Inference on the Champagne Model using a Gaussian Process

TODO

- Change outputs

Setting up the Champagne Model

Imports

```
import pandas as pd
import numpy as np
from typing import Any
import matplotlib.pyplot as plt

from scipy.stats import qmc
from scipy.stats import norm

import tensorflow as tf
import tensorflow_probability as tfp
from tensorflow_probability.python.distributions import normal

tfb = tfp.bijectors
tfd = tfp.distributions
tfk = tfp.math.psd_kernels
tfp_acq = tfp.experimental.bayesopt.acquisition

gpu_devices = tf.config.experimental.list_physical_devices("GPU")
```

```
for device in gpu_devices:
    tf.config.experimental.set_memory_growth(device, True)
```

```
2024-05-13 16:10:39.941689: I tensorflow/core/platform/cpu_feature_guard.cc:210] This Tensorl
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with
2024-05-13 16:10:40.748725: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Wa
2024-05-13 16:10:43.532653: I external/local_xla/xla/stream_executor/cuda/cuda_executor.cc:98
2024-05-13 16:10:43.687259: W tensorflow/core/common_runtime/gpu/gpu_device.cc:2251] Cannot c
Skipping registering GPU devices...
```

Model itself

```
np.random.seed(590154)

population = 1000
initial_infecteds = 10
epidemic_length = 1000
number_of_events = 15000

pv_champ_alpha = 0.4  # prop of effective care
pv_champ_beta = 0.4  # prop of radical cure
pv_champ_gamma_L = 1 / 223  # liver stage clearance rate
pv_champ_delta = 0.05  # prop of imported cases
pv_champ_lambda = 0.04  # transmission rate
pv_champ_f = 1 / 72  # relapse frequency
pv_champ_r = 1 / 60  # blood stage clearance rate

gamma_L_max = 1/30
lambda_max = 0.1
f_max = 1/14
r_max = 1/14

num_lhc_samples = 128
initial_repeats = 1
```

```
def champagne_stochastic(
    alpha_,
    beta_,
    gamma_L,
```

```
    lambda_,
    f,
    r,
    N=population,
    I_L=initial_infecteds,
    I_0=0,
    S_L=0,
    delta_=0,
    end_time=epidemic_length,
    num_events=number_of_events,
):
    if (0 > (alpha_ or beta_)) or (1 < (alpha_ or beta_)):
        return "Alpha or Beta out of bounds"
    if 0 > (gamma_L or lambda_ or f or r):
        return "Gamma, lambda, f or r out of bounds"

    t = 0
    S_0 = N - I_L - I_0 - S_L
    inc_counter = 0

    list_of_outcomes = [
        {"t": 0, "S_0": S_0, "S_L": S_L, "I_0": I_0, "I_L": I_L, "inc_counter": 0}
    ]

    prop_new = alpha_ * beta_ * f / (alpha_ * beta_ * f + gamma_L)
    i = 0

    while (i < num_events) or (t < 30):
        i += 1
        if S_0 == N:
            while t < 31:
                t += 1
                new_stages = {
                    "t": t,
                    "S_0": N,
                    "S_L": 0,
                    "I_0": 0,
                    "I_L": 0,
                    "inc_counter": inc_counter,
                }
                list_of_outcomes.append(new_stages)
            break
```

```
        S_0_to_I_L = (1 - alpha_) * lambda_ * (I_L + I_0) / N * S_0
        S_0_to_S_L = alpha_ * (1 - beta_) * lambda_ * (I_0 + I_L) / N * S_0
        I_0_to_S_0 = r * I_0 / N
        I_0_to_I_L = lambda_ * (I_L + I_0) / N * I_0
        I_L_to_I_0 = gamma_L * I_L
        I_L_to_S_L = r * I_L
        S_L_to_S_0 = (gamma_L + (f + lambda_ * (I_0 + I_L) / N) * alpha_ * beta_) * S_L
        S_L_to_I_L = (f + lambda_ * (I_0 + I_L) / N) * (1 - alpha_) * S_L

        total_rate = (
            S_0_to_I_L
            + S_0_to_S_L
            + I_0_to_S_0
            + I_0_to_I_L
            + I_L_to_I_0
            + I_L_to_S_L
            + S_L_to_S_0
            + S_L_to_I_L
        )

        delta_t = np.random.exponential(1 / total_rate)
        new_stages_prob = [
            S_0_to_I_L / total_rate,
            S_0_to_S_L / total_rate,
            I_0_to_S_0 / total_rate,
            I_0_to_I_L / total_rate,
            I_L_to_I_0 / total_rate,
            I_L_to_S_L / total_rate,
            S_L_to_S_0 / total_rate,
            S_L_to_I_L / total_rate,
        ]
        t += delta_t
        silent_incidences = np.random.poisson(
            delta_t * alpha_ * beta_ * lambda_ * (I_L + I_0) * S_0 / N
        )

        new_stages = np.random.choice(
            [
                {
                    "t": t,
                    "S_0": S_0 - 1,
                    "S_L": S_L,
```

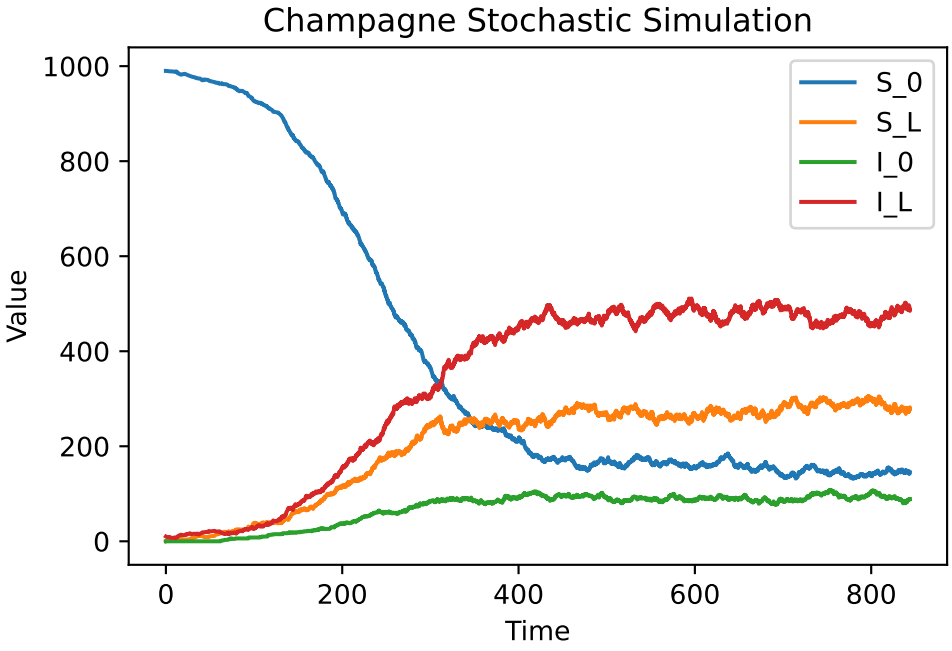
```
                "I_0": I_0,
                "I_L": I_L + 1,
                "inc_counter": inc_counter + silent_incidences + 1,
            },
            {
                "t": t,
                "S_0": S_0 - 1,
                "S_L": S_L + 1,
                "I_0": I_0,
                "I_L": I_L,
                "inc_counter": inc_counter + silent_incidences + 1,
            },
            {
                "t": t,
                "S_0": S_0 + 1,
                "S_L": S_L,
                "I_0": I_0 - 1,
                "I_L": I_L,
                "inc_counter": inc_counter + silent_incidences,
            },
            {
                "t": t,
                "S_0": S_0,
                "S_L": S_L,
                "I_0": I_0 - 1,
                "I_L": I_L + 1,
                "inc_counter": inc_counter + silent_incidences,
            },
            {
                "t": t,
                "S_0": S_0,
                "S_L": S_L,
                "I_0": I_0 + 1,
                "I_L": I_L - 1,
                "inc_counter": inc_counter + silent_incidences,
            },
            {
                "t": t,
                "S_0": S_0,
                "S_L": S_L + 1,
                "I_0": I_0,
                "I_L": I_L - 1,
```

```
                    "inc_counter": inc_counter + silent_incidences,
                },
                {
                    "t": t,
                    "S_0": S_0 + 1,
                    "S_L": S_L - 1,
                    "I_0": I_0,
                    "I_L": I_L,
                    "inc_counter": inc_counter
                    + silent_incidences
                    + np.random.binomial(1, prop_new),
                },
                {
                    "t": t,
                    "S_0": S_0,
                    "S_L": S_L - 1,
                    "I_0": I_0,
                    "I_L": I_L + 1,
                    "inc_counter": inc_counter + silent_incidences + 1,
                },
            ],
            p=new_stages_prob,
        )

        list_of_outcomes.append(new_stages)

        S_0 = new_stages["S_0"]
        I_0 = new_stages["I_0"]
        I_L = new_stages["I_L"]
        S_L = new_stages["S_L"]
        inc_counter = new_stages["inc_counter"]

    outcome_df = pd.DataFrame(list_of_outcomes)
    return outcome_df


champ_samp = champagne_stochastic(
    pv_champ_alpha,
    pv_champ_beta,
    pv_champ_gamma_L,
    pv_champ_lambda,
    pv_champ_f,
```

```
    pv_champ_r,
)  # .melt(id_vars='t')
```

Plotting outcome

```
champ_samp.drop("inc_counter", axis=1).plot(x="t", legend=True)
plt.xlabel("Time")
plt.ylabel("Value")
plt.title("Champagne Stochastic Simulation")
plt.savefig("champagne_GP_images/champagne_simulation.pdf")
plt.show()
```



Function that Outputs Final Prevalence

```
def incidence(df, start, days):
    start_ind = df[df["t"].le(start)].index[-1]
    end_ind = df[df["t"].le(start + days)].index[-1]
    incidence_week = df.iloc[end_ind]["inc_counter"] - df.iloc[start_ind]["inc_counter"]
```

```
    return incidence_week


def champ_sum_stats(alpha_, beta_, gamma_L, lambda_, f, r):
    champ_df_ = champagne_stochastic(alpha_, beta_, gamma_L, lambda_, f, r)
    fin_t = champ_df_.iloc[-1]["t"]
    first_month_inc = incidence(champ_df_, 0, 30)
    fin_t = champ_df_.iloc[-1]["t"]
    fin_week_inc = incidence(champ_df_, fin_t - 7, 7)
    fin_prev = champ_df_.iloc[-1]["I_0"] + champ_df_.iloc[-1]["I_L"]

    return np.array([fin_prev, first_month_inc, fin_week_inc])


observed_sum_stats = champ_sum_stats(
    pv_champ_alpha,
    pv_champ_beta,
    pv_champ_gamma_L,
    pv_champ_lambda,
    pv_champ_f,
    pv_champ_r,
)


def discrepency_fn(alpha_, beta_, gamma_L, lambda_, f, r, mean_of = 20):  # best is L1 norm
    mean_obs = 0
    for i in range(mean_of):
        x = champ_sum_stats(alpha_, beta_, gamma_L, lambda_, f, r)
        mean_obs += (
            1
            / mean_of
            * np.log(np.linalg.norm((x - observed_sum_stats) / observed_sum_stats))
        )
    # return np.sum(np.abs((x - observed_sum_stats) / observed_sum_stats))
    # return np.linalg.norm((x - observed_sum_stats) / observed_sum_stats)
    return mean_obs
```

Gaussian Process Regression on Final Prevalence Discrepency

```
my_seed = np.random.default_rng(seed=1795)  # For replicability

variables_names = ["alpha", "beta", "gamma_L", "lambda", "f", "r"]

LHC_sampler = qmc.LatinHypercube(d=6, seed=my_seed)
LHC_samples = LHC_sampler.random(n=num_lhc_samples)

# Using Champagne Initialisation table 2
LHC_samples[:, 2] = gamma_L_max * LHC_samples[:, 2]
LHC_samples[:, 3] = lambda_max * LHC_samples[:, 3]
LHC_samples[:, 4] = f_max * LHC_samples[:, 4]
LHC_samples[:, 5] = r_max * LHC_samples[:, 5]


# LHC_samples[:, 2] = 1/50* LHC_samples[:, 2]
# LHC_samples[:, 3] = 0.2 * LHC_samples[:, 3]
# LHC_samples[:, 4] = 1/10 * LHC_samples[:, 4]
# LHC_samples[:, 5] = 1/10 * LHC_samples[:, 5]
# LHC_samples[:, 2] = -pv_champ_gamma_L * np.log(LHC_samples[:, 2])
# LHC_samples[:, 3] = -pv_champ_lambda * np.log(LHC_samples[:, 3])
# LHC_samples[:, 4] = -pv_champ_f * np.log(LHC_samples[:, 4])
# LHC_samples[:, 5] = -pv_champ_r * np.log(LHC_samples[:, 5])

LHC_samples = np.repeat(LHC_samples, initial_repeats, axis = 0)

LHC_indices_df = pd.DataFrame(LHC_samples, columns=variables_names)

print(LHC_indices_df.head())
```

```
      alpha      beta   gamma_L    lambda         f         r
0  0.882816  0.500918  0.014975  0.089356  0.001630  0.019512
1  0.070010  0.816779  0.021772  0.042146  0.064415  0.056634
2  0.540452  0.705079  0.024414  0.088323  0.050239  0.030581
3  0.535875  0.652563  0.026679  0.055069  0.041084  0.008563
4  0.262114  0.172826  0.010177  0.072764  0.021838  0.052903
```

Generate Discrepencies

```
random_discrepencies = LHC_indices_df.apply(
    lambda x: discrepency_fn(
        x["alpha"], x["beta"], x["gamma_L"], x["lambda"], x["f"], x["r"]
    ),
    axis=1,
)

print(random_discrepencies.head())
```

```
0    0.150908
1    0.234689
2    1.059515
3   -0.093519
4    0.613619
dtype: float64
```

Differing Methods to Iterate Function

```
# import timeit

# def function1():
#     np.vectorize(champ_sum_stats)(random_indices_df['alpha'],
#     random_indices_df['beta'], random_indices_df['gamma_L'],
#     random_indices_df['lambda'], random_indices_df['f'], random_indices_df['r'])
#     pass

# def function2():
#     random_indices_df.apply(
#         lambda x: champ_sum_stats(
#             x['alpha'], x['beta'], x['gamma_L'], x['lambda'], x['f'], x['r']),
#             axis = 1)
#     pass

# # Time function1
# time_taken_function1 = timeit.timeit(
#     "function1()", globals=globals(), number=100)
```

```
# # Time function2
# time_taken_function2 = timeit.timeit(
#     "function2()", globals=globals(), number=100)

# print("Time taken for function1:", time_taken_function1)
# print("Time taken for function2:", time_taken_function2)
```

