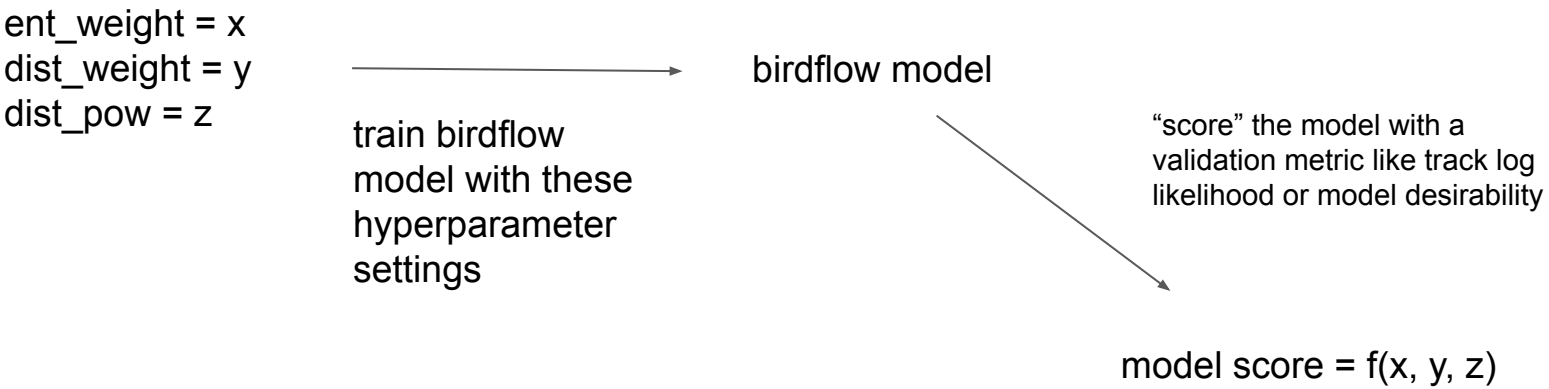
Bayesian Optimization for Birdflow Hyperparameter Tuning

Jacob Epstein

Problem Formulation: BirdFlow Hyperparameter Tuning

Model Score Function: f(x, y, z)



Problem: Maximize the model score function!

Potential Solutions

- Grid Search
  - evaluate on predetermined grid
- Random Search
  - Select hyperparameters randomly within the optimization bounds
- Bayesian Optimization (BO)!
  - selects points to evaluate one-by-one using information from previous evaluations

Motivation for Bayesian Optimization Experiments

Compare the efficiency and hyperparameter quality of **Bayesian Optimization** to **Grid Search** and **Random Search.**

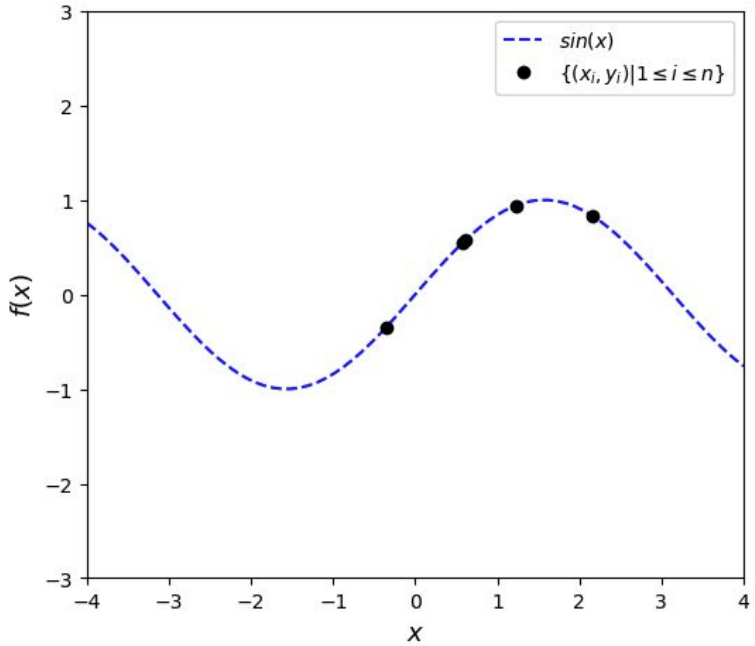
But before we begin discussing experiment methodology and results…

What is Bayesian Optimization?

- A technique for maximizing a **black box function**
- Does not use derivative information
- Iteratively selects points to evaluate by fitting a **Gaussian Process** and optimizing an **acquisition function**

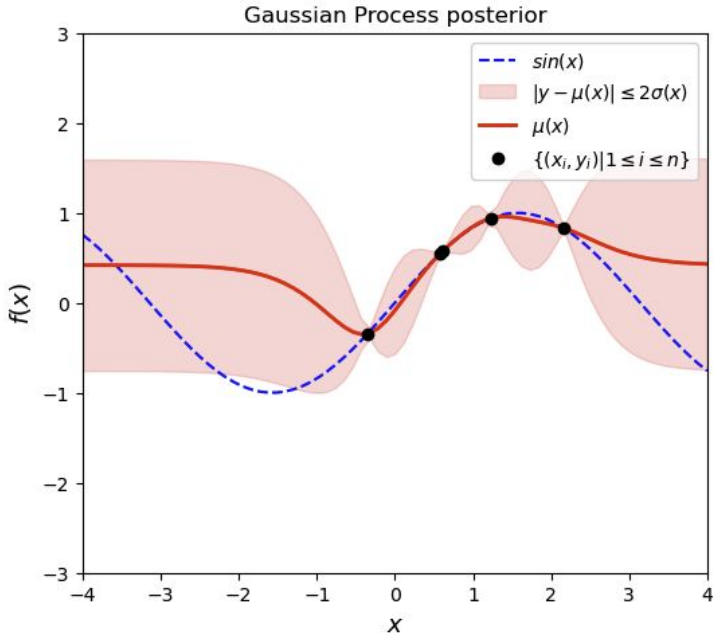
Trace through a Single Iteration of BO: Initial Points

Given that we have evaluated our black box function **f** at $\mathbf{x}_1 \ldots \mathbf{x}_n$, choose an **x** to evaluate at next.



Next: fit a gaussian process to these points. But first…

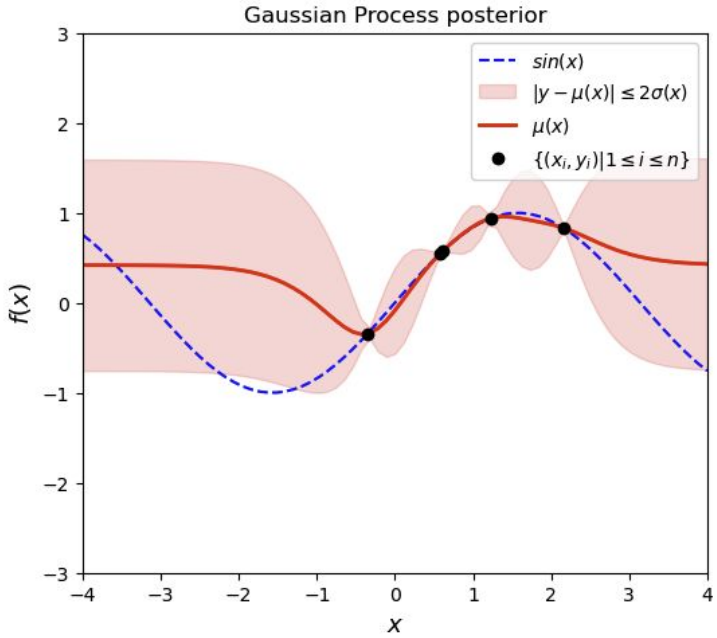
What is a Gaussian Process?



A **Gaussian Process** is a distribution over functions such that every subset of function values comes from a multivariate normal distribution defined by a mean function and a kernel function

$f \sim GP(m, \kappa)$
m is the **mean function** (0 for our purposes)
$\kappa(x_i, x_j) = \exp\left(-\frac{|x_i - x_j|^2}{2\sigma^2}\right)$ is the **kernel** (this is the RBF kernel)
Let $X = \{x_1, x_2, ..., x_n\}$. Want to know about $y = \{f(x_1), ..., f(x_n)\}$
$y \sim \mathcal{N}(\mu, \Sigma)$ with mean vector μ and covariance matrix Σ given by
$\mu = m(X) = 0$
$\Sigma_{i,j} = \kappa(x_i, x_j)$.

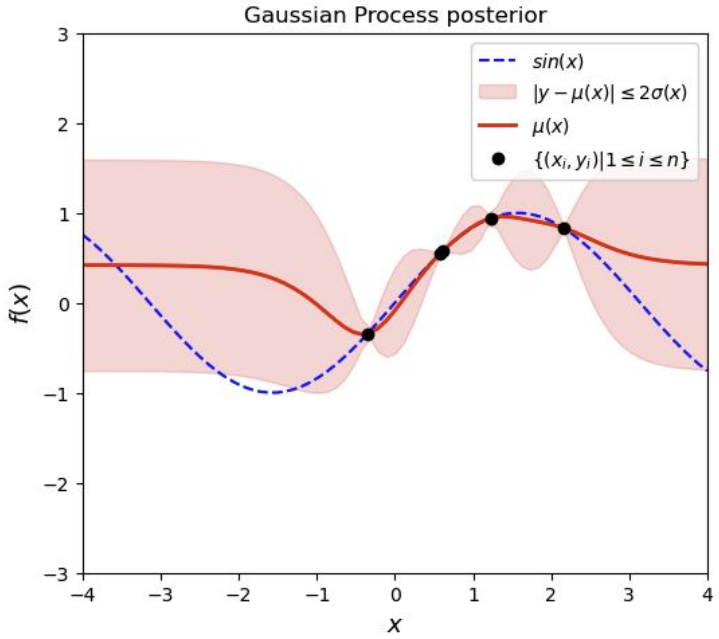
Fitting a Gaussian Process to the Evaluated Points



Bottom Line:
Choose kernel hyperparameters that maximize the likelihood of evaluated points.

