Supplementary Material
Bayesian Optimization by Density Ratio Estimation

Louis C. Tiao[1,3] **Aaron Klein**[2] **Cédric Archambeau**[2]
Edwin V. Bonilla[3,1] **Matthias Seeger**[2] **Fabio Ramos**[1,4]
[1]University of Sydney [2]Amazon Web Services [3]CSIRO's Data61 [4]NVIDIA

A Expected improvement

For completeness, we reproduce the derivations of [1]. Recall that expected improvement (EI) is defined as the expectation of the improvement utility function $I_\gamma(\mathbf{x})$ under the posterior predictive distribution $p(y \,|\, \mathbf{x}, \mathcal{D}_N)$. Expanding this out, we have

$$\begin{aligned}
\alpha_\gamma(\mathbf{x}; \mathcal{D}_N) = \mathbb{E}_{p(y \,|\, \mathbf{x}, \mathcal{D}_N)}[I_\gamma(\mathbf{x})] &= \int_{-\infty}^{\infty} I_\gamma(\mathbf{x}) p(y \,|\, \mathbf{x}, \mathcal{D}_N) \, \mathrm{d}y && (1) \\
&= \int_{-\infty}^{\tau} (\tau - y) p(y \,|\, \mathbf{x}, \mathcal{D}_N) \, \mathrm{d}y && (2) \\
&= \frac{1}{p(\mathbf{x} \,|\, \mathcal{D}_N)} \int_{-\infty}^{\tau} (\tau - y) p(\mathbf{x} \,|\, y, \mathcal{D}_N) p(y \,|\, \mathcal{D}_N) \, \mathrm{d}y. && (3)
\end{aligned}$$

Note we have used Bayes' rule in the last step above. Next, the denominator evaluates to

$$\begin{aligned}
p(\mathbf{x} \,|\, \mathcal{D}_N) &= \int_{-\infty}^{\infty} p(\mathbf{x} \,|\, y, \mathcal{D}_N) p(y \,|\, \mathcal{D}_N) \, \mathrm{d}y && (4) \\
&= \ell(\mathbf{x}) \int_{-\infty}^{\tau} p(y \,|\, \mathcal{D}_N) \, \mathrm{d}y + g(\mathbf{x}) \int_{\tau}^{\infty} p(y \,|\, \mathcal{D}_N) \, \mathrm{d}y && (5) \\
&= \gamma \ell(\mathbf{x}) + (1 - \gamma) g(\mathbf{x}), && (6)
\end{aligned}$$

since $\gamma = \Phi(\tau) = p(y < \tau \,|\, \mathcal{D}_N)$, by definition. Finally, we evaluate the numerator,

$$\begin{aligned}
\int_{-\infty}^{\tau} (\tau - y) p(\mathbf{x} \,|\, y, \mathcal{D}_N) p(y \,|\, \mathcal{D}_N) \, \mathrm{d}y &= \ell(\mathbf{x}) \int_{-\infty}^{\tau} (\tau - y) p(y \,|\, \mathcal{D}_N) \, \mathrm{d}y && (7) \\
&= \ell(\mathbf{x}) \tau \int_{-\infty}^{\tau} p(y \,|\, \mathcal{D}_N) \, \mathrm{d}y - \ell(\mathbf{x}) \int_{-\infty}^{\tau} y p(y \,|\, \mathcal{D}_N) \, \mathrm{d}y && (8) \\
&= \gamma \tau \ell(\mathbf{x}) - \ell(\mathbf{x}) \int_{-\infty}^{\tau} y p(y \,|\, \mathcal{D}_N) \, \mathrm{d}y && (9) \\
&= K \cdot \ell(\mathbf{x}), && (10)
\end{aligned}$$

where

$$K = \gamma \tau - \int_{-\infty}^{\tau} y p(y \,|\, \mathcal{D}_N) \, \mathrm{d}y. \qquad (11)$$

4th Workshop on Meta-Learning at NeurIPS2020, Vancouver, Canada.