Time taken for function1: 187.48960775700016 Time taken for function2: 204.06618941299985

Constrain Variables to be Positive

```
constrain_positive = tfb.Shift(np.finfo(np.float64).tiny)(tfb.Exp())
```

Custom Quadratic Mean Function

```
class quad_mean_fn(tf.Module):
    def __init__(self):
        super(quad_mean_fn, self).__init__()
        # self.amp_alpha_mean = tfp.util.TransformedVariable(
        #     bijector=constrain_positive,
        #     initial_value=1.0,
        #     dtype=np.float64,
        #     name="amp_alpha_mean",
        # )
        # self.alpha_tp = tf.Variable(pv_champ_alpha, dtype=np.float64, name="alpha_tp")
        # self.amp_beta_mean = tfp.util.TransformedVariable(
        #     bijector=constrain_positive,
        #     initial_value=0.5,
        #     dtype=np.float64,
        #     name="amp_beta_mean",
        # )
        # self.beta_tp = tf.Variable(pv_champ_beta, dtype=np.float64, name="beta_tp")
        self.amp_gamma_L_mean = tfp.util.TransformedVariable(
            bijector=constrain_positive,
            initial_value=1.0,
            dtype=np.float64,
            name="amp_gamma_L_mean",
        )
```

```
        # self.gamma_L_tp = tfp.util.TransformedVariable(
        #     bijector=constrain_positive,
        #     initial_value=1.0,
        #     dtype=np.float64,
        #     name="gamma_L_tp",
        # )
        self.amp_lambda_mean = tfp.util.TransformedVariable(
            bijector=constrain_positive,
            initial_value=1.0,
            dtype=np.float64,
            name="amp_lambda_mean",
        )
        # self.lambda_tp = tfp.util.TransformedVariable(
        #     bijector=constrain_positive,
        #     initial_value=1.0,
        #     dtype=np.float64,
        #     name="lambda_tp",
        # )
        self.amp_f_mean = tfp.util.TransformedVariable(
            bijector=constrain_positive,
            initial_value=1.0,
            dtype=np.float64,
            name="amp_f_mean",
        )
        # self.f_tp = tfp.util.TransformedVariable(
        #     bijector=constrain_positive,
        #     initial_value=1.0,
        #     dtype=np.float64,
        #     name="f_tp",
        # )
        self.amp_r_mean = tfp.util.TransformedVariable(
            bijector=constrain_positive,
            initial_value=1.0,
            dtype=np.float64,
            name="amp_r_mean",
        )
        # self.r_tp = tfp.util.TransformedVariable(
        #     bijector=constrain_positive,
        #     initial_value=1.0,
        #     dtype=np.float64,
        #     name="r_tp",
        # )
```

```
        # self.bias_mean = tfp.util.TransformedVariable(
        #     bijector=constrain_positive,
        #     initial_value=1.0,
        #     dtype=np.float64,
        #     name="bias_mean",
        # )
        self.bias_mean = tf.Variable(-1.5, dtype=np.float64, name="bias_mean")

    def __call__(self, x):
        return (
            self.bias_mean
            # + self.amp_alpha_mean * (x[..., 0] - self.alpha_tp) ** 2
            # + self.amp_beta_mean * (x[..., 1] - self.beta_tp) ** 2
            # + self.amp_gamma_L_mean * (x[..., 2] - self.gamma_L_tp) ** 2
            # + self.amp_lambda_mean * (x[..., 3] - self.lambda_tp) ** 2
            # + self.amp_f_mean * (x[..., 4] - self.f_tp) ** 2
            # + self.amp_r_mean * (x[..., 5] - self.r_tp) ** 2
            + self.amp_gamma_L_mean * (x[..., 2]) ** 2
            + self.amp_lambda_mean * (x[..., 3]) ** 2
            + self.amp_f_mean * (x[..., 4]) ** 2
            + self.amp_r_mean * (x[..., 5]) ** 2
        )


quad_mean_fn().__call__(x=np.array([[1.0, 1.0, 1.0, 1.0, 1.0, 1.0]]))  # should return 1
```

```
<tf.Tensor: shape=(1,), dtype=float64, numpy=array([2.5])>
```

Custom Linear Mean Function

```
class lin_mean_fn(tf.Module):
    def __init__(self):
        super(lin_mean_fn, self).__init__()
        # self.amp_alpha_lin = tfp.util.TransformedVariable(
        #     bijector=constrain_positive,
        #     initial_value=1.0,
        #     dtype=np.float64,
        #     name="amp_alpha_lin",
        # )
        # self.amp_beta_lin = tfp.util.TransformedVariable(
```

```
        #     bijector=constrain_positive,
        #     initial_value=0.5,
        #     dtype=np.float64,
        #     name="amp_beta_lin",
        # )
        self.amp_gamma_L_lin = tfp.util.TransformedVariable(
            bijector=constrain_positive,
            initial_value=1.0,
            dtype=np.float64,
            name="amp_gamma_L_lin",
        )
        self.amp_lambda_lin = tfp.util.TransformedVariable(
            bijector=constrain_positive,
            initial_value=1.0,
            dtype=np.float64,
            name="amp_lambda_lin",
        )
        self.amp_f_lin = tfp.util.TransformedVariable(
            bijector=constrain_positive,
            initial_value=1.0,
            dtype=np.float64,
            name="amp_f_lin",
        )
        self.amp_r_lin = tfp.util.TransformedVariable(
            bijector=constrain_positive,
            initial_value=1.0,
            dtype=np.float64,
            name="amp_r_lin",
        )
        # self.bias_lin = tfp.util.TransformedVariable(
        #     bijector=constrain_positive,
        #     initial_value=1.0,
        #     dtype=np.float64,
        #     name="bias_lin",
        # )
        self.bias_lin = tf.Variable(0.0, dtype=np.float64, name="bias_mean")

    def __call__(self, x):
        return (
            self.bias_lin
            # + self.amp_alpha_lin * (x[..., 0])
            # + self.amp_beta_lin * (x[..., 1])
```

```
            + self.amp_gamma_L_lin * (x[..., 2])
            + self.amp_lambda_lin * (x[..., 3])
            + self.amp_f_lin * (x[..., 4])
            + self.amp_r_lin * (x[..., 5])
        )
```

```
class const_mean_fn(tf.Module):
    def __init__(self):
        super(const_mean_fn, self).__init__()
        self.bias_lin = tf.Variable(0.0, dtype=np.float64, name="bias_mean")

    def __call__(self, x):
        return self.bias_lin
```

Making the ARD Kernel

```
index_vals = LHC_indices_df.values
obs_vals = random_discrepencies.values

amplitude_champ = tfp.util.TransformedVariable(
    bijector=constrain_positive,
    initial_value=4.0,
    dtype=np.float64,
    name="amplitude_champ",
)

observation_noise_variance_champ = tfp.util.TransformedVariable(
    bijector=constrain_positive,
    initial_value=1.,
    dtype=np.float64,
    name="observation_noise_variance_champ",
)
```

```
length_scales_champ = tfp.util.TransformedVariable(
    bijector=tfb.Sigmoid(
        np.float64(0.0),
        [1.0 / 4, 1.0 / 4, gamma_L_max / 4, lambda_max / 4, f_max / 4, r_max / 4],
    ),
    initial_value=[1 / 8, 1 / 8, gamma_L_max / 8, lambda_max / 8, f_max / 8, r_max / 8],
    dtype=np.float64,
```

```
    name="length_scales_champ",
)
```

```
kernel_champ = tfk.FeatureScaled(
    tfk.MaternFiveHalves(amplitude=amplitude_champ),
    scale_diag=length_scales_champ,
)
```

Define the Gaussian Process with Quadratic Mean Function and ARD Kernel

```
# Define Gaussian Process with the custom kernel
champ_GP = tfd.GaussianProcess(
    kernel=kernel_champ,
    observation_noise_variance=observation_noise_variance_champ,
    index_points=index_vals,
    mean_fn=const_mean_fn(),
)

print(champ_GP.trainable_variables)

Adam_optim = tf.keras.optimizers.Adam(learning_rate=0.01)
```

```
(<tf.Variable 'amplitude_champ:0' shape=() dtype=float64, numpy=1.3862943611198906>, <tf.Vari
```

Train the Hyperparameters

Leave One Out Predictive Log-likelihood

```
# predictive log stuff
# @tf.function(autograph=False, jit_compile=False)
# def optimize():
#     with tf.GradientTape() as tape:
#         K = (
#             champ_GP.kernel.matrix(index_vals, index_vals)
#             + tf.eye(index_vals.shape[0], dtype=np.float64)
#             * observation_noise_variance_champ
#         )
#         means = champ_GP.mean_fn(index_vals)
```

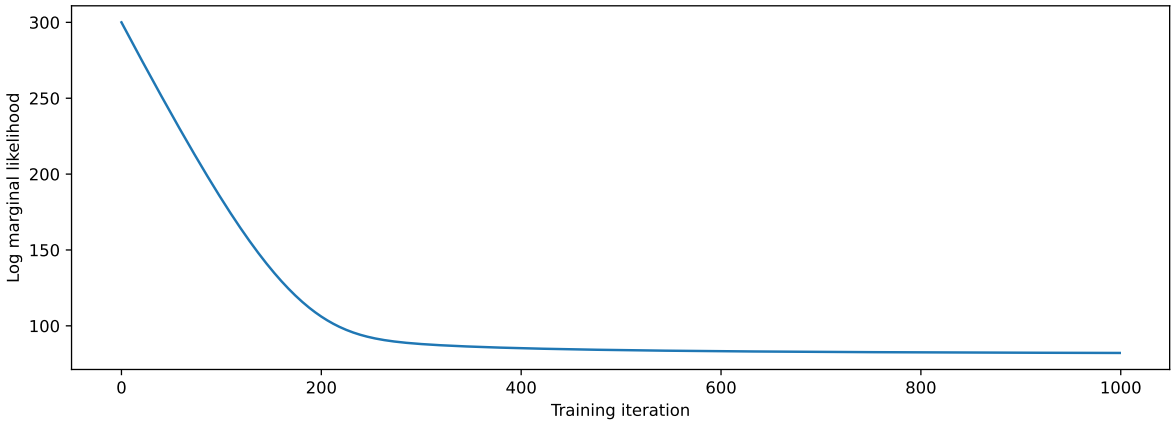
```
#         K_inv = tf.linalg.inv(K)
#         K_inv_y = K_inv @ tf.reshape(obs_vals - means, shape=[obs_vals.shape[0], 1])
#         K_inv_diag = tf.linalg.diag_part(K_inv)
#         log_var = tf.math.log(K_inv_diag)
#         log_mu = tf.reshape(K_inv_y, shape=[-1]) ** 2
#         loss = -tf.math.reduce_sum(log_var - log_mu)
#     grads = tape.gradient(loss, champ_GP.trainable_variables)
#     Adam_optim.apply_gradients(zip(grads, champ_GP.trainable_variables))
#     return loss


# num_iters = 10000

# lls_ = np.zeros(num_iters, np.float64)
# tolerance = 1e-6  # Set your desired tolerance level
# previous_loss = float("inf")

# for i in range(num_iters):
#     loss = optimize()
#     lls_[i] = loss

#     # Check if change in loss is less than tolerance
#     if abs(loss - previous_loss) < tolerance:
#         print(f"Hyperparameter convergence reached at iteration {i+1}.")
#         lls_ = lls_[range(i + 1)]
#         break

#     previous_loss = loss
```

Maximum Likelihood Estimation

```
# Now we optimize the model parameters.
num_iters = 1000

# Use `tf.function` to trace the loss for more efficient evaluation.
@tf.function(autograph=False, jit_compile=False)
def train_model():
    with tf.GradientTape() as tape:
        loss = -champ_GP.log_prob(obs_vals)
    grads = tape.gradient(loss, champ_GP.trainable_variables)
```

```
    Adam_optim.apply_gradients(zip(grads, champ_GP.trainable_variables))
    return loss


# Store the likelihood values during training, so we can plot the progress
lls_ = np.zeros(num_iters, np.float64)
for i in range(num_iters):
    loss = train_model()
    lls_[i] = loss

print("Trained parameters:")
print("amplitude: {}".format(amplitude_champ._value().numpy()))
print("length_scales: {}".format(length_scales_champ._value().numpy()))
print(
    "observation_noise_variance: {}".format(
        observation_noise_variance_champ._value().numpy()
    )
)

# Plot the loss evolution
plt.figure(figsize=(12, 4))
plt.plot(lls_)
plt.xlabel("Training iteration")
plt.ylabel("Log marginal likelihood")
plt.show()
```

```
Trained parameters:
amplitude: 0.4852664671997231
length_scales: [0.24905499 0.24924018 0.00831123 0.02017699 0.01780131 0.01779375]
observation_noise_variance: 0.004872773344395279
```

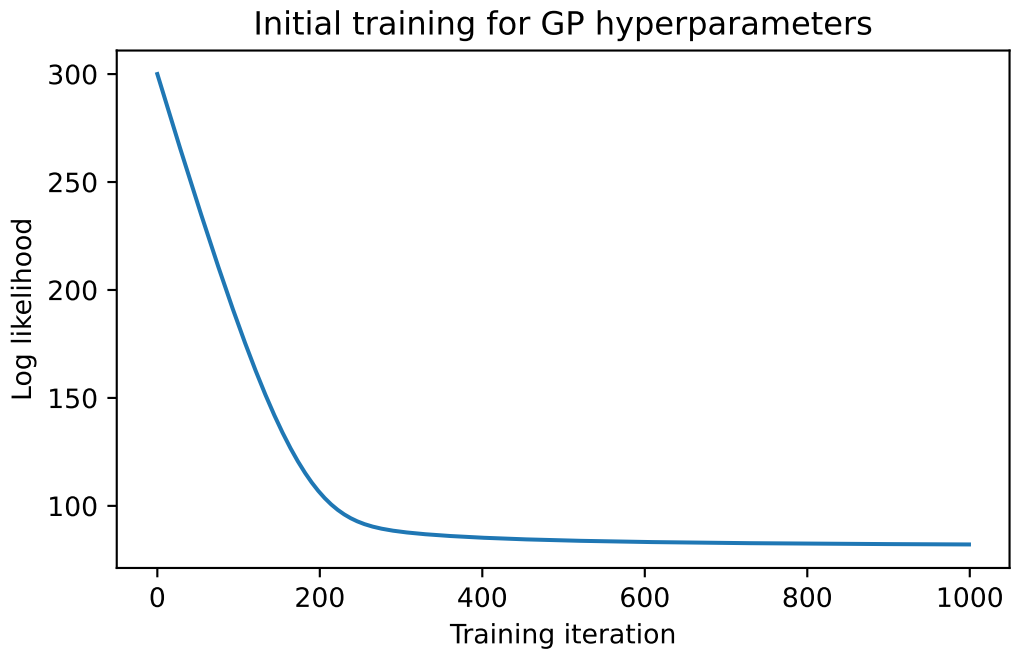



```
print("Trained parameters:")
for var in champ_GP.trainable_variables:
    if "bias" in var.name:
        print("{} is {}\n".format(var.name, var.numpy().round(3)))
    else:
        if "length" in var.name:
            print(
                "{} is {}\n".format(
                    var.name,
                    tfb.Sigmoid(
                        np.float64(0.0),
                        [
                            1.0 / 4,
                            1.0 / 4,
                            gamma_L_max / 4,
                            lambda_max / 4,
                            f_max / 4,
                            r_max / 4,
                        ],
                    )
                    .forward(var)
                    .numpy()
                    .round(3),
                )
            )
        else:
            print(
                "{} is {}\n".format(
                    var.name, constrain_positive.forward(var).numpy().round(3)
```

```
                )
            )
```

```
Trained parameters:
amplitude_champ:0 is 0.485

length_scales_champ:0 is [0.249 0.249 0.008 0.02  0.018 0.018]

observation_noise_variance_champ:0 is 0.005

bias_mean:0 is 0.315
```

```
plt.figure(figsize=(6, 3.5))
plt.plot(lls_)
plt.title("Initial training for GP hyperparameters")
plt.xlabel("Training iteration")
plt.ylabel("Log likelihood")
plt.savefig("champagne_GP_images/hyperparam_loss_log_discrep.pdf")
plt.show()
```