Assume $\{f(x_1), f(x_2), ..., f(x_n)\} = y \sim \mathcal{N}(0, \Sigma)$
Pick kernel paramaters (σ in this case) that maximize the likelihood $p(y|X, \sigma)$

Computing the Gaussian Process Posterior



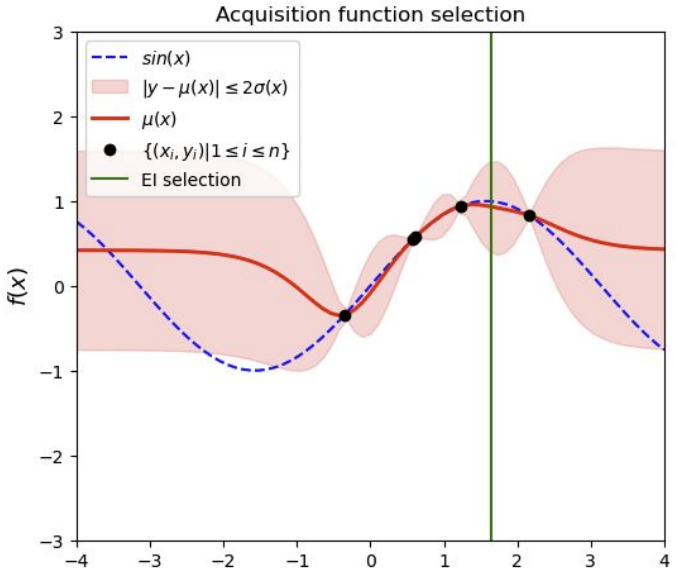
Given arbitrary x, we want to estimate $f(x)$.
Get a posterior probability distribution for $f(x)$
Idea: take a conditional of $p(f(x), f(x_1), ..., f(x_n))$
Namely: $p(f(x)|f(x_1), ..., f(x_n))$
Math tells us that this conditional is a normal distribution, and can be computed with our fitted Gaussian Process' kernel!

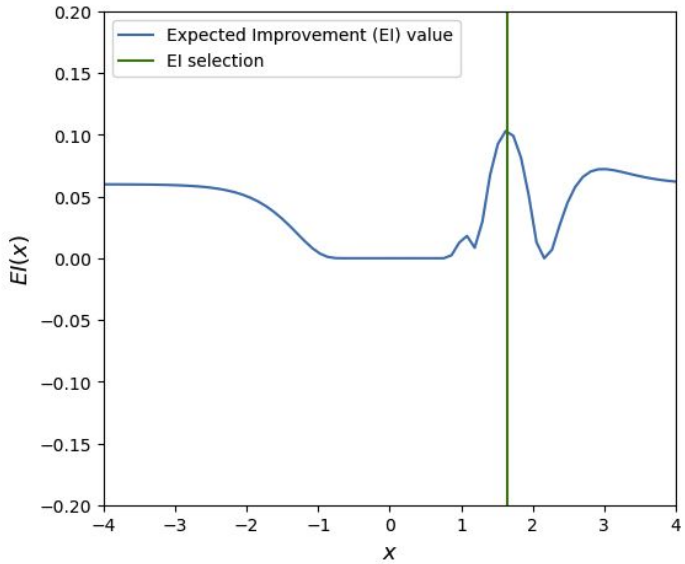
Bottom Line: We can compute a good estimate for the mean μ(x) and the standard deviation σ(x) of f(x) using our fitted Gaussian Process!

Selecting a New Point to Evaluate!

Then, select the next point to evaluate at by optimizing an *acquisition function* of the mean and standard deviation of f(**x**).

$$\mathrm{EI}(x) = \mathbb{E}\left[\max(y - f^*, 0) \mid y \sim \mathcal{N}(\mu(x), \sigma^2(x))\right]$$





Bayesian Optimization Pseudocode

$\kappa \leftarrow$ gaussian process kernel function
$f \leftarrow$ hyperparameter scoring function
$N \leftarrow$ number of iterations
Observe f at n_0 points selected uniformly at random in hyperparameter bounds B
$n \leftarrow n_0$
while $n \leq N$ **do**
    Select hyperparameters for κ that maximize the exact marginal log likelihood of data points observed so far
    Compute posterior
    $x' \leftarrow$ maximizer of the acquisition function over $x \in B$ (calculated w.r.t the current posterior)
    Observe $y' = f(x')$
    $n \leftarrow n + 1$
end
Return the point x with the largest evaluated $f(x)$

Experiment Methodology

Bayesian Optimization Algorithms Used:

- Matérn kernel, Upper Confidence Bound (UCB) acquisition function
- Matérn kernel, Log-Expected Improvement (Log EI) acquisition function
- Matérn kernel, Log-Probability of Improvement (Log PI) acquisition function

- Use American Woodcock s&t / track data, 48 km grid cell resolution.
- Grid searches of 64 points
- 20 trials of 100-point random searches (points selected from a 300-point pool of pre-selected hyperparameters), plot median, and quantiles 0.05 and 0.95 over all trials

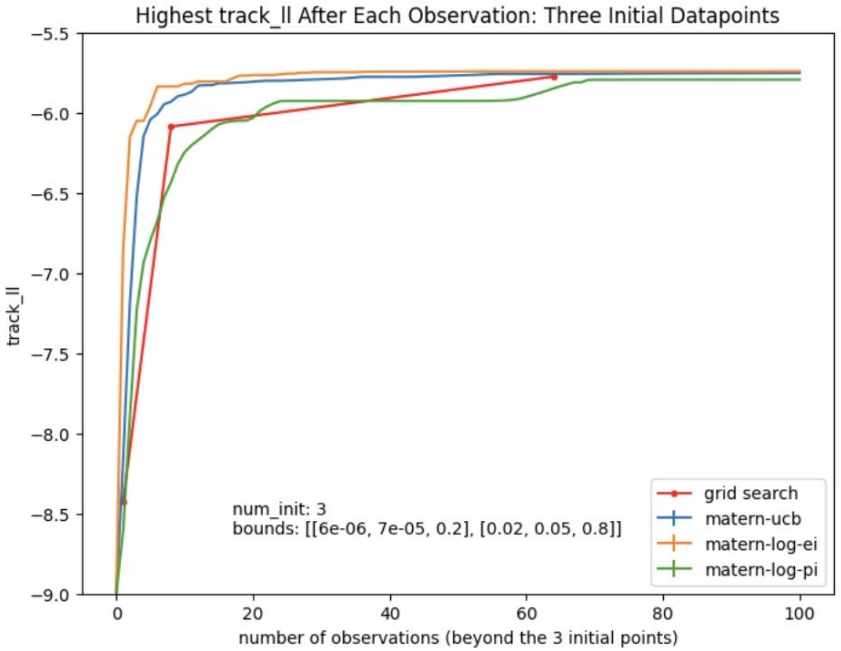
Things that vary across experiments

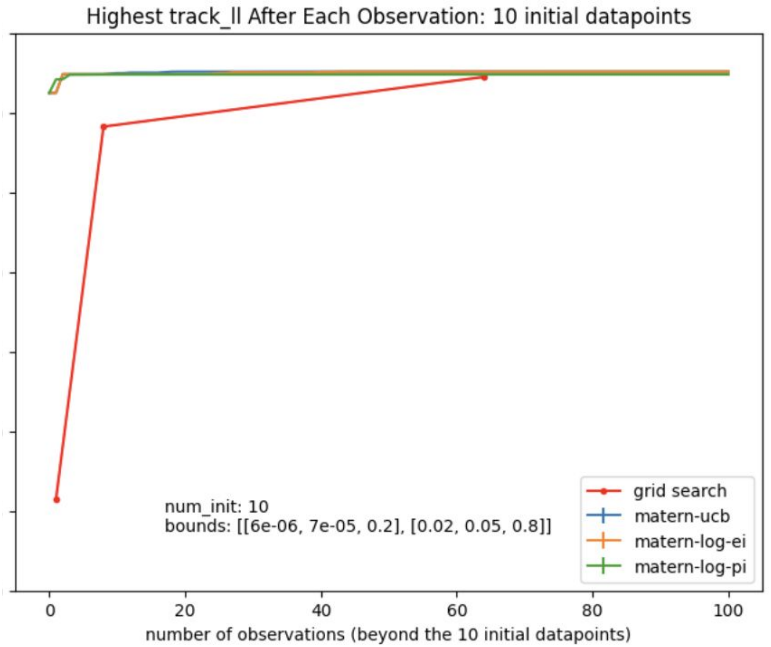
- The hyperparameter bounds
- Number of hyperparameters to search over
- Number of initial randomly selected data points
- Hyperparameter scoring function!
    - Track Log Likelihood
    - Model Desirability (straightness & end traverse correlation)

Summary of Experiments

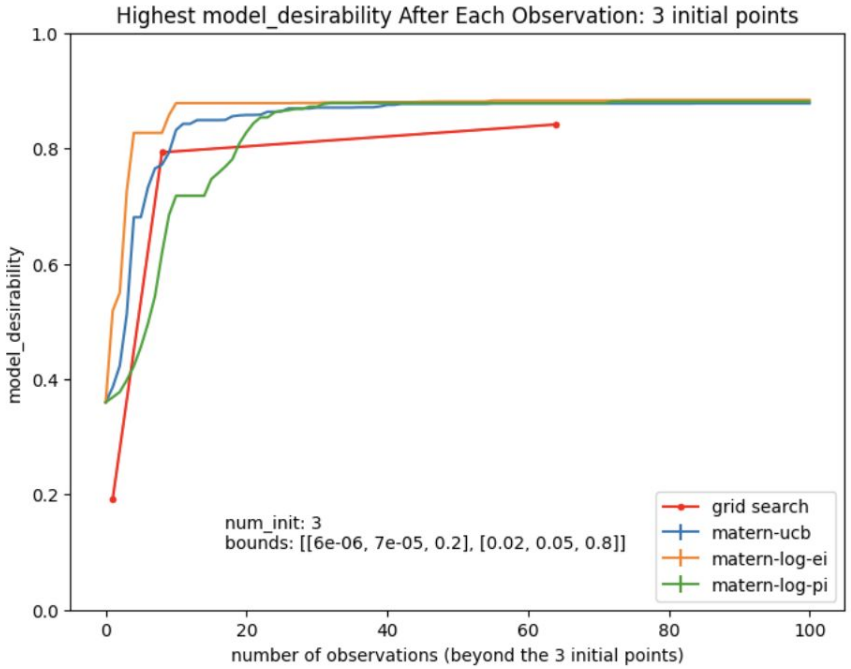
- Track Log Likelihood score function
  - 1, 3, 5, 10 initial data points
  - Wider bounds
  - 2-D optimization space
- Model desirability score function
  - 1, 3, 5, 10 initial data points
  - Wider bounds
  - 2-D optimization space