Hence, this shows that the EI function is equivalent to the γ-relative density ratio [3] up to a constant factor K,

$$\alpha_\gamma(\mathbf{x}; \mathcal{D}_N) \propto \frac{\ell(\mathbf{x})}{\gamma \ell(\mathbf{x}) + (1-\gamma) g(\mathbf{x})} \tag{12}$$

$$= \left(\gamma + \frac{g(\mathbf{x})}{\ell(\mathbf{x})}(1-\gamma) \right)^{-1}. \tag{13}$$

B Class-posterior probability

We provide an unabridged derivation of the identity in eq. 13. First, the ordinary density ratio is given by

$$r_0(\mathbf{x}) = \frac{\ell(\mathbf{x})}{g(\mathbf{x})} = \frac{p(\mathbf{x} \,|\, z=1)}{p(\mathbf{x} \,|\, z=0)} \tag{14}$$

$$= \left(\frac{p(z=1 \,|\, \mathbf{x}) p(\mathbf{x})}{p(z=1)} \right) \left(\frac{p(z=0 \,|\, \mathbf{x}) p(\mathbf{x})}{p(z=0)} \right)^{-1} \tag{15}$$

$$= \frac{p(z=0)}{p(z=1)} \cdot \frac{p(z=1 \,|\, \mathbf{x})}{p(z=0 \,|\, \mathbf{x})}. \tag{16}$$

By construction, we have,

$$\frac{p(z=0)}{p(z=1)} = \frac{p(y \geq \tau)}{p(y < \tau)} = \frac{1-\gamma}{\gamma} = \left(\frac{\gamma}{1-\gamma} \right)^{-1}. \tag{17}$$

Furthermore,

$$\frac{p(z=1 \,|\, \mathbf{x})}{p(z=0 \,|\, \mathbf{x})} = \frac{p(z=1 \,|\, \mathbf{x})}{1 - p(z=1 \,|\, \mathbf{x})} = \frac{\pi(\mathbf{x})}{1-\pi(\mathbf{x})}. \tag{18}$$

Therefore,

$$r_0(\mathbf{x}) = \left(\frac{\gamma}{1-\gamma} \right)^{-1} \frac{\pi(\mathbf{x})}{1-\pi(\mathbf{x})}. \tag{19}$$

Plugging this into the expression of eq. 4, we have

$$r_\gamma(\mathbf{x}) = \left(\gamma + (1-\gamma) \left(\frac{\gamma}{1-\gamma} \right) \left(\frac{\pi(\mathbf{x})}{1-\pi(\mathbf{x})} \right)^{-1} \right)^{-1} \tag{20}$$

$$= \gamma^{-1} \left(1 + \left(\frac{\pi(\mathbf{x})}{1-\pi(\mathbf{x})} \right)^{-1} \right)^{-1} \tag{21}$$

$$= \gamma^{-1} \pi(\mathbf{x}), \tag{22}$$

as required.

C Binary cross-entropy

The binary cross-entropy (BCE) loss is given by

$$\mathcal{L}(\boldsymbol{\theta}) = -\beta \cdot \mathbb{E}_{\ell(\mathbf{x})}[\log \pi_{\boldsymbol{\theta}}(\mathbf{x})] - (1-\beta) \cdot \mathbb{E}_{g(\mathbf{x})}[\log(1 - \pi_{\boldsymbol{\theta}}(\mathbf{x}))], \tag{23}$$

where β denotes the class balance rate. That is, let N_ℓ and N_g be the sizes of the support of $\ell(\mathbf{x})$ and $g(\mathbf{x})$, respectively. Then, we have

$$\beta = \frac{N_\ell}{N}, \qquad \text{and} \qquad 1 - \beta = \frac{N_g}{N}, \tag{24}$$

where $N = N_\ell + N_g$.

C.1 Empirical risk minimization

We show that the BCE loss can be approximated by the empirical risk,

$$\mathcal{L}(\boldsymbol{\theta}) \simeq -\frac{1}{N}\left(\sum_{n=1}^{N} z_n \log \pi_{\boldsymbol{\theta}}(\mathbf{x}_n) + (1 - z_n)\log\left(1 - \pi_{\boldsymbol{\theta}}(\mathbf{x}_n)\right)\right). \tag{25}$$