Creating slices across one variable dimension

```
plot_samp_no = 21
plot_gp_no = 100
gp_samp_no = 30
```

```
slice_samples_dict = {
    "alpha_slice_samples": np.repeat(np.concatenate(
        (
            np.linspace(0, 1, plot_samp_no, dtype=np.float64).reshape(-1, 1),  # alpha
            np.repeat(pv_champ_beta, plot_samp_no).reshape(-1, 1),  # beta
            np.repeat(pv_champ_gamma_L, plot_samp_no).reshape(-1, 1),  # gamma_L
            np.repeat(pv_champ_lambda, plot_samp_no).reshape(-1, 1),  # lambda
            np.repeat(pv_champ_f, plot_samp_no).reshape(-1, 1),  # f
            np.repeat(pv_champ_r, plot_samp_no).reshape(-1, 1),  # r
        ),
        axis=1,
    ), 5, axis = 0),
    "alpha_gp_samples": np.concatenate(
        (
            np.linspace(0, 1, plot_gp_no, dtype=np.float64).reshape(-1, 1),  # alpha
            np.repeat(pv_champ_beta, plot_gp_no).reshape(-1, 1),  # beta
            np.repeat(pv_champ_gamma_L, plot_gp_no).reshape(-1, 1),  # gamma_L
            np.repeat(pv_champ_lambda, plot_gp_no).reshape(-1, 1),  # lambda
            np.repeat(pv_champ_f, plot_gp_no).reshape(-1, 1),  # f
            np.repeat(pv_champ_r, plot_gp_no).reshape(-1, 1),  # r
        ),
        axis=1,
    ),
    "beta_slice_samples": np.repeat(np.concatenate(
        (
            np.repeat(pv_champ_alpha, plot_samp_no).reshape(-1, 1),  # alpha
            np.linspace(0, 1, plot_samp_no, dtype=np.float64).reshape(-1, 1),  # beta
            np.repeat(pv_champ_gamma_L, plot_samp_no).reshape(-1, 1),  # gamma_L
            np.repeat(pv_champ_lambda, plot_samp_no).reshape(-1, 1),  # lambda
            np.repeat(pv_champ_f, plot_samp_no).reshape(-1, 1),  # f
            np.repeat(pv_champ_r, plot_samp_no).reshape(-1, 1),  # r
        ),
        axis=1,
    ), 5, axis = 0),
    "beta_gp_samples": np.concatenate(
```

```
        (
            np.repeat(pv_champ_alpha, plot_gp_no).reshape(-1, 1),  # alpha
            np.linspace(0, 1, plot_gp_no, dtype=np.float64).reshape(-1, 1),  # beta
            np.repeat(pv_champ_gamma_L, plot_gp_no).reshape(-1, 1),  # gamma_L
            np.repeat(pv_champ_lambda, plot_gp_no).reshape(-1, 1),  # lambda
            np.repeat(pv_champ_f, plot_gp_no).reshape(-1, 1),  # f
            np.repeat(pv_champ_r, plot_gp_no).reshape(-1, 1),  # r
        ),
        axis=1,
    ),
    "gamma_L_slice_samples": np.repeat(np.concatenate(
        (
            np.repeat(pv_champ_alpha, plot_samp_no).reshape(-1, 1),  # alpha
            np.repeat(pv_champ_beta, plot_samp_no).reshape(-1, 1),  # beta
            np.linspace(0, gamma_L_max, plot_samp_no, dtype=np.float64).reshape(-1, 1),  # ga
            np.repeat(pv_champ_lambda, plot_samp_no).reshape(-1, 1),  # lambda
            np.repeat(pv_champ_f, plot_samp_no).reshape(-1, 1),  # f
            np.repeat(pv_champ_r, plot_samp_no).reshape(-1, 1),  # r
        ),
        axis=1,
    ), 5, axis = 0),
    "gamma_L_gp_samples": np.concatenate(
        (
            np.repeat(pv_champ_alpha, plot_gp_no).reshape(-1, 1),  # alpha
            np.repeat(pv_champ_beta, plot_gp_no).reshape(-1, 1),  # beta
            np.linspace(0, gamma_L_max, plot_gp_no, dtype=np.float64).reshape(-1, 1),  # gamm
            np.repeat(pv_champ_lambda, plot_gp_no).reshape(-1, 1),  # lambda
            np.repeat(pv_champ_f, plot_gp_no).reshape(-1, 1),  # f
            np.repeat(pv_champ_r, plot_gp_no).reshape(-1, 1),  # r
        ),
        axis=1,
    ),
    "lambda_slice_samples": np.repeat(np.concatenate(
        (
            np.repeat(pv_champ_alpha, plot_samp_no).reshape(-1, 1),  # alpha
            np.repeat(pv_champ_beta, plot_samp_no).reshape(-1, 1),  # beta
            np.repeat(pv_champ_gamma_L, plot_samp_no).reshape(-1, 1),  # gamma_L
            np.linspace(0, lambda_max, plot_samp_no, dtype=np.float64).reshape(-1, 1), # lamb
            np.repeat(pv_champ_f, plot_samp_no).reshape(-1, 1),  # f
            np.repeat(pv_champ_r, plot_samp_no).reshape(-1, 1),  # r
        ),
        axis=1,
```

```
    ), 5, axis = 0),
    "lambda_gp_samples": np.concatenate(
        (
            np.repeat(pv_champ_alpha, plot_gp_no).reshape(-1, 1),  # alpha
            np.repeat(pv_champ_beta, plot_gp_no).reshape(-1, 1),  # beta
            np.repeat(pv_champ_gamma_L, plot_gp_no).reshape(-1, 1),  # gamma_L
            np.linspace(0, lambda_max, plot_gp_no, dtype=np.float64).reshape(-1, 1), # lambda
            np.repeat(pv_champ_f, plot_gp_no).reshape(-1, 1),  # f
            np.repeat(pv_champ_r, plot_gp_no).reshape(-1, 1),  # r
        ),
        axis=1,
    ),
    "f_slice_samples": np.repeat(np.concatenate(
        (
            np.repeat(pv_champ_alpha, plot_samp_no).reshape(-1, 1),  # alpha
            np.repeat(pv_champ_beta, plot_samp_no).reshape(-1, 1),  # beta
            np.repeat(pv_champ_gamma_L, plot_samp_no).reshape(-1, 1),  # gamma_L
            np.repeat(pv_champ_lambda, plot_samp_no).reshape(-1, 1),  # lambda
            np.linspace(0, f_max, plot_samp_no, dtype=np.float64).reshape(-1, 1), # f
            np.repeat(pv_champ_r, plot_samp_no).reshape(-1, 1),  # r
        ),
        axis=1,
    ), 5, axis = 0),
    "f_gp_samples": np.concatenate(
        (
            np.repeat(pv_champ_alpha, plot_gp_no).reshape(-1, 1),  # alpha
            np.repeat(pv_champ_beta, plot_gp_no).reshape(-1, 1),  # beta
            np.repeat(pv_champ_gamma_L, plot_gp_no).reshape(-1, 1),  # gamma_L
            np.repeat(pv_champ_lambda, plot_gp_no).reshape(-1, 1),  # lambda
            np.linspace(0, f_max, plot_gp_no, dtype=np.float64).reshape(-1, 1), # f
            np.repeat(pv_champ_r, plot_gp_no).reshape(-1, 1),  # r
        ),
        axis=1,
    ),
    "r_slice_samples": np.repeat(np.concatenate(
        (
            np.repeat(pv_champ_alpha, plot_samp_no).reshape(-1, 1),  # alpha
            np.repeat(pv_champ_beta, plot_samp_no).reshape(-1, 1),  # beta
            np.repeat(pv_champ_gamma_L, plot_samp_no).reshape(-1, 1),  # gamma_L
            np.repeat(pv_champ_lambda, plot_samp_no).reshape(-1, 1),  # lambda
            np.repeat(pv_champ_f, plot_samp_no).reshape(-1, 1),  # f
            np.linspace(0, r_max, plot_samp_no, dtype=np.float64).reshape(-1, 1), # r
```

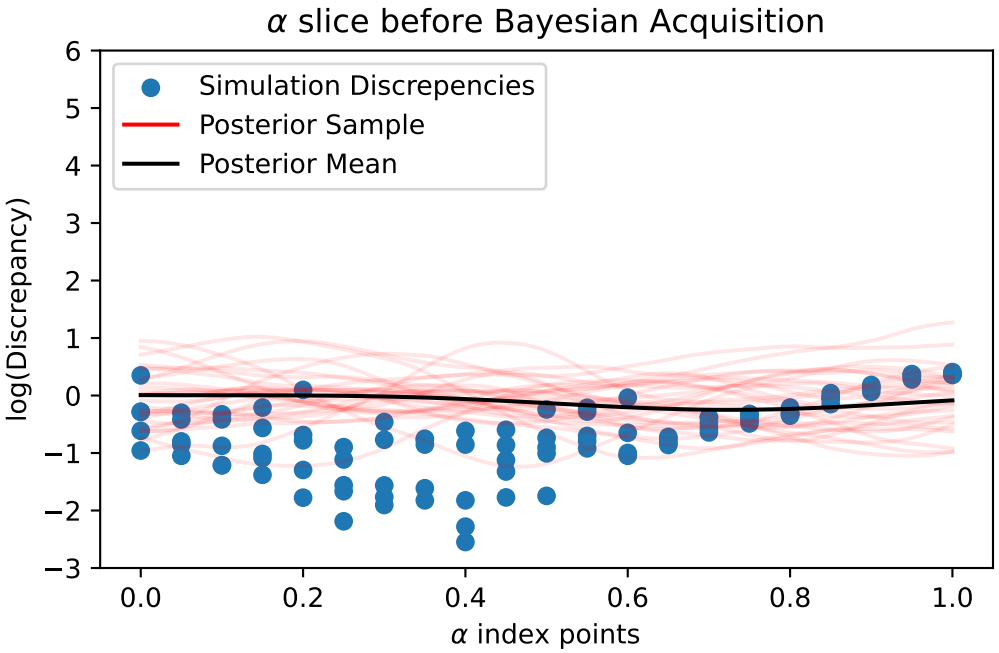
```
        ),
        axis=1,
    ), 5, axis = 0),
    "r_gp_samples": np.concatenate(
        (
            np.repeat(pv_champ_alpha, plot_gp_no).reshape(-1, 1),  # alpha
            np.repeat(pv_champ_beta, plot_gp_no).reshape(-1, 1),  # beta
            np.repeat(pv_champ_gamma_L, plot_gp_no).reshape(-1, 1),  # gamma_L
            np.repeat(pv_champ_lambda, plot_gp_no).reshape(-1, 1),  # lambda
            np.repeat(pv_champ_f, plot_gp_no).reshape(-1, 1),  # f
            np.linspace(0, r_max, plot_gp_no, dtype=np.float64).reshape(-1, 1), # r
        ),
        axis=1,
    ),
}
```

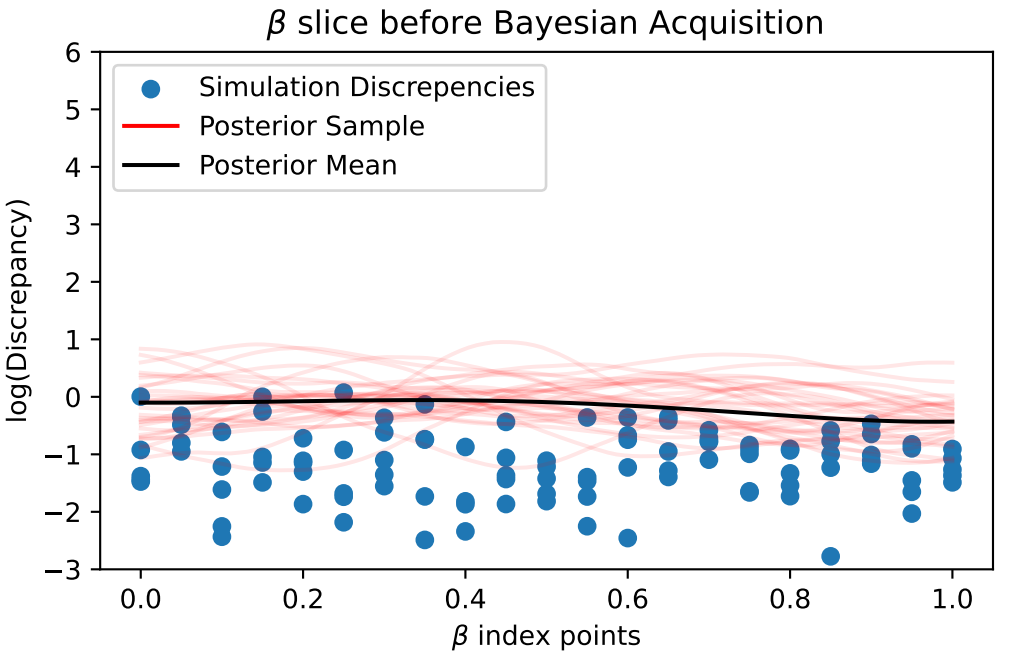
Plotting the GPs across different slices