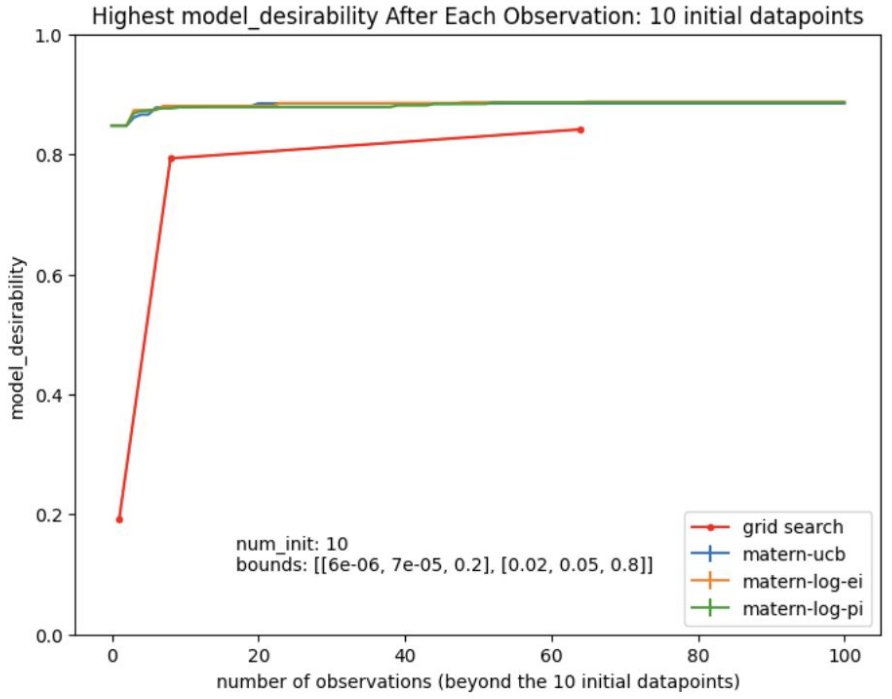
More Initial Points => Higher Quality of Starting Location (track log likelihood)



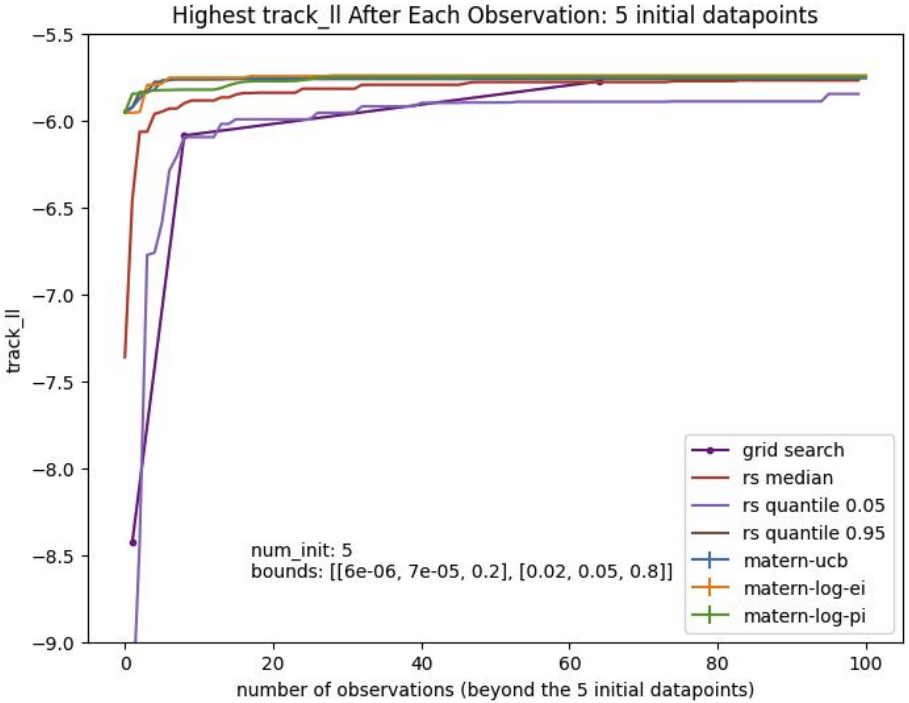


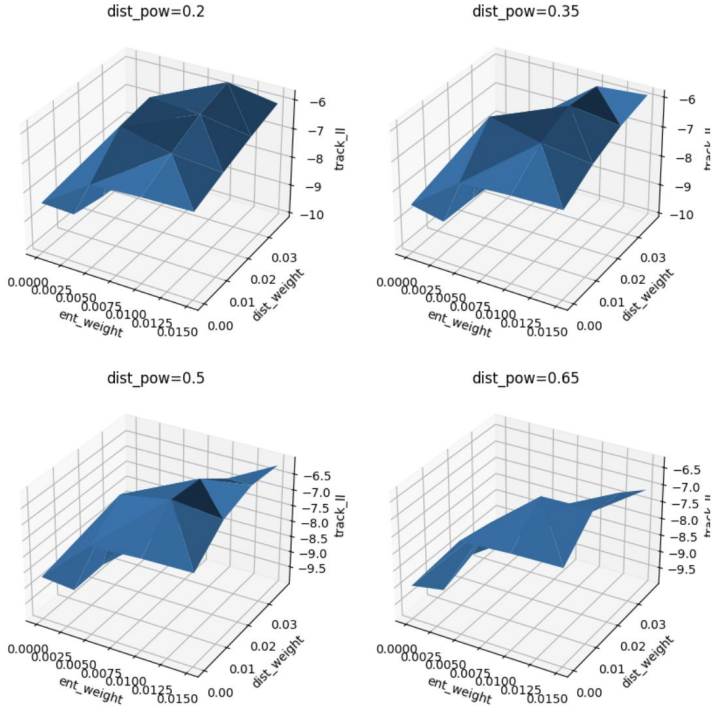
More Initial Points => Higher Quality of Starting Location (model desirability)





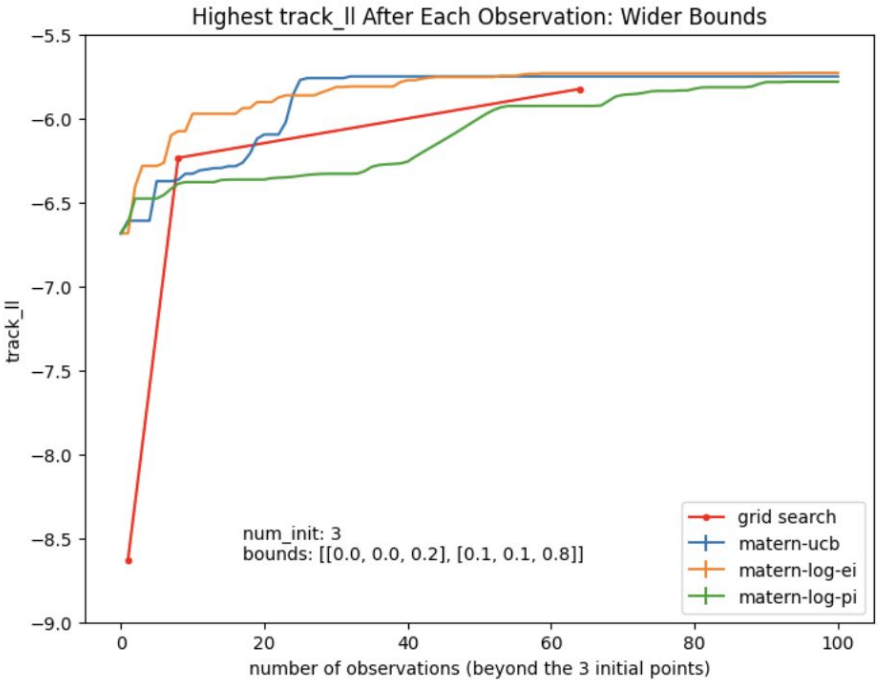
Effectiveness of Random Search and Grid Search (track log likelihood black box function)

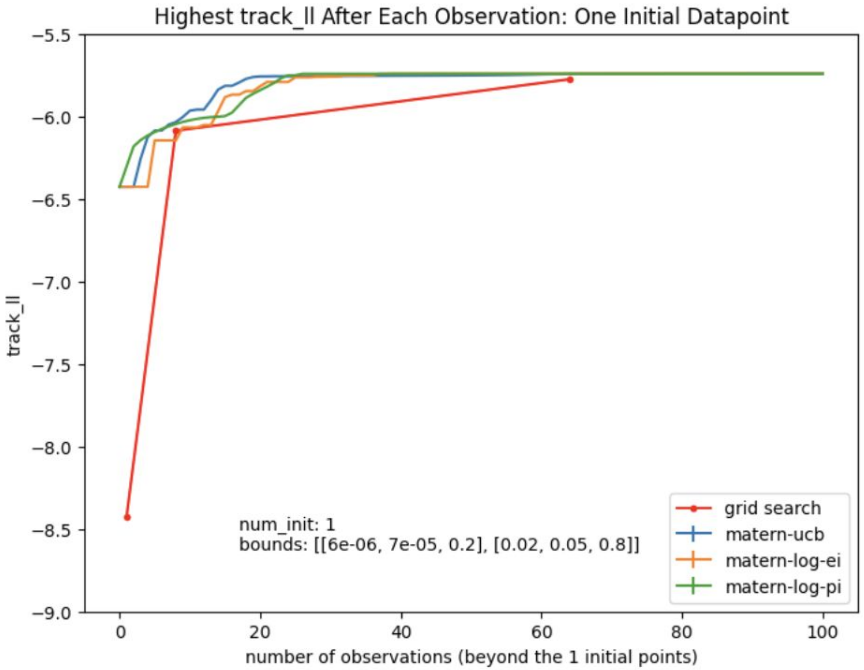




Both Grid and Random Search seem to be effective on the flat track log likelihood hyperparameter score function