Let ρ be the permutation of the set $\{1, \dots, N\}$, i.e. the bijection from $\{1, \dots, N\}$ to itself, such that $y_{\rho(n)} < \tau$ if $0 < \rho(n) \leq N_\ell$, and $y_{\rho(n)} \geq \tau$ if $N_\ell < \rho(n) \leq N_g$. That is to say,

$$\mathbf{x}_{\rho(n)} \sim \begin{cases} \ell(\mathbf{x}) & \text{if } 0 < \rho(n) \leq N_\ell, \\ g(\mathbf{x}) & \text{if } N_\ell < \rho(n) \leq N_g. \end{cases} \quad \text{and} \quad z_{\rho(n)} = \begin{cases} 1 & \text{if } 0 < \rho(n) \leq N_\ell, \\ 0 & \text{if } N_\ell < \rho(n) \leq N_g. \end{cases} \tag{26}$$

Then, we have

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}) &= -\frac{1}{N}\left(N_\ell \cdot \mathbb{E}_{\ell(\mathbf{x})}[\log \pi_{\boldsymbol{\theta}}(\mathbf{x})] + N_g \cdot \mathbb{E}_{g(\mathbf{x})}[\log\left(1 - \pi_{\boldsymbol{\theta}}(\mathbf{x})\right)]\right) & (27)\\
&\simeq -\frac{1}{N}\left(\cancel{N_\ell} \cdot \frac{1}{\cancel{N_\ell}} \sum_{n=1}^{N_\ell} \log \pi_{\boldsymbol{\theta}}(\mathbf{x}_{\rho(n)}) + \cancel{N_g} \cdot \frac{1}{\cancel{N_g}} \sum_{n=N_\ell+1}^{N_g} \log\left(1 - \pi_{\boldsymbol{\theta}}(\mathbf{x}_{\rho(n)})\right)\right) & (28)\\
&= -\frac{1}{N}\left(\sum_{n=1}^{N} z_{\rho(n)} \log \pi_{\boldsymbol{\theta}}(\mathbf{x}_{\rho(n)}) + (1 - z_{\rho(n)})\log\left(1 - \pi_{\boldsymbol{\theta}}(\mathbf{x}_{\rho(n)})\right)\right) & (29)\\
&= -\frac{1}{N}\left(\sum_{n=1}^{N} z_n \log \pi_{\boldsymbol{\theta}}(\mathbf{x}_n) + (1 - z_n)\log\left(1 - \pi_{\boldsymbol{\theta}}(\mathbf{x}_n)\right)\right). & (30)
\end{aligned}$$

C.2 Optimum

We show the identity of eq. 16. Taking the functional derivative and setting it to zero, we get

$$\begin{aligned}
0 = \frac{\partial \mathcal{L}}{\partial \pi_{\boldsymbol{\theta}}} &= -\mathbb{E}_{\ell(\mathbf{x})}\left[\frac{\beta}{\pi_{\boldsymbol{\theta}}(\mathbf{x})}\right] + \mathbb{E}_{g(\mathbf{x})}\left[\frac{1-\beta}{1 - \pi_{\boldsymbol{\theta}}(\mathbf{x})}\right] & (31)\\
&= \int -\frac{\beta \ell(\mathbf{x})}{\pi_{\boldsymbol{\theta}}(\mathbf{x})} + \frac{(1-\beta) g(\mathbf{x})}{1 - \pi_{\boldsymbol{\theta}}(\mathbf{x})} \mathrm{d}\mathbf{x} & (32)
\end{aligned}$$

This integral evaluates to zero iff the integrand itself evaluates to zero. Hence, we solve the following for $\pi_{\boldsymbol{\theta}}(\mathbf{x})$,

$$\frac{\beta \ell(\mathbf{x})}{\pi_{\boldsymbol{\theta}}(\mathbf{x})} = \frac{(1-\beta) g(\mathbf{x})}{1 - \pi_{\boldsymbol{\theta}}(\mathbf{x})}. \tag{33}$$

We re-arrange this expression to give

$$\frac{1 - \pi_{\boldsymbol{\theta}}(\mathbf{x})}{\pi_{\boldsymbol{\theta}}(\mathbf{x})} = \left(\frac{1-\beta}{\beta}\right)\frac{g(\mathbf{x})}{\ell(\mathbf{x})} \quad \Leftrightarrow \quad \frac{1}{\pi_{\boldsymbol{\theta}}(\mathbf{x})} - 1 = \frac{\beta \ell(\mathbf{x}) + (1-\beta) g(\mathbf{x})}{\beta \ell(\mathbf{x})} - 1. \tag{34}$$