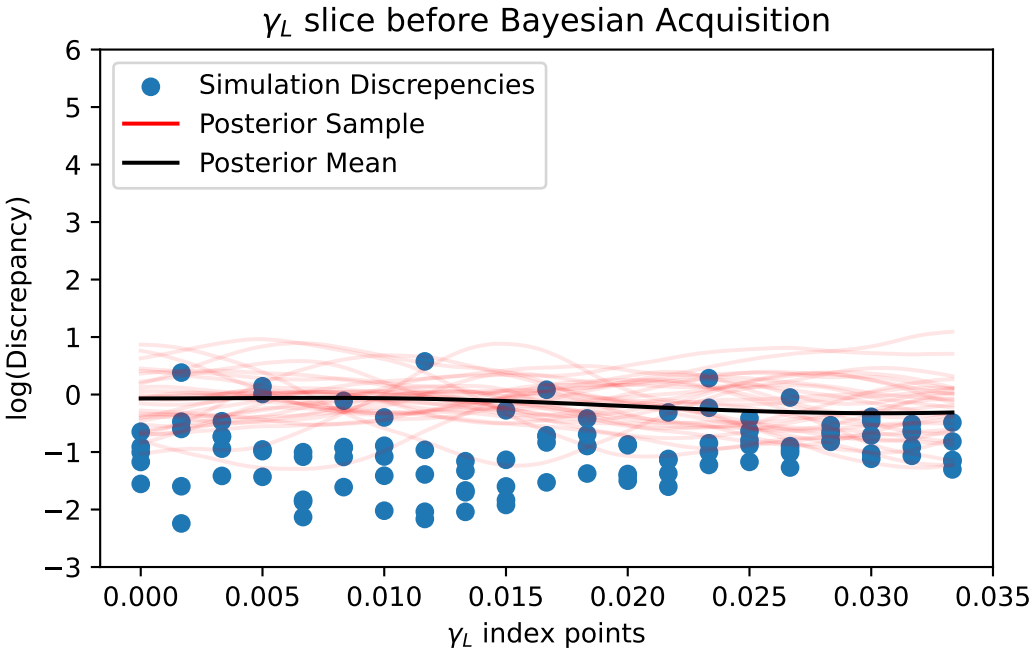
```
GP_seed = tfp.random.sanitize_seed(4362)
vars = ["alpha", "beta", "gamma_L", "lambda", "f", "r"]
slice_indices_dfs_dict = {}
slice_index_vals_dict = {}
slice_discrepencies_dict = {}

for var in vars:
    val_df = pd.DataFrame(
        slice_samples_dict[var + "_slice_samples"], columns=variables_names
    )
    slice_indices_dfs_dict[var + "_slice_indices_df"] = val_df
    slice_index_vals_dict[var + "_slice_index_vals"] = val_df.values
    discreps = val_df.apply(
        lambda x: discrepency_fn(
            x["alpha"], x["beta"], x["gamma_L"], x["lambda"], x["f"], x["r"], mean_of = 1
        ),
        axis=1,
    )
    slice_discrepencies_dict[var + "_slice_discrepencies"] = discreps


    gp_samples_df = pd.DataFrame(
```

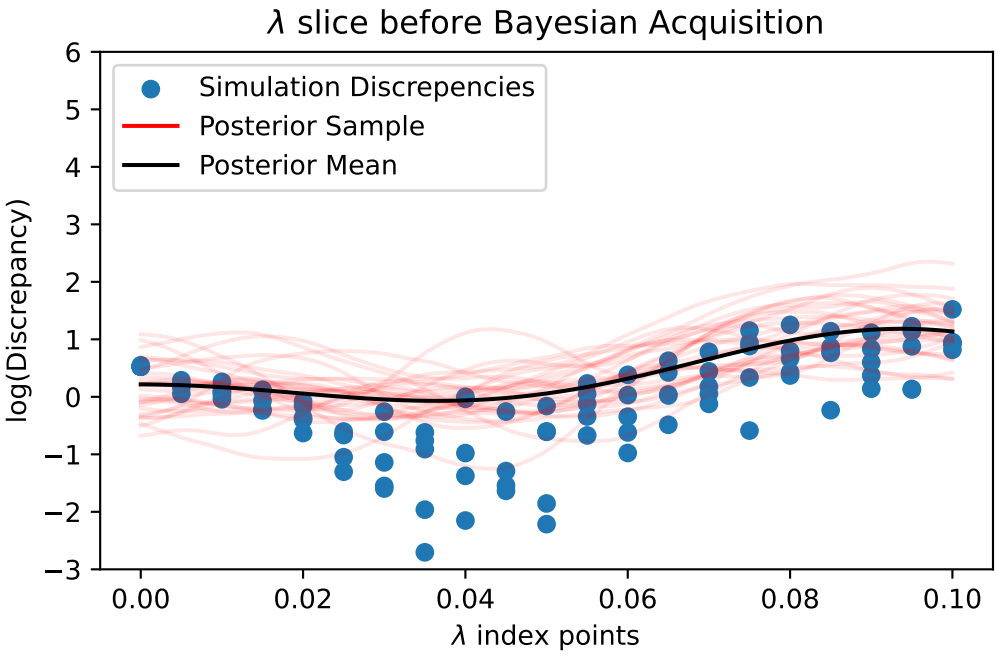
```
        slice_samples_dict[var + "_gp_samples"], columns=variables_names
    )
    slice_indices_dfs_dict[var + "_gp_indices_df"] = gp_samples_df
    slice_index_vals_dict[var + "_gp_index_vals"] = gp_samples_df.values

    champ_GP_reg_plot = tfd.GaussianProcessRegressionModel(
        kernel=kernel_champ,
        index_points=gp_samples_df.values,
        observation_index_points=index_vals,
        observations=obs_vals,
        observation_noise_variance=observation_noise_variance_champ,
        predictive_noise_variance=0.0,
        mean_fn=const_mean_fn(),
    )
    GP_samples = champ_GP_reg_plot.sample(gp_samp_no, seed=GP_seed)

    plt.figure(figsize=(6, 3.5))
    plt.scatter(
        val_df[var].values,
        discreps,
        label = "Simulation Discrepencies",
    )
    for i in range(gp_samp_no):
        plt.plot(
            gp_samples_df[var].values,
            GP_samples[i, :],
            c="r",
            alpha=0.1,
            label="Posterior Sample" if i == 0 else None,
        )
    plt.plot(
        slice_indices_dfs_dict[var + "_gp_indices_df"][var].values,
        champ_GP_reg_plot.mean_fn(slice_indices_dfs_dict[var + "_gp_indices_df"].values),
        c="black",
        alpha=1,
        label="Posterior Mean",
    )
    leg = plt.legend(loc="upper left")
    for lh in leg.legend_handles:
        lh.set_alpha(1)
    if var in ["f", "r"]:
        plt.xlabel("$" + var + "$ index points")
```

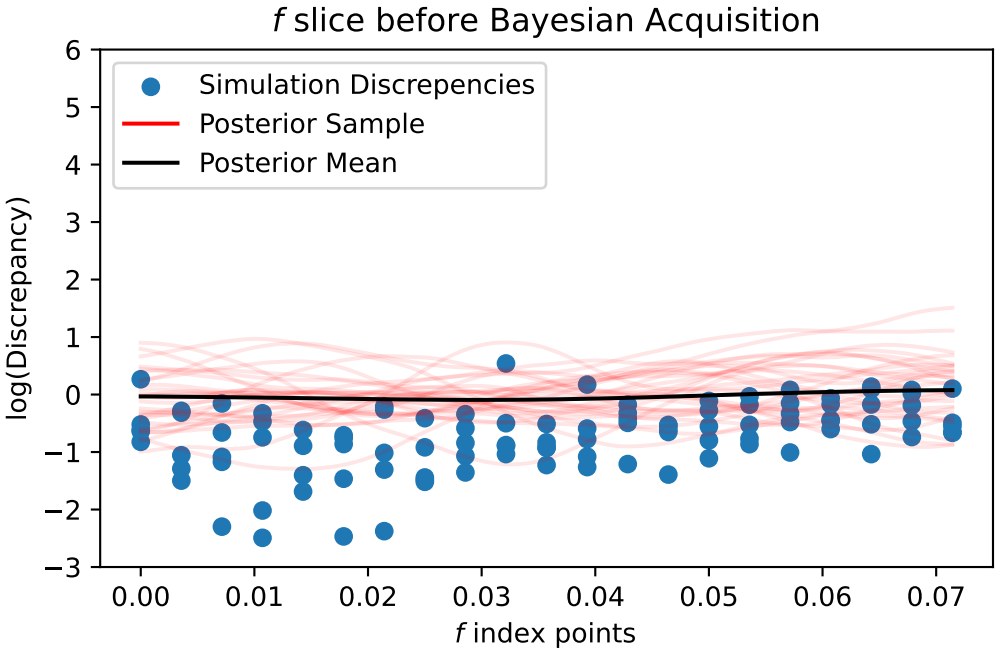
```
        plt.title("$" + var + "$ slice before Bayesian Acquisition")
    else:
        plt.xlabel("$\\" + var + "$ index points")
        plt.title("$\\" + var + "$ slice before Bayesian Acquisition")
    # if var not in ["alpha", "beta"]:
    #     plt.xscale("log", base=np.e)
    plt.ylabel("log(Discrepancy)")
    plt.ylim((-3, 6))
    plt.savefig("champagne_GP_images/initial_" + var + "_slice_log_discrep.pdf")
    plt.show()
```

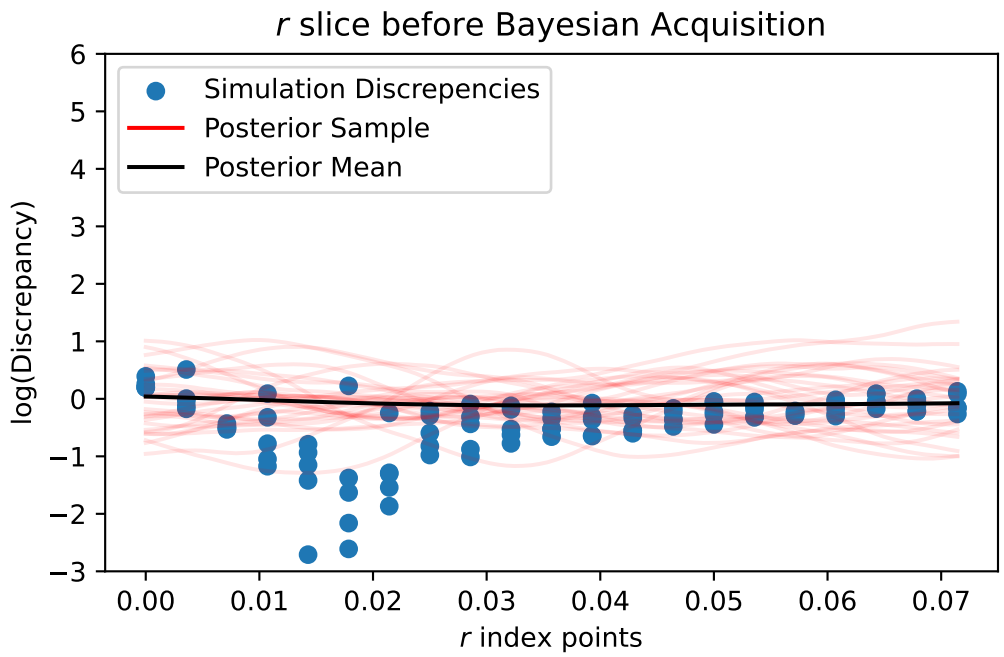



β slice before Bayesian Acquisition
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
β index points


γ_L slice before Bayesian Acquisition
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
γ_L index points


λ slice before Bayesian Acquisition
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
λ index points


f slice before Bayesian Acquisition
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
f index points



Acquiring the next datapoint to test

Proof that .variance returns what we need in acquisition function

```
champ_GP_reg = tfd.GaussianProcessRegressionModel(
    kernel=kernel_champ,
    observation_index_points=index_vals,
    observations=obs_vals,
    observation_noise_variance=observation_noise_variance_champ,
    mean_fn=const_mean_fn(),
)

new_guess = np.array([0.4, 0.4, 0.004, 0.04, 0.01, 0.17])
mean_t = champ_GP_reg.mean_fn(new_guess)
variance_t = champ_GP_reg.variance(index_points=[new_guess])

kernel_self = kernel_champ.apply(new_guess, new_guess)
kernel_others = kernel_champ.apply(new_guess, index_vals)
K = kernel_champ.matrix(
    index_vals, index_vals
```

```
) + observation_noise_variance_champ * np.identity(index_vals.shape[0])
inv_K = np.linalg.inv(K)
print("Self Kernel is {}".format(kernel_self.numpy().round(3)))
print("Others Kernel is {}".format(kernel_others.numpy().round(3)))
print(inv_K)
my_var_t = kernel_self - kernel_others.numpy() @ inv_K @ kernel_others.numpy()

print("Variance function is {}".format(variance_t.numpy().round(3)))
print("Variance function is {}".format(my_var_t.numpy().round(3)))
```

```
Self Kernel is 0.235
Others Kernel is [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0.]
[[ 5.13273674e+00 -5.96135849e-04 -3.11955890e-03 ... -2.80237180e-02
   1.78521706e-02 -1.18501103e-02]
 [-5.96135849e-04  4.38989512e+00  2.27069886e-02 ... -6.76215662e-03
   1.02255674e-03  2.32570724e-03]
 [-3.11955890e-03  2.27069886e-02  5.77459397e+00 ...  3.74078079e-02
   2.27154551e-03  5.09392507e-03]
 ...
 [-2.80237180e-02 -6.76215662e-03  3.74078079e-02 ...  4.41600916e+00
   9.38375714e-03 -2.43437171e-02]
 [ 1.78521706e-02  1.02255674e-03  2.27154551e-03 ...  9.38375714e-03
   5.09278409e+00  3.19185945e-03]
 [-1.18501103e-02  2.32570724e-03  5.09392507e-03 ... -2.43437171e-02
   3.19185945e-03  4.98016671e+00]]
Variance function is [0.24]
Variance function is 0.235
```

