Joint Entropy Search for Maximally-Informed Bayesian Optimization

Carl Hvarfner
carl.hvarfner@cs.lth.se
Lund University

Frank Hutter
fh@cs.uni-freiburg.de
University of Freiburg
Bosch Center for Artificial Intelligence

Luigi Nardi
luigi.nardi@cs.lth.se
Lund University
Stanford University
DBtune

Abstract

Information-theoretic Bayesian optimization techniques have become popular for optimizing expensive-to-evaluate black-box functions due to their non-myopic qualities. Entropy Search and Predictive Entropy Search both consider the entropy over the optimum in the input space, while the recent Max-value Entropy Search considers the entropy over the optimal value in the output space. We propose Joint Entropy Search (JES), a novel information-theoretic acquisition function that considers an entirely new quantity, namely the entropy over the joint optimal probability density over both input and output space. To incorporate this information, we consider the reduction in entropy from conditioning on fantasized optimal input/output pairs. The resulting approach primarily relies on standard GP machinery and removes complex approximations typically associated with information-theoretic methods. With minimal computational overhead, JES shows superior decision-making, and yields state-of-the-art performance for information-theoretic approaches across a wide suite of tasks. As a light-weight approach with superior results, JES provides a new go-to acquisition function for Bayesian optimization.

1 Introduction

The optimization of expensive black-box functions is a prominent task, arising across a wide range of applications. *Bayesian optimization* (BO) [24, 34] is a sample-efficient approach, and has been successfully applied to various problems, including machine learning hyperparameter optimization [2, 19, 32, 36], robotics [3, 6, 22, 23], hardware design [10, 26], and tuning reinforcement learning agents like AlphaGo [7]. In BO, a probabilistic surrogate model is used for modeling the (unknown) objective. The selection policy employed by the BO algorithm is dictated by an acquisition function, which draws on the uncertainty of the surrogate to guide the selection of the next query.

The choice of acquisition function is significant for the success of the BO algorithm. A popular line of acquisition functions takes an information-theoretic angle, and considers the *expected information gain* regarding the location of the optimum that is obtained from an upcoming query. *Entropy Search* (ES) [14], *Predictive Entropy Search* (PES) [15] and the earlier work of IAGO [45] select queries by maximizing this quantity. While ES and PES are efficient in the number of queries to optimize the objective, they both require significant computational effort and complex approximations of the expected information gain, which impacts their performance and practical use [15, 46].

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

A related information-theoretic family of approaches considers the information gain on the optimal objective value [17, 30, 46]. *Max-value Entropy Search* (MES) [46] was the first information-theoretic approach to have a proven convergence rate, albeit only in a noiseless problem setting. Moreover, its consideration of a one-dimensional density over the output space as opposed to a D-dimensional input space and a reduction in intricate approximations yielded a computationally efficient alternative to the ES/PES line of approaches. Despite its empirical success, some crucial shortcomings of MES have been highlighted in recent works. Its convergence rate has been disputed [41], and crucially, it does not differentiate between the (unobserved) maximal objective value f^* and the observed noisy maximum y_{max} [25, 27, 40, 41]. As such, its assumption on the posterior distribution of the output $p(y|\mathcal{D}, \boldsymbol{x})$ does not hold in any setting where noise is present, and several follow-ups have been proposed to address the noisy problem setting [25, 27, 40, 41].

We propose an approach which merges the ES/PES and MES lines of work, and provides an all-encompassing perspective on information gain regarding optimality. We introduce Joint Entropy Search (JES), a novel acquisition function which has the following advantages over existing infomation-theoretic approaches:

1. It utilizes two sources of information, by considering the entropy over both the optimum and the noiseless optimal value;
2. It utilizes the full optimal observation, allowing it to rely primarily on exact computation through standard GP machinery instead of complex approximations; and
3. It is computationally light-weight, requiring minimal pre-computation relative to other information-theoretic approaches which consider the input space.

Simultaneously to our work, a similar approach aimed at the multi-objective setting, was proposed by Tu et al. [43]. The authors independently came up with the same JES acquisition function, with a subtly different approximation scheme to the one we present. We see our work as being complementary to theirs because we both demonstrate the effectiveness of this new acquisition function in different settings - theirs being multi-objective and batch evaluations, ours being single-objective and large levels of output noise. Our code for reproducing the experiments is available at `https://github.com/hvarfner/JointEntropySearch`.

2 Background and related work

Bayesian optimization. We consider the problem of optimizing a black-box function f across a set of feasible inputs $\mathcal{X} \subset \mathbb{R}^d$:

$$\boldsymbol{x}^* \in \arg\max_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}). \tag{1}$$

We assume that $f(\boldsymbol{x})$ is expensive to evaluate, and can potentially only be observed through a noise-corrupted estimate, y, where $y = f(\boldsymbol{x}) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$ for some noise level σ_ϵ^2. In this setting, we wish to maximize f in an efficient manner, typically while adhering to a budget which sets a cap on the number of points that can be evaluated. BO aims to globally maximize f by an initial design and thereafter sequentially choosing new points $\boldsymbol{x}_n$ for some iteration n, creating the data $\mathcal{D}_n = \mathcal{D}_{n-1} \cup \{(\boldsymbol{x}_n, y_n)\}$. After each new observation, BO constructs a probabilistic surrogate model $p(f|\mathcal{D}_n)$ and uses that surrogate to build an acquisition function $\alpha(\boldsymbol{x}, \mathcal{D}_n)$. The combination of surrogate model and acquisition function encodes the strategy for selecting the next point $\boldsymbol{x}_{n+1}$. After the full budget of N iterations is exhausted, a best configuration $\boldsymbol{x}_N^*$ is returned as either the $\arg\max$ of the observed values, or the optimum as predicted by the surrogate model.

Gaussian processes. When constructing the surrogate, the most common choice is *Gaussian processes* (GPs) [29]. Formally, a GP is an infinite collection of random variables, such that every finite subset of those variables follows a multivariate Gaussian distribution. The GP utilizes a covariance function k, which encodes a prior belief for the smoothness of f, and determines how previous observations influence prediction. Given observations $\mathcal{D}_n$ at iteration n, the posterior $p(f|\mathcal{D}_n)$ over the objective is characterized by the posterior mean m_n and variance s_n of the GP:

$$m_n(\boldsymbol{x}) = \mathbf{k}_n(\boldsymbol{x})^\top (\mathbf{K}_n + \sigma_\epsilon^2 \mathbf{I})^{-1} \mathbf{y}, \quad s_n(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}) - \mathbf{k}_n(\boldsymbol{x})^\top (\mathbf{K}_n + \sigma_\epsilon^2 \mathbf{I})^{-1} \mathbf{k}_n(\boldsymbol{x}), \tag{2}$$

where $(\mathbf{K}_n)_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $\mathbf{k}_n(\boldsymbol{x}) = [k(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, k(\boldsymbol{x}, \boldsymbol{x}_n)]^\top$ and σ_ϵ^2 is the noise variance. Alternative surrogate models include random forests [18] and Bayesian neural networks [37, 38].

Acquisition functions. The *acquisition function* acts on the surrogate model to quantify the attractiveness of a point in the search space. Acquisition functions employ a trade-off between exploration and exploitation, typically using a greedy heuristic to do so. Simple, computationally cheap heuristics are Expected Improvement (EI) [5, 20]. For a noiseless function, EI selects the next point $\boldsymbol{x}_{n+1}$ as

$$\boldsymbol{x}_{n+1} \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}\left[(y_n^* - y_{n+1}^*)^+\right] = \arg\max_{\boldsymbol{x}\in\mathcal{X}} Z s_n(\boldsymbol{x})\Phi(Z) + s_n(\boldsymbol{x})\phi(Z), \tag{3}$$

where $Z = (y_n^* - m_n(\boldsymbol{x}))/s_n(\boldsymbol{x})$. Other acquisition functions which use similar heuristics are the Upper Confidence Bound (UCB) [39], and Probability of Improvement (PI) [21]. A more sophisticated approach related to EI is Knowledge Gradient (KG) [11].

Information-theoretic acquisition functions. Information-theoretic acquisition functions [14, 15, 31, 46] and their various adaptations [1, 16, 33] seek to maximize the expected information gain I from observing a subsequent query $(\boldsymbol{x}, y)$ regarding the optimum, $\boldsymbol{x}^*$. This equates to reducing the uncertainty of the density over the optimum, $p(\boldsymbol{x}^*|\mathcal{D}) = \mathbb{P}(\boldsymbol{x} = \arg\max_{\boldsymbol{x}'\in\mathcal{X}} f(\boldsymbol{x}')|\mathcal{D})$, using the information obtained through $(\boldsymbol{x}, y)$. By quantifying uncertainty through the differential entropy H, design points are selected based on the expected reduction in this quantity over $p(\boldsymbol{x}^*|\mathcal{D})$. Formally, this is expressed as the difference between the current entropy over $p(\boldsymbol{x}^*|\mathcal{D})$, and the expected entropy of that density after observing the next query:

$$\alpha_{\texttt{ES}}(\boldsymbol{x}) = I((\boldsymbol{x}, y); \boldsymbol{x}^*|\mathcal{D}) = \mathrm{H}[p(\boldsymbol{x}^*|\mathcal{D})] - \mathbb{E}_y\left[\mathrm{H}[p(\boldsymbol{x}^*|\mathcal{D}\cup(\boldsymbol{x}, y)]\right]. \tag{4}$$

By utilizing the symmetric property of the mutual information, one can arrive at an equivalent expression, where the entropy is computed with regard to the density over the output y,

$$\alpha_{\texttt{PES}}(\boldsymbol{x}) = I(y; (\boldsymbol{x}, \boldsymbol{x}^*)|\mathcal{D}) = \mathrm{H}[p(y|\mathcal{D}, \boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}^*}\left[\mathrm{H}[p(y|\mathcal{D}, \boldsymbol{x}, \boldsymbol{x}^*)]\right]. \tag{5}$$

Eq. 4 is the original formulation used in ES [14] and Eq. 5 is the formulation introduced with PES [15]. Both formulations require a series of approximations and expensive computational steps to compute the entropy in the second term. For PES specifically, with n data points of dimension d, the second term is estimated through Monte Carlo (MC) methods by computing Cholesky decompositions of size $\mathcal{O}(n + d^2/2)^3$, and approximating the Hessian at the optimum for each MC sample.