Finally, we add one to both sides and invert the result to give

$$\pi_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{\beta \ell(\mathbf{x})}{\beta \ell(\mathbf{x}) + (1-\beta) g(\mathbf{x})}, \tag{35}$$

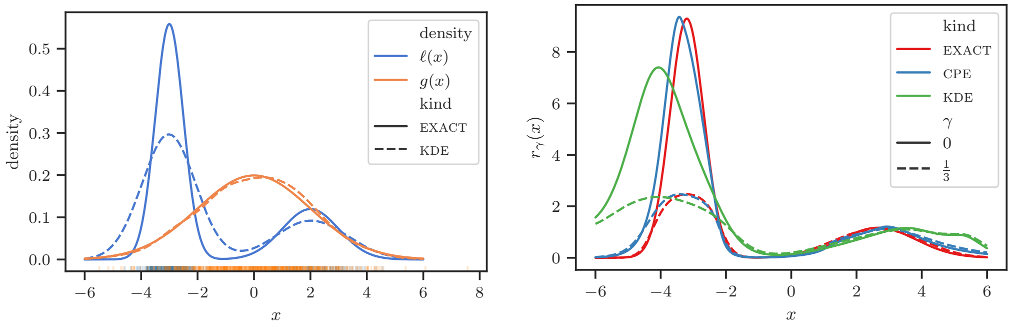
as required.

D Toy example

Consider the following toy example where the densities $\ell(x)$ and $g(x)$ are *known* and given exactly by the following (mixture of) Gaussians,

$$\ell(x) = 0.3\mathcal{N}(2, 1^2) + 0.7\mathcal{N}(-3, 0.5^2), \quad \text{and} \quad g(x) = \mathcal{N}(0, 2^2), \tag{36}$$

as illustrated by the *solid blue* and *orange* lines in Figure 1a, respectively. We draw a total of



(a) Densities $\ell(x)$ and $g(x)$.

(b) γ-relative density ratios $r_\gamma(x)$.

Figure 1: Synthetic toy example with (mixtures of) Gaussians.

$N = 1000$ samples from these distributions, with a fraction $\gamma = 1/3$ drawn from $\ell(x)$ and the remainder from $g(x)$. These are represented by the vertical markers along the bottom of the x-axis (a so-called "rug plot"). Then, two kernel density estimations (KDEs), shown with *dashed* lines, are fit on these respective sample sets, with kernel bandwidths selected according to the "normal reference" rule-of-thumb. We see that, for both densities, the modes are recovered well, while for $\ell(x)$, the variances are overestimated in both of its mixture components. As we shall see, this has deleterious effects on the resulting density ratio estimate.

In Figure 1b, we show the ordinary and γ-relative density ratios with the *solid* and *dashed red* lines, respectively. The density ratio estimates resulting from taking the ratio of the KDEs are shown in *green*. Those resulting from the class-probability estimation (CPE) method described in § 3 are shown in *blue*. The probabilistic classifier consists of a simple multi-layer perceptron (MLP) with 3 hidden layers, each with and 32 units and `elu` activations.

The CPE method appears, at least visually, to recover the exact density ratios well, while the KDE method does so quite poorly. Perhaps the more important quality to focus on, for the purposes of Bayesian optimization (BO), is the *global maximum* of the density ratio functions. In the case of the KDE method, we can see that this deviates significantly from that of the true density ratios. In this instance, even though KDE fit $g(x)$ well and recovered the modes of $\ell(x)$ accurately, a slight overestimation of the variance in the latter led to a significant shift in the maximum of the resulting density ratio functions.

E Implementation details

Software. Our method is implemented as a *configuration generator* plugin for the `HpBandSter` library of Falkner et al. [2].

Epochs per iteration. To ensure the training time on BO iteration N is nonincreasing as a function of N, instead of directly specifying the number of epochs (i.e. full passes over the data), we specify the number of (batchwise gradient) steps S to train for in each iteration. Since the number of steps per epoch is $M = \lceil N/B \rceil$, the effective number of epochs on the N-th BO iteration is then $E = \lfloor S/M \rfloor$. For example, if $S = 1024$ and $B = 64$, the number of epochs for iteration $N = 512$ would be $E = 128$. As another example, for all $0 < N \leq B$, we have $E = S = 1024$. We set $S = 100$.

F Qualitative analysis

In Figures 2 and 3, we show a scatterplot of the locations suggested by tree-structured Parzen estimator (TPE) and BO, respectively, across 20 runs on the SIX-HUMP CAMEL problem.