Loss function

```
next_alpha = tfp.util.TransformedVariable(
    initial_value=0.5,
    bijector=tfb.Sigmoid(),
    dtype=np.float64,
    name="next_alpha",
```

```
)

next_beta = tfp.util.TransformedVariable(
    initial_value=0.5,
    bijector=tfb.Sigmoid(),
    dtype=np.float64,
    name="next_beta",
)

next_gamma_L = tfp.util.TransformedVariable(
    initial_value=gamma_L_max/2,
    bijector=tfb.Sigmoid(np.float64(0.), gamma_L_max),
    dtype=np.float64,
    name="next_gamma_L",
)

next_lambda = tfp.util.TransformedVariable(
    initial_value=lambda_max/2,
    bijector=tfb.Sigmoid(np.float64(0.), lambda_max),
    dtype=np.float64,
    name="next_lambda",
)

next_f = tfp.util.TransformedVariable(
    initial_value=f_max/2,
    bijector=tfb.Sigmoid(np.float64(0.), f_max),
    dtype=np.float64,
    name="next_f",
)

next_r = tfp.util.TransformedVariable(
    initial_value=r_max/2,
    bijector=tfb.Sigmoid(np.float64(0.), r_max),
    dtype=np.float64,
    name="next_r",
)

next_vars = (
    (next_alpha.trainable_variables[0],
    next_beta.trainable_variables[0],
    next_gamma_L.trainable_variables[0],
    next_lambda.trainable_variables[0],
```

```
    next_f.trainable_variables[0],
    next_r.trainable_variables[0],)
)

next_vars
```

