \forall \mathbf{h}' \in D_{i-1}^* \cup \{\mathbf{h}_i^{(0)}\}$). In layman's terms, the function $g_i(\mathbf{h})$ is a function that passes through all the maxima of f and is a sample from the auxiliary global $GP(0, k_h)$. It is worth noting that our assumption is more relaxed than the assumption in [Srinivas et al. \(2010\)](#) where it is assumed that the objective function f must be sampled from the global $GP(0, k_h)$. Specifically, it can be seen that if the assumption in [Srinivas et al. \(2010\)](#) holds, our assumption also holds because if $f(\mathbf{h})$ is a sample from $GP(0, k_h)$, then a choice for $g_i(\mathbf{h})$ is $f(\mathbf{h})$, thus, our assumption holds.

Using Lemmas 5.1 and 5.2 in [Srinivas et al. \(2010\)](#) for the function g_i , when $\beta_i = 2 \log(|\mathcal{H}| i^2 \pi^2 / 6\zeta)$, for all i , with probability $1 - \zeta$, we have,

$$\begin{aligned} \mu_{gl}(\mathbf{h}_i^{(0)}; D_{i-1}^*) + \sqrt{\beta_i} \sigma_{gl}(\mathbf{h}_i^{(0)}; D_{i-1}^*) \\ \geq \mu_{gl}(\mathbf{h}^*; D_{i-1}^*) + \sqrt{\beta_i} \sigma_{gl}(\mathbf{h}^*; D_{i-1}^*) \\ \geq g_i(\mathbf{h}^*). \end{aligned}$$

Thus, with probability $1 - \zeta$,

$$\begin{aligned} g_i(\mathbf{h}^*) - g_i(\mathbf{h}_i^{(0)}) \\ \leq \mu_{gl}(\mathbf{h}_i^{(0)}; D_{i-1}^*) + \sqrt{\beta_i} \sigma_{gl}(\mathbf{h}_i^{(0)}; D_{i-1}^*) - g_i(\mathbf{h}_i^{(0)}) \\ \leq 2\sqrt{\beta_i} \sigma_{gl}(\mathbf{h}_i^{(0)}; D_{i-1}^*). \end{aligned}$$

Combining this inequality with the fact that $g_i(\mathbf{h}_i^{(0)}) = f(\mathbf{h}_i^{(0)})$, and $g_i(\mathbf{h}_i^*) = f(\mathbf{h}_i^*)$, we have, with probability $1 - \zeta$,

$$f(\mathbf{h}^*) - f(\mathbf{h}_i^{(0)}) \leq 2\sqrt{\beta_i} \sigma_{gl}(\mathbf{h}_i^{(0)}; D_{i-1}^*).$$

Let us denote \mathbf{h}_i^* as the local maxima found by CASMOPOLITAN at the i -th restart. As $f(\mathbf{h}_i^{(0)}) \leq f(\mathbf{h}_i^*)$, therefore,

$$f(\mathbf{h}^*) - f(\mathbf{h}_i^*) \leq 2\sqrt{\beta_i} \sigma_{gl}(\mathbf{h}_i^{(0)}; D_{i-1}^*).$$

¹⁰The operator $\lfloor \cdot \rfloor$ denotes the floor function

This results that, with probability $1 - \zeta$,

$$R_I = \sum_{i=1}^I (f(\mathbf{h}^*) - f(\mathbf{h}_i^*)) \leq \sum_{i=1}^I 2\sqrt{\beta_i} \sigma_{gl}(\mathbf{h}_i^{(0)}; D_{i-1}^*).$$

Finally, using Lemmas 5.3 and 5.4 in Srinivas et al. (2010), we can bound R_I as $R_I \leq \sqrt{IC_1\beta_I\gamma(I; k_h, \mathcal{H})}$ with $C_1 = 8/\log(1 + \sigma^{-2})$ and $\gamma(I; k_h, \mathcal{H})$ being the maximum information gain for the categorical kernel derived in Theorem 3.1. \square

D.5. Proof of Theorem 3.4

Similar to the proof for categorical setting in Section D.4, let us first remind our restart strategy in the mixed space setting. Suppose we are restarting the i -th time, we first fit the global GP model on a subset of data $D_{i-1}^* = \{\mathbf{z}_j^*, f(\mathbf{z}_j^*)\}_{j=1}^{i-1}$, where \mathbf{z}_j^* is the local maxima found after the j -th restart, or, a random data point, if the found local maxima after the j -th restart is same as one of previous restart. Let us also denote $\mu_{gl}(\mathbf{z}; D_{i-1}^*)$ and $\sigma_{gl}^2(\mathbf{z}; D_{i-1}^*)$ as the posterior mean and variance of the global GP learned from D_{i-1}^* . Then, at the i -th restart, we select the following location $\mathbf{z}_i^{(0)}$ as the initial centre of the new TR:

$$\mathbf{z}_i^{(0)} = \arg \max_{\mathbf{z} \in [\mathcal{H}, \mathcal{X}]} \mu_{gl}(\mathbf{z}; D_{i-1}^*) + \sqrt{\beta_i} \sigma_{gl}(\mathbf{z}; D_{i-1}^*),$$

where β_i is the trade-off parameter in GP-UCB (Srinivas et al., 2010).