MES [46] avoids this computational hurdle by considering the information gain $I((\boldsymbol{x}, y); y^*|\mathcal{D})$ regarding the optimal value y^*. As such, it computes the entropy reduction for a one-dimensional density:

$$\alpha_{\texttt{MES}}(\boldsymbol{x}) = I(y; (\boldsymbol{x}, y^*)|\mathcal{D}) = \mathrm{H}[p(y|\mathcal{D}, \boldsymbol{x})] - \mathbb{E}_{y^*}\left[\mathrm{H}[p(y|\mathcal{D}, \boldsymbol{x}, y^*)]\right]. \tag{6}$$

Here, it is assumed that the posterior predictive distribution $p(y|\mathcal{D}, \boldsymbol{x}, y^*)$ is a truncated Gaussian distribution, for which the entropy can be computed in closed form. However, $p(y|\mathcal{D}, \boldsymbol{x}, y^*)$ takes this form only in a strictly noiseless setting [27, 40], where it holds true that $f^* = y_{max}$, i.e. when the maximal observation and the optimal value of the objective function coincide. For noisy applications, this assumption leads to an overestimation of the entropy reduction [27].

3 Joint Entropy Search

We now present Joint Entropy Search (JES), a novel information-theoretic approach for Bayesian optimization. As for other information-theoretic acquisition functions, JES considers a mutual information quantity. However, unlike its predecessors, JES adds an additional piece of information: compared to ES/PES, it adds the density over the noiseless optimal value f^*, and compared to MES, it adds the density over $\boldsymbol{x}^*$. It utilizes a novel two-step reduction in the predictive entropy from conditioning on sampled optima and their associated values. Throughout the section, we will refer to a sampled optimum and its associated value, $(\boldsymbol{x}^*, f^*)$, as an *optimal pair*.

3.1 Joint density over the optimum and optimal value: pictorial

JES considers the joint probability density $p(\boldsymbol{x}^*, f^*)$ over both the optimum $\boldsymbol{x}^*$ and the true, noiseless optimal value f^*. Fig. 1 visualizes the densities $p(\boldsymbol{x}^*)$ and $p(f^*)$, considered by ES/PES and MES, respectively, and the joint density $p(\boldsymbol{x}^*, f^*)$, considered by JES. As highlighted by the vertical dashed lines for the point selection of each strategy (bottom), PES chooses strictly to reduce the

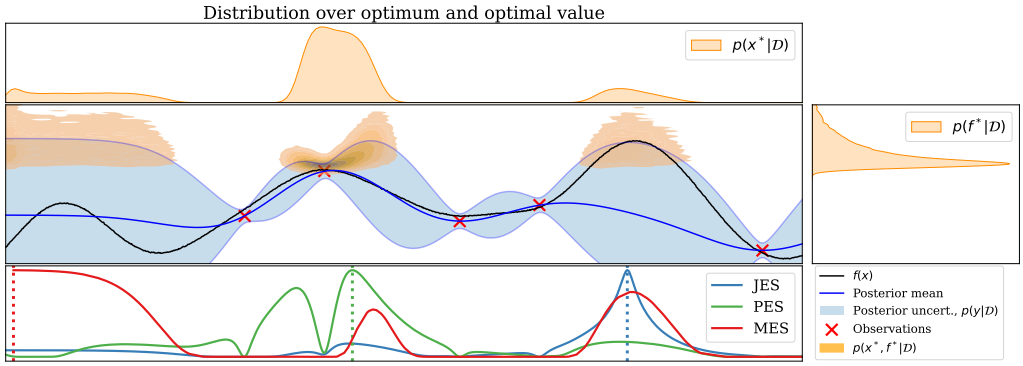

Figure 1: The densities considered by ES/PES (top), MES (right) and JES (center) on a one-dimensional toy example. The multimodal density $p(\boldsymbol{x}^*, f^*)$ is reduced to a heavy-tailed density over f^* for the density used by MES (right), which does not capture the multi-modality of the density over the optimum. The density over $\boldsymbol{x}^*$ used by PES (top) does not capture the apparent exploration/exploitation trade-off that exists between the modes. The acquisition functions and their next point selections are highlighted with dashed lines (bottom).

uncertainty over $\boldsymbol{x}^*$, and as such, considers a region where the uncertainty over the optimal value is low. However, it can effectively determine that the right side of the local optimum is more promising to query next. MES seeks to reduce the tail of the probability density over f^* (right), which in this case leads to an exploratory query. JES' joint probability density over optimum and optimal value captures uncertainties over both "where" and "how large" the optimum will be. As such, it selects a point which is uncertain under both measures. As such, JES will learn about likely locations for the optimum, while simultaneously learning probable lower and upper bounds for the optimal value, which by itself yields an effective query strategy [46] and provides valuable knowledge for future queries. For the selected query in Fig. 1, JES will learn substantially about both $\boldsymbol{x}^*$ and f^* by querying it, whereas PES and MES learn only about one of them.

3.2 The Joint Entropy Search acquisition function

We consider the mutual information between the random variables $(\boldsymbol{x}^*, f^*)$ and a future query $(\boldsymbol{x}, y)$:

$$\alpha_{\texttt{JES}}(\boldsymbol{x}) = I((\boldsymbol{x}, y); (\boldsymbol{x}^*, f^*)|\mathcal{D}_n) \tag{7}$$
$$= \mathrm{H}[p(y|\mathcal{D}, \boldsymbol{x})] - \mathbb{E}_{(\boldsymbol{x}^*, f^*)}\left[\mathrm{H}[p(y|\mathcal{D}, \boldsymbol{x}, \boldsymbol{x}^*, f^*)]\right] \tag{8}$$
$$= \mathrm{H}[p(y|\mathcal{D}, \boldsymbol{x})] - \mathbb{E}_{(\boldsymbol{x}^*, f^*)}\left[\mathrm{H}[p(y|\mathcal{D} \cup (\boldsymbol{x}^*, f^*), \boldsymbol{x}, f^*)]\right]. \tag{9}$$

Eq. 8 is similar to Eq. 6 but with the addition of $\boldsymbol{x}^*$ and the replacement of y^* with f^* in the conditioning of the second term. The expectation is computed with respect to a $D+1$-dimensional joint probability density over $\boldsymbol{x}^*$ and f^*. In Eq. 9, we make it explicit that the conditional density inside the expectation is obtained after 1. conditioning the GP on the previous data $\mathcal{D}$, plus one additional noiseless optimal pair $(\boldsymbol{x}^*, f^*)$, and 2. knowing that the noiseless optimal value is in fact f^*. By utilizing the complete observation $(\boldsymbol{x}^*, f^*)$, we can treat it like any (noiseless) observation. As such, we quantify much of the entropy reduction by utilizing standard GP conditioning functionality. For 2., we cannot globally condition on $f(\boldsymbol{x}') \leq f^*, \forall \boldsymbol{x}'$. As such, we follow previous work [25, 27, 41, 46] and enforce the condition *locally* at the current query $\boldsymbol{x}$. The resulting effect is to truncate the GP's posterior over f locally at $\boldsymbol{x}$, upper bounding it to f^*. Notably, utilizing the fantasized observation $(\boldsymbol{x}^*, f^*)$ guarantees that the conditioned optimal value f^* in JES is actually obtained, rather than serving as a possibly unattained upper bound, which is typical in the MES family of acquisition functions. The expectation in Eq. 9 is approximated through MC by sampling L optimal pairs $\{(\boldsymbol{x}_\ell^*, f_\ell^*)\}_{\ell=1}^L$ from $p(\boldsymbol{x}^*, f^*)$ using an approximate version of *Thompson Sampling* (TS) [42], as explained in Sec. 3.3. In Fig. 2, the resulting posterior distribution of the two-step conditioning is shown in greater detail. As pointed out in [27, 40], after conditioning on f^*, the posterior predictive density over y is a sum of a truncated Gaussian distribution over f and the Gaussian noise ϵ. The entropy reduction from the two-step conditioning yields two separate variance reduction steps over $p(y|\mathcal{D}, \boldsymbol{x})$: a conditioning term and a truncation term. The former is computed exactly, while the latter, generally smaller term, requires approximation, as shown in Sec. 3.4.

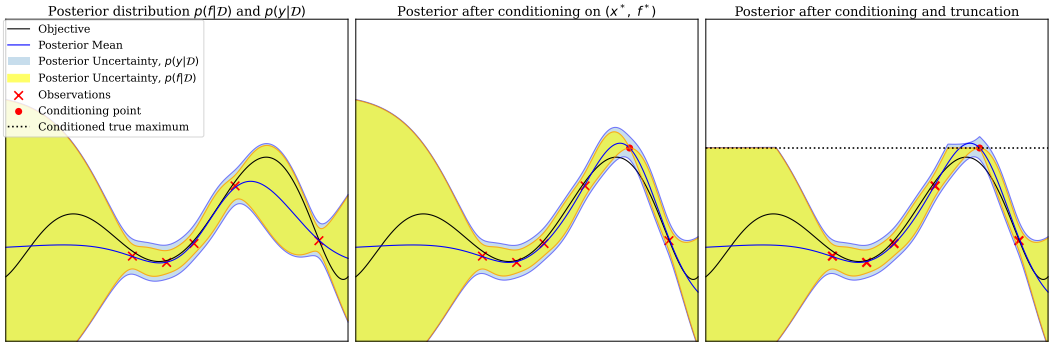

Figure 2: Step-by-step modeling when conditioning on one optimal pair $(\boldsymbol{x}^*, f^*)$. The posterior with noise $p(y|\mathcal{D})$ and without noise $p(f|\mathcal{D})$ are illustrated in blue and yellow, respectively. The GP after 5 (noisy) observations, before conditioning on $(\boldsymbol{x}^*, f^*)$ is shown on the left. In the middle panel, we draw $(\boldsymbol{x}^*, f^*)$ and condition on it, making $p(f|\mathcal{D} \cup (\boldsymbol{x}^*, f^*))$ a delta distribution at the conditioning point as the fantasized observation f^* is noiseless. Since f^* is also the presumed noiseless maximum, we truncate its posterior $p(f|\mathcal{D} \cup (\boldsymbol{x}^*, f^*), f^*)$ globally in the right panel. The observation noise allows for non-zero density on $p(y > f^*|\mathcal{D} \cup (\boldsymbol{x}^*, f^*), f^*)$. We note that, while the noise is homoscedastic, its relative contribution to the total variance differs over the input space. As such, and since we're plotting standard deviations (not variances), the blue region is wider near observed data, where $p(f|\mathcal{D})$ has lower variance.