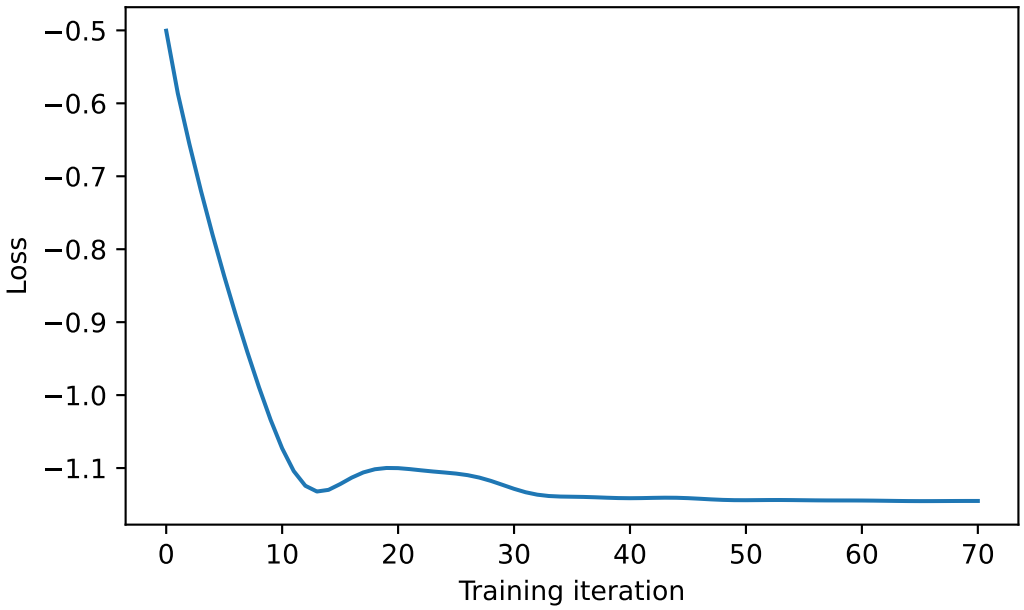
```
(<tf.Variable 'next_alpha:0' shape=() dtype=float64, numpy=0.0>,
 <tf.Variable 'next_beta:0' shape=() dtype=float64, numpy=0.0>,
 <tf.Variable 'next_gamma_L:0' shape=() dtype=float64, numpy=0.0>,
 <tf.Variable 'next_lambda:0' shape=() dtype=float64, numpy=0.0>,
 <tf.Variable 'next_f:0' shape=() dtype=float64, numpy=0.0>,
 <tf.Variable 'next_r:0' shape=() dtype=float64, numpy=0.0>)
```

```
eta_t = tf.constant(1.0, dtype=np.float64)

def UCB_loss(champ_GP_reg):
    next_guess = tf.reshape(
        tf.stack([next_alpha, next_beta, next_gamma_L, next_lambda, next_f, next_r]),
        [1, 6],
    )
    mean_t = champ_GP_reg.mean_fn(next_guess)
    std_t = tf.math.sqrt(
        champ_GP_reg.variance(index_points=next_guess)
        - observation_noise_variance_champ
    )
    return tf.squeeze(mean_t - std_t)


optimizer_fast = tf.keras.optimizers.Adam(learning_rate=0.1)

@tf.function(autograph=False, jit_compile=False)
def opt_var():
    with tf.GradientTape() as tape:
        loss = UCB_loss(champ_GP_reg)
    grads = tape.gradient(loss, next_vars)
    optimizer_fast.apply_gradients(zip(grads, next_vars))
    return loss

num_iters = 10000

lls_ = np.zeros(num_iters, np.float64)
```

```
tolerance = 1e-6  # Set your desired tolerance level
previous_loss = float("inf")

for i in range(num_iters):
    loss = opt_var()
    lls_[i] = loss

    # Check if change in loss is less than tolerance
    if abs(loss - previous_loss) < tolerance:
        print(f"Acquisition function convergence reached at iteration {i+1}.")
        lls_ = lls_[range(i + 1)]
        break

    previous_loss = loss

print("Trained parameters:")
for var in [next_alpha, next_beta, next_gamma_L, next_lambda, next_f, next_r]:
    print("{} is {}".format(var.name, (var.bijector.forward(var).numpy().round(3))))
```

```
Acquisition function convergence reached at iteration 71.
Trained parameters:
next_alpha is 0.574
next_beta is 0.569
next_gamma_L is 0.017
next_lambda is 0.051
next_f is 0.036
next_r is 0.036
```

```
plt.figure(figsize=(6, 3.5))
plt.plot(lls_)
plt.xlabel("Training iteration")
plt.ylabel("Loss")
plt.savefig("champagne_GP_images/bolfi_optim_loss_log_discrep.pdf")
plt.show()
```




```
def update_GP_LOO():
    @tf.function(autograph=False, jit_compile=False)
    def opt_GP():
        with tf.GradientTape() as tape:
            K = (
                champ_GP.kernel.matrix(index_vals, index_vals)
                + tf.eye(index_vals.shape[0], dtype=np.float64)
                * observation_noise_variance_champ
            )
            means = champ_GP.mean_fn(index_vals)
            K_inv = tf.linalg.inv(K)
            K_inv_y = K_inv @ tf.reshape(obs_vals - means, shape=[obs_vals.shape[0], 1])
            K_inv_diag = tf.linalg.diag_part(K_inv)
            log_var = tf.math.log(K_inv_diag)
            log_mu = tf.reshape(K_inv_y, shape=[-1]) ** 2
            loss = -tf.math.reduce_sum(log_var - log_mu)
        grads = tape.gradient(loss, champ_GP.trainable_variables)
        optimizer_slow.apply_gradients(zip(grads, champ_GP.trainable_variables))
        return loss

    num_iters = 10000

    lls_ = np.zeros(num_iters, np.float64)
    tolerance = 1e-6  # Set your desired tolerance level
```

```
    previous_loss = float("inf")

    for i in range(num_iters):
        loss = opt_GP()

        # Check if change in loss is less than tolerance
        if abs(loss - previous_loss) < tolerance:
            print(f"Hyperparameter convergence reached at iteration {i+1}.")
            break

        previous_loss = loss
    for var in optimizer_slow.variables:
        var.assign(tf.zeros_like(var))


def update_GP_MLE(champ_GP):
    @tf.function(autograph=False, jit_compile=False)
    def train_model():
        with tf.GradientTape() as tape:
            loss = -champ_GP.log_prob(obs_vals)
        grads = tape.gradient(loss, champ_GP.trainable_variables)
        optimizer_slow.apply_gradients(zip(grads, champ_GP.trainable_variables))
        return loss

    num_iters = 10000

    lls_ = np.zeros(num_iters, np.float64)
    tolerance = 1e-6  # Set your desired tolerance level
    previous_loss = float("inf")

    for i in range(num_iters):
        loss = train_model()

        # Check if change in loss is less than tolerance
        if abs(loss - previous_loss) < tolerance:
            print(f"Hyperparameter convergence reached at iteration {i+1}.")
            break

        previous_loss = loss
    for var in optimizer_slow.variables:
        var.assign(tf.zeros_like(var))
```

```
# def UCB_loss(eta_t, champ_GP_reg):
#     next_guess = tf.reshape(
#         tf.stack([next_alpha, next_beta, next_gamma_L, next_lambda, next_f, next_r]),
#         [1, 6],
#     )
#     mean_t = champ_GP_reg.mean_fn(next_guess)
#     std_t = champ_GP_reg.stddev(index_points=next_guess)
#     return tf.squeeze(mean_t - eta_t * std_t)


def update_var_UCB(eta_t, champ_GP_reg, next_vars):
    optimizer_fast = tf.keras.optimizers.Adam(learning_rate=0.1)

    @tf.function(autograph=False, jit_compile=False)
    def opt_var():
        with tf.GradientTape() as tape:
            loss = UCB_loss(eta_t, champ_GP_reg)
        grads = tape.gradient(loss, next_vars)
        optimizer_fast.apply_gradients(zip(grads, next_vars))
        return loss

    num_iters = 10000

    lls_ = np.zeros(num_iters, np.float64)
    tolerance = 1e-3  # Set your desired tolerance level
    previous_loss = float("inf")

    for i in range(num_iters):
        loss = opt_var()
        lls_[i] = loss

        # Check if change in loss is less than tolerance
        if abs(loss - previous_loss) < tolerance:
            print(f"Acquisition function convergence reached at iteration {i+1}.")
            break

        previous_loss = loss

    next_guess = tf.reshape(
        tf.stack([next_alpha, next_beta, next_gamma_L, next_lambda, next_f, next_r]),
        [1, 6],
    )
```

```
    print(
        "The final UCB loss was {}".format(loss.numpy().round(3))
        + " with predicted mean of {}".format(
            champ_GP_reg.mean_fn(next_guess).numpy().round(3)
        )
    )
    for var in optimizer_fast.variables:
        var.assign(tf.zeros_like(var))


def update_var_EI(GP_reg, alpha, beta, gamma_L, lambda_, f, r, min_obs):
    def EI_loss(alpha, beta, gamma_L, lambda_, f, r, min_obs):
        next_guess = tf.reshape(
            tf.stack([alpha, beta, gamma_L, lambda_, f, r]),
            [1, 6],
        )
        mean_t = GP_reg.mean_fn(next_guess)
        std_t = GP_reg.stddev(index_points=next_guess)
        delt = min_obs - mean_t
        return -tf.squeeze(
            delt * tfd.Normal(0, std_t).cdf(delt)
            + std_t * GP_reg.prob(delt, index_points=next_guess)
        )

    optimizer_fast = tf.keras.optimizers.Adam(learning_rate=0.1)

    @tf.function(autograph=False, jit_compile=False)
    def opt_var():
        with tf.GradientTape() as tape:
            loss = EI_loss(alpha, beta, gamma_L, lambda_, f, r, min_obs)
        grads = tape.gradient(loss, next_vars)
        optimizer_fast.apply_gradients(zip(grads, next_vars))
        return loss

    num_iters = 10000

    lls_ = np.zeros(num_iters, np.float64)
    tolerance = 1e-6  # Set your desired tolerance level
    previous_loss = np.float64("inf")

    for i in range(num_iters):
        loss = opt_var()
```

```
        lls_[i] = loss

        # Check if change in loss is less than tolerance
        if abs(loss - previous_loss) < tolerance:
            print(f"Acquisition function convergence reached at iteration {i+1}.")
            lls_ = lls_[range(i + 1)]
            break

        previous_loss = loss

    next_guess = tf.reshape(
        tf.stack([next_alpha, next_beta, next_gamma_L, next_lambda, next_f, next_r]),
        [1, 6],
    )
    print(
        "The final EI loss was {}".format(loss.numpy().round(3))
        + " with predicted mean of {}".format(
            champ_GP_reg.mean_fn(next_guess).numpy().round(3)
        )
    )


# update_var_EI(
#     champ_GP_reg, next_alpha, next_beta, next_gamma_L, next_lambda, next_f, next_r
# )
# EI = tfp_acq.GaussianProcessExpectedImprovement(champ_GP_reg, obs_vals)


def new_eta_t(t, d, exploration_rate):
    # return np.log((t + 1) ** (d * 2 + 2) * np.pi**2 / (3 * exploration_rate))
    return np.sqrt(np.log((t + 1) ** (d * 2 + 2) * np.pi**2 / (3 * exploration_rate)))
```

```
# optimizer_fast = tf.keras.optimizers.Adam(learning_rate=1.)
# update_var_EI()
# plt.figure(figsize=(6, 3.5))
# plt.plot(lls_)
# plt.xlabel("Training iteration")
# plt.ylabel("Loss")
# plt.show()
```

```
exploration_rate = 1
d = 6
update_GP_hp_freq = 20  # how many iterations before updating GP hyperparams
eta_t = tf.Variable(0, dtype=np.float64, name="eta_t")
min_obs = tf.Variable(100, dtype=np.float64, name="min_obs", shape=())
min_index = index_vals[
    champ_GP_reg.mean_fn(index_vals) == min(champ_GP_reg.mean_fn(index_vals))
][0]
simulation_reps = 20

for t in range(401):
    # min_index = index_vals[
    #     champ_GP_reg.mean_fn(index_vals) == min(champ_GP_reg.mean_fn(index_vals))
    # ][
    #     0,
    # ]
    optimizer_slow = tf.keras.optimizers.Adam()
    eta_t.assign(new_eta_t(t, d, exploration_rate))
    min_obs.assign(min(champ_GP_reg.mean_fn(index_vals)))
    print("Iteration " + str(t))
    # print(eta_t)

    ###########################################################################

    for var in [next_alpha, next_beta, next_gamma_L, next_lambda, next_f, next_r]:
        var.assign(
            var.bijector.forward(np.float64(100000000.0))
            * np.float64(np.random.uniform())
        )

    # update_var_UCB(eta_t, champ_GP_reg)
    update_var_EI(
        champ_GP_reg,
        next_alpha,
        next_beta,
        next_gamma_L,
        next_lambda,
        next_f,
        next_r,
        min_obs,
    )
```

```
    new_params = np.array(
        [
            next_alpha.numpy(),
            next_beta.numpy(),
            next_gamma_L.numpy(),
            next_lambda.numpy(),
            next_f.numpy(),
            next_r.numpy(),
        ]
    ).reshape(1, -1)
    print("The next parameters to simulate from are {}".format(new_params.round(3)))

    new_discrepency = discrepency_fn(
        next_alpha.numpy(),
        next_beta.numpy(),
        next_gamma_L.numpy(),
        next_lambda.numpy(),
        next_f.numpy(),
        next_r.numpy(),
    )

    index_vals = np.append(index_vals, new_params, axis=0)
    obs_vals = np.append(obs_vals, new_discrepency)

    print("The mean of the samples was {}".format(new_discrepency.round(3)))
    ##########################################################################

    champ_GP_reg = tfd.GaussianProcessRegressionModel(
        kernel=kernel_champ,
        observation_index_points=index_vals,
        observations=obs_vals,
        observation_noise_variance=observation_noise_variance_champ,
        predictive_noise_variance=0.0,
        mean_fn=const_mean_fn(),
    )

    if t % update_GP_hp_freq == 0:
        champ_GP = tfd.GaussianProcess(
            kernel=kernel_champ,
            observation_noise_variance=observation_noise_variance_champ,
            index_points=index_vals,
            mean_fn=const_mean_fn(),
```

```
        )
        # update_GP_LOO()
        update_GP_MLE(champ_GP)
        min_value = min(champ_GP_reg.mean_fn(index_vals))
        min_index = index_vals[champ_GP_reg.mean_fn(index_vals) == min_value][0,]
        print(
            "The minimum predicted mean of the observed indices is {}".format(
                min_value.numpy().round(3)
            )
            + " at the point \n{}".format(min_index.round(3))
        )

    if (t > 0) & (t % 50 == 0):
        print("Trained parameters:")
        for train_var in champ_GP.trainable_variables:
            if "bias" in train_var.name:
                print("{} is {}\n".format(train_var.name, train_var.numpy().round(3)))
            else:
                if "length" in train_var.name:
                    print(
                        "{} is {}\n".format(
                            train_var.name,
                            tfb.Sigmoid(
                                np.float64(0.0),
                                [
                                    1.0 / 4,
                                    1.0 / 4,
                                    gamma_L_max / 4,
                                    lambda_max / 4,
                                    f_max / 4,
                                    r_max / 4,
                                ],
                            )
                            .forward(train_var)
                            .numpy()
                            .round(3),
                        )
                    )
                else:
                    print(
                        "{} is {}\n".format(
                            train_var.name,
```

```
                        constrain_positive.forward(train_var).numpy().round(3),
                    )
                )

    for var in vars:
        champ_GP_reg_plot = tfd.GaussianProcessRegressionModel(
            kernel=kernel_champ,
            index_points=slice_indices_dfs_dict[var + "_gp_indices_df"].values,
            observation_index_points=index_vals,
            observations=obs_vals,
            observation_noise_variance=observation_noise_variance_champ,
            predictive_noise_variance=0.0,
            mean_fn=const_mean_fn(),
        )
        GP_samples = champ_GP_reg_plot.sample(gp_samp_no, seed=GP_seed)

        plt.figure(figsize=(6, 3.5))
        plt.scatter(
            slice_indices_dfs_dict[var + "_slice_indices_df"][var].values,
            slice_discrepencies_dict[var + "_slice_discrepencies"],
            label="Simulation Discrepencies",
        )
        for i in range(gp_samp_no):
            plt.plot(
                slice_indices_dfs_dict[var + "_gp_indices_df"][var].values,
                GP_samples[i, :],
                c="r",
                alpha=0.1,
                label="Posterior Sample" if i == 0 else None,
            )
        plt.plot(
            slice_indices_dfs_dict[var + "_gp_indices_df"][var].values,
            champ_GP_reg_plot.mean_fn(
                slice_indices_dfs_dict[var + "_gp_indices_df"].values
            ),
            c="black",
            alpha=1,
            label="Posterior Mean",
        )
        leg = plt.legend(loc="upper left")
        for lh in leg.legend_handles:
            lh.set_alpha(1)
```

```
            if var in ["f", "r"]:
                plt.xlabel("$" + var + "$ index points")
                plt.title(
                    "$" + var + "$ slice after " + str(t) + " Bayesian acquisitions"
                )
            else:
                plt.xlabel("$\\" + var + "$ index points")
                plt.title(
                    "$\\" + var + "$ slice after " + str(t) + " Bayesian acquisitions"
                )
            plt.ylabel("log(Discrepancy)")
            plt.ylim((-3, 6))
            plt.savefig(
                "champagne_GP_images/"
                + var
                + "_slice_"
                + str(t)
                + "_bolfi_updates_log_discrep.pdf"
            )
            plt.show()
```

```
Iteration 0
Acquisition function convergence reached at iteration 190.
The final EI loss was -0.351 with predicted mean of [-0.434]
The next parameters to simulate from are [[0.603 0.738 0.032 0.063 0.017 0.023]]
The mean of the samples was -0.134
The minimum predicted mean of the observed indices is -0.893 at the point
[0.237 0.288 0.032 0.023 0.032 0.037]
Iteration 1
Acquisition function convergence reached at iteration 234.
The final EI loss was -0.399 with predicted mean of [-0.447]
The next parameters to simulate from are [[0.601 0.754 0.032 0.061 0.011 0.022]]
The mean of the samples was -0.352
Iteration 2
Acquisition function convergence reached at iteration 183.
The final EI loss was -0.399 with predicted mean of [-0.446]
The next parameters to simulate from are [[0.473 0.035 0.022 0.061 0.01  0.042]]
The mean of the samples was -0.506
Iteration 3
Acquisition function convergence reached at iteration 85.
The final EI loss was -0.391 with predicted mean of [-0.444]
The next parameters to simulate from are [[0.541 0.919 0.007 0.041 0.016 0.034]]
```

```
The mean of the samples was -0.333
Iteration 4
Acquisition function convergence reached at iteration 70.
The final EI loss was -0.399 with predicted mean of [-0.446]
The next parameters to simulate from are [[0.741 0.049 0.024 0.058 0.004 0.027]]
The mean of the samples was -0.454
Iteration 5
Acquisition function convergence reached at iteration 14.
The final EI loss was 0.0 with predicted mean of [1.099]
The next parameters to simulate from are [[0.22  0.601 0.007 0.083 0.038 0.047]]
The mean of the samples was 1.156
Iteration 6
Acquisition function convergence reached at iteration 9.
The final EI loss was 0.0 with predicted mean of [1.246]
The next parameters to simulate from are [[0.147 0.051 0.003 0.098 0.042 0.037]]
The mean of the samples was 1.668
Iteration 7
Acquisition function convergence reached at iteration 179.
The final EI loss was -0.399 with predicted mean of [-0.447]
The next parameters to simulate from are [[0.777 0.062 0.02  0.05  0.069 0.048]]
The mean of the samples was -0.463
Iteration 8
Acquisition function convergence reached at iteration 249.
The final EI loss was -0.037 with predicted mean of [0.016]
The next parameters to simulate from are [[1.    0.996 0.    0.1   0.071 0.071]]
The mean of the samples was 0.399
Iteration 9
Acquisition function convergence reached at iteration 43.
The final EI loss was -0.328 with predicted mean of [-0.414]
The next parameters to simulate from are [[0.272 0.497 0.03  0.011 0.048 0.038]]
The mean of the samples was -0.164
Iteration 10
Acquisition function convergence reached at iteration 219.
The final EI loss was -0.399 with predicted mean of [-0.446]
The next parameters to simulate from are [[0.528 0.971 0.006 0.042 0.015 0.032]]
The mean of the samples was -0.371
Iteration 11
Acquisition function convergence reached at iteration 125.
The final EI loss was -0.392 with predicted mean of [-0.449]
The next parameters to simulate from are [[0.786 0.603 0.009 0.019 0.038 0.014]]
The mean of the samples was -0.084
Iteration 12
Acquisition function convergence reached at iteration 203.
```

```
The final EI loss was -0.399 with predicted mean of [-0.446]
The next parameters to simulate from are [[0.484 0.017 0.021 0.062 0.009 0.044]]
The mean of the samples was -0.422
Iteration 13