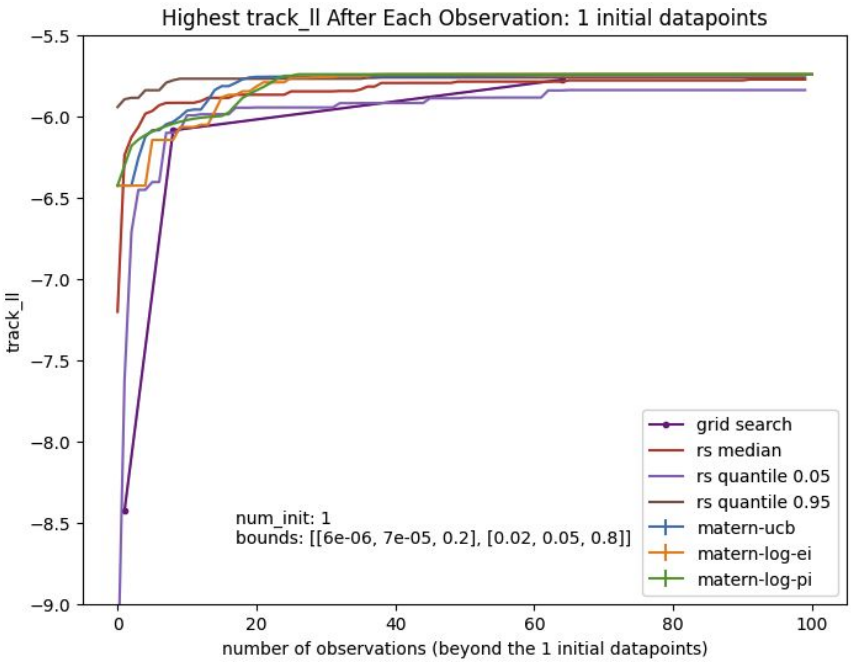
Increased Efficiency of BO over Grid Search (track log likelihood)

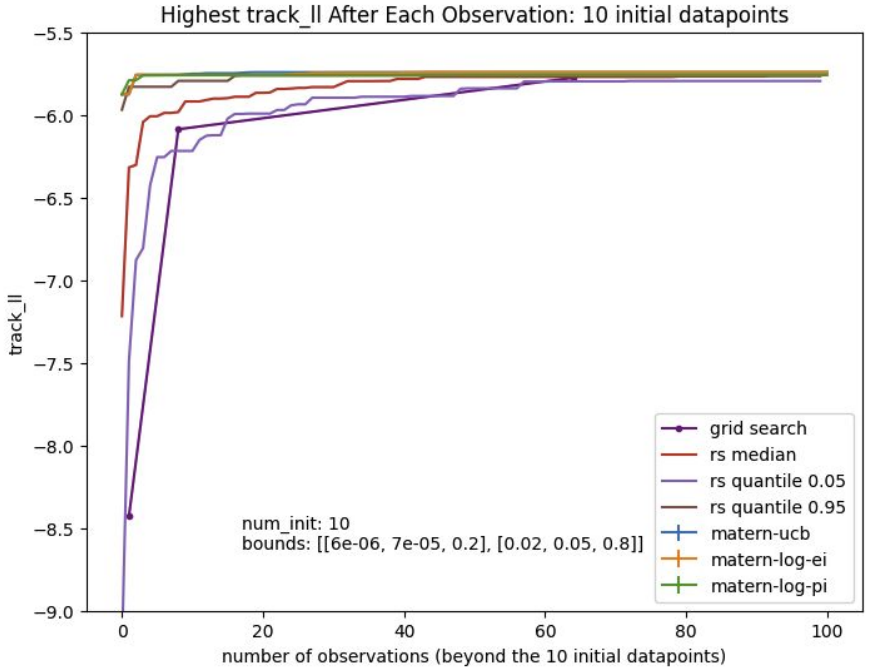




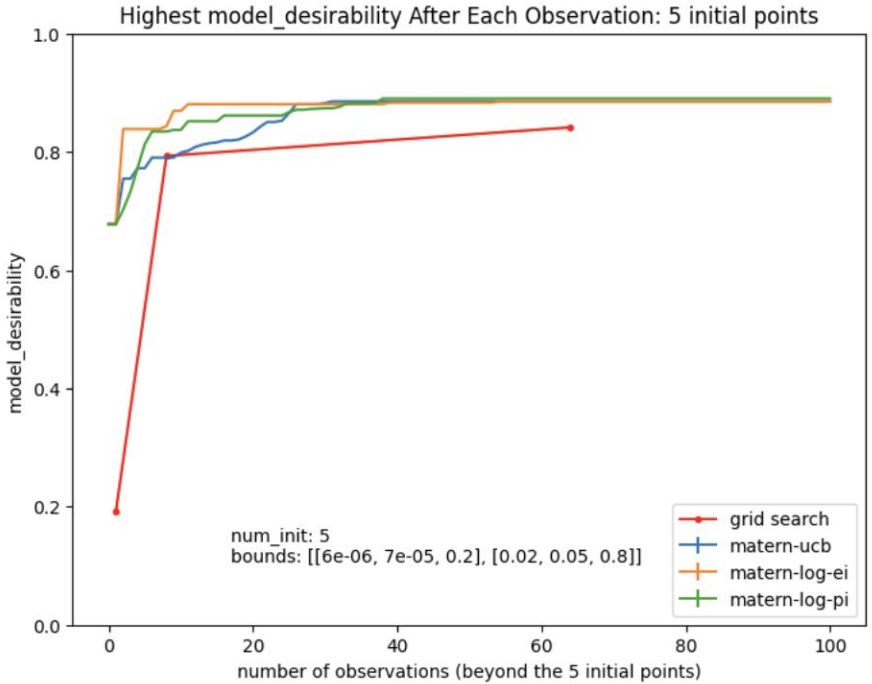
In the majority of Track Log Likelihood Experiments, BO finds the optimum before grid search does.

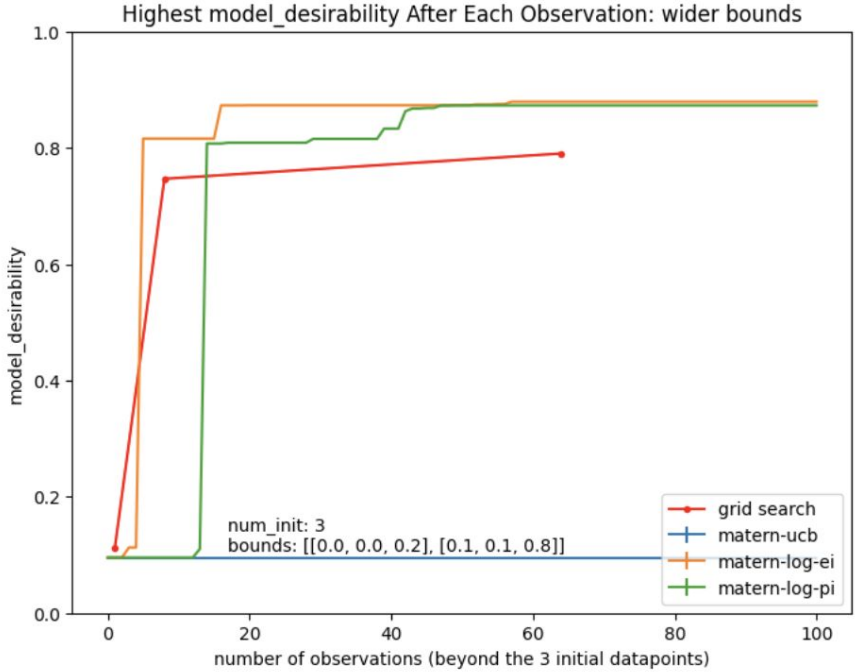
Increased Efficiency of BO over Random Search (track log likelihood)