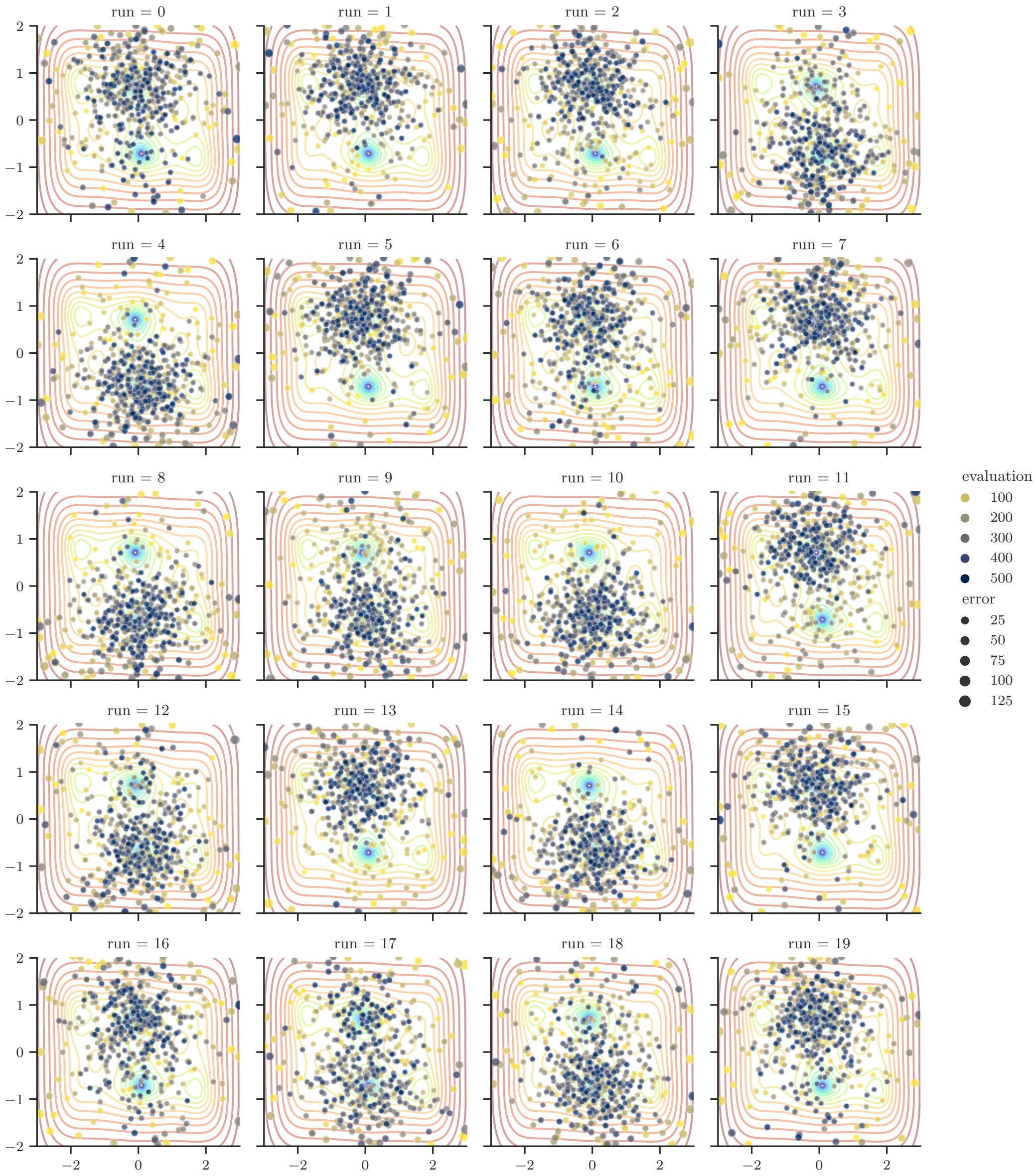


Figure 2: Candidates suggested by TPE on the SIX-HUMP CAMEL problem across 20 runs.

References

[1] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, pages 2546–2554, 2011.

[2] S. Falkner, A. Klein, and F. Hutter. Bohb: Robust and efficient hyperparameter optimization at scale. In *International Conference on Machine Learning*, pages 1437–1446, 2018.