Acquisition function convergence reached at iteration 137.
The final EI loss was -0.395 with predicted mean of [-0.438]
The next parameters to simulate from are [[0.059 0.677 0.028 0.016 0.036 0.046]]
The mean of the samples was -0.536
Iteration 14
Acquisition function convergence reached at iteration 249.
The final EI loss was -0.399 with predicted mean of [-0.447]
The next parameters to simulate from are [[0.279 0.231 0.028 0.021 0.028 0.052]]
The mean of the samples was -0.531
Iteration 15
Acquisition function convergence reached at iteration 141.
The final EI loss was -0.392 with predicted mean of [-0.444]
The next parameters to simulate from are [[0.253 0.452 0.031 0.016 0.043 0.037]]
The mean of the samples was -0.545
Iteration 16
Acquisition function convergence reached at iteration 484.
The final EI loss was -0.397 with predicted mean of [-0.453]
The next parameters to simulate from are [[0.798 0.629 0.008 0.023 0.043 0.012]]
The mean of the samples was -0.296
Iteration 17
Acquisition function convergence reached at iteration 162.
The final EI loss was -0.399 with predicted mean of [-0.446]
The next parameters to simulate from are [[0.392 0.805 0.031 0.022 0.011 0.026]]
The mean of the samples was -0.421
Iteration 18
Acquisition function convergence reached at iteration 195.
The final EI loss was -0.399 with predicted mean of [-0.447]
The next parameters to simulate from are [[0.557 0.984 0.006 0.04  0.015 0.029]]
The mean of the samples was -0.325
Iteration 19
Acquisition function convergence reached at iteration 167.
The final EI loss was -0.397 with predicted mean of [-0.442]
The next parameters to simulate from are [[0.776 0.673 0.008 0.026 0.046 0.009]]
The mean of the samples was -0.66
Iteration 20
Acquisition function convergence reached at iteration 101.
The final EI loss was -0.399 with predicted mean of [-0.446]
The next parameters to simulate from are [[0.103 0.36  0.007 0.009 0.038 0.05 ]]
The mean of the samples was -0.427
```

```
Hyperparameter convergence reached at iteration 3215.
The minimum predicted mean of the observed indices is -0.893 at the point
[0.237 0.288 0.032 0.023 0.032 0.037]
Iteration 21
Acquisition function convergence reached at iteration 109.
The final EI loss was -0.399 with predicted mean of [-0.447]
The next parameters to simulate from are [[0.524 0.921 0.007 0.045 0.015 0.032]]
The mean of the samples was -0.428
Iteration 22
Acquisition function convergence reached at iteration 201.
The final EI loss was -0.399 with predicted mean of [-0.446]
The next parameters to simulate from are [[0.519 0.893 0.008 0.05  0.015 0.03 ]]
The mean of the samples was -0.475
Iteration 23
Acquisition function convergence reached at iteration 8.
The final EI loss was 0.0 with predicted mean of [1.173]
The next parameters to simulate from are [[0.197 0.63  0.009 0.085 0.004 0.01 ]]
The mean of the samples was 1.268
Iteration 24
Acquisition function convergence reached at iteration 99.
The final EI loss was -0.399 with predicted mean of [-0.447]
The next parameters to simulate from are [[0.447 0.247 0.027 0.034 0.051 0.013]]
The mean of the samples was -0.517
Iteration 25
Acquisition function convergence reached at iteration 118.
The final EI loss was -0.399 with predicted mean of [-0.446]
The next parameters to simulate from are [[0.339 0.146 0.027 0.023 0.031 0.058]]
The mean of the samples was -0.67
Iteration 26
Acquisition function convergence reached at iteration 120.
The final EI loss was -0.399 with predicted mean of [-0.447]
The next parameters to simulate from are [[0.432 0.799 0.032 0.023 0.012 0.027]]
The mean of the samples was -0.433
Iteration 27
Acquisition function convergence reached at iteration 116.
The final EI loss was -0.399 with predicted mean of [-0.446]
The next parameters to simulate from are [[0.34  0.821 0.031 0.022 0.012 0.027]]
The mean of the samples was -0.4
Iteration 28
Acquisition function convergence reached at iteration 74.
The final EI loss was -0.399 with predicted mean of [-0.444]
The next parameters to simulate from are [[0.529 0.923 0.007 0.042 0.012 0.032]]
The mean of the samples was -0.372
```

```
Iteration 29
Acquisition function convergence reached at iteration 96.
The final EI loss was -0.396 with predicted mean of [-0.448]
The next parameters to simulate from are [[0.44  0.244 0.014 0.026 0.038 0.063]]
The mean of the samples was -0.411
Iteration 30
Acquisition function convergence reached at iteration 129.
The final EI loss was -0.399 with predicted mean of [-0.446]
The next parameters to simulate from are [[0.382 0.849 0.032 0.022 0.014 0.027]]
The mean of the samples was -0.376
Iteration 31
Acquisition function convergence reached at iteration 110.
The final EI loss was -0.399 with predicted mean of [-0.444]
The next parameters to simulate from are [[0.378 0.736 0.031 0.021 0.009 0.025]]
The mean of the samples was -0.383
Iteration 32
Acquisition function convergence reached at iteration 17.
The final EI loss was 0.0 with predicted mean of [0.996]
The next parameters to simulate from are [[0.109 0.618 0.019 0.085 0.069 0.065]]
The mean of the samples was 1.397
Iteration 33
Acquisition function convergence reached at iteration 11.
The final EI loss was 0.0 with predicted mean of [0.801]
The next parameters to simulate from are [[0.258 0.25  0.012 0.076 0.025 0.039]]
The mean of the samples was 0.669
Iteration 34
Acquisition function convergence reached at iteration 198.
The final EI loss was -0.399 with predicted mean of [-0.446]
The next parameters to simulate from are [[0.061 0.704 0.029 0.014 0.039 0.047]]
The mean of the samples was -0.41
Iteration 35
Acquisition function convergence reached at iteration 3.
The final EI loss was 0.0 with predicted mean of [1.28]
The next parameters to simulate from are [[0.235 0.516 0.005 0.083 0.024 0.   ]]
The mean of the samples was 1.655
Iteration 36
Acquisition function convergence reached at iteration 414.
The final EI loss was -0.399 with predicted mean of [-0.446]
The next parameters to simulate from are [[0.41  0.77  0.032 0.025 0.009 0.026]]
The mean of the samples was -0.577
Iteration 37
Acquisition function convergence reached at iteration 101.
The final EI loss was -0.399 with predicted mean of [-0.447]
```

```
The next parameters to simulate from are [[0.491 0.875 0.008 0.046 0.016 0.033]]
The mean of the samples was -0.411
Iteration 38
Acquisition function convergence reached at iteration 12.
The final EI loss was 0.0 with predicted mean of [0.897]
The next parameters to simulate from are [[0.125 0.224 0.003 0.083 0.036 0.067]]
The mean of the samples was 1.236
Iteration 39
Acquisition function convergence reached at iteration 8.
The final EI loss was 0.0 with predicted mean of [1.085]
The next parameters to simulate from are [[0.128 0.633 0.016 0.089 0.03  0.05 ]]
The mean of the samples was 1.276
Iteration 40
Acquisition function convergence reached at iteration 119.
The final EI loss was -0.399 with predicted mean of [-0.447]
The next parameters to simulate from are [[0.796 0.065 0.021 0.051 0.07  0.049]]
The mean of the samples was -0.436
Hyperparameter convergence reached at iteration 1584.
The minimum predicted mean of the observed indices is -0.893 at the point
[0.237 0.288 0.032 0.023 0.032 0.037]
Iteration 41
Acquisition function convergence reached at iteration 146.
The final EI loss was -0.399 with predicted mean of [-0.446]
The next parameters to simulate from are [[0.512 0.938 0.007 0.051 0.015 0.033]]
The mean of the samples was -0.404
Iteration 42
Acquisition function convergence reached at iteration 108.
The final EI loss was -0.398 with predicted mean of [-0.443]
The next parameters to simulate from are [[0.483 0.135 0.029 0.053 0.019 0.061]]
The mean of the samples was -0.382
Iteration 43
Acquisition function convergence reached at iteration 258.
The final EI loss was -0.399 with predicted mean of [-0.445]
The next parameters to simulate from are [[0.468 0.093 0.024 0.062 0.01  0.04 ]]
The mean of the samples was -0.357
Iteration 44
Acquisition function convergence reached at iteration 182.
The final EI loss was -0.399 with predicted mean of [-0.446]
The next parameters to simulate from are [[0.55  0.896 0.007 0.053 0.017 0.029]]
The mean of the samples was -0.256
Iteration 45
Acquisition function convergence reached at iteration 902.
The final EI loss was -0.399 with predicted mean of [-0.444]
```

```
The next parameters to simulate from are [[0.734 0.598 0.009 0.018 0.039 0.01 ]]
The mean of the samples was -0.355
Iteration 46
Acquisition function convergence reached at iteration 234.
The final EI loss was -0.399 with predicted mean of [-0.447]
The next parameters to simulate from are [[0.382 0.091 0.027 0.022 0.03  0.06 ]]
The mean of the samples was -0.444
Iteration 47
Acquisition function convergence reached at iteration 98.
The final EI loss was -0.396 with predicted mean of [-0.448]
The next parameters to simulate from are [[0.68  0.11  0.001 0.048 0.035 0.022]]
The mean of the samples was -0.551
Iteration 48
Acquisition function convergence reached at iteration 9.
The final EI loss was 0.0 with predicted mean of [0.627]
The next parameters to simulate from are [[0.61  0.95  0.03  0.086 0.038 0.02 ]]
The mean of the samples was 0.701
Iteration 49
Acquisition function convergence reached at iteration 174.
The final EI loss was -0.382 with predicted mean of [-0.447]
The next parameters to simulate from are [[0.087 0.611 0.004 0.047 0.011 0.031]]
The mean of the samples was -0.081
Iteration 50
Acquisition function convergence reached at iteration 117.
The final EI loss was -0.399 with predicted mean of [-0.447]
The next parameters to simulate from are [[0.764 0.029 0.022 0.052 0.069 0.052]]
The mean of the samples was -0.373
Trained parameters:
amplitude_champ:0 is 0.457

length_scales_champ:0 is [0.25  0.25  0.008 0.019 0.018 0.018]

observation_noise_variance_champ:0 is 0.0

bias_mean:0 is 0.323

Iteration 51
Acquisition function convergence reached at iteration 101.
The final EI loss was -0.385 with predicted mean of [-0.444]
The next parameters to simulate from are [[0.187 0.883 0.02  0.03  0.046 0.013]]
The mean of the samples was -0.495
Iteration 52
Acquisition function convergence reached at iteration 8.
```

```
The final EI loss was 0.0 with predicted mean of [1.048]
The next parameters to simulate from are [[0.269 0.687 0.019 0.087 0.053 0.031]]
The mean of the samples was 1.254
Iteration 53
Acquisition function convergence reached at iteration 343.
The final EI loss was -0.399 with predicted mean of [-0.446]
The next parameters to simulate from are [[0.835 0.685 0.008 0.025 0.047 0.007]]
The mean of the samples was -0.519
Iteration 54
Acquisition function convergence reached at iteration 9.
The final EI loss was 0.0 with predicted mean of [1.058]
The next parameters to simulate from are [[0.171 0.641 0.026 0.08  0.041 0.04 ]]
The mean of the samples was 0.949
Iteration 55
Acquisition function convergence reached at iteration 78.
The final EI loss was -0.399 with predicted mean of [-0.446]
The next parameters to simulate from are [[0.737 0.02  0.021 0.049 0.071 0.05 ]]
The mean of the samples was -0.427
Iteration 56
Acquisition function convergence reached at iteration 193.
The final EI loss was -0.399 with predicted mean of [-0.447]
The next parameters to simulate from are [[0.394 0.237 0.014 0.027 0.037 0.064]]
The mean of the samples was -0.411
Iteration 57
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [1.294]
The next parameters to simulate from are [[0.216 0.655 0.013 0.085 0.047 0.029]]
The mean of the samples was 1.226
Iteration 58
Acquisition function convergence reached at iteration 32.
The final EI loss was -0.391 with predicted mean of [-0.446]
The next parameters to simulate from are [[0.18  0.766 0.022 0.021 0.061 0.017]]
The mean of the samples was -0.435
Iteration 59
Acquisition function convergence reached at iteration 105.
The final EI loss was -0.399 with predicted mean of [-0.446]
The next parameters to simulate from are [[0.378 0.212 0.027 0.034 0.047 0.013]]
The mean of the samples was -0.506
Iteration 60
Acquisition function convergence reached at iteration 107.
The final EI loss was -0.396 with predicted mean of [-0.447]
The next parameters to simulate from are [[0.734 0.355 0.001 0.069 0.004 0.007]]
The mean of the samples was -0.069
```

```
Hyperparameter convergence reached at iteration 901.
The minimum predicted mean of the observed indices is -0.893 at the point
[0.237 0.288 0.032 0.023 0.032 0.037]
Iteration 61
Acquisition function convergence reached at iteration 15.
The final EI loss was 0.0 with predicted mean of [0.78]
The next parameters to simulate from are [[0.704 0.445 0.016 0.092 0.045 0.021]]
The mean of the samples was 0.75
Iteration 62
Acquisition function convergence reached at iteration 508.
The final EI loss was -0.399 with predicted mean of [-0.446]
The next parameters to simulate from are [[0.212 0.273 0.03  0.02  0.023 0.048]]
The mean of the samples was -0.47
Iteration 63
Acquisition function convergence reached at iteration 98.
The final EI loss was -0.399 with predicted mean of [-0.446]
The next parameters to simulate from are [[0.115 0.382 0.007 0.013 0.035 0.051]]
The mean of the samples was -0.535
Iteration 64
Acquisition function convergence reached at iteration 6.
The final EI loss was 0.0 with predicted mean of [0.735]
The next parameters to simulate from are [[0.56  0.55  0.025 0.082 0.039 0.026]]
The mean of the samples was 0.833
Iteration 65
Acquisition function convergence reached at iteration 4.
The final EI loss was 0.0 with predicted mean of [0.996]
The next parameters to simulate from are [[0.201 0.702 0.016 0.078 0.06  0.026]]
The mean of the samples was 1.214
Iteration 66
Acquisition function convergence reached at iteration 303.
The final EI loss was -0.03 with predicted mean of [0.007]
The next parameters to simulate from are [[0.011 0.998 0.033 0.001 0.071 0.071]]
The mean of the samples was 0.164
Iteration 67
Acquisition function convergence reached at iteration 5.
The final EI loss was 0.0 with predicted mean of [1.069]
The next parameters to simulate from are [[0.218 0.292 0.001 0.097 0.032 0.022]]
The mean of the samples was 1.713
Iteration 68
Acquisition function convergence reached at iteration 127.
The final EI loss was -0.399 with predicted mean of [-0.447]
The next parameters to simulate from are [[0.775 0.059 0.021 0.049 0.067 0.046]]
The mean of the samples was -0.432
```

```
Iteration 69
Acquisition function convergence reached at iteration 112.
The final EI loss was -0.399 with predicted mean of [-0.447]
The next parameters to simulate from are [[0.672 0.025 0.021 0.047 0.071 0.051]]
The mean of the samples was -0.264
Iteration 70
Acquisition function convergence reached at iteration 136.
The final EI loss was -0.399 with predicted mean of [-0.447]
The next parameters to simulate from are [[0.343 0.771 0.03  0.024 0.006 0.026]]
The mean of the samples was -0.36
Iteration 71
Acquisition function convergence reached at iteration 121.
The final EI loss was -0.399 with predicted mean of [-0.447]
The next parameters to simulate from are [[0.106 0.426 0.007 0.009 0.04  0.05 ]]
The mean of the samples was -0.486
Iteration 72
Acquisition function convergence reached at iteration 561.
The final EI loss was -0.399 with predicted mean of [-0.447]
The next parameters to simulate from are [[0.088 0.628 0.003 0.041 0.01  0.03 ]]
The mean of the samples was -0.655
Iteration 73
Acquisition function convergence reached at iteration 166.
The final EI loss was -0.399 with predicted mean of [-0.447]
The next parameters to simulate from are [[0.38  0.762 0.031 0.027 0.002 0.025]]
The mean of the samples was -0.566
Iteration 74
Acquisition function convergence reached at iteration 13.
The final EI loss was 0.0 with predicted mean of [0.408]
The next parameters to simulate from are [[0.895 0.531 0.008 0.016 0.023 0.028]]
The mean of the samples was 0.353
Iteration 75
Acquisition function convergence reached at iteration 238.
The final EI loss was -0.399 with predicted mean of [-0.448]
The next parameters to simulate from are [[0.657 0.122 0.001 0.052 0.038 0.024]]
The mean of the samples was -0.277
Iteration 76
Acquisition function convergence reached at iteration 113.
The final EI loss was -0.399 with predicted mean of [-0.446]
The next parameters to simulate from are [[0.148 0.338 0.006 0.01  0.033 0.052]]
The mean of the samples was -0.292
Iteration 77
Acquisition function convergence reached at iteration 137.
The final EI loss was -0.399 with predicted mean of [-0.446]
```

```
The next parameters to simulate from are [[0.765 0.003 0.021 0.046 0.071 0.046]]
The mean of the samples was -0.694
Iteration 78
Acquisition function convergence reached at iteration 208.
The final EI loss was -0.399 with predicted mean of [-0.447]
The next parameters to simulate from are [[0.349 0.198 0.027 0.035 0.046 0.01 ]]
The mean of the samples was -0.222
Iteration 79
Acquisition function convergence reached at iteration 93.
The final EI loss was -0.399 with predicted mean of [-0.446]
The next parameters to simulate from are [[0.094 0.335 0.006 0.013 0.032 0.051]]
The mean of the samples was -0.61
Iteration 80
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [1.216]
The next parameters to simulate from are [[0.206 0.248 0.005 0.078 0.029 0.027]]
The mean of the samples was 1.012
Hyperparameter convergence reached at iteration 3744.
The minimum predicted mean of the observed indices is -0.892 at the point
[0.237 0.288 0.032 0.023 0.032 0.037]
Iteration 81
Acquisition function convergence reached at iteration 94.
The final EI loss was -0.399 with predicted mean of [-0.446]
The next parameters to simulate from are [[0.045 0.417 0.007 0.008 0.044 0.05 ]]
The mean of the samples was -0.535
Iteration 82
Acquisition function convergence reached at iteration 17.
The final EI loss was 0.0 with predicted mean of [0.987]
The next parameters to simulate from are [[0.233 0.062 0.009 0.087 0.039 0.059]]
The mean of the samples was 1.138
Iteration 83
Acquisition function convergence reached at iteration 165.
The final EI loss was -0.399 with predicted mean of [-0.447]
The next parameters to simulate from are [[0.456 0.016 0.021 0.059 0.007 0.047]]
The mean of the samples was -0.415
Iteration 84
Acquisition function convergence reached at iteration 16.
The final EI loss was 0.0 with predicted mean of [0.707]
The next parameters to simulate from are [[0.16  0.458 0.017 0.065 0.032 0.013]]
The mean of the samples was 0.797
Iteration 85
Acquisition function convergence reached at iteration 127.