Fig. 3 shows the difference in log variance over $p(y|\mathcal{D}, \boldsymbol{x})$ resulting from conditioning (in blue) and truncation (in orange) for the scenario in Fig. 2. The overall reduction is largest close to the point of conditioning, and the truncation term mainly contributes at uncertain regions far away from the conditioned point. Moreover, the magnitude of the conditioning term will rely on the prior variance at the conditioned point, as a larger prior variance will lead to a larger reduction in entropy from conditioning. As we average over optimal pairs, many such entropy reduction terms accumulate.

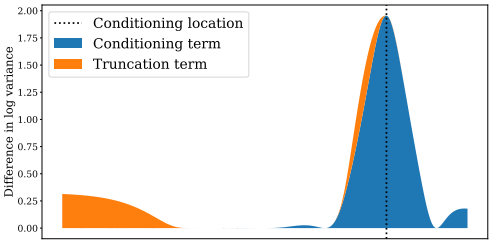

Figure 3: Reduction in log variance from the conditioning step and the truncation step as visualized in Fig. 2. The local conditioning term (blue), and the globally variance-reducing truncation term (orange).

3.3 Incorporating optimal pairs

To obtain samples $(\boldsymbol{x}^*, f^*)$, we utilize an approximate variant of TS [42], originally proposed in PES [15]. We utilize Bochner's theorem [4], which, for any stationary kernel k, asserts the existence of its Fourier dual $s(\boldsymbol{w})$. By normalizing $s(\boldsymbol{w})$, we obtain the spectral density $p(\boldsymbol{w}) = s(\boldsymbol{w})/\alpha$, where α is a normalization constant. We can then write the kernel as an expectation,

$$k(\boldsymbol{x}, \boldsymbol{x}') = \alpha \mathbb{E}_{\boldsymbol{w}}[e^{i\boldsymbol{w}^\intercal(\boldsymbol{x}-\boldsymbol{x}')}] = 2\alpha \mathbb{E}_{\boldsymbol{w},\boldsymbol{b}}[\cos(\boldsymbol{w}^\intercal \boldsymbol{x} + \boldsymbol{b}) \cos(\boldsymbol{w}^\intercal \boldsymbol{x}' + \boldsymbol{b})], \tag{10}$$

where $\boldsymbol{b} \sim \mathcal{U}(\boldsymbol{0}, 2\pi \boldsymbol{I})$. Following Rahimi and Recht [28], we sample $\boldsymbol{b}$ and $\boldsymbol{w}$ to obtain an unbiased estimate of the kernel k. From this approximation, approximate sample paths can be drawn as a weighted sum of basis functions. This form allows for fast and dense querying of the sample paths – the $\arg\max$ and $\max$ of which is an approximate draw from $p(\boldsymbol{x}^*, f^*)$. In PES, each sample $\boldsymbol{x}^*_\ell$ along with its inverted Hessian is required for computing the acquisition function. To obtain the Hessian, each sample needs to be thoroughly optimized through gradient-based optimization. JES on the other hand, only requires $(\boldsymbol{x}^*, f^*)$. As such, it can rely on cheap, approximate optimization of these samples, e.g., by densely querying sample points on a non-uniform grid.

After obtaining a set of optimal pairs $\{(\boldsymbol{x}^*_\ell, y^*_\ell)\}_{\ell=1}^L$, JES computes the conditional entropy quantity over the output y. Concretely, we generate L GPs, each modeling a posterior density $\{p(y|\mathcal{D} \cup (\boldsymbol{x}^*_\ell, f^*_\ell), \boldsymbol{x})\}_{\ell=1}^L$ conditioned on an optimal pair and previously observed data $\mathcal{D}$. Since each optimal pair is drawn from the current GP hyperparameter set, we know that the current hyperparameter set is the correct one even after adding the optimal pair to the data. By consequence, JES can compute the updated inverse Gram matrix, $(K + \sigma_\epsilon^2)^{-1}$, through a rank-1 update, instead of solving a linear system

of equations. Utilizing the Sherman–Morrison formula [35], we obtain updated Gram matrices in $\mathcal{O}(n^2)$ for each sample, as opposed to $\mathcal{O}(n^3)$ for solving the linear system of equations.

3.4 Approximating the truncated entropy

As highlighted in the right panel of Fig. 2, conditioning on f^* yields a truncated normal distribution $p(f|\mathcal{D} \cup (\boldsymbol{x}^*, f^*), \boldsymbol{x}, f^*)$ after having locally enforced the inequality $f(\boldsymbol{x}) \leq f^*$. The entropy, however, is computed with regard to the density over noisy observations, $y = f + \epsilon$, which follows an Extended Skew distribution [27] and as such, does not have tractable entropy. We approximate this quantity through moment matching [25] of the truncated Gaussian distribution over f, which yields a valid lower bound on the information gain [25]. Consequently, we obtain two Gaussian densities $\hat{p}(f|\mathcal{D} \cup (\boldsymbol{x}^*, f^*), \boldsymbol{x}, f^*) \sim \mathcal{N}(m_{f|f^*}, \sigma^2_{f|f^*})$ and $p(\epsilon) \sim \mathcal{N}(0, \sigma^2_\epsilon)$, where $m_{f|f^*}$ and $\sigma^2_{f|f^*}$ are the mean and variance of the truncated Gaussian posterior $p(f|\mathcal{D} \cup (\boldsymbol{x}^*, f^*), \boldsymbol{x}, f^*)$. Due to independence between f and ϵ and the linearity of Gaussian distributions, we can then compute the entropy of the approximate density $\hat{p}_y$ exactly as $\mathrm{H}[\hat{p}(y|\mathcal{D} \cup (\boldsymbol{x}^*, f^*), \boldsymbol{x}, f^*)] = \log(2\pi(\sigma^2_\epsilon + \sigma^2_{f|f^*}))$. Moreover, the variance of the truncated Gaussian $\sigma^2_{f|f^*}$ is computed as

$$\sigma^2_{f|f^*}(\boldsymbol{x}; \mathcal{D} \cup (\boldsymbol{x}^*_\ell, f^*_\ell)) = \sigma^2_T(f^*; m^\ell_n(\boldsymbol{x}), s^\ell_n(\boldsymbol{x})) \tag{11}$$

where $\sigma^2_T(\alpha; \mu, \sigma^2)$ is the variance of an upper truncated Gaussian distribution with parameters (μ, σ^2), truncated at α, and $m^\ell_n(\boldsymbol{x})$ and $s^\ell_n(\boldsymbol{x})$ are the mean and covariance functions of the GP which is conditioned on the optimal pair $(\boldsymbol{x}^*_\ell, f^*_\ell)$. The quality of the moment matching approximation is studied in greater detail in Appendix E.

3.5 Exploitative selection to guard against model misspecification

As with all information-theoretic approaches, JES aims to reduce the uncertainty over the location of the optimum. With this strategy, the incentive to query the perceived optimum is often lower than for heuristic approaches, such as EI. In cases where the surrogate model is misspecified, information-theoretic approaches risk reducing the entropy based on a faulty belief of the optimum, which can drastically impact their performance. As a remedy, we utilize γ-exploit approach: with probability γ, JES will query the $\arg\max$ of the posterior mean to confirm its belief of the location of the optimum. If the model is misspecified, these exploitative steps enable the algorithm to reconsider its beliefs, rather than continuing to act based on faulty ones. In Appendix C.2, we show how this approach can substantially improve performance in cases of surrogate model misspecification, while having negligible impact on performance in the worst case.

3.6 Putting it all together: The JES algorithm

For a sampled set of size L, containing optimal pairs $\{(\boldsymbol{x}^*_\ell, y^*_\ell)\}_{\ell=1}^L$ and GPs with mean and covariance functions $\{m^\ell_n(\boldsymbol{x}), s^\ell_n(\boldsymbol{x})\}_{\ell=1}^L$, the expression for the JES acquisition function is

$$\alpha(\boldsymbol{x})_{\mathtt{JES}} = \mathrm{H}[p(y|\mathcal{D}, \boldsymbol{x})] - \mathbb{E}_{(\boldsymbol{x}^*, f^*)}\left[\mathrm{H}[p(y|\mathcal{D} \cup (\boldsymbol{x}^*, f^*), \boldsymbol{x}, f^*)]\right] \tag{12}$$

$$\approx \log(2\pi(s_n(\boldsymbol{x}) + \sigma^2_\epsilon)) - \frac{1}{L}\sum_{\ell=1}^{L} \log(2\pi(\sigma^2_\epsilon + \sigma^2_{f|f^*}(\boldsymbol{x}; \mathcal{D} \cup (\boldsymbol{x}^*_\ell, f^*_\ell))), \tag{13}$$

The first term in 13 is simply the entropy of a Gaussian that can be computed in closed form. The second term contains both the conditioning term, which is exact, and the truncation, which is approximated as described in Sec. 3.4. Algorithm 1 outlines pseudocode for JES in its entirety.