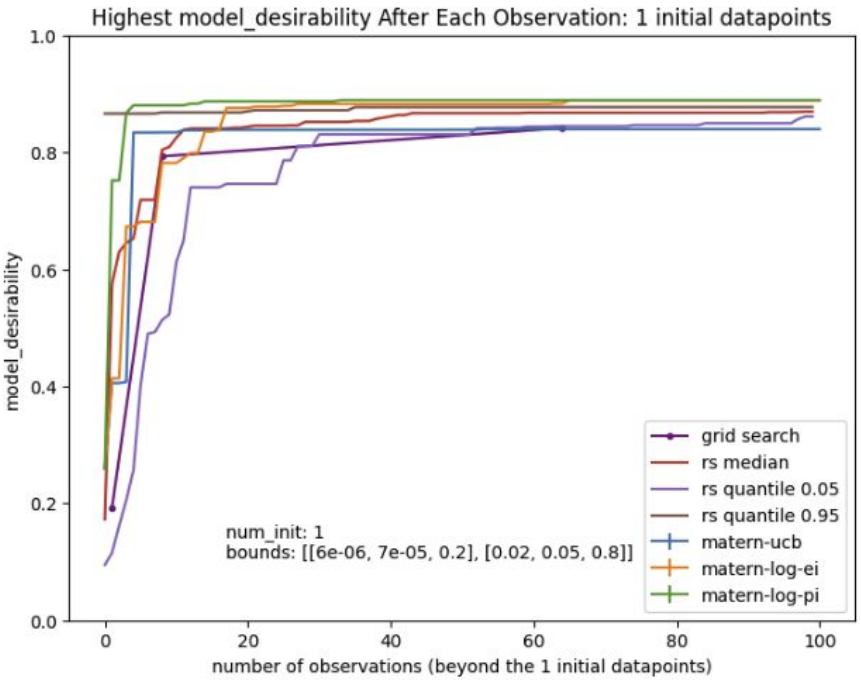


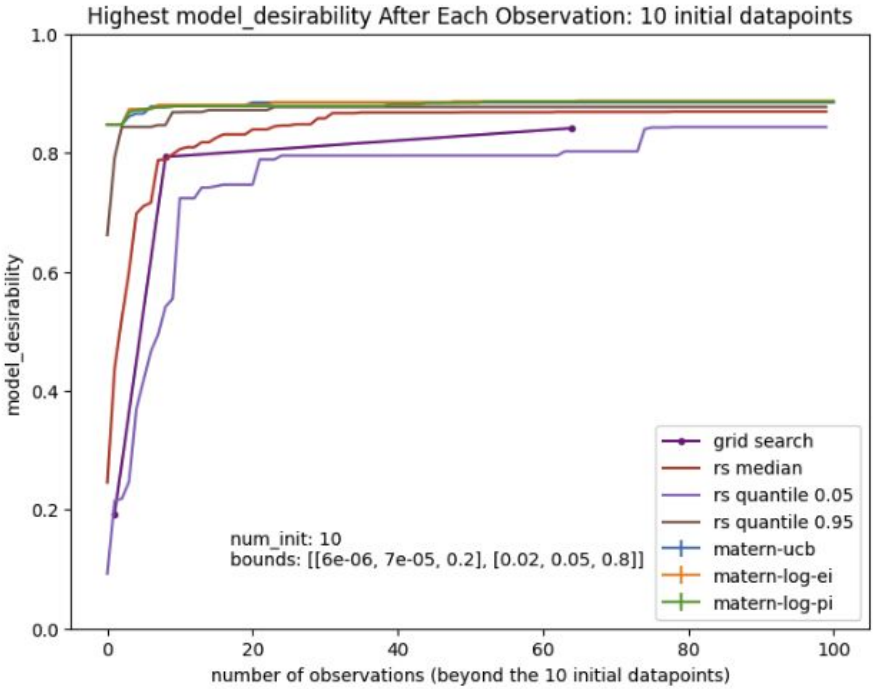
Increased Efficiency of BO over Grid Search (model desirability)





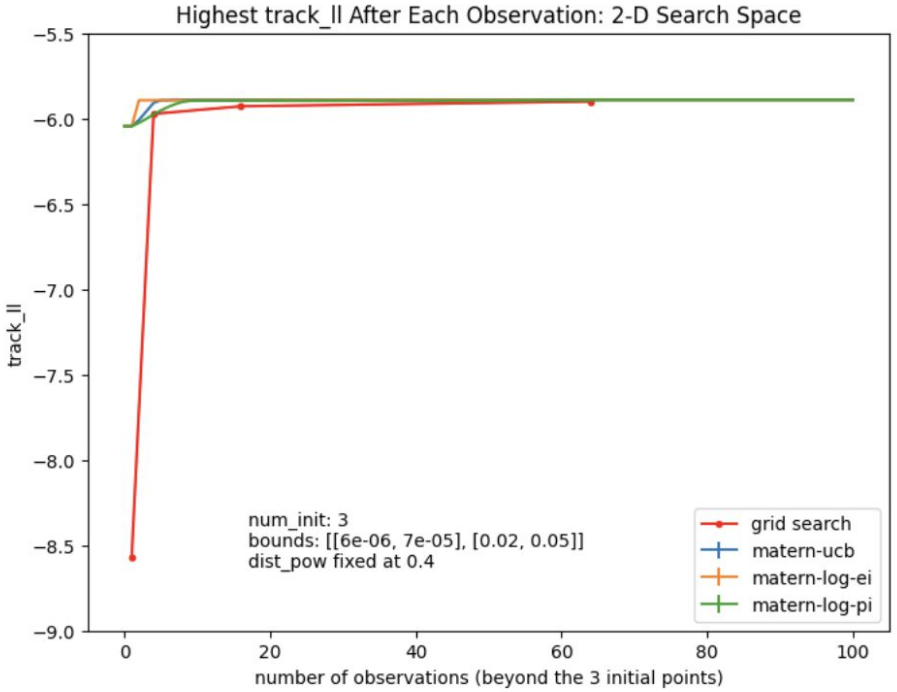
Increased Efficiency of BO over Random Search (model desirability)

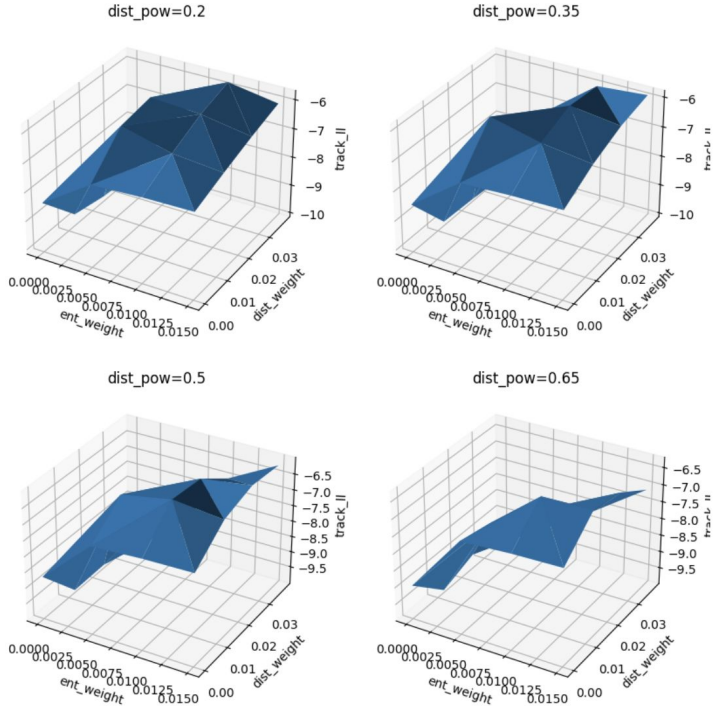




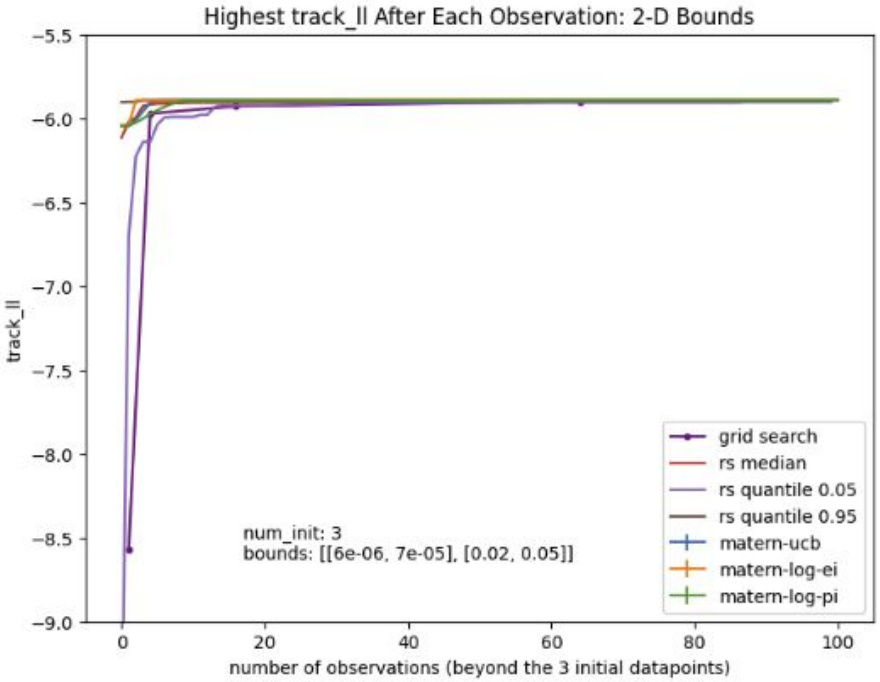
Similarity in Hyperparameter Quality of BO and Grid Search when Optimizing over Track Log Likelihood

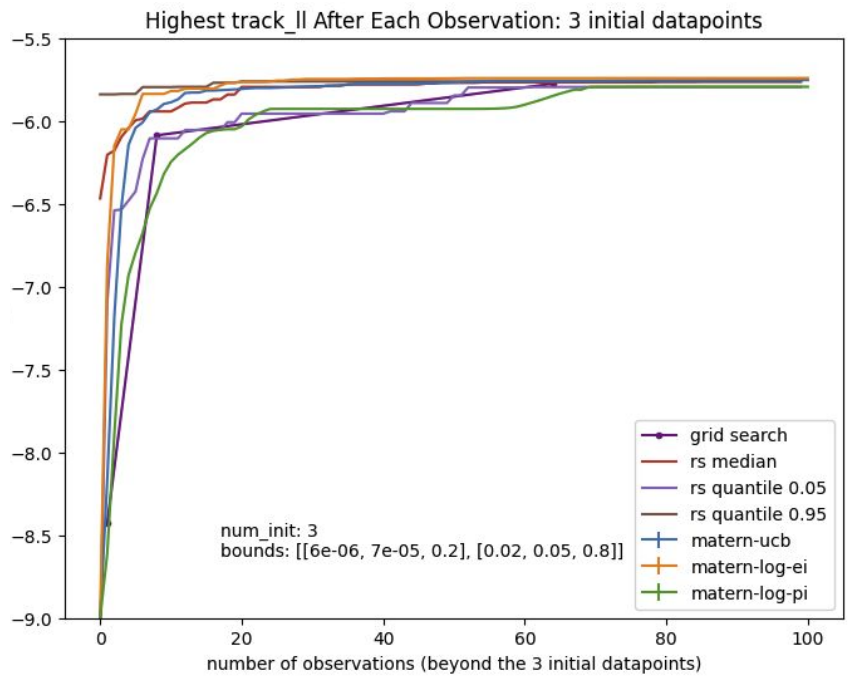
Could be due to the flatness of the track log likelihood model score function.



