

Project Report

Alexis Park, Jonathan Chen, Kevin Xie
CSE515T: Bayesian Methods in Machine Learning

December 3, 2019

Data visualization

The Branin function is defined as follows:

$$f(\mathbf{x}) = a(x_2 - bx_1^2 + cx_1 - r)^2 + s(1 - t)\cos(x_1) + s \quad (0.1)$$

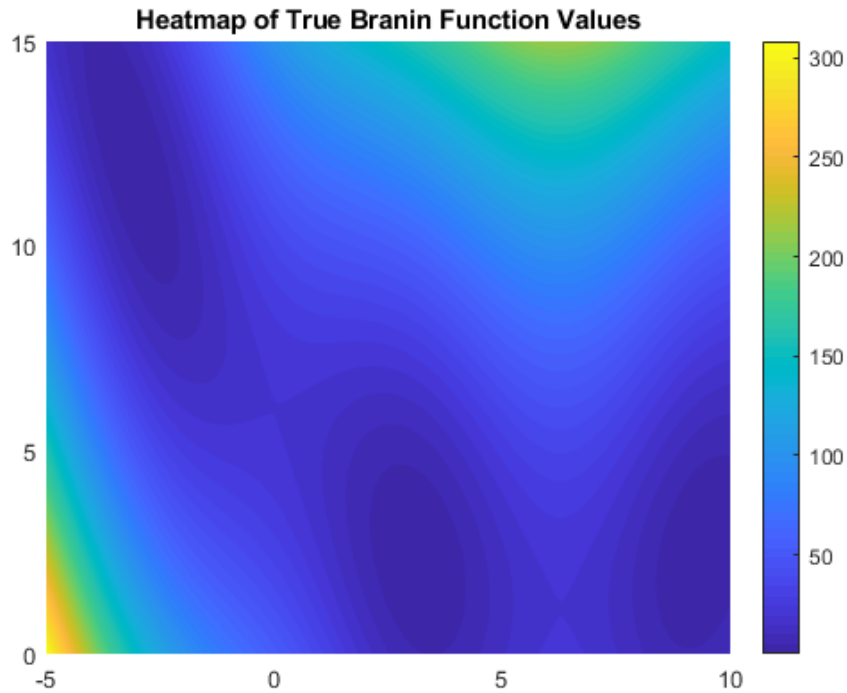


Figure 1: Heatmap of True Branin Function values with domain $\mathcal{X} = [-5, 10] \times [0, 15]$ with 1000 values per dimension.

From 1, one can see that the function's values fluctuate in a somewhat sinusoidal manner, with a large minimal "trench" spanning diagonally across the center of the domain with steadily increasing regions along the trench's sides. Therefore the function does not appear to be stationary.

In order to make the function more stationary, we tried couple of transformations. By analyzing the equation, we note that the sinusoidal fluctuations can be attributed to both the added

cosine term and the 4th order polynomial term. We thus attempted to pass the data through a transformation that combined an inverse cosine function (*arc cos*) with the cumulative density function of the normal probability distribution (*normal cdf*). However, taking the *log* of the data gave us the most stationary result as shown in 2.

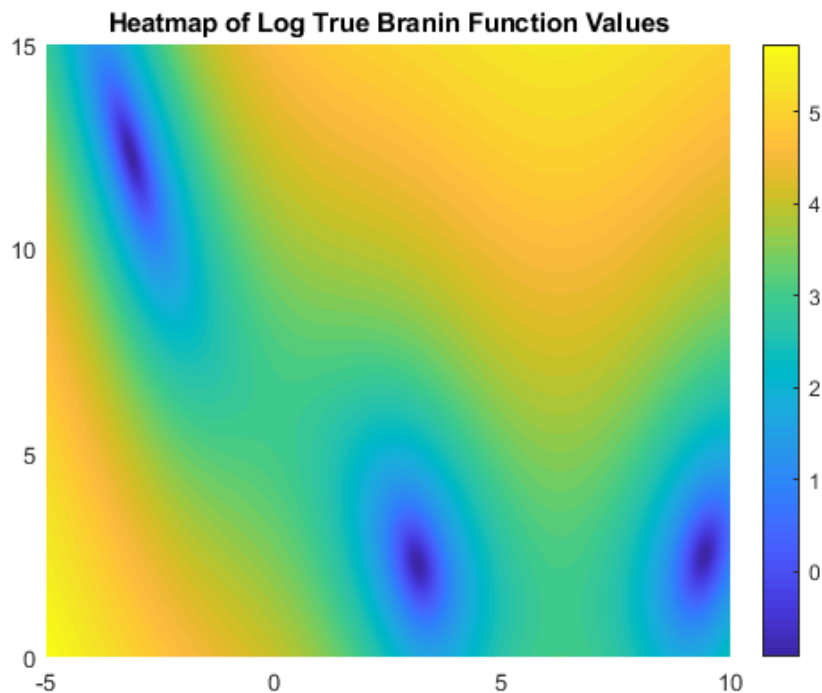


Figure 2: Heatmap of logged Branin Function values with domain $\mathcal{X} = [-5, 10] \times [0, 15]$ with 1000 values per dimension. The plot shows the behavior of the Branin Function relatively more constant throughout the domain compare to the original function behavior.

Now, we plotted kernel density estimates of LDA and SVM benchmarks. It can be seen that the two estimates have relatively similar relative behavior but on significantly different scales.