4 Experimental evaluation

Benchmarks. We now evaluate JES on a suite of diverse tasks. We consider three different types of benchmarks: samples drawn from a GP prior, commonly used synthetic test functions [15], and a collection of classification tasks on tabular data using an MLP, provided through HPOBench [9]. For the GP prior tasks, the hyperparameters are known for all methods to evaluate the effect of the acquisition function in isolation. Consequently, we do not use the γ-exploit approach from Sec. 3.5

Algorithm 1 JES Algorithm

Input: Black-box function f, input space $\mathcal{X}$, size M of the initial design, max number of optimization iterations N, number of posterior MC samples L, fraction of greedy samples γ.
Output: Optimized design $\boldsymbol{x}^*$.
$\mathcal{D}_M \leftarrow \{(\boldsymbol{x}_i, y_i)\}_{i=1}^M$ ▷ initial design
for $\{n = M+1, \ldots, M+N\}$ **do**
  $m(\boldsymbol{x}), s^2(\boldsymbol{x}) \leftarrow \text{FitGP}(\mathcal{D}_{n-1})$
  if $\text{Rand}(0,1) < \gamma$ **then** $\boldsymbol{x}_n \leftarrow \arg\max_{\boldsymbol{x}\in\mathcal{X}} m_n(\boldsymbol{x})$ ▷ as described in Sec. 3.5
  else
    for $\{\ell = 1, \ldots, L\}$ **do**
      $(\boldsymbol{x}_\ell^*, y_\ell^*) \leftarrow \texttt{TS}(f)$ ▷ as described in Sec. 3.3
      $p(y|\mathcal{D}_{n-1} \cup (\boldsymbol{x}_\ell^*, f_\ell^*), \boldsymbol{x}) \leftarrow \text{UpdateGP}(\boldsymbol{x}_\ell^*, f_\ell^*)$ ▷ as described in Sec. 3.3
    end for
    $\boldsymbol{x}_n = \arg\max_{\mathcal{X}} \alpha_{\texttt{JES}}(\boldsymbol{x})$ ▷ defined in Eq. 12
  end if
  $y_n = f(\boldsymbol{x}_n) + \epsilon, \quad \mathcal{D}_n \leftarrow \mathcal{D}_{n-1} \cup \{(\boldsymbol{x}_n, y_n)\}$ ▷ observe next query
end for
return $\boldsymbol{x}^* \leftarrow \arg\max_{\boldsymbol{x}\in\mathcal{X}} m_n(\boldsymbol{x})$

in this case (i.e., we set $\gamma = 0$ in Algorithm 1). For the synthetic and MLP tasks, we marginalize over the GP hyperparameters, and set $\gamma = 0.1$. The hyperparameters of the GP prior experiments can be found in Appendix B, and ablation studies on γ in Appendix C.

Evaluation criteria. We use two types of evaluation criteria as in [46]: *simple regret* and *inference regret*. The simple regret $r_n = \max_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x}) - \max_{t\in[1,n]} f(\boldsymbol{x}_t)$ measures the value of the best queried point so far. After a query, we may infer an $\arg\max$ of the function, which is chosen as $\boldsymbol{x}_n^* = \arg\max_{\boldsymbol{x}\in\mathcal{X}} m_n(\boldsymbol{x})$ [14, 15, 46]. We denote the inference regret as $r_n = \max_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x}) - f(\boldsymbol{x}_n^*)$. Since information-theoretic approaches do not necessarily seek to query the optimum, but only to know its location, inference regret characterizes how satisfying our belief of the $\arg\max$ is. Notably, this metric is non-monotonic, meaning that the best guess can worsen with time. We use this metric in the ideal model benchmarking setting, when we sample tasks from a GP with known hyperparameters. We use simple regret for the synthetic test functions, as it constitutes a metric that is more robust to surrogate model misspecification. Inference regret for these tasks can be found in Appendix F. For the HPOBench tasks, inference regret is unobtainable.

The experimental setup. We compare against other state-of-the-art information-theoretic approaches: PES [15] and MES [46], as well as EI [20]. The acquisition functions are all run in the same framework written in MATLAB, created for the original PES implementation by Hernández-Lobato et al. [15]. All synthetic experiments were run for $50D$ iterations. In the main paper, we fix the number of MC samples for MES, PES and JES to 100 each. In Appendix D we assess the sensitivity of JES to this number and quantify the computational expense. In Appendix B, we provide all details on our experimental setup, including the runtime tests.

4.1 GP prior samples

We consider samples from a GP prior for four different dimensionalities: 2D, 4D, 6D, and 12D, with a noise standard deviation of 0.1 for a range of outputs spanning roughly $[-10, 10]$. These tasks constitute an optimal setting for each algorithm, as the surrogate perfectly models the task at hand. In Fig. 5, JES demonstrates empirically the value of the additional source of information, substantially outperforming PES and MES on all tasks.

Fig. 4 compares JES (top left) against PES, MES and EI in terms of point selection for one repetition on a two-dimensional sample task, where all runs are initialized with $D+1$ identical random samples. We observe that JES succeeds in finding all attractive regions of the search space, and queries the region around the optimum densely, which is sensible in a noisy setting. We further notice that EI (bottom right) fails to query the two circled local optima. PES (bottom left) also ignores two local optima to various degrees, and tends to circle the (perceived) optimum densely, which is expensive in terms of number of evaluations. We believe this showcases a shortcoming of only

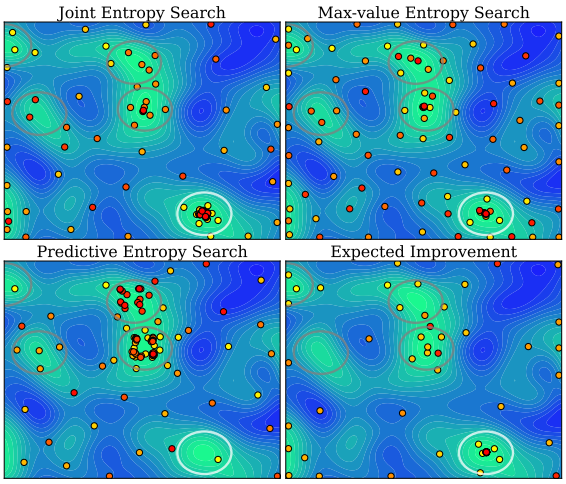

Figure 4: Comparison of queries for JES (top left), MES (top right), PES (bottom left) and EI (bottom right) on a sample of a 2D GP after a 100 function evaluations. The global optimum is circled in white, and four local optima in grey. Earlier queries are colored yellow, and later queries red.

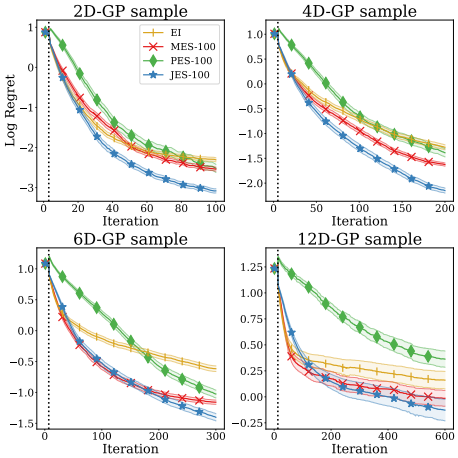

Figure 5: Comparison of JES, MES, PES and EI on GP prior samples. We run 1000 repetitions each for 2, 4 and 6D, and 250 on 12D. Mean and 2 standard errors of log regret are displayed for each acquisition function. The vertical dashed line shows the end of the initial design phase.

considering the density over the optimum: PES circles the optimum, but does not query its value. Lastly, MES (top right) successfully queries all attractive regions of the space, but also samples regions that are evidently poor the most densely out of the four approaches, despite information given by earlier (brighter) samples. Since JES considers the information conveyed by both MES and PES, it successfully excludes the apparent suboptimal regions of the space, finds all relevant optima, and queries these optima in a desirable manner.

We additionally evaluate the performance of all approaches on GP sample tasks that have a substantial amount of noise - its standard deviation roughly accounting for 10% of the total output range. We run these tasks with the GP hyperparameters fixed a priori for a larger number of iterations, 125D, to display the stagnation of some approaches. While MES and PES slow down approximately at the halfway point for both tasks, JES steadily improves for the entire length of the run. This robustness to large noise magnitudes highlights the importance of intrinsically handling noisy objectives in JES.

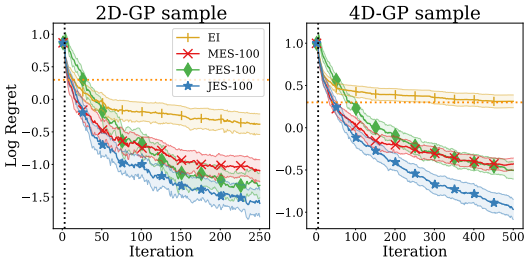

Figure 6: Evaluation of JES, MES, PES and EI on noisy ($\sigma_\epsilon^2 = 4$, orange) GP sample tasks across 100 repetitions. Mean and 2 standard errors of log regret are displayed for each acquisition function.

Task	JES-100	MES-100	PES-100	EI
2D	1.40 ± 0.32	1.03 ± 0.19	17.39 ± 4.95	0.23 ± 0.13
4D	1.50 ± 0.37	1.21 ± 0.3	34.53 ± 8.3	0.3 ± 0.17
6D	1.56 ± 0.39	1.26 ± 0.37	62.92 ± 13.54	0.35 ± 0.2

Table 1: Runtime of JES, MES, PES and EI on GP sample tasks of varying dimensionalities. JES is only marginally slower than MES, and orders of magnitude faster than PES.

In Table 1, we display the runtime of each acquisition function on these tasks when marginalizing over 10 sets hyperparameters, and sampling 10 optima per set. We time each iteration from after hyperparameters have been sampled, up until (but excluding) the query of the black-box function. Thus, acquisition function pre-computation and optimization are included. The runtime of JES is only marginally slower than that of MES with Gumbel sampling, while being at least an order of magnitude faster than PES for all displayed dimensionalities.

4.2 Synthetic test functions

Next, in Fig. 7, we study the performance of JES on three optimization test functions: Branin (2D), Hartmann (3D) and Hartmann (6D). For these tasks, we follow convention [15, 30] and marginalize over GP hyperparameters. On Branin, JES starts out slightly slower than MES but reaches the same performance in 100 iterations; and on the two Hartmann functions, JES performs amongst the best in the beginning and clearly best in the end. We note that PES experienced numerical issues on Branin, and as such, we acknowledge that its performance should be better than what is reported.

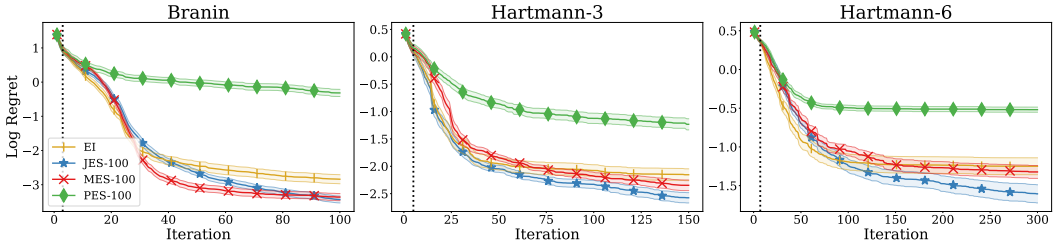

Figure 7: Comparison of JES, MES, PES and EI on Branin and Hartmann-6, $\sigma_n^2 = 0.10$. Mean and 2 standard errors of log regret are displayed for each acquisition function across 100 repetitions. The vertical dashed line represents the end of the initial design phase.