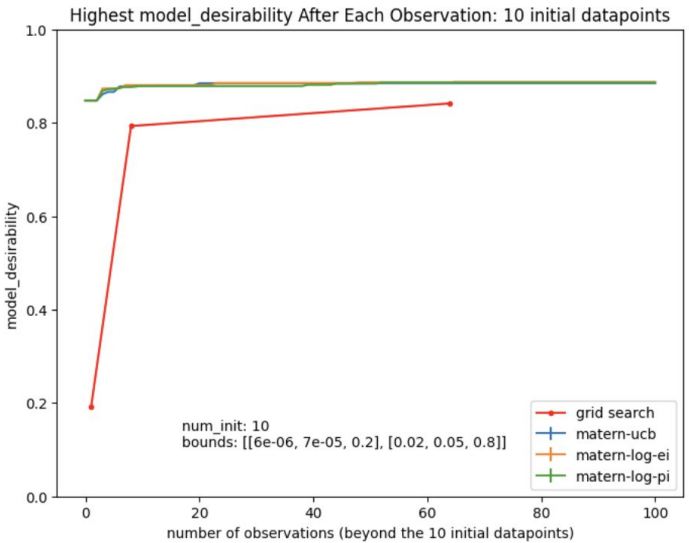


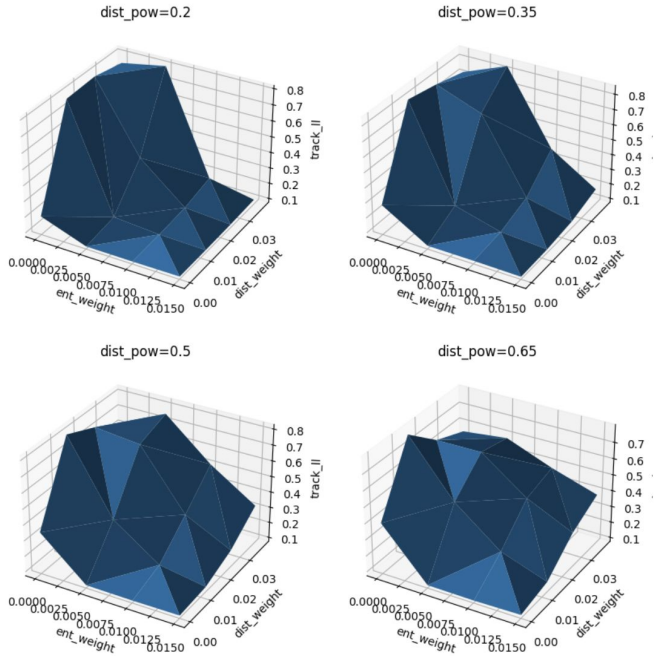
Similarity in Hyperparameter Quality of BO and Random Search when optimizing over Track Log Likelihood





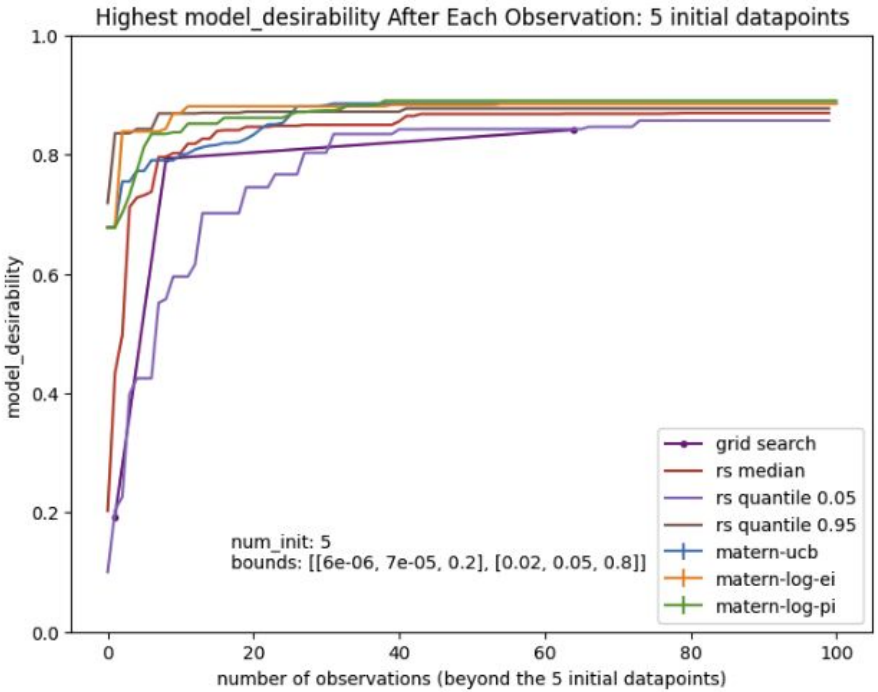
Hyperparameter Quality of Bayesian Optimization consistently Greater than that of Grid Search when Optimizing Over Model Desirability

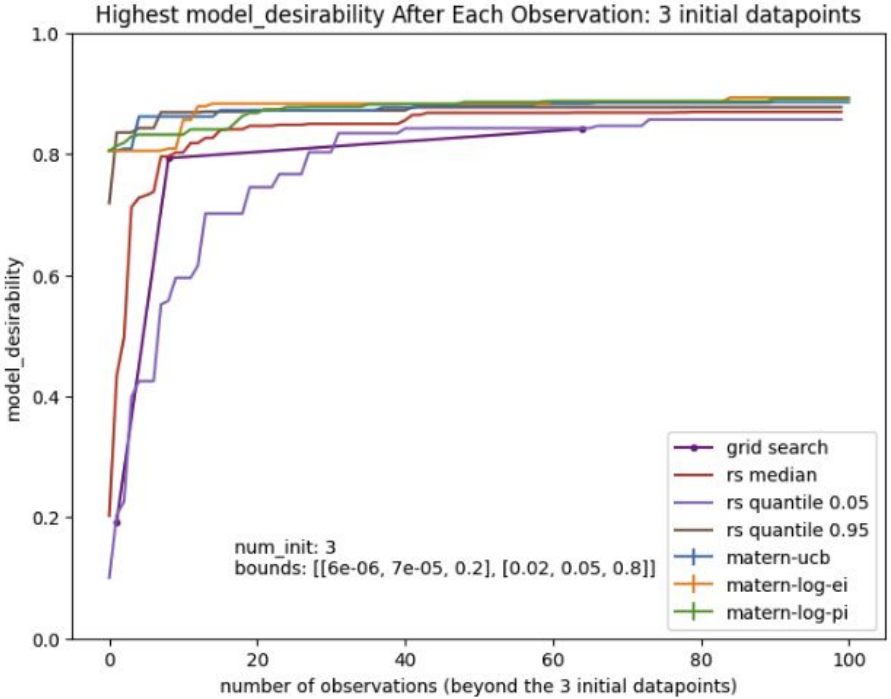




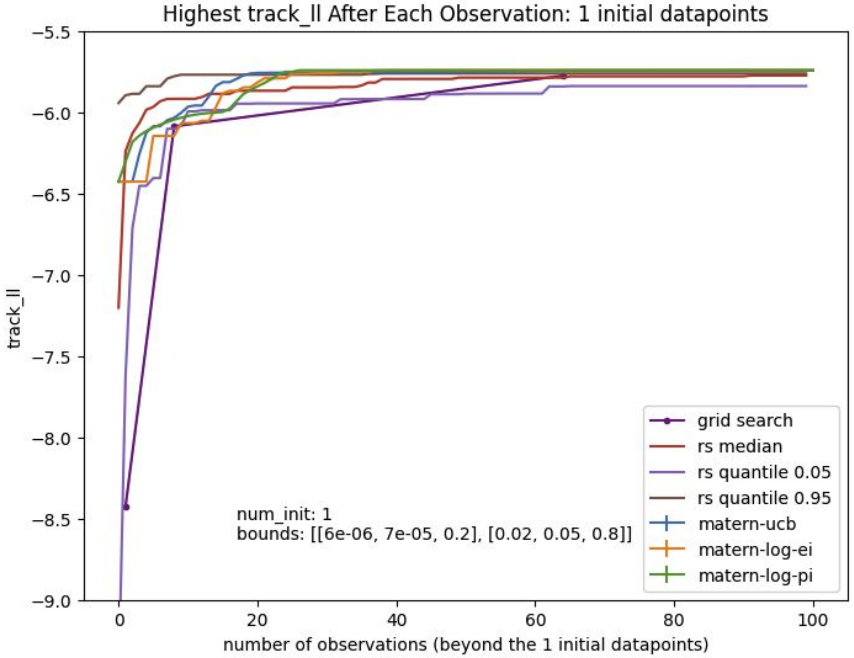
Potential Explanation: Unlike the track log likelihood model score function, the model desirability model score function does not appear to have a plateau at the optimum.

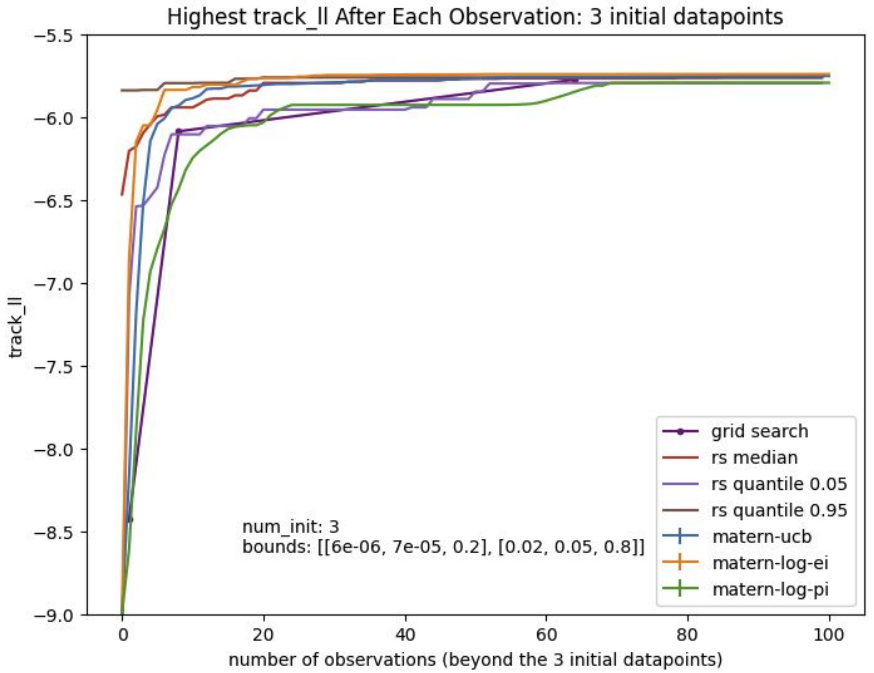
Hyperparameter Quality of Bayesian Optimization Slightly Greater than that of Random Search when Optimizing Over Model Desirability