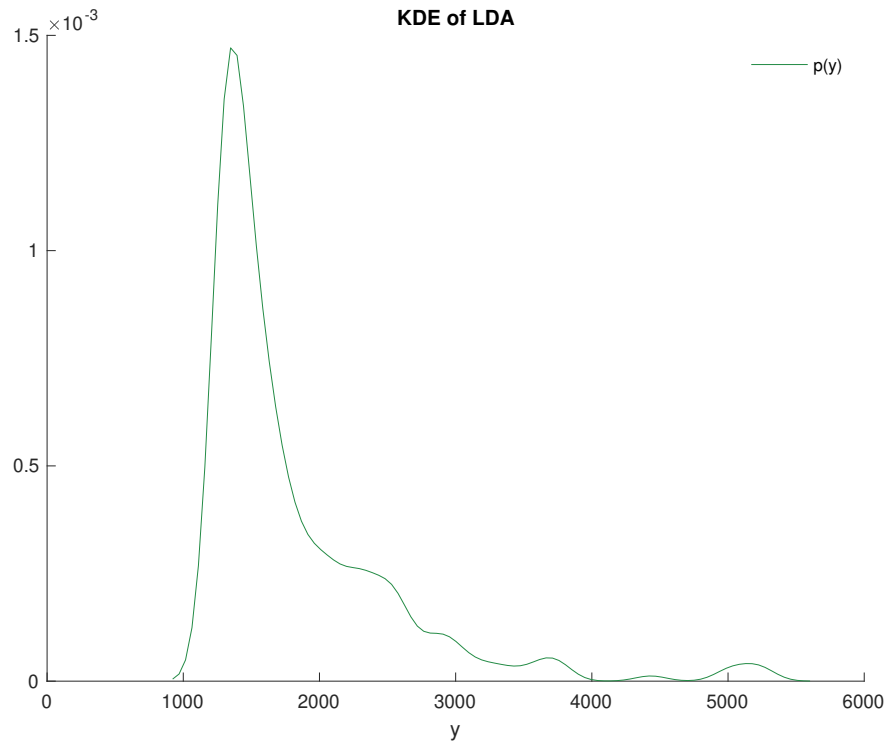


Figure 3: Kernel Density Estimate of the LDA values.

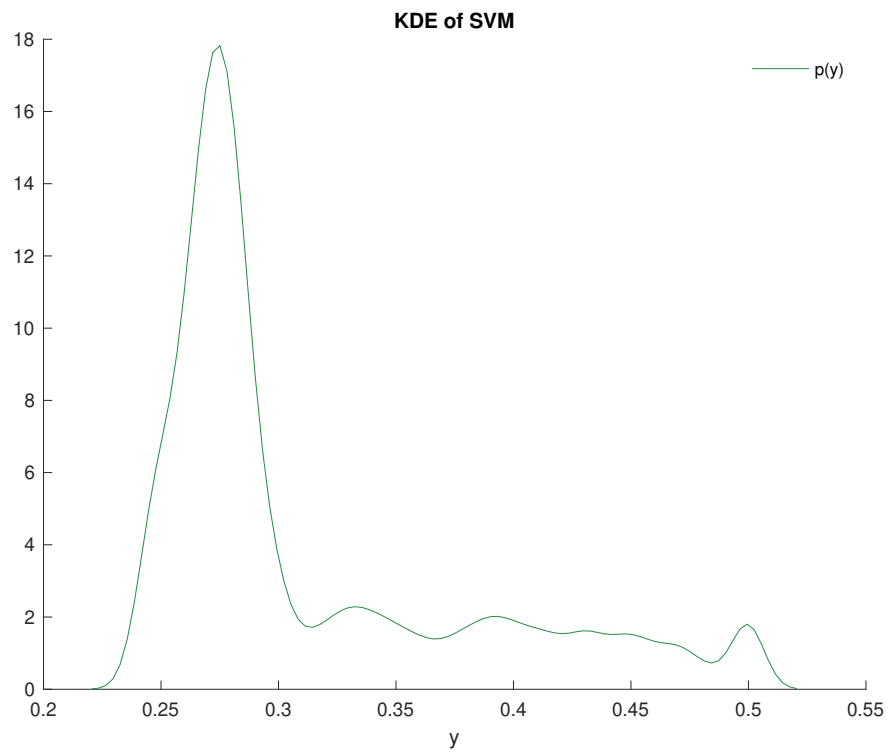


Figure 4: Kernel Density Estimate of the SVM values.

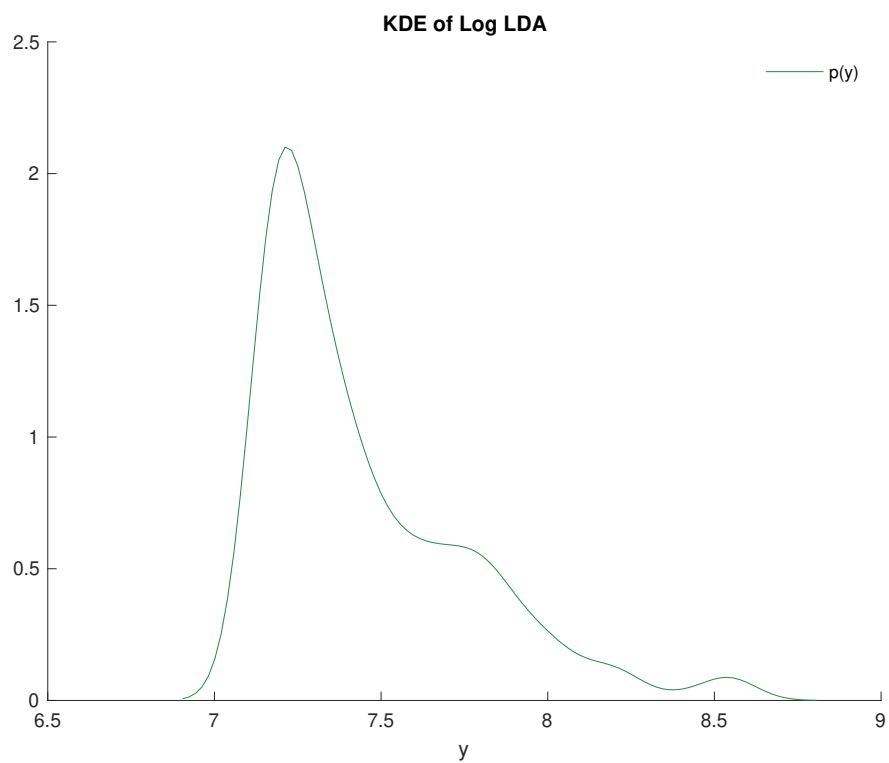


Figure 5:

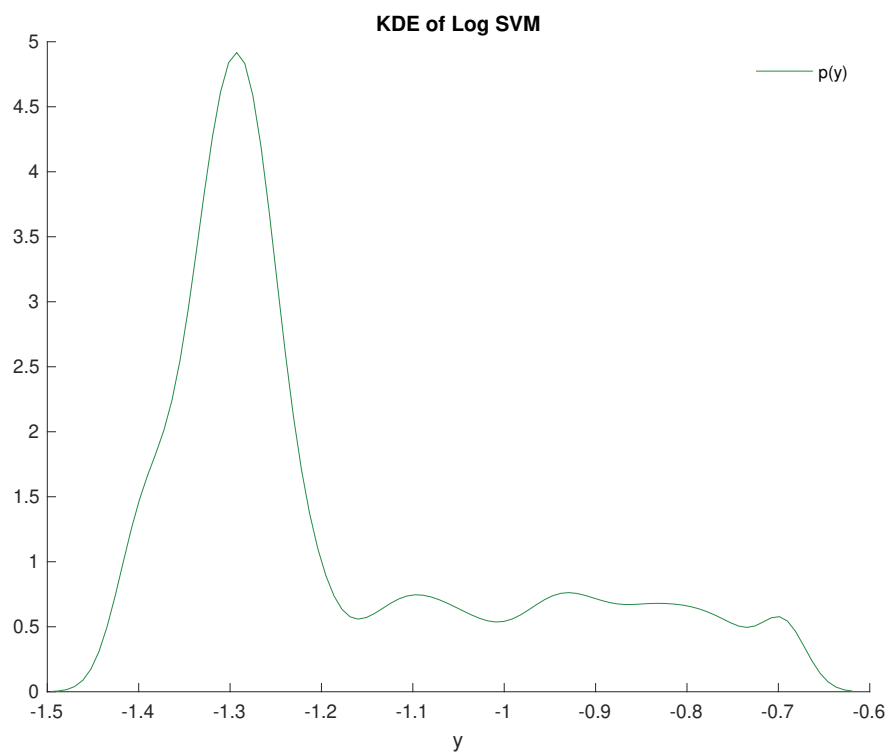


Figure 6:

Model fitting

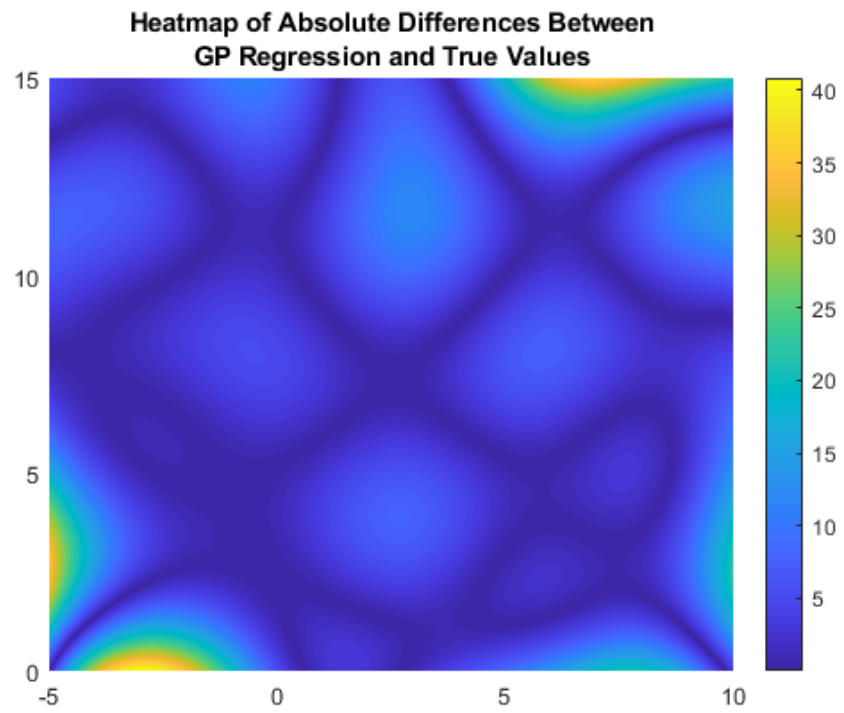


Figure 7:

Q: Compare the predicted values with the true values. Do you see systematic errors?



Figure 8:

Q: Do the values make sense? Does the scale make sense? Does the standard deviation drop to near zero at your data points?

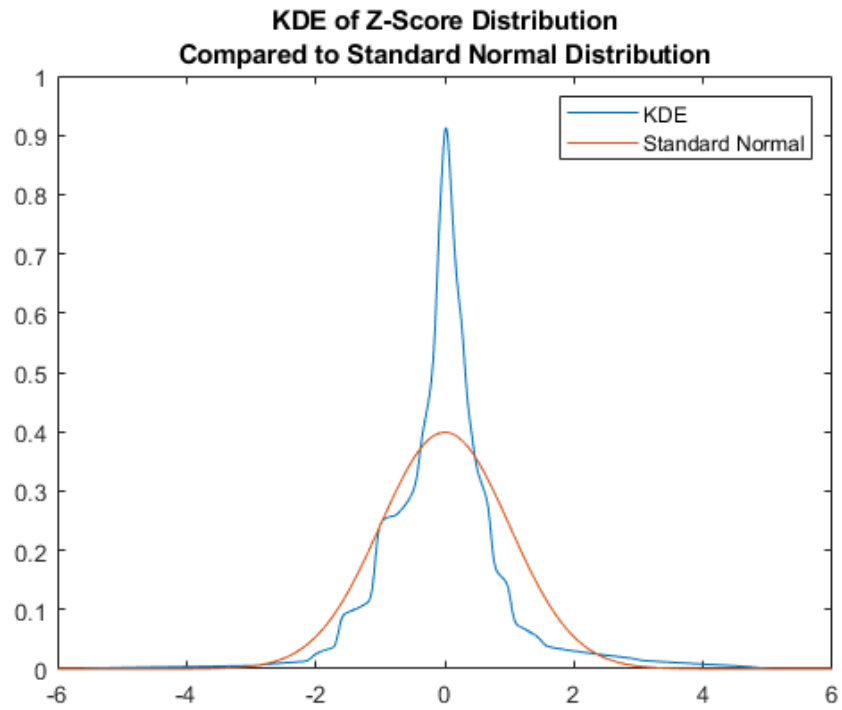


Figure 9:

Based on figure 9, the KDE of Z-score distribution follows approximately standard normal distribution with more concentrated peak in the middle.

Now, we repeated model fitting using a log transformation to the output of the Branin function.