4.3 MLP tasks

Lastly, we evaluate the performance of JES on the tuning an MLP model's 4 hyperparameters for 20D iterations on three datasets. These tasks are part of the OpenML[1] library of tasks, and the HPO benchmark is provided through the HPOBench [9] suite. The tasks, which are classification problems with tabular data, regard classifying: (1) vehicles into one of four different categories, (2) blood donors for a blood transfusion service center, and (3) anonymous credit card applications. We measure the best observed classification accuracy. Notably, these tasks have a large amount of noise, which causes the performance to fluctuate substantially between repetitions. We observe that JES performs almost identically to MES and EI on vehicle silhouette and credit application, and substantially better on blood transfusion. PES lags behind significantly on all tasks.

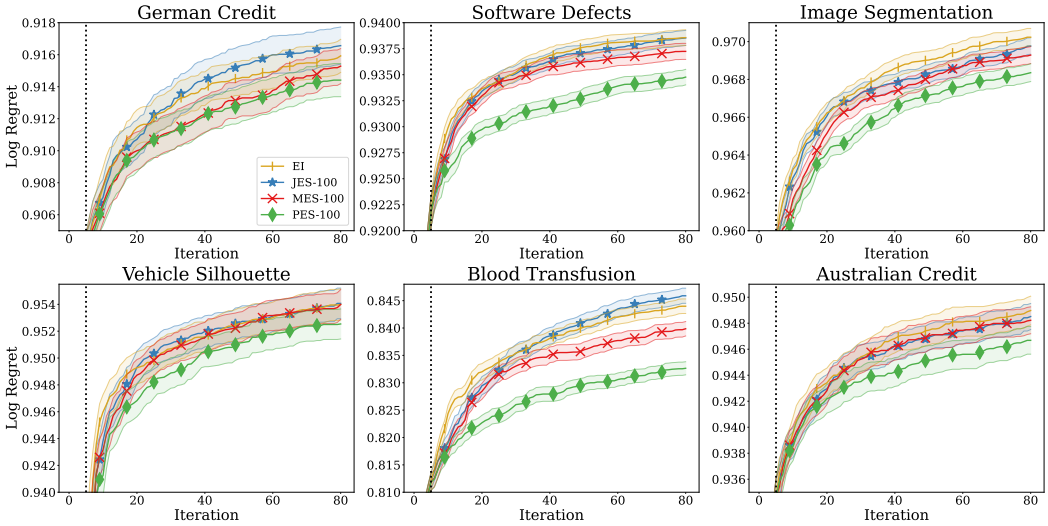

Figure 8: Comparison of JES, MES, PES and EI on six different MLP tuning tasks from the HPOBench suite. Mean and 1 standard error of best observed accuracy are displayed for each acquisition function across 100 repetitions. The vertical dashed line represents the end of the initial design phase.

[1] https://www.openml.org/

5 Conclusions

We have presented Joint Entropy Search, an information-theoretic acquisition function that considers an entirely new quantity, namely the joint density over the optimum and optimal value. By utilizing the entropy reduction from fantasized optimal observations, JES obtains a simple form for the entropy reduction regarding the joint distribution. As such, the additional information considered comes with minimal computational overhead, avoids restrictive assumptions on the objective, and yields state-of-the-art performance along with superior decision-making. We believe JES to be a new go-to acquisition function for BO, and to establish a new standard for subsequent information-theoretic techniques.

6 Limitations and Future Work

The main contribution of this paper is to provide a novel information-theoretic acquisition function which, given a sufficiently accurate model, yields impressive results. However, the non-myopic, speculative nature of information-theoretic approaches lend them to be susceptible to model misspecification, such as a poor choice of GP kernel or GP hyperparameters. In our view, information-theoretic approaches are possibly more susceptible to this issue than their myopic counterparts (EI, UCB, TS). While we propose a remedy to stabilize and improve the acquisition function under model misspecification with the γ-exploit approach, this technique only serves to *discover* misspecification and adjust accordingly, but not to inherently fix the misspecification. We believe misspecification can only be remedied by altering the surrogate model. It is thus very promising to combine advanced modelling techniques with information-theoretic acquisition functions, as already done with the additive GP approach utilized in conjunction with MES by Wang and Jegelka [46]; further promising additions would be to tackle heterogeneous noise and input warping as done by HEBO [8].

We also note that, since JES computes the entropy reduction from conditioning on the optimal pair, it relies on some level of noise in the objective. A surrogate model with zero noise will result in an infinite information gain for every optimal pair, which (by utilizing some random tie-breaking strategy) would make JES equivalent to TS. However, if JES is to be used in a completely noiseless setting, we argue that a small noise term should be added as a remedy. As this is done by default in many prominent GP frameworks [12, 13, 44], we do not view this as a major limitation of our approach. Nevertheless, improving upon this strategy would be interesting in future work.

For future work, we also envision work on the adaptation of JES to various different domains, such as multi-fidelity [47] and multi-objective optimization [1], as well as the integration of user prior knowledge over the location of the optimum [19] to accelerate optimization.

Acknowledgments and Disclosure of Funding

Luigi Nardi was supported in part by affiliate members and other supporters of the Stanford DAWN project — Ant Financial, Facebook, Google, Intel, Microsoft, NEC, SAP, Teradata, and VMware. Carl Hvarfner and Luigi Nardi were partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. Luigi Nardi was partially supported by the Wallenberg Launch Pad (WALP) grant Dnr 2021.0348. Frank Hutter acknowledges support through TAILOR, a project funded by the EU Horizon 2020 research and innovation programme under GA No 952215, by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant number 417962828, by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant no INST 39/963-1 FUGG, and by the European Research Council (ERC) Consolidator Grant "Deep Learning 2.0" (grant no. 101045765). The computations were also enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at LUNARC partially funded by the Swedish Research Council through grant agreement no. 2018-05973. Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the ERC. Neither the European Union nor the ERC can be held responsible for them.

Funded by the European Union

References

[1] S. Belakaria, A. Deshwal, and J. R. Doppa. Max-value entropy search for multi-objective bayesian optimization. *Advances in neural information processing systems*, 32, 2019.

[2] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for Hyper-Parameter Optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 24. Curran Associates, Inc., 2011.

[3] F. Berkenkamp, A. Krause, and A. Schoellig. Bayesian optimization with safety constraints: Safe and automatic parameter tuning in robotics. *Machine Learning*, 06 2021. doi: 10.1007/s10994-021-06019-1.

[4] S. Bochner et al. *Lectures on Fourier integrals*, volume 42. Princeton University Press, 1959.

[5] A. D. Bull. Convergence rates of efficient global optimization algorithms. 12:2879–2904, 2011.

[6] R. Calandra, N. Gopalan, A. Seyfarth, J. Peters, and M. Deisenroth. Bayesian gait optimization for bipedal locomotion. In P. Pardalos and M. Resende, editors, *Proceedings of the Eighth International Conference on Learning and Intelligent Optimization (LION'14)*, 2014.

[7] Y. Chen, A. Huang, Z. Wang, I. Antonoglou, J. Schrittwieser, D. Silver, and N. de Freitas. Bayesian optimization in alphago. *CoRR*, abs/1812.06855, 2018. URL `http://arxiv.org/abs/1812.06855`.

[8] A. I. Cowen-Rivers, W. Lyu, Z. Wang, R. Tutunov, J. Hao, J. Wang, and H. Bou-Ammar. HEBO: heteroscedastic evolutionary bayesian optimisation. *CoRR*, abs/2012.03826, 2020. URL `https://arxiv.org/abs/2012.03826`.

[9] K. Eggensperger, P. Müller, N. Mallik, M. Feurer, R. Sass, A. Klein, N. Awad, M. Lindauer, and F. Hutter. HPOBench: A collection of reproducible multi-fidelity benchmark problems for HPO. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL `https://openreview.net/forum?id=1k4rJYEwda-`.

[10] A. Ejjeh, L. Medvinsky, A. Councilman, H. Nehra, S. Sharma, V. Adve, L. Nardi, E. Nurvitadhi, and R. A. Rutenbar. Hpvm2fpga: Enabling true hardware-agnostic fpga programming. In *Proceedings of the 33rd IEEE International Conference on Application-specific Systems, Architectures, and Processors*, 2022.

[11] P. I. Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[12] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson. Gpytorch: Black-box matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, 2018.

[13] GPy. GPy: A gaussian process framework in python. `http://github.com/SheffieldML/GPy`, since 2012.

[14] P. Hennig and C. J. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(1):1809–1837, June 2012. ISSN 1532-4435.

[15] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems*, 2014. URL `https://proceedings.neurips.cc/paper/2014/file/069d3bb002acd8d7dd095917f9efe4cb-Paper.pdf`.

[16] J. M. Hernández-Lobato, M. Gelbart, M. Hoffman, R. Adams, and Z. Ghahramani. Predictive entropy search for bayesian optimization with unknown constraints. In *International conference on machine learning*, pages 1699–1707. PMLR, 2015.

[17] M. W. Hoffman and Z. Ghahramani. Output-space predictive entropy search for flexible global optimization. 2016.

[18] F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization*, 2011.

[19] C. Hvarfner, D. Stoll, A. Souza, L. Nardi, M. Lindauer, and F. Hutter. PiBO: Augmenting Acquisition Functions with User Beliefs for Bayesian Optimization. In *International Conference on Learning Representations*, 2022.

[20] D. Jones, M. Schonlau, and W. Welch. Efficient global optimization of expensive black box functions. 13:455–492, 1998.

[21] H. J. Kushner. A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise. *Journal of Basic Engineering*, 86(1):97–106, 03 1964. ISSN 0021-9223. doi: 10.1115/1.3653121. URL `https://doi.org/10.1115/1.3653121`.

[22] M. Mayr, F. Ahmad, K. I. Chatzilygeroudis, L. Nardi, and V. Krüger. Skill-based Multi-objective Reinforcement Learning of Industrial Robot Tasks with Planning and Knowledge Integration. *CoRR*, abs/2203.10033, 2022. URL `https://doi.org/10.48550/arXiv.2203.10033`.

[23] M. Mayr, C. Hvarfner, K. Chatzilygeroudis, L. Nardi, and V. Krueger. Learning skill-based industrial robot tasks with user priors. *IEEE 18th International Conference on Automation Science and Engineering*, 2022. URL `https://arxiv.org/abs/2208.01605`.