[3] M. Yamada, T. Suzuki, T. Kanamori, H. Hachiya, and M. Sugiyama. Relative density-ratio estimation for robust distribution comparison. In *Advances in Neural Information Processing Systems*, pages 594–602, 2011.

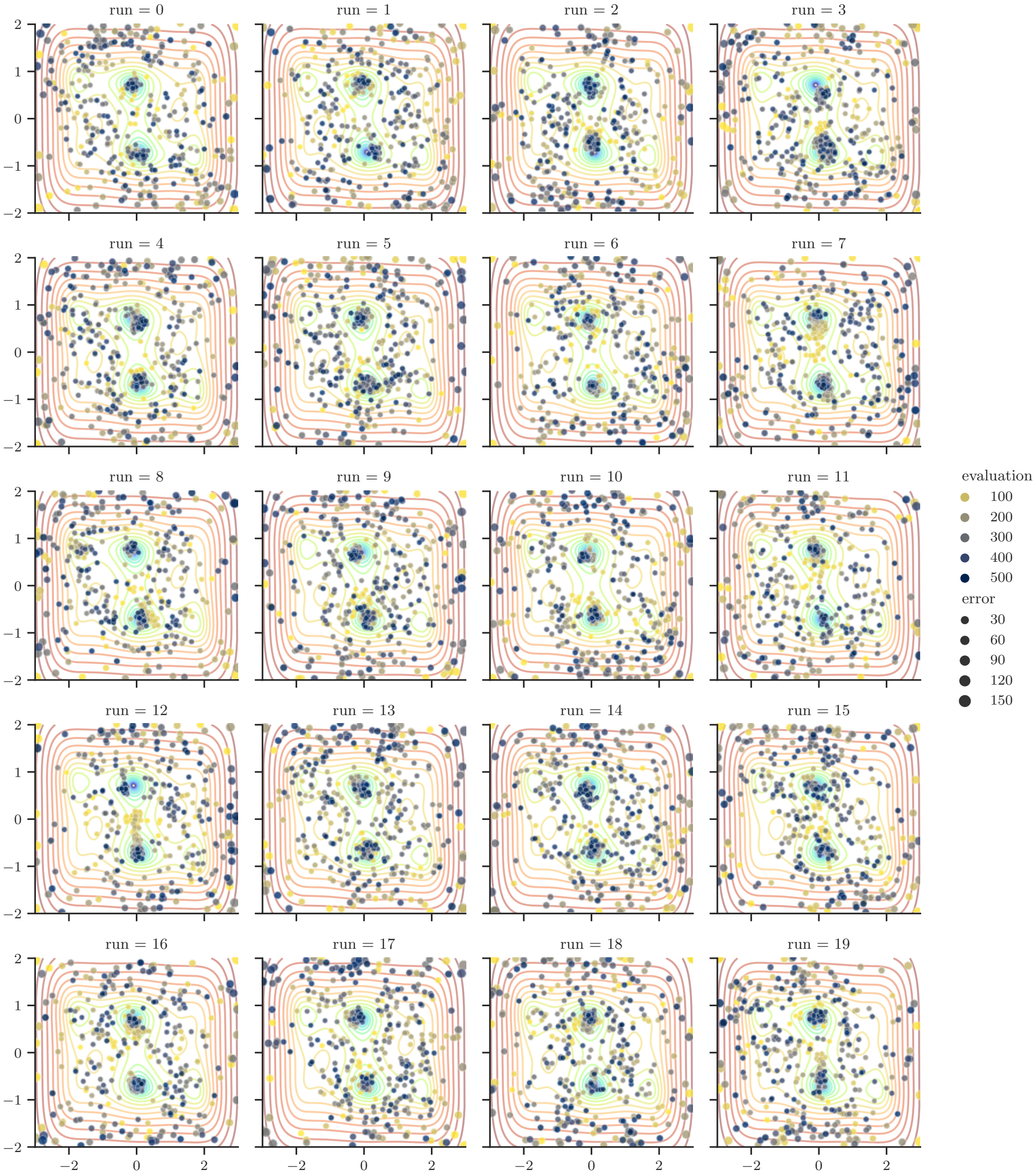


Figure 3: Candidates suggested by BORE on the SIX-HUMP CAMEL problem across 20 runs.