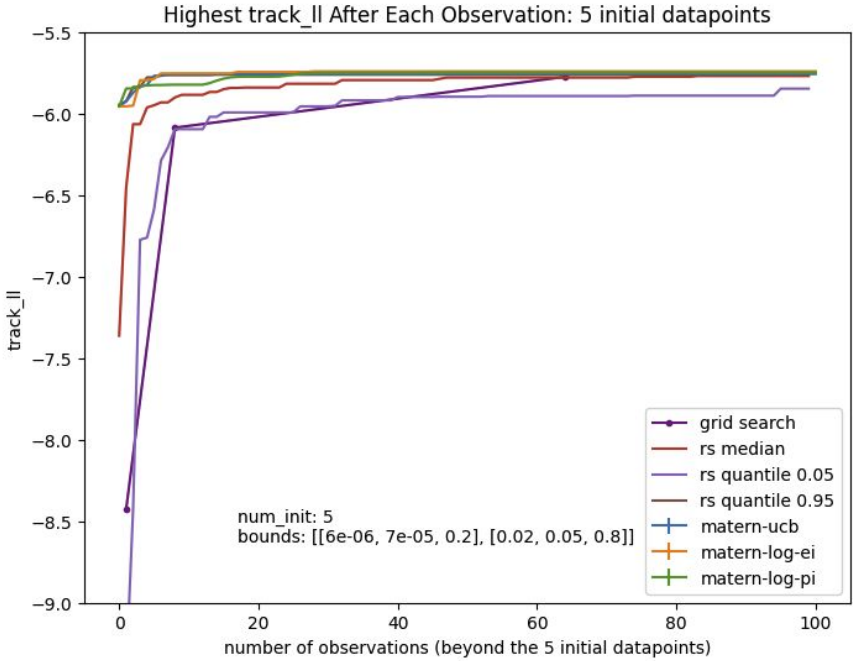


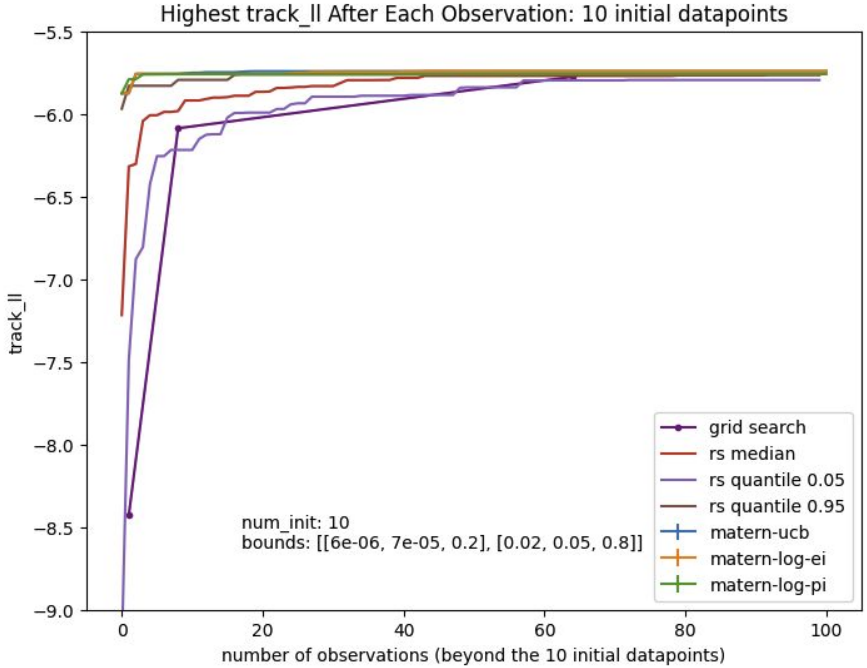
Experiments with 1 and 3 Initial Datapoints (track log likelihood)



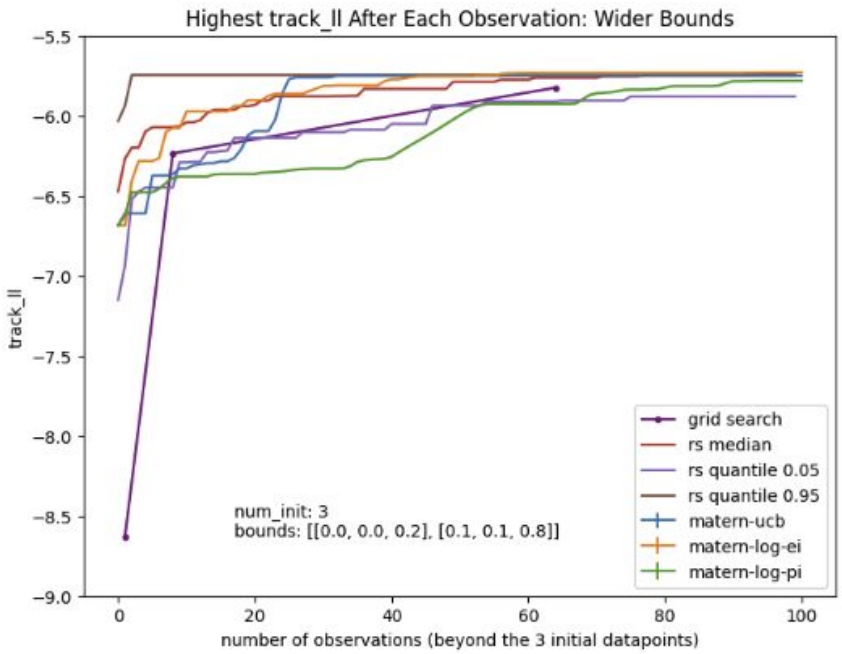


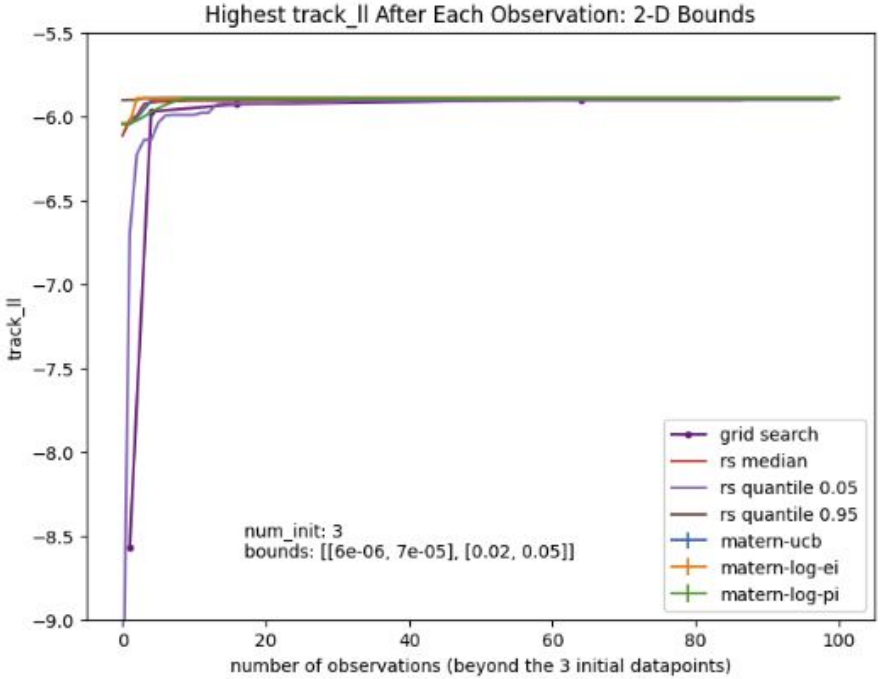
Experiment with 5 and 10 Initial Datapoints (track log likelihood)