To prove the convergent property of CASMOPOLITAN in the mixed space setting, apart from Assumptions 3.1 & 3.2, let us also assume that at the i -th restart, there exists a function $g_i(\mathbf{z})$: (a) lies in the RKHS $\mathcal{G}_k([\mathcal{H}, \mathcal{X}])$ and $\|g_i\|_k^2 \leq B$, (b) shares the same global maximum \mathbf{z}^* with f , and, (c) passes through all the local maxima of f and any data point \mathbf{z}' in $D_{i-1}^* \cup \{\mathbf{z}_i^{(0)}\}$ which are not local maxima (i.e. $g_i(\mathbf{z}') = f(\mathbf{z}') \forall \mathbf{z}' \in D_{i-1}^* \cup \{\mathbf{z}_i^{(0)}\}$). In layman's terms, the function $g_i(\mathbf{z})$ is a function that passes through the maxima of f whilst lying in the RKHS $\mathcal{G}_k([\mathcal{H}, \mathcal{X}])$ and satisfying $\|g_i\|_k^2 \leq B$. Our assumption is more relaxed than Srinivas et al. (2010) which assumed that the objective function f lies in the RKHS $\mathcal{G}_k([\mathcal{H}, \mathcal{X}])$. Specifically, it can be seen that if the assumption in Srinivas et al. (2010) holds, our assumption also holds because if $f(\mathbf{z})$ lies in the RKHS $\mathcal{G}_k([\mathcal{H}, \mathcal{X}])$, then a choice for $g_i(\mathbf{z})$ is $f(\mathbf{z})$, thus, our assumption holds.

Using Theorem 6 in Srinivas et al. (2010) for function g_i , when $\beta_i = 2\|g_i\|_k^2 + 300\gamma_i \log(i/\zeta)^3$, $\forall i, \forall z \in [\mathcal{H}, \mathcal{X}]$, we have,

$$\Pr\{|\mu_{gl}(\mathbf{z}; D_{i-1}^*) - g_i(\mathbf{z})| \leq \sqrt{\beta_i} \sigma_{gl}(\mathbf{z}; D_{i-1}^*)\} \geq 1 - \zeta. \quad (18)$$

Note since $\|g_i\|_k^2 \leq B$, Eq. (18) is also correct using $\beta_i = 2B + 300\gamma_i \log(i/\zeta)^3$. By using the inequality in Eq. (18),

the proof technique is similar to that in Section D.4. In particular, with probability $1 - \zeta$, we have that,

$$\begin{aligned} & \mu_{gl}(\mathbf{z}_i^{(0)}; D_{i-1}^*) + \sqrt{\beta_i} \sigma_{gl}(\mathbf{z}_i^{(0)}; D_{i-1}^*) \\ & \geq \mu_{gl}(\mathbf{h}^*; D_{i-1}^*) + \sqrt{\beta_i} \sigma_{gl}(\mathbf{z}^*; D_{i-1}^*) \geq g_i(\mathbf{z}^*). \end{aligned} \quad (19)$$

Thus, with probability $1 - \zeta$, we have

$$\begin{aligned} & g_i(\mathbf{z}^*) - g_i(\mathbf{z}_i^{(0)}) \\ & \leq \mu_{gl}(\mathbf{z}_i^{(0)}; D_{i-1}^*) + \sqrt{\beta_i} \sigma_{gl}(\mathbf{z}_i^{(0)}; D_{i-1}^*) - g_i(\mathbf{z}_i^{(0)}) \\ & \leq 2\sqrt{\beta_i} \sigma_{gl}(\mathbf{z}_i^{(0)}; D_{i-1}^*). \end{aligned}$$

Since $g_i(\mathbf{z}_i^{(0)}) = f(\mathbf{z}_i^{(0)})$, and $g_i(\mathbf{z}^*) = f(\mathbf{z}^*)$, hence, $f(\mathbf{z}^*) - f(\mathbf{z}_i^{(0)}) \leq 2\sqrt{\beta_i} \sigma_{gl}(\mathbf{z}_i^{(0)}; D_{i-1}^*)$ with probability $1 - \zeta$. With \mathbf{z}_i^* as the local maxima found by CASMOPOLITAN at the i -th restart. As $f(\mathbf{z}_i^{(0)}) \leq f(\mathbf{z}_i^*)$, therefore,

$$f(\mathbf{z}^*) - f(\mathbf{z}_i^*) \leq 2\sqrt{\beta_i} \sigma_{gl}(\mathbf{z}_i^{(0)}; D_{i-1}^*).$$

This results, with probability $1 - \zeta$,

$$R_I = \sum_{i=1}^I (f(\mathbf{z}^*) - f(\mathbf{z}_i^*)) \leq \sum_{i=1}^I 2\sqrt{\beta_i} \sigma_{gl}(\mathbf{z}_i^{(0)}; D_{i-1}^*).$$