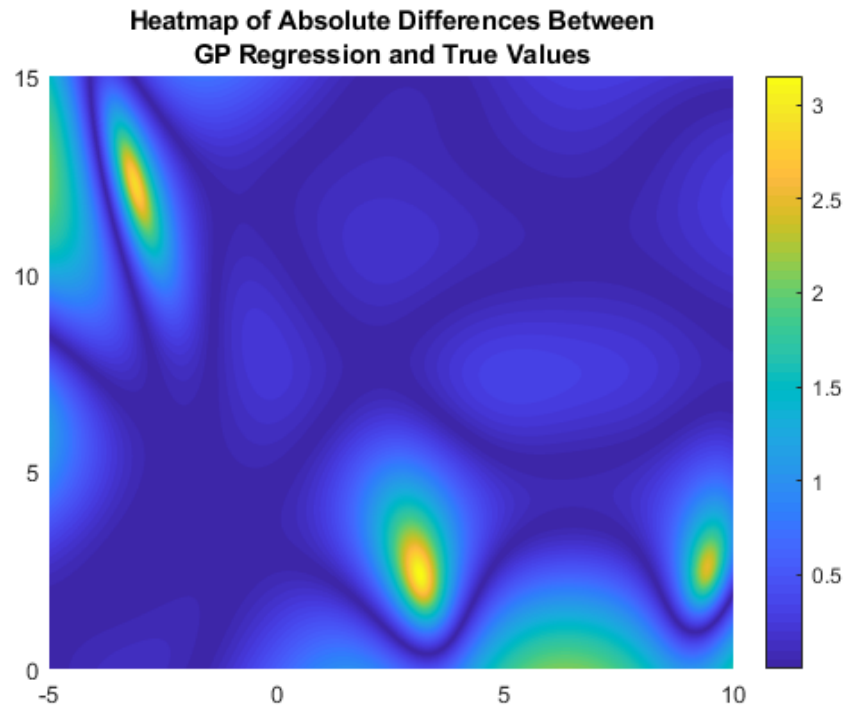


Figure 10:

Q: Compare the predicted values with the true values. Do you see systematic errors?

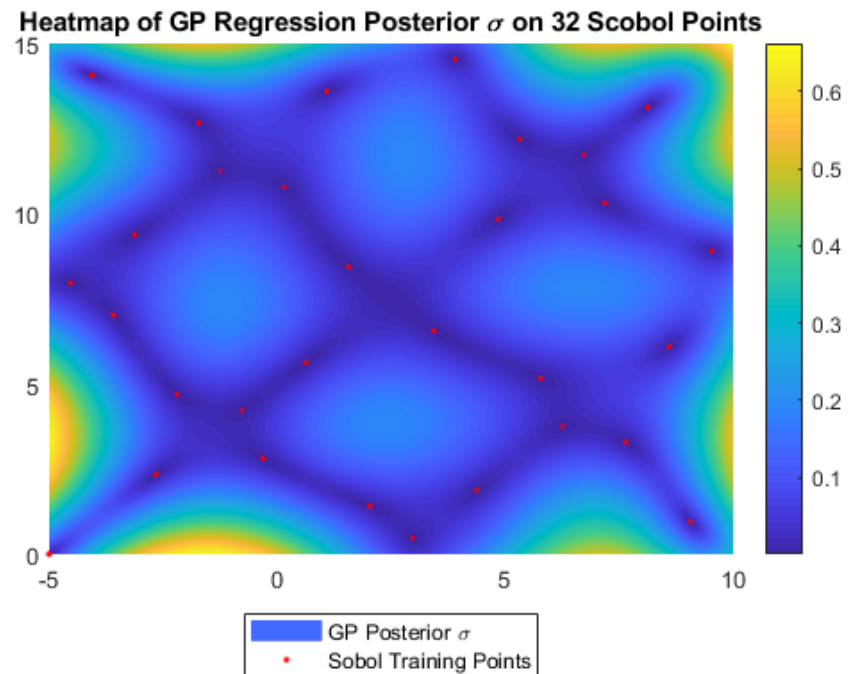


Figure 11:

Q: Do the values make sense? Does the scale make sense? Does the standard deviation drop to near zero at your data points?

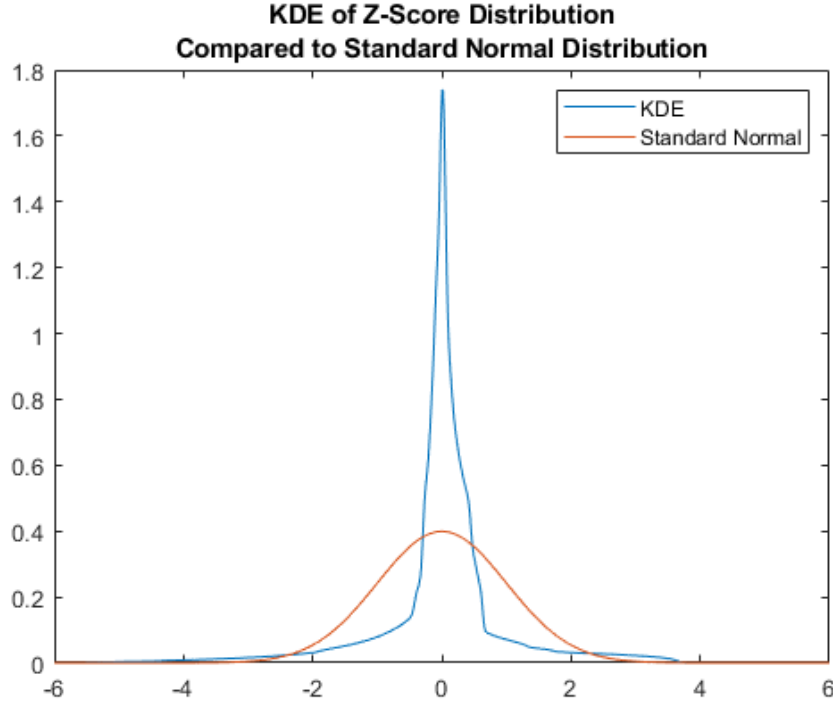


Figure 12:

LOGS: Q: Does the marginal likelihood improve? Does the model appear better calibrated?

Bayesian optimization

The best-fitting models were selected from the previous experiments. Specifically, the Branin Model used a log-transformed dataset, a Constant Mean function, and a Squared Exponential covariance function. Similarly, the LDA Model used a log-transformed dataset, a Constant Mean Function, and a Rational Quadratic covariance Function. Finally, the SVM Model used a normal dataset with a Constant Mean Function and a product of the Rational Quadratic and Squared Exponential functions as its covariance function.