[24] J. Mockus, V. Tiesis, and A. Zilinskas. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2(117-129):2, 1978.

[25] H. B. Moss, D. S. Leslie, J. Gonzalez, and P. Rayson. Gibbon: General-purpose information-based bayesian optimisation. *Journal of Machine Learning Research*, 22(235):1–49, 2021. URL `http://jmlr.org/papers/v22/21-0120.html`.

[26] L. Nardi, D. Koeplinger, and K. Olukotun. Practical design space exploration. In *2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 347–358. IEEE, 2019.

[27] Q. P. Nguyen, B. K. H. Low, and P. Jaillet. Rectified max-value entropy search for bayesian optimization, 2022. URL `https://arxiv.org/abs/2202.13597`.

[28] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, Advances in Neural Information Processing Systems, page 1177–1184, Red Hook, NY, USA, 2007. Curran Associates Inc. ISBN 9781605603520.

[29] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[30] B. Ru, M. A. Osborne, M. Mcleod, and D. Granziol. Fast information-theoretic Bayesian optimisation. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4384–4392. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/ru18a.html`.

[31] D. Russo and B. Van Roy. Learning to optimize via information-directed sampling. *Advances in Neural Information Processing Systems*, 27, 2014.

[32] K. Šehić, A. Gramfort, J. Salmon, and L. Nardi. LassoBench: A High-Dimensional Hyperparameter Optimization Benchmark Suite for Lasso. *arXiv preprint arXiv:2111.02790*, 2021.

[33] A. Shah and Z. Ghahramani. Parallel predictive entropy search for batch global optimization of expensive objective functions. *Advances in neural information processing systems*, 28, 2015.

[34] B. Shahriari, K. Swersky, Z. Wang, R. Adams, and N. de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

[35] J. Sherman and W. J. Morrison. Adjustment of an Inverse Matrix Corresponding to a Change in One Element of a Given Matrix. *The Annals of Mathematical Statistics*, 21(1):124 – 127, 1950. doi: 10.1214/aoms/1177729893. URL `https://doi.org/10.1214/aoms/1177729893`.

[36] J. Snoek, H. Larochelle, and R. Adams. Practical Bayesian optimization of machine learning algorithms. In P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Proceedings of the 26th International Conference on Advances in Neural Information Processing Systems (NeurIPS'12)*, pages 2960–2968, 2012.

[37] J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M. Patwary, Prabhat, and R. Adams. Scalable Bayesian optimization using deep neural networks. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML'15)*, volume 37, pages 2171–2180. Omnipress, 2015.

[38] J. Springenberg, A. Klein, S. Falkner, and F. Hutter. Bayesian optimization with robust Bayesian neural networks. In D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, editors, *Proceedings of the 30th International Conference on Advances in Neural Information Processing Systems (NeurIPS'16)*, 2016.

[39] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, May 2012. ISSN 1557-9654. doi: 10.1109/tit.2011.2182033. URL `http://dx.doi.org/10.1109/TIT.2011.2182033`.

[40] S. Takeno, H. Fukuoka, Y. Tsukada, T. Koyama, M. Shiga, I. Takeuchi, and M. Karasuyama. Multi-fidelity Bayesian optimization with max-value entropy search and its parallelization. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9334–9345. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/takeno20a.html`.

[41] S. Takeno, T. Tamura, K. Shitara, and M. Karasuyama. Sequential and parallel constrained max-value entropy search via information lower bound. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 20960–20986. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/takeno22a.html`.

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

[43] B. Tu, A. Gandy, N. Kantas, and B. Shafei. Joint entropy search for multi-objective bayesian optimization. *arXiv preprint arXiv:2210.02905*, 2022.

[44] J. Vanhatalo, J. Riihimäki, J. Hartikainen, P. Jylänki, V. Tolvanen, and A. Vehtari. Gpstuff: Bayesian modeling with gaussian processes. *Journal of Machine Learning Research*, 14: 1175–1179, Apr. 2013. ISSN 1532-4435.

[45] J. Villemonteix, E. Vazquez, and E. Walter. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 44, 12 2006. doi: 10.1007/s10898-008-9354-2.

[46] Z. Wang and S. Jegelka. Max-value entropy search for efficient bayesian optimization. In *International Conference on Machine Learning (ICML)*, 2017.

[47] Y. Zhang, T. N. Hoang, B. K. H. Low, and M. Kankanhalli. Information-based multi-fidelity bayesian optimization. In *NeurIPS Workshop on Bayesian Optimization*, 2017.

A Broader impact

Our work proposes a novel acquisition function for Bayesian optimization. The approach is foundational and does not have direct societal or ethical consequences. However, JES will be used in the development of applications for a wide range of areas and thus indirectly contribute to their impacts on society. As an algorithm that can be used for HPO, JES intends to cut resource expenditure associated with model training, while increasing their performance. This can help reduce the environmental footprint of machine learning research.

B Experimental setup

Frameworks. For all tasks and acquisition functions, we use the original PES implementation in MATLAB by Hernández-Lobato et al. [15], which uses the GPStuff [44] library. The implementation optimizes the acquisition function, and the posterior mean, by sampling a dense grid of points, and uses a gradient-based optimizer to further optimize the single best point. For better accuracy, we substantially increased the number of grid points. The JES implementation can be found at `https://github.com/jointentropysearch/JointEntropySearch`.

GP sample tasks. To generate the GP sample tasks, we use a random Fourier feature [28] with weights drawn from the spectral density of a squared exponential kernel. For dimension-wise length scale θ_d, output scale σ^2, and noise variance σ_ϵ^2, the hyperparameters per task are shown in Table 2. The range in the last column is a rough approximation of the magnitude of the output spanned by

D	θ_d	σ^2	σ_ϵ^2	Approximate range
2	0.1	10	0.01	$[-9, 9]$
4	0.2	10	0.01	$[-11, 11]$
6	0.3	10	0.01	$[-13, 13]$
12	0.6	10	0.01	$[-18, 18]$

Table 2: Hyperparameters for the generated GP sample tasks.

each GP sample. The length scales of the samples are gradually increased with each dimensionality to maintain a reasonable level of difficulty for all tasks. Since the optimal values for these tasks are unavailable, they are approximated through a dense random search, followed by local search on the most promising subset of points.

Runtime tests. For the runtime tests, we run each acquisition function on 100 seeds on the 2, 4 and 6-dimensional GP sample tasks for 100 iterations each. To consider the runtime induced by each acquisition function in a realistic setting, we marginalize over 10 hyperparameter sets (including noise variance), and sample 10 optima each for JES, MES and PES. For these experiments, the runtime of an acquisition function is considered to be the time the time from after GP hyperparameters are sampled and the GP covariance has been inverted, until the query has been selected. Thus, only acquisition function setup and acquisition function optimization are considered as part of the runtime. All acquisition functions are optimized using an identical budget of 10000 raw samples, and a subsequent gradient-based optimization around the single best point.

Synthetic test functions. For the synthetic test functions, 100 sampled optimal pairs are used for each acquisition function. GP hyperparameters are marginalized over for these tasks, so an equal number of optimal pairs are sampled for each hyperparameter set. The hyperparameters are re-sampled on a fixed schedule throughout the run. Naturally, the sampled maxima were updated at each iteration. Moreover, each test function was also given a fixed amount of noise. Regret was computed not from the noisy observed value, but from the true, noiseless function value.

MLP classification tasks. All the classification tasks have a substantial amount of noise. As noiseless objective values are unavailable, we report the observed classification accuracy. 5 hyperparameters are available in HPOBench for these tasks. However, one of them (number of layers, *depth*) is held fixed due to its small integer-valued domain and the lack of integer hyperparameter support in

Task	No. hyperparameter sets	No. maxima per set	Update frequency	σ_ϵ^2
Branin	20	5	5	0.01
Hartmann (3D)	20	5	10	0.01
Hartmann (6D)	20	5	10	0.01
MLP classification	20	5	5	0.0

Table 3: GP Hyperparameter sets and updates for the synthetic test functions and MLP tasks.

Name	Type	Range
Alpha (L2)	Continuous	$[10^{-8}, 10^{-3}]$
Batch size	Integer	$[2^2, 2^8]$
Depth	Fixed to 2	$\{1, 2, 3\}$
Initial learning rate	Continuous	$[10^{-5}, 1]$
Width	Integer	$[2^4, 2^{10}]$

Table 4: Search space for the MLP tasks.

the MATLAB framework. As seen in Tab. 4, the two other integer-valued hyperparameters batch size and *width* have orders of magnitude larger domains, and can therefore reasonably be treated as continuous hyperparameters. All tasks are optimized in the [0, 1] range, and are scaled, transformed, and rounded to the nearest integer in the objective function where necessary. All non-fixed parameters are evaluated in log scale. The three tasks evaluated are *Australian*, *Blood-transfusion-service-center*, and *Vehicle*, with HPOBench task numbers #146818, #10101, #53, respectively.

Compute resources. All experiments are carried out on *Intel Xeon Gold 6130* CPUs. Each repetition is run on a single core. In total, approximately $50,000$ core hours are used for the experiments in the main paper, and an additional $20,000$ for the appendix.

C Ablation studies and model misspecification

We provide ablation studies for the hyperparameter controlling the ratio of exploitative selections γ. Moreover, we show how γ-exploit improves inference regret by recognizing regions believed to be optimal as poor. Lastly, we display the robustness of `JES` to the noise variance σ_ϵ^2.

C.1 Ablation studies

We provide an ablation study on γ in terms of both the simple and inference regret of JES. While simple regret may be a more practically relevant metric, inference regret helps understand the ability of the acquisition function to successfully locate the optimum. Fig. 9 shows that $\gamma > 0$ improves simple regret, which is to be expected from its occasional greedy selection. However, as is shown in Fig. 10, a moderate fraction $\gamma \in \{0.05, 0.1\}$ also yields comparable or even improved inference regret on all tasks. Notably, $\gamma = 0.2$ yields slightly worse performance on Hartmann (6D), but yields marginally improved performance on Branin and Hartmann (3D). As such, the γ-exploit approach not only improves performance in terms of simple regret, but yields improved inference as well.