Finally, using Lemmas 5.3 and 5.4 in Srinivas et al. (2010), we can bound R_I as $R_I \leq \sqrt{IC_1\beta_I\gamma(I; k, [\mathcal{H}, \mathcal{X}])}$ with $C_1 = 8/\log(1 + \sigma^{-2})$ and $\gamma(I; k, [\mathcal{H}, \mathcal{X}])$ is the maximum information gain for the mixed kernel derived in Theorem 3.1. \square

Discussion It is worth emphasizing that the assumption of the existence of such a function $g_i(\mathbf{z})$ (at the i -th restart) that satisfies our requirements generally does not need to hold when $i \rightarrow \infty$. In fact, if this assumption needs to satisfy $\forall i$ when $i \rightarrow \infty$ then it will be same as the assumption in Srinivas et al. (2010). We will show that generally this assumption only needs to hold for a finite number of restarts. In particular, it is common that for the objective function f , there exists a local maxima $\tilde{\mathbf{z}}^*$ which is larger than all other local maxima and only smaller than the global maximum. Then as $\lim_{I \rightarrow \infty} R_I/I = 0$, there exists a finite number I_0 that the function value of the TR center at the I_0 -th restart will be larger than $f(\tilde{\mathbf{z}}^*)$, and thus the ‘local maxima’ found after the I_0 -th restart is actually the global maximum, and CASMOPOLITAN converges. Therefore, our assumption regarding the existence of $g_i(\mathbf{z})$ only needs to hold until the I_0 -th restart. This discussion is applicable for both categorical and mixed space settings.

Table 2. Configurations of the test problems.

Objective f	Type	Inputs
Contamination (Hu et al., 2010)	real, cat, min, 25-dim	Choices on whether to use control at each stage $\mathbf{h} \in \{\text{True}, \text{False}\}^{25}$
Pest (Oh et al., 2019)	real, cat, min, 25-dim	Pesticide choice at each stage (or use no pesticide) $\mathbf{h} \in \{\text{No pesticide}, 1, 2, 3, 4\}^{25}$
DifficultPest	real, cat, min, 80-dim	Pesticide choice at each stage (or use no pesticide) $\mathbf{h} \in \{\text{No pesticide}, 1, 2, 3, 4\}^{80}$
MAXSAT	real, cat, min, 60-dim	$\mathbf{h} \in \{0, 1\}^{60}$
Func2C (Ru et al., 2020a)	synthetic, mixed, max, 4-dim	$\mathbf{x} \in [-1, 1]^2$ $h_1 = \{\text{ros}(\mathbf{x}), \text{cam}(\mathbf{x}), \text{bea}(\mathbf{x})\}$ $h_2 = \{+\text{ros}(\mathbf{x}), +\text{cam}(\mathbf{x}), +\text{bea}(\mathbf{x}), +\text{bea}(\mathbf{x}), +\text{bea}(\mathbf{x})\}$
Func3C (Ru et al., 2020a)	synthetic, mixed, max, 5-dim	$\mathbf{x} \in [-1, 1]^2$ $h_1 = \{\text{ros}(\mathbf{x}), \text{cam}(\mathbf{x}), \text{bea}(\mathbf{x})\}$ $h_2 = \{+\text{ros}(\mathbf{x}), +\text{cam}(\mathbf{x}), +\text{bea}(\mathbf{x}), +\text{bea}(\mathbf{x}), +\text{bea}(\mathbf{x})\}$ $h_3 = \{+5 \times \text{ros}(\mathbf{x}), +2 \times \text{cam}(\mathbf{x}), +2 \times \text{bea}(\mathbf{x}), +3 \times \text{bea}(\mathbf{x})\}$
XG-MNIST	real, mixed, max, 8-dim	h_1 (booster type type) $\in \{\text{gbtree}, \text{dart}\}$ h_2 (grow policies) $\in \{\text{depthwise}, \text{loss}\}$ h_3 (training objective) $\in \{\text{softmax}, \text{softprob}\}$ x_1 (learning rate) $\in [0, 1]$ x_2 (max depth) $\in [1, 10]$ x_3 (minimum split loss) $\in [0, 10]$ x_4 (subsample) $\in [0.001, 1]$ x_5 (amount of regularisation) $\in [0, 5]$
Ackley-53 (Bliedt et al., 2020)	synthetic, mixed, min, 53-dim	$\mathbf{h} \in \{0, 1\}^{50}$ $\mathbf{x} \in [-1, 1]^3$
Rosen-200	synthetic, mixed, min, 200-dim	$\mathbf{h} \in \{0, 1\}^{100}$ $\mathbf{x} \in [-2, 2]^{100}$
Black-box adversarial attack	real, mixed, max, 85-dim	Choice on the location of the pixel $h_i \in \{0, 1, \dots, 13\} \forall i \in [1, 42], i \in \mathbb{Z}$ Upsampling technique $h_{43} \in \{\text{bilinear}, \text{nearest}, \text{bicubic}\}$ Amount of perturbation $\mathbf{x} \in [-1, 1]^{42}$

Note: real/synthetic: whether the problem is/simulates a real-life task or whether it is a standard benchmark function.

cat/mixed: categorical or mixed categorical-continuous problem.

max/min: maximisation or minimisation problem. We flip the sign of the objective function values where appropriate.