We then used the Expected Improvement (EI) Acquisition Function, defined from the course notes as:

$$a_{ei}(\mathbf{x}) = (f' - \mu(\mathbf{x}))\Phi(f'; \mu(\mathbf{x}), K(\mathbf{x}, \mathbf{x})) + K(\mathbf{x}, \mathbf{x})\mathcal{N}(f'; \mu(\mathbf{x}), K(\mathbf{x}, \mathbf{x})) \quad (0.2)$$

Where $\Phi(\mathbf{x})$ is the Cumulative Probability Density of the Normal Distribution, and f' is the minimum value of the current observations. Note that the EI acquisition function combines both exploitation and exploration by using both the posterior mean and known minimum value, and the posterior standard deviation, respectively.

Using the previously selected 32 points, a GP model was fit using the aforementioned optimized settings and the following heatmaps of the posterior mean and standard deviation of the Branin function were created:

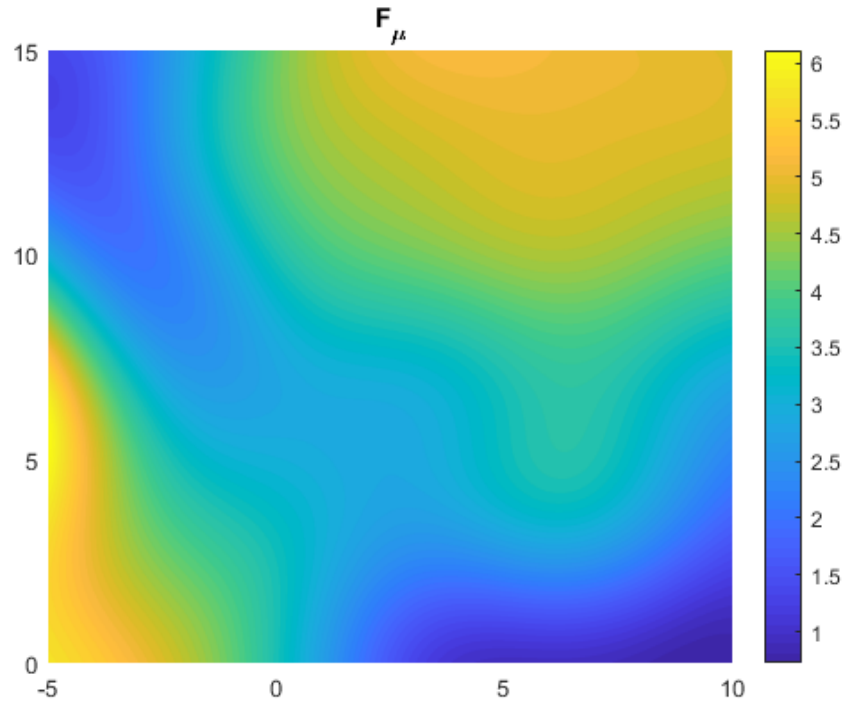


Figure 13: Predictive Posterior Mean of the log Branin Function, calculated using a previously optimized GP model trained on 32 Sobol Sequence points. Warmer colors indicate higher values. Note the predicted minimum areas in dark blue at the top left and bottom right corners of the plot.

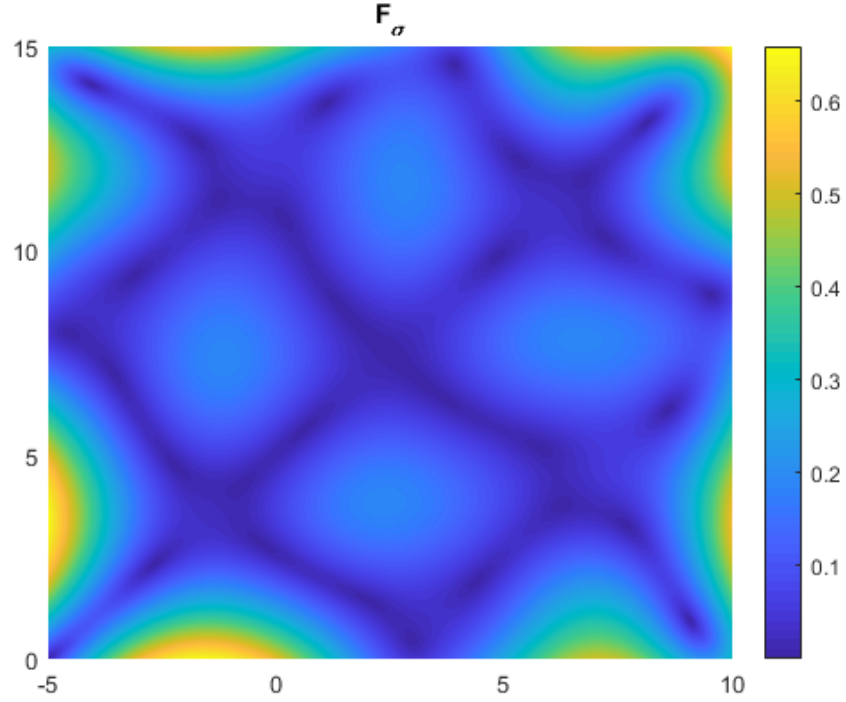


Figure 14: Predictive Posterior Standard Deviation of the log Branin Function, calculated using a previously optimized GP model trained on 32 Sobol Sequence points. Warmer colors indicate higher deviations and imply greater uncertainty in the prediction.