C.2 Model misspecification

As mentioned in Sec. 3.5, the performance of information-theoretic methods can suffer substantially from model misspecification. In Fig. 11, we show for `JES`, `PES` and `MES` how the γ-exploit approach helps stabilize inference and improve inference regret, and yields substantially better simple regret for all methods. For Michalewicz (10D), we observe that the inference regret of `MES` gets substantially worse after iteration 150. With the γ-exploit approach, this issue is severely reduced. In Fig. 12, the corresponding simple regrets for `MES` deviate at iteration 150. The same behavior can be observed for `JES` on Levy (8D) around iteration 200. Across all test functions in Fig. 11 and Fig. 12, an γ-exploit strategy yields comparable or improved inference regret, and strictly improved simple regret for all acquisition functions. Moreover, we observe that `MES` is generally the top-performing acquisition

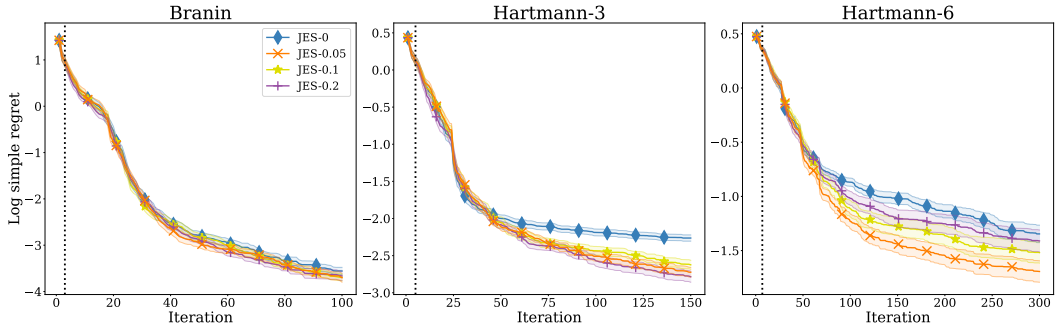

Figure 9: Comparison of JES with varying fraction γ of exploitative selections. Mean and 1 standard error of log simple regret is displayed for all tasks.

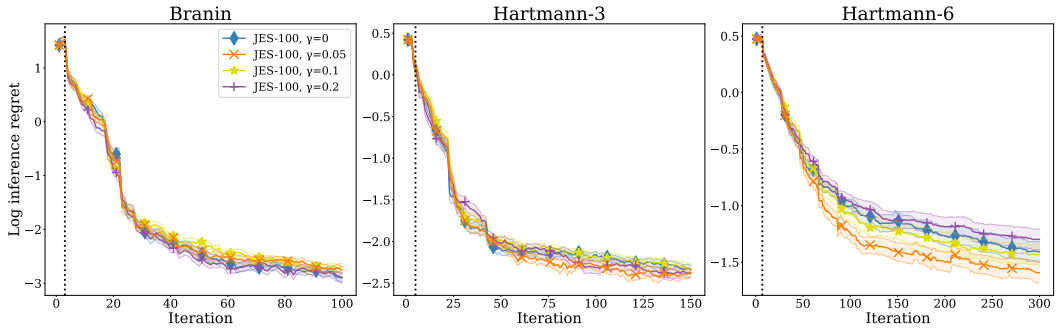

Figure 10: Comparison of JES with varying fraction γ of exploitative selections. Mean and 1 standard error of log inference regret is displayed for all tasks.

function, implying that it is the most robust to model misspecification. Using the γ-exploit approach, the acquisition function verifies whether the belief over the location of the optimum is correct under the current model hyperparameters. If it is not, it re-calibrates its belief.

C.3 Ablation study on σ_ϵ^2

For the noise variance ablation study, we once again consider the GP sample tasks. We fix the GP noise hyperparameter σ_ϵ^2 to the correct value prior to the start of the experiment. In Fig. 13, we show that the performance of JES is robust with respect to the noise level, while the performance of MES and PES decrease more drastically as the level of noise increases.

D Dependence on the number of MC samples

We show in Fig. 14 the dependence of JES on the number of MC samples for the GP sample tasks. JES displays a slight dependence on the number of GP samples, as initial performance improves marginally for larger number of samples. This is more prominent for higher-dimensional tasks, where a lower number of samples causes a substantially slower start. This can be explained by the fact that a larger number of sampled optimal pairs are required to accurately model a higher-dimensional density. Notably, the final regret on the 12-D benchmark is better for lower numbers of samples, such as JES-4. One potential explanation for this is its inability to model the joint distribution accurately. If all realized optimal pairs are close to the perceived optimum, JES is almost certainly going to sample there. Moreover, since the information gain for each sample is relatively low for samples in a well-explored region, the information gain of a few samples in an unexplored region may outweight the information gain of a much larger number of samples in a well-explored region. For JES-4 and JES-20, it is more likely that all of the sampled optimal pairs are close to the optimum in this manner, while there is still small positive density on other parts of the search space. As such, it is possible that JES-4 and JES-20 over-exploit slightly in the 12-dimensional benchmark.

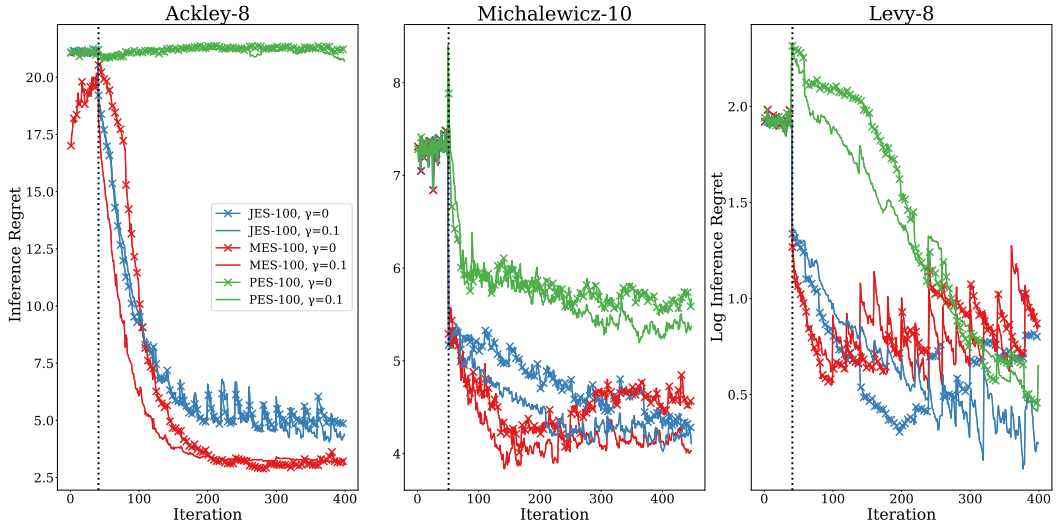

Figure 11: Mean of inference regret on high-dimensional synthetic functions for vanilla γ-exploit versions of JES, MES and PES. Upward spikes signify points where the GP hyperparameters are re-sampled, and the inference regret getting worse as a result. Error bars are omitted to increase legibility.

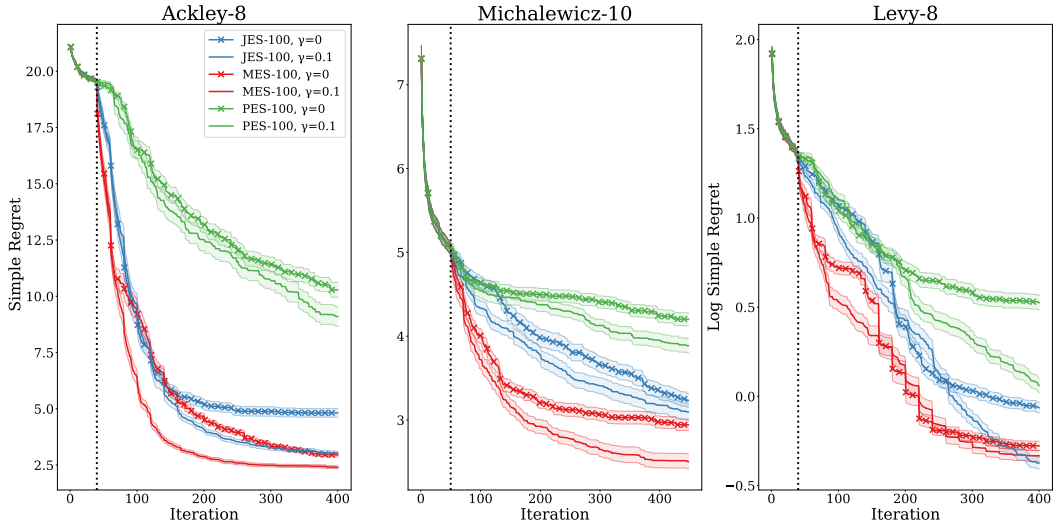

Figure 12: Mean and 1 standard error of simple regret on high-dimensional synthetic functions for vanilla γ-exploit versions of JES, MES and PES.

E Approximation Quality

As stated in Sec. 3.4, we approximate the entropy of the posterior over observations conditioned on the data and the optimal pair $\mathrm{H}\left[p(y|\mathcal{D} \cup (\boldsymbol{x}^*, f^*), \boldsymbol{x}, f^*)\right]$ – the entropy of the sum of a Gaussian and a truncated Gaussian variable – by moment matching of the truncated distribution. We now show the quality of this approximation, and what impact it can potentially have on point selection. To do so, we utilize the results from Nguyen et al. [27] regarding the density of $p(y|\mathcal{D}, \boldsymbol{x}, f^*)$. We note that the approximation regards the *truncation* from knowing the optimal value f^*, which constitutes an additional reduction in entropy after having conditioned the GP on the new observation as in Sec. 3.3. As such, the approximation considers only a fraction of the total entropy reduction, as visualized in Fig. 3.