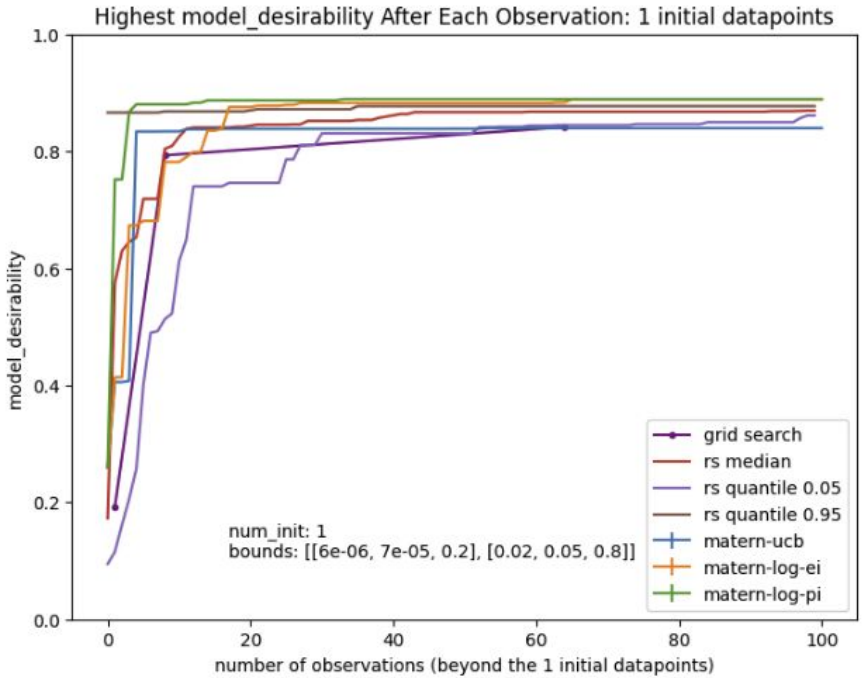


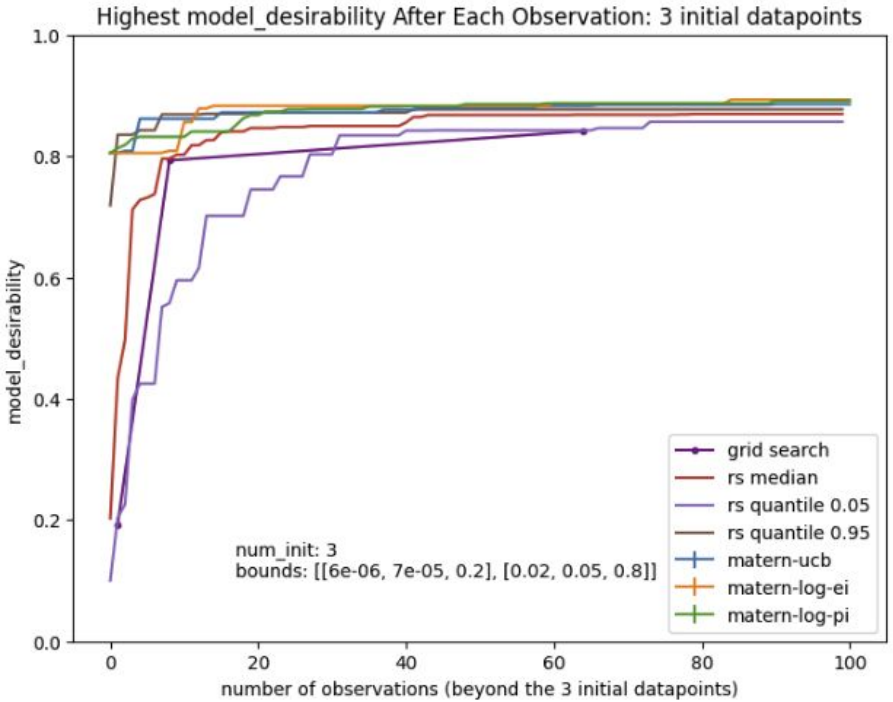
Experiments with different Search Spaces (Track Log Likelihood)



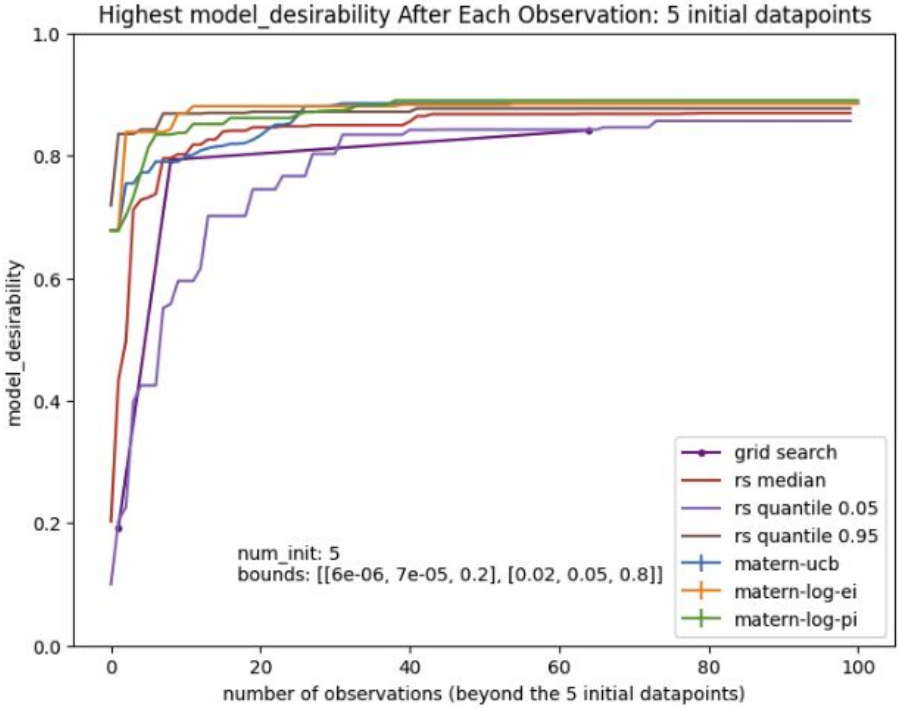


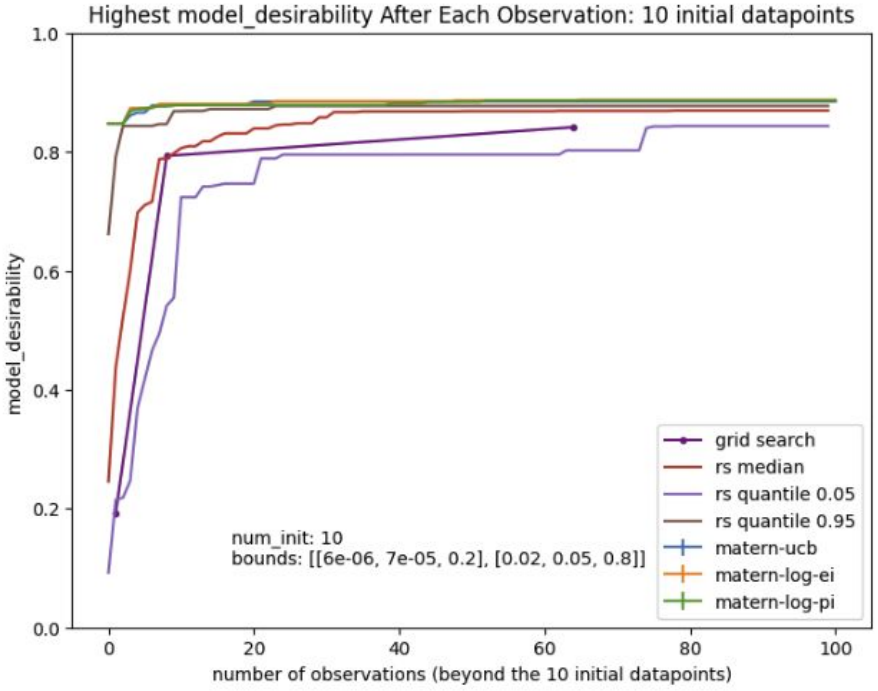
Experiments with 1 and 3 Initial Datapoints (Model Desirability)



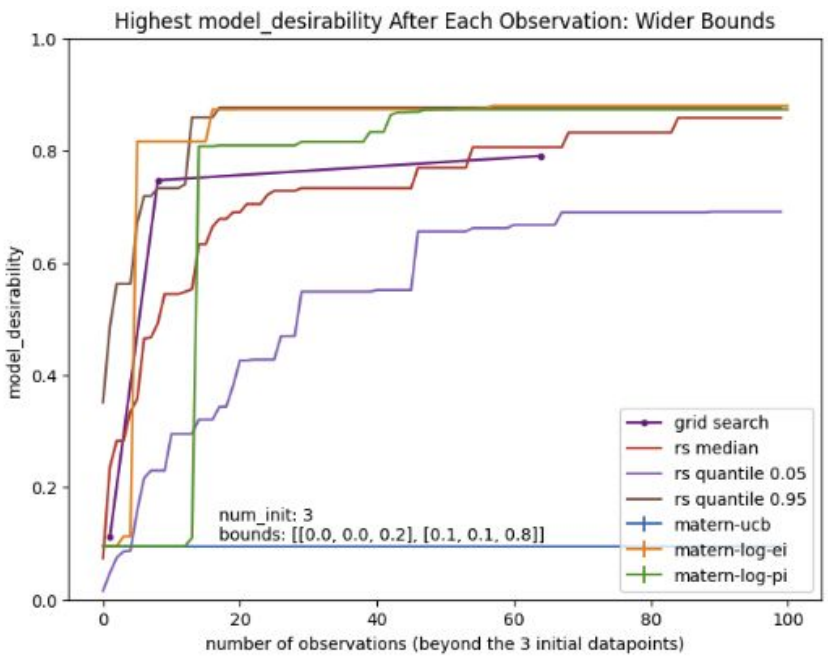


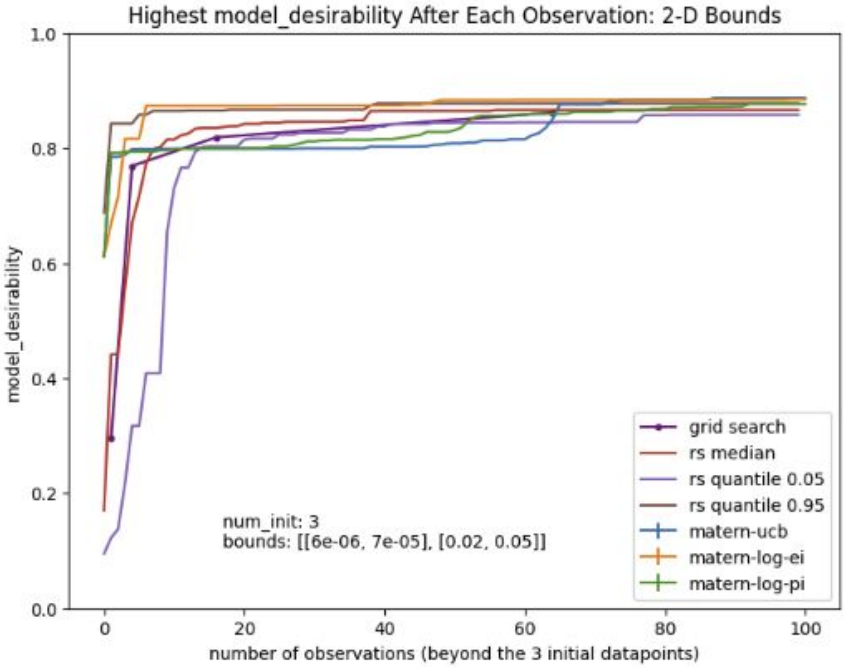
Experiments with 5 and 10 Initial Datapoints (Model Desirability)





Experiments with different search spaces (Model Desirability)





Conclusions and Further Questions

- Bayesian Optimization seems to find similar quality hyperparameters slightly more efficiently than both grid and random search in most track log likelihood optimization scenarios
- Bayesian Optimization is slightly more efficient and finds slightly better hyperparameters than both grid and random search in model desirability optimization scenarios

Further Questions:

- Comparison with Bayesian Optimization's best hyperparameters and Dave's current best hyperparameters found via grid search?
- Open Discussion!