The EI value was then calculated using the posteriors and identified the point $[7.658, 0]$ as the optimal point to test next. A heatmap of the EI distribution is shown below:

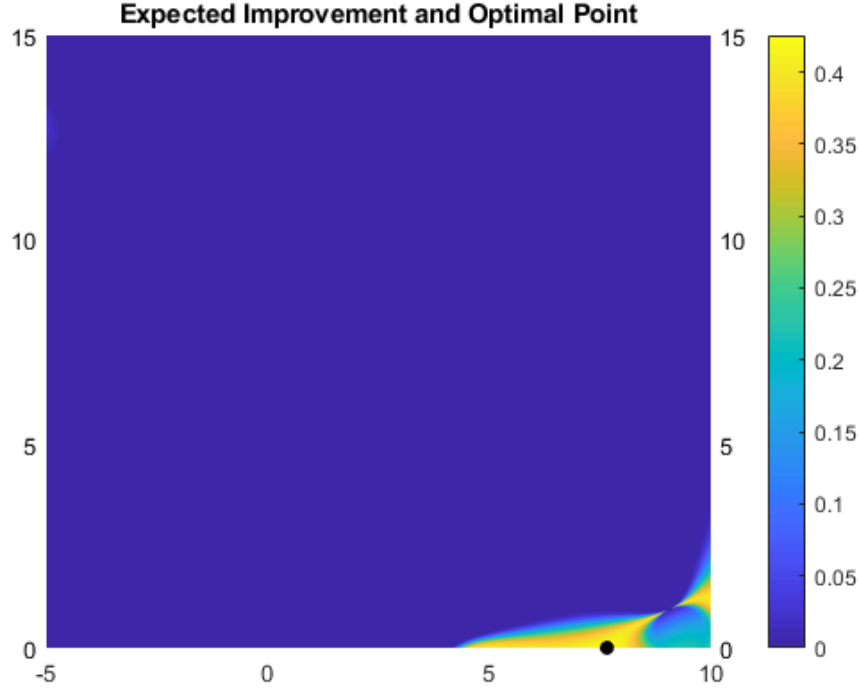


Figure 15: Expected Improvement acquisition values of the optimized GP model trained on 32 Sobol Sequence points. Warmer colors indicate higher expected improvement. The maximum expected improvement is marked as a black point and denotes the optimal location for the next observation.

Analyzing 13, 14, and 15 above, we reason that the proposed optimal testing point is ideal. From the posterior mean, it can be seen that the point $[7.658, 0]$ is within a region of predicted minimal values. Furthermore, from the posterior standard deviation, it can be seen that the point is also within a region of higher uncertainty and therefore reduced predictive confidence. Thus, it is plausible that the EI acquisition function would seek to test this area and point in order to identify a possible global minimum and improve confidence in an area of uncertainty.

The following bayesian active learning experiment was then applied independently to the Branin, LDA and SVM functions:

1. 5 initial observations were randomly selected, constituting the initial dataset \mathcal{D}
2. For the Branin Function only, a dense grid of 250,000 points was generated within the domain of the function
3. A GP model using the respective aforementioned optimized settings was fit to \mathcal{D}
4. A new point x was found using the EI acquisition function and the GP predictive posterior
5. The function value $f(x)$ was calculated and the point $(x, f(x))$ was added to \mathcal{D}
6. Steps 3-5 were repeated 30 times, resulting in a final dataset \mathcal{D} of 35 points

The performance of each of the above experiments was evaluated using the "gap" measure, defined for minimization as:

$$\text{gap} = \frac{f(\text{best found}) - f(\text{best initial})}{f(\text{maximum}) - f(\text{best initial})} \quad (0.3)$$

The gaps for the Branin, LDA, and SVM models were calculated to be 1.0000, 0.7932, and 0.9604, respectively. This implies that EI successfully found a global minimum of the Branin function, but missed the global minimum of the LDA and SVM functions.

The above bayesian active learning experiment was then modified as such:

1. A seed for the random number generator (RNG) was chosen
2. 5 initial observations were randomly selected, constituting the initial dataset \mathcal{D}
3. For the Branin Function only, a dense grid of 250,000 points was generated within the domain of the function for use in calculating the GP predictive posterior
4. A GP model using the respective aforementioned optimized settings was fit to \mathcal{D}
5. A new point x was found using the EI acquisition function and the GP predictive posterior
6. The function value $f(x)$ was calculated and the point $(x, f(x))$ was added to \mathcal{D}
7. Steps 4-6 were repeated 150 times, resulting in a final dataset \mathcal{D} of 155 points
8. A new GP model using the respective aforementioned optimized setting was fit to the original initial dataset \mathcal{D} , now called \mathcal{D}'
9. A new point x was found using the Random Search (RS) acquisition function, which randomly selects a new point
10. The function value $f(x)$ was calculated and the point $(x, f(x))$ was added to \mathcal{D}'
11. Steps 8-10 were repeated 150 times, resulting in a final dataset \mathcal{D}' of 155 points

The above experiment was repeated 20 times with 20 different RNG seeds to create different random initializations for the Branin, LDA, and SVM functions, resulting in a total of 60 experiments. The learning curves of the 60 experiments using only the first 30 new observations for both EI and RS acquisition are shown in the figures below:

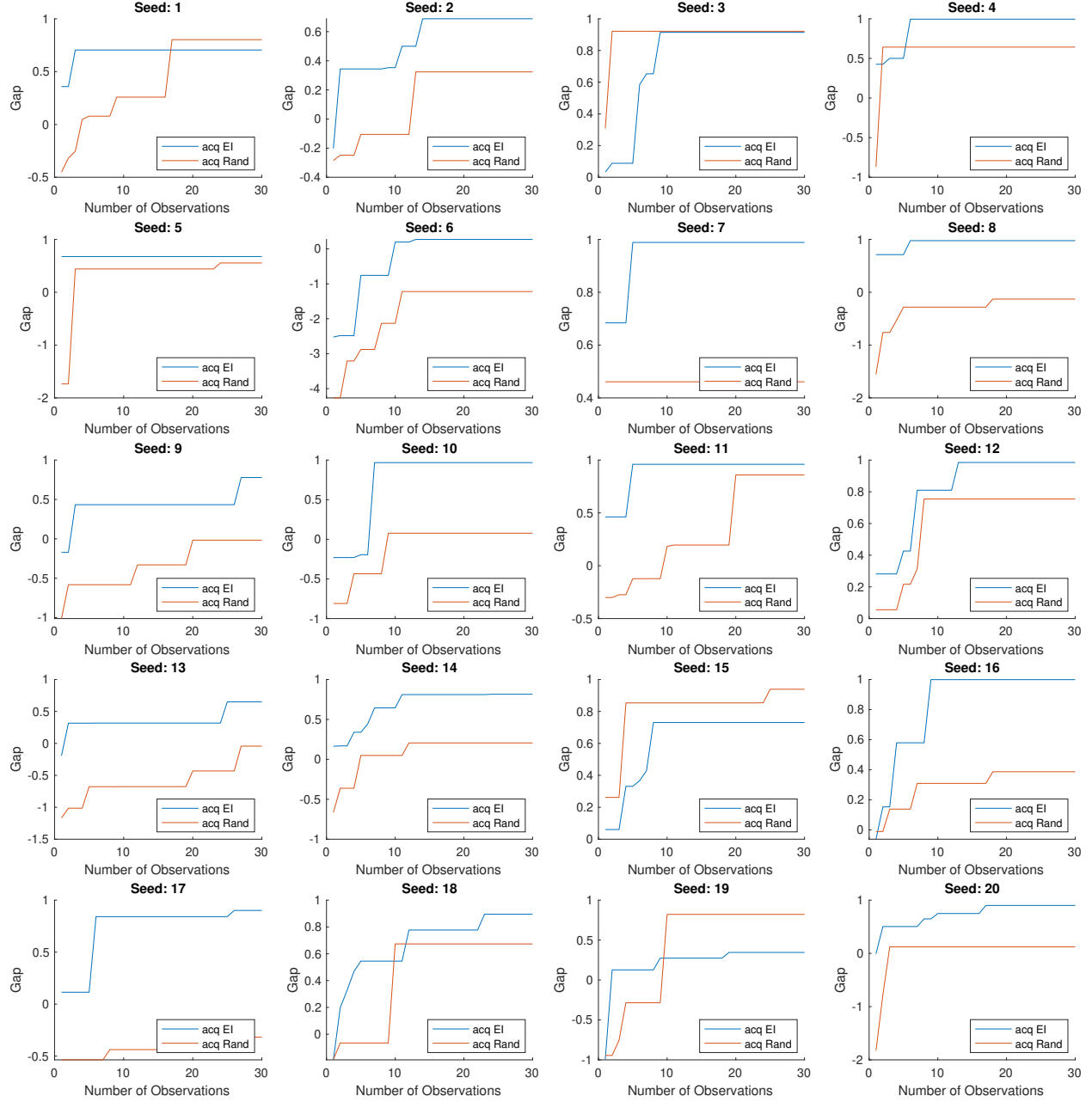


Figure 16: Learning curves for 20 branin function experiments. The EI acquisition function is shown in blue, and the RS acquisition function is shown in orange. Observe that EI consistently outperforms RS in both rate of GAP increase, and final GAP convergence at 30 observations.

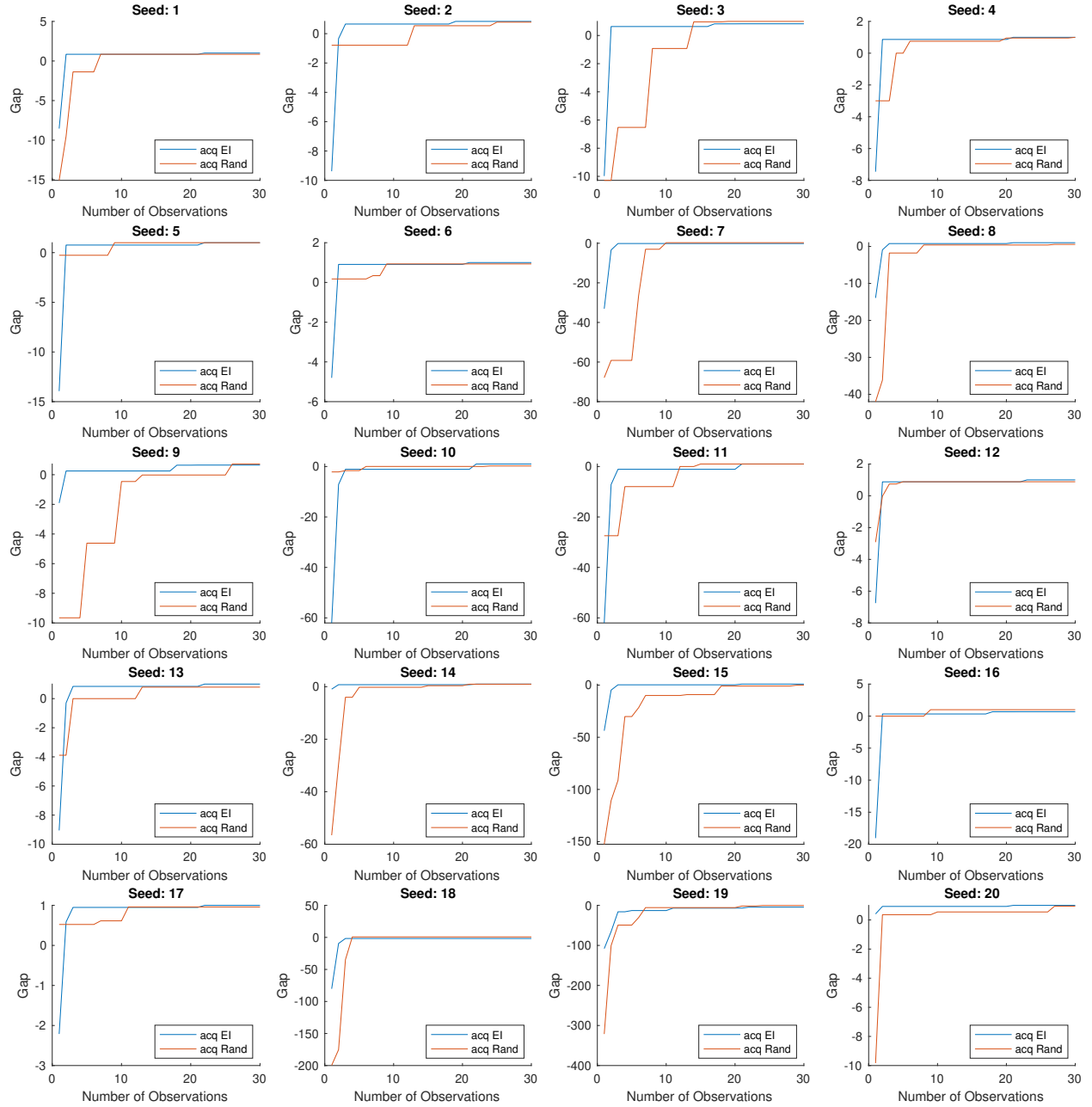


Figure 17: Learning curves for 20 LDA function experiments. The EI acquisition function is shown in blue, and the RS acquisition function is shown in orange. Note that EI and RS both consistently converge towards the same GAP value at 30 observations