To establish the quality of the approximation, we compare our approach to approximating the entropy by MC. Naturally, the MC approach is more computationally expensive than moment matching, but

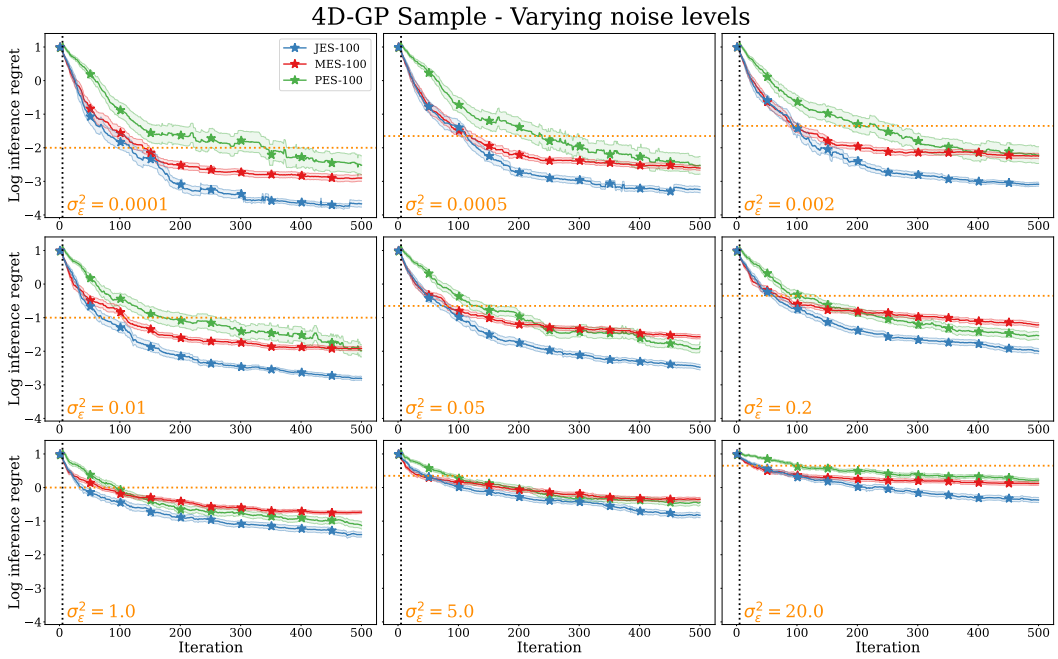

Figure 13: Evaluation of JES, MES and PES on noisy 4D GP sample tasks across 50 repetitions for 9 different noise levels. The noise variance σ_ϵ^2 ranges from 10^{-4} (top left) to 20 (bottom right). Log noise standard deviation $\log(\sigma_\epsilon)$ is marked in dashed orange.

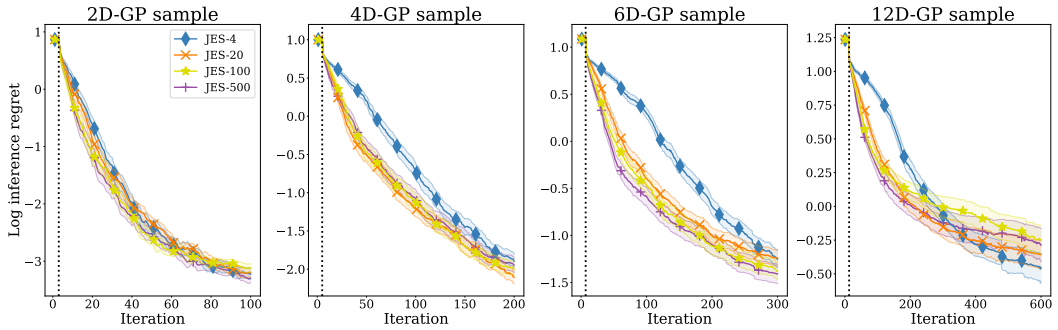

Figure 14: Comparison of JES with varying numbers of MC samples on GP tasks of varying dimensionalities - 10 repetitions on 10 unique tasks, for a total of 100 repetitions. Mean and 1 standard error of log inference regret is displayed for all tasks.

yields an asymptotically correct result. In Fig. 15, we show for varying noise levels, expressed as the noise variance ratio of the total variance, and truncation quantiles $\Phi^{-1}(\alpha)$, the difference in entropy between an MC and a moment matching approach. For example, a truncation quantile of 10^{-2} means that the upper 99% of the density of the posterior distribution is removed as a result of truncation. We see that the approximate entropy reduction from moment matching is consistently lower than for the asymptotically exact MC approach. Moreover, the approximation error seemingly increases logarithmically with the truncation quantile. As such, we can expect modest underestimates of the entropy reduction when we truncate the posterior to an extreme degree, and the level of noise is low. In the right image of Fig 2, the posterior is severely truncated left of the conditioned location. However, the noise variance constitutes a large fraction of the total variance at this location, which means that the approximation is still accurate with respect to the true entropy reduction. The scenario represented in the bottom left corner of the grid, where we severely truncate the posterior *and* the noise variance is low, does not reasonably occur in practice. The aforementioned scenario would entail sampling an optimal pair which is several orders of magnitudes *worse* than the mean of the GP at uncertain (in the sense that $\sigma >> \sigma_\epsilon$) locations. Lastly, the blue region in the upper left corner of

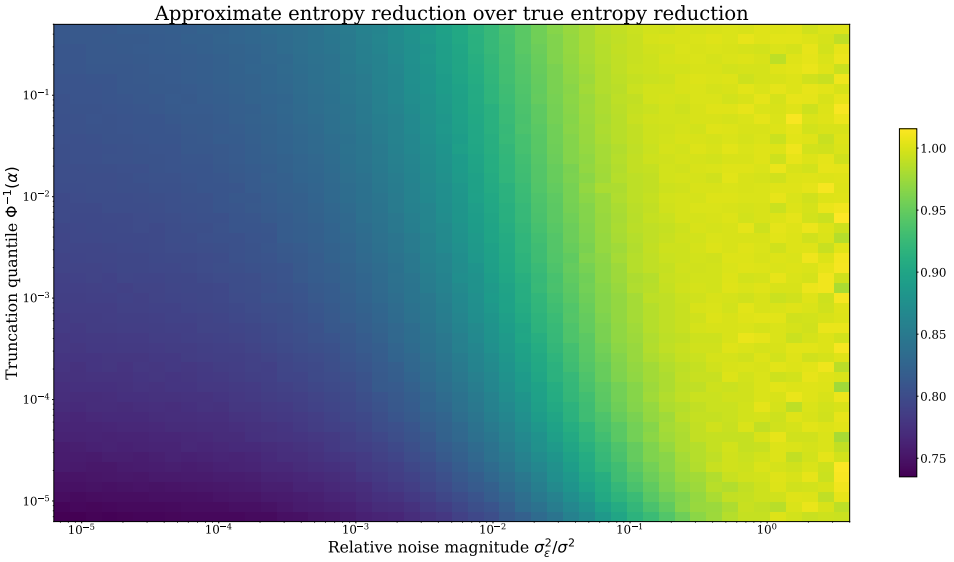

Figure 15: Visualization of approximation error from the moment matching approach compared to an asymptotically exact MC approach. The colormap represents the fraction of the entropy reduction resulting from truncation as approximated from moment matching divided by the entropy reduction as computed through MC. The inconsistencies in coloring in the rightmost part of the image are caused by inconsistent MC approximation.

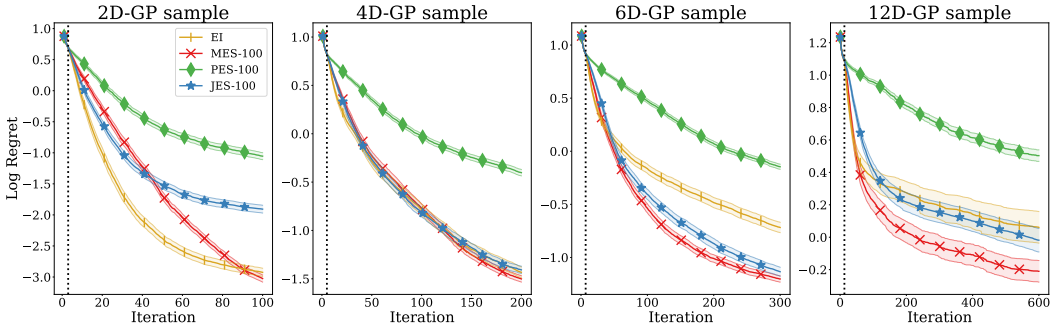

Figure 16: Comparison of JES, MES, PES and EI on GP prior samples using simple regret. We run 1000 repetitions each for 2, 4 and 6D, and 250 on 12D. Mean and 2 standard errors of log regret are displayed for each acquisition function. The vertical dashed line shows the end of the initial design phase.

the grid represents a region where we have less truncation, and the noise is a relatively insignificant part of the total variance. Fig. 15 shows that the entropy reduction from truncation in this region is underestimated by approximately 15%. As such, the approximation error leads to a slightly less explorative strategy than what a strategy with exact computation of the truncation term would provide.

F Regret Measures

We display the regret measures for the GP sample and synthetic test functions.

F.1 GP sample tasks

In Fig. 16, we show the simple regret for the GP sample tasks. We note that the simple and inference regrets for MES are approximately equal, while there is a substantial difference for JES. For PES, the difference between simple and inference regret is the most pronounced at approximately two order of magnitude at the most.

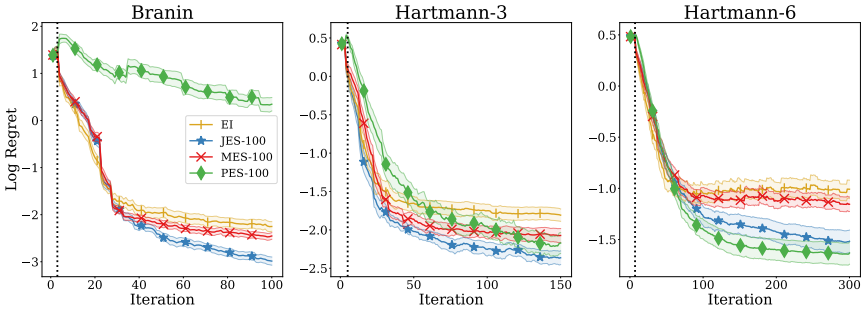

Figure 17: Comparison of JES, MES, PES and EI on Branin and Hartmann-6, $\sigma_n^2 = 0.10$. Mean and 2 standard errors of log regret are displayed for each acquisition function across 100 repetitions. The vertical dashed line represents the end of the initial design phase.

F.2 Synthetic test functions

Next, we show the inference regret for the synthetic test functions in Fig. 17. We note that he simple and inference regrets for JES are approximately equal. For PES, the difference between simple and inference regret is once again very pronounced. Notably, the simple regret is significantly better than the inference regret for MES on Branin, implying that it does not yet have full knowledge of where the optimum is. We once again note the numerical issues of PES on Branin. Moreover, the inference regret of MES gets marginally worse for Hartmann (6D) from approximately iteration 100 until the end of the run, implying that its knowledge about the location of the optimum gets worse.