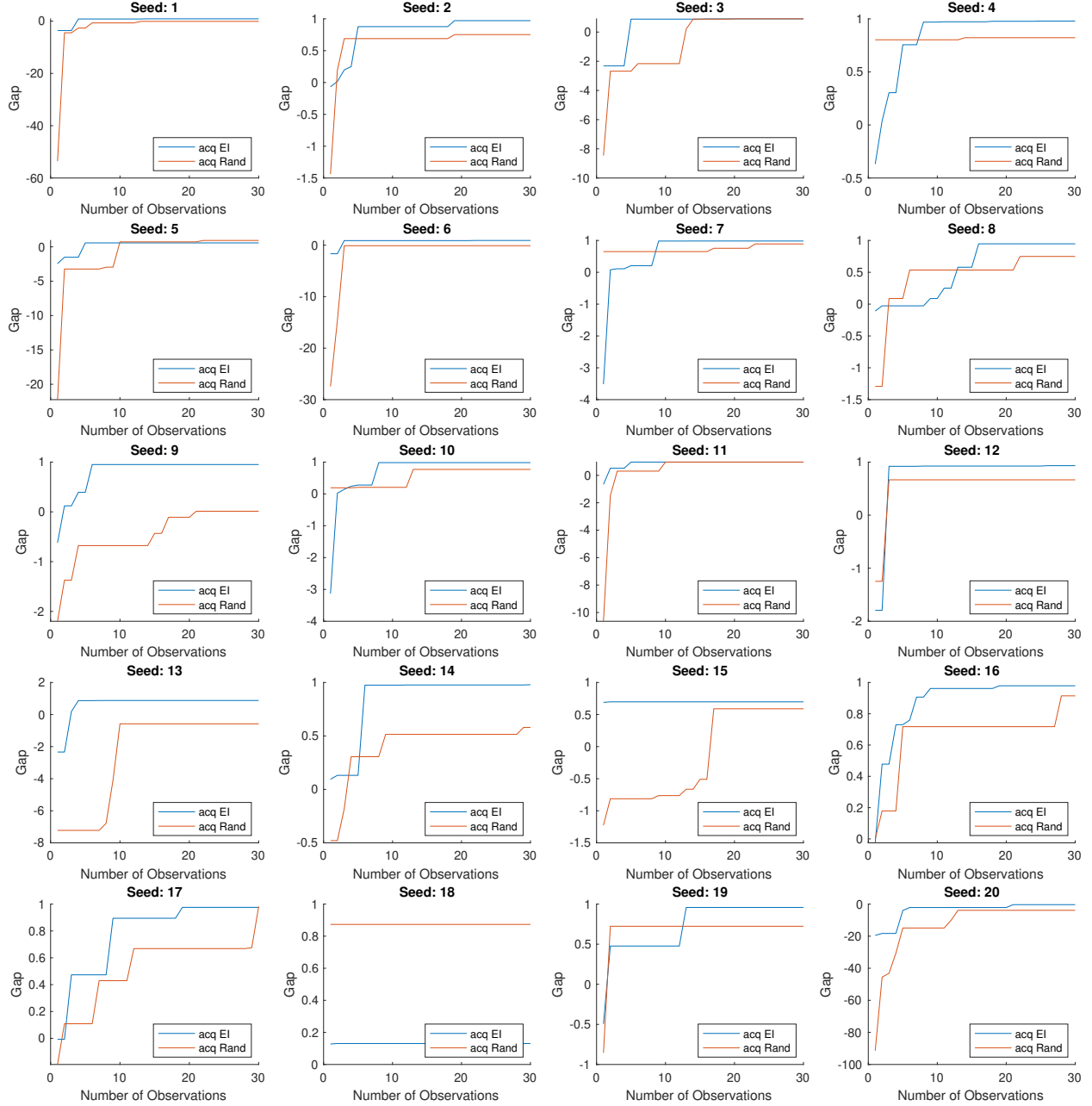


Figure 18: Learning curves for 20 SVM function experiments. The EI acquisition function is shown in blue, and the RS acquisition function is shown in orange. It can be seen that EI performs at least as well as RS in the majority of experiments, but can find and get stuck in a local minima, as seen in Seed 18.

From figures 16, and 18 above, it can be seen that EI significantly outperforms RS on both the Branin Function and the SVM Function. Note that in these experiments, EI often converges to its final GAP value in significantly fewer observations than RS does, and that its final GAP values at 30 observations are generally higher than those of RS. This behavior was expected, as EI combines both exploitation and exploration together to identify locations with higher expected benefit, whereas RS only applies (random) exploration. The addition of exploitation and non-random exploration in EI allows it to actually navigate towards optima, whereas RS can only approach

optima through luck. However, there are cases where EI and RS have similar performance: from figure 17, it can be seen that the two acquisition functions perform similarly to each other in the majority of experiments at 30 observations. We believe that EI's lack of performance on the LDA function can be attributed to a large amount of local minima; EI can often become stuck in local minima, thereby decreasing its performance, whereas RS is purely exploratory and therefore unhindered by the presence of any local optima.

The difference in performance between EI and RS can be further seen when one considers their mean gaps at 30, 60, 90, 120 and 150 observations, as shown in the table below: In all cases, it is evident that EI outperforms RS on all three functions, including LDA, at 150 observations. Additionally, EI requires less observations to achieve higher gap values, implying a significantly faster learning rate than RS. We also note that EI likely spends considerable time navigating local minima in the LDA function, as evident by its noticeably worse performance at 30 observations when compared to that of RS. However, EI for LDA rapidly improves following this local-minima exploration and quickly outperforms RS. It is also important to note that on average, no method was able to find the global minima. This result was expected for RS, which would need to be extraordinarily lucky to find the global minima, but was somewhat surprising for EI. It appears that in the limit, EI may become entrenched in local minima and thus converges towards a suboptimal result.

We more rigorously compare the performance of EI and RS using a paired t-test. The null hypothesis of the test

Bonus

For the bonus section, we implemented two more acquisition functions: Lower Confidence Bound (LCB) and Max Variance. We then compared their performances with EI when used on their own and with two heuristics.

First, we implemented a wrapper for acquisition functions that selects a point at random with probability $p = 0.1$; we dubbed this heuristic "random restarts" (RR). We also tried each acquisition function while minimizing the model's hyperparameters after each iteration; we called this "online optimization" (OO).