The final EI loss was -0.399 with predicted mean of [-0.446]
```

```
The next parameters to simulate from are [[0.038 0.676 0.029 0.012 0.034 0.05 ]]
The mean of the samples was -0.296
Iteration 86
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [1.002]
The next parameters to simulate from are [[0.376 0.558 0.024 0.084 0.044 0.04 ]]
The mean of the samples was 1.0
Iteration 87
Acquisition function convergence reached at iteration 87.
The final EI loss was -0.368 with predicted mean of [-0.437]
The next parameters to simulate from are [[0.361 0.891 0.009 0.041 0.025 0.016]]
The mean of the samples was -1.06
Iteration 88
Acquisition function convergence reached at iteration 8.
The final EI loss was 0.0 with predicted mean of [0.813]
The next parameters to simulate from are [[0.596 0.428 0.001 0.091 0.041 0.028]]
The mean of the samples was 0.921
Iteration 89
Acquisition function convergence reached at iteration 10.
The final EI loss was 0.0 with predicted mean of [0.957]
The next parameters to simulate from are [[0.255 0.442 0.001 0.082 0.047 0.061]]
The mean of the samples was 1.161
Iteration 90
Acquisition function convergence reached at iteration 246.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.196 0.286 0.032 0.018 0.025 0.046]]
The mean of the samples was -0.514
Iteration 91
Acquisition function convergence reached at iteration 153.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.874 0.705 0.007 0.026 0.049 0.003]]
The mean of the samples was -0.783
Iteration 92
Acquisition function convergence reached at iteration 114.
The final EI loss was -0.399 with predicted mean of [-0.53]
The next parameters to simulate from are [[0.264 0.118 0.02  0.033 0.027 0.011]]
The mean of the samples was -0.329
Iteration 93
Acquisition function convergence reached at iteration 137.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.194 0.272 0.03  0.024 0.022 0.05 ]]
The mean of the samples was -0.744
Iteration 94
```

```
Acquisition function convergence reached at iteration 316.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.914 0.679 0.008 0.023 0.047 0.001]]
The mean of the samples was -0.407
Iteration 95
Acquisition function convergence reached at iteration 135.
The final EI loss was -0.399 with predicted mean of [-0.53]
The next parameters to simulate from are [[0.382 0.191 0.028 0.03  0.049 0.012]]
The mean of the samples was -0.361
Iteration 96
Acquisition function convergence reached at iteration 211.
The final EI loss was -0.399 with predicted mean of [-0.531]
The next parameters to simulate from are [[0.766 0.042 0.021 0.044 0.063 0.045]]
The mean of the samples was -0.549
Iteration 97
Acquisition function convergence reached at iteration 91.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.031 0.657 0.028 0.018 0.036 0.05 ]]
The mean of the samples was -0.666
Iteration 98
Acquisition function convergence reached at iteration 12.
The final EI loss was 0.0 with predicted mean of [0.841]
The next parameters to simulate from are [[0.227 0.879 0.008 0.08  0.048 0.044]]
The mean of the samples was 0.982
Iteration 99
Acquisition function convergence reached at iteration 3.
The final EI loss was 0.0 with predicted mean of [0.45]
The next parameters to simulate from are [[0.716 0.49  0.029 0.078 0.041 0.019]]
The mean of the samples was 0.703
Iteration 100
Acquisition function convergence reached at iteration 380.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.552 0.984 0.005 0.039 0.014 0.021]]
The mean of the samples was -0.611
Hyperparameter convergence reached at iteration 881.
The minimum predicted mean of the observed indices is -1.058 at the point
[0.361 0.891 0.009 0.041 0.025 0.016]
Trained parameters:
amplitude_champ:0 is 0.447

length_scales_champ:0 is [0.25  0.25  0.008 0.017 0.018 0.018]

observation_noise_variance_champ:0 is 0.0
```

```
bias_mean:0 is 0.313

Iteration 101
Acquisition function convergence reached at iteration 414.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.951 0.708 0.009 0.025 0.051 0.002]]
The mean of the samples was -0.364
Iteration 102
Acquisition function convergence reached at iteration 177.
The final EI loss was -0.399 with predicted mean of [-0.526]
The next parameters to simulate from are [[0.477 0.773 0.03  0.026 0.016 0.024]]
The mean of the samples was -0.545
Iteration 103
Acquisition function convergence reached at iteration 152.
The final EI loss was -0.399 with predicted mean of [-0.531]
The next parameters to simulate from are [[0.448 0.247 0.027 0.038 0.051 0.016]]
The mean of the samples was -0.605
Iteration 104
Acquisition function convergence reached at iteration 202.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.604 0.77  0.033 0.06  0.009 0.017]]
The mean of the samples was -0.585
Iteration 105
Acquisition function convergence reached at iteration 99.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.172 0.862 0.02  0.027 0.055 0.014]]
The mean of the samples was -0.51
Iteration 106
Acquisition function convergence reached at iteration 284.
The final EI loss was -0.399 with predicted mean of [-0.53]
The next parameters to simulate from are [[0.931 0.729 0.009 0.029 0.055 0.002]]
The mean of the samples was -0.924
Iteration 107
Acquisition function convergence reached at iteration 168.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.232 0.29  0.03  0.017 0.03  0.042]]
The mean of the samples was -0.465
Iteration 108
Acquisition function convergence reached at iteration 364.
The final EI loss was -0.397 with predicted mean of [-0.525]
The next parameters to simulate from are [[0.732 0.605 0.008 0.027 0.043 0.016]]
The mean of the samples was -0.49
```

```
Iteration 109
Acquisition function convergence reached at iteration 149.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.291 0.131 0.019 0.03  0.027 0.013]]
The mean of the samples was -0.536
Iteration 110
Acquisition function convergence reached at iteration 213.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.099 0.643 0.003 0.038 0.008 0.04 ]]
The mean of the samples was -0.517
Iteration 111
Acquisition function convergence reached at iteration 3.
The final EI loss was 0.0 with predicted mean of [1.017]
The next parameters to simulate from are [[0.036 0.661 0.017 0.076 0.03  0.009]]
The mean of the samples was 1.213
Iteration 112
Acquisition function convergence reached at iteration 74.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.025 0.367 0.006 0.009 0.04  0.052]]
The mean of the samples was -0.514
Iteration 113
Acquisition function convergence reached at iteration 139.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.379 0.222 0.013 0.03  0.037 0.059]]
The mean of the samples was -0.555
Iteration 114
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.835]
The next parameters to simulate from are [[0.514 0.484 0.015 0.098 0.041 0.007]]
The mean of the samples was 1.332
Iteration 115
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.9]
The next parameters to simulate from are [[0.202 0.219 0.004 0.077 0.028 0.035]]
The mean of the samples was 0.856
Iteration 116
Acquisition function convergence reached at iteration 92.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.123 0.629 0.002 0.037 0.006 0.04 ]]
The mean of the samples was -0.367
Iteration 117
Acquisition function convergence reached at iteration 205.
The final EI loss was -0.399 with predicted mean of [-0.529]
```

```
The next parameters to simulate from are [[0.182 0.281 0.031 0.022 0.017 0.049]]
The mean of the samples was -0.766
Iteration 118
Acquisition function convergence reached at iteration 170.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.231 0.275 0.031 0.021 0.026 0.052]]
The mean of the samples was -0.558
Iteration 119
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.939]
The next parameters to simulate from are [[0.222 0.829 0.017 0.083 0.068 0.05 ]]
The mean of the samples was 1.314
Iteration 120
Acquisition function convergence reached at iteration 134.
The final EI loss was -0.394 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.325 0.882 0.011 0.026 0.019 0.025]]
The mean of the samples was -0.7
Hyperparameter convergence reached at iteration 882.
The minimum predicted mean of the observed indices is -1.057 at the point
[0.361 0.891 0.009 0.041 0.025 0.016]
Iteration 121
Acquisition function convergence reached at iteration 5.
The final EI loss was 0.0 with predicted mean of [0.923]
The next parameters to simulate from are [[0.438 0.503 0.018 0.086 0.064 0.048]]
The mean of the samples was 1.116
Iteration 122
Acquisition function convergence reached at iteration 8.
The final EI loss was 0.0 with predicted mean of [0.767]
The next parameters to simulate from are [[0.739 0.368 0.017 0.094 0.032 0.01 ]]
The mean of the samples was 0.775
Iteration 123
Acquisition function convergence reached at iteration 147.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.201 0.299 0.033 0.015 0.032 0.042]]
The mean of the samples was -0.316
Iteration 124
Acquisition function convergence reached at iteration 160.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.15  0.433 0.008 0.016 0.036 0.049]]
The mean of the samples was -0.591
Iteration 125
Acquisition function convergence reached at iteration 271.
The final EI loss was -0.399 with predicted mean of [-0.529]
```

```
The next parameters to simulate from are [[0.676 0.575 0.009 0.017 0.04  0.007]]
The mean of the samples was -0.431
Iteration 126
Acquisition function convergence reached at iteration 115.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.189 0.439 0.006 0.019 0.035 0.049]]
The mean of the samples was -0.53
Iteration 127
Acquisition function convergence reached at iteration 5.
The final EI loss was 0.0 with predicted mean of [0.945]
The next parameters to simulate from are [[0.343 0.72  0.021 0.082 0.057 0.067]]
The mean of the samples was 0.985
Iteration 128
Acquisition function convergence reached at iteration 223.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.61  0.786 0.031 0.061 0.01  0.016]]
The mean of the samples was -0.384
Iteration 129
Acquisition function convergence reached at iteration 12.
The final EI loss was 0.0 with predicted mean of [0.812]
The next parameters to simulate from are [[0.126 0.405 0.023 0.074 0.036 0.009]]
The mean of the samples was 1.205
Iteration 130
Acquisition function convergence reached at iteration 111.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.394 0.011 0.022 0.054 0.008 0.048]]
The mean of the samples was -0.605
Iteration 131
Acquisition function convergence reached at iteration 97.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.497 0.066 0.025 0.057 0.01  0.042]]
The mean of the samples was -0.666
Iteration 132
Acquisition function convergence reached at iteration 157.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.739 0.634 0.008 0.018 0.039 0.005]]
The mean of the samples was -0.456
Iteration 133
Acquisition function convergence reached at iteration 121.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.547 0.96  0.004 0.037 0.015 0.021]]
The mean of the samples was -0.516
Iteration 134
```

```
Acquisition function convergence reached at iteration 11.
The final EI loss was 0.0 with predicted mean of [0.87]
The next parameters to simulate from are [[0.569 0.442 0.01  0.084 0.052 0.048]]
The mean of the samples was 0.753
Iteration 135
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [1.384]
The next parameters to simulate from are [[0.07  0.606 0.02  0.097 0.016 0.014]]
The mean of the samples was 1.918
Iteration 136
Acquisition function convergence reached at iteration 191.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.2   0.874 0.02  0.026 0.051 0.012]]
The mean of the samples was -0.387
Iteration 137
Acquisition function convergence reached at iteration 128.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.453 0.27  0.028 0.032 0.054 0.013]]
The mean of the samples was -0.472
Iteration 138
Acquisition function convergence reached at iteration 61.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.49  0.269 0.028 0.035 0.058 0.015]]
The mean of the samples was -0.705
Iteration 139
Acquisition function convergence reached at iteration 214.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.295 0.159 0.028 0.022 0.025 0.054]]
The mean of the samples was -0.505
Iteration 140
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [1.717]
The next parameters to simulate from are [[0.383 0.373 0.011 0.091 0.025 0.008]]
The mean of the samples was 1.412
Hyperparameter convergence reached at iteration 903.
The minimum predicted mean of the observed indices is -1.058 at the point
[0.361 0.891 0.009 0.041 0.025 0.016]
Iteration 141
Acquisition function convergence reached at iteration 4.
The final EI loss was 0.0 with predicted mean of [0.41]
The next parameters to simulate from are [[0.782 0.168 0.029 0.095 0.023 0.007]]
The mean of the samples was 0.698
Iteration 142
```

```
Acquisition function convergence reached at iteration 150.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.832 0.015 0.023 0.05  0.069 0.055]]
The mean of the samples was -0.384
Iteration 143
Acquisition function convergence reached at iteration 154.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.495 0.993 0.006 0.046 0.006 0.027]]
The mean of the samples was -0.434
Iteration 144
Acquisition function convergence reached at iteration 111.
The final EI loss was -0.009 with predicted mean of [0.024]
The next parameters to simulate from are [[1.    0.999 0.033 0.1   0.071 0.071]]
The mean of the samples was 0.655
Iteration 145
Acquisition function convergence reached at iteration 7.
The final EI loss was 0.0 with predicted mean of [0.732]
The next parameters to simulate from are [[0.526 0.296 0.006 0.087 0.047 0.065]]
The mean of the samples was 0.713
Iteration 146
Acquisition function convergence reached at iteration 9.
The final EI loss was 0.0 with predicted mean of [0.745]
The next parameters to simulate from are [[0.209 0.008 0.003 0.079 0.027 0.055]]
The mean of the samples was 0.874
Iteration 147
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.697]
The next parameters to simulate from are [[0.722 0.473 0.021 0.097 0.041 0.023]]
The mean of the samples was 0.864
Iteration 148
Acquisition function convergence reached at iteration 276.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.497 0.254 0.028 0.041 0.054 0.013]]
The mean of the samples was -0.515
Iteration 149
Acquisition function convergence reached at iteration 395.
The final EI loss was -0.399 with predicted mean of [-0.531]
The next parameters to simulate from are [[0.352 0.084 0.027 0.024 0.031 0.064]]
The mean of the samples was -0.408
Iteration 150
Acquisition function convergence reached at iteration 184.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.468 0.094 0.028 0.051 0.017 0.055]]
```

```
The mean of the samples was -0.498
Trained parameters:
amplitude_champ:0 is 0.438

length_scales_champ:0 is [0.25  0.25  0.008 0.017 0.018 0.018]

observation_noise_variance_champ:0 is 0.0

bias_mean:0 is 0.321

Iteration 151
Acquisition function convergence reached at iteration 143.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.396 0.236 0.014 0.03  0.043 0.061]]
The mean of the samples was -0.555
Iteration 152
Acquisition function convergence reached at iteration 465.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.686 0.03  0.023 0.057 0.006 0.032]]
The mean of the samples was -0.51
Iteration 153
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.9]
The next parameters to simulate from are [[0.033 0.675 0.003 0.095 0.035 0.053]]
The mean of the samples was 1.553
Iteration 154
Acquisition function convergence reached at iteration 78.
The final EI loss was -0.399 with predicted mean of [-0.53]
The next parameters to simulate from are [[0.412 0.238 0.014 0.03  0.048 0.059]]
The mean of the samples was -0.482
Iteration 155
Acquisition function convergence reached at iteration 318.
The final EI loss was -0.399 with predicted mean of [-0.53]
The next parameters to simulate from are [[0.766 0.335 0.001 0.064 0.005 0.011]]
The mean of the samples was -0.462
Iteration 156
Acquisition function convergence reached at iteration 113.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.164 0.463 0.007 0.017 0.042 0.049]]
The mean of the samples was -0.584
Iteration 157
Acquisition function convergence reached at iteration 7.
The final EI loss was 0.0 with predicted mean of [0.884]
```

```
The next parameters to simulate from are [[0.327 0.578 0.028 0.09  0.034 0.048]]
The mean of the samples was 0.945
Iteration 158
Acquisition function convergence reached at iteration 497.
The final EI loss was -0.399 with predicted mean of [-0.53]
The next parameters to simulate from are [[0.506 0.257 0.028 0.044 0.059 0.015]]
The mean of the samples was -0.326
Iteration 159
Acquisition function convergence reached at iteration 102.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.447 0.028 0.02  0.051 0.008 0.045]]
The mean of the samples was -0.502
Iteration 160
Acquisition function convergence reached at iteration 122.
The final EI loss was -0.399 with predicted mean of [-0.53]
The next parameters to simulate from are [[0.185 0.239 0.031 0.021 0.026 0.055]]
The mean of the samples was -0.669
Hyperparameter convergence reached at iteration 952.
The minimum predicted mean of the observed indices is -1.058 at the point
[0.361 0.891 0.009 0.041 0.025 0.016]
Iteration 161
Acquisition function convergence reached at iteration 3.
The final EI loss was 0.0 with predicted mean of [0.995]
The next parameters to simulate from are [[0.293 0.603 0.006 0.083 0.051 0.008]]
The mean of the samples was 1.406
Iteration 162
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.339]
The next parameters to simulate from are [[0.958 0.472 0.007 0.022 0.021 0.025]]
The mean of the samples was 0.378
Iteration 163
Acquisition function convergence reached at iteration 105.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.197 0.827 0.021 0.03  0.044 0.012]]
The mean of the samples was -0.434
Iteration 164
Acquisition function convergence reached at iteration 145.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.617 0.795 0.032 0.06  0.004 0.013]]
The mean of the samples was -0.628
Iteration 165
Acquisition function convergence reached at iteration 226.
The final EI loss was -0.399 with predicted mean of [-0.529]
```

```
The next parameters to simulate from are [[0.499 0.927 0.002 0.037 0.017 0.023]]
The mean of the samples was -0.421
Iteration 166
Acquisition function convergence reached at iteration 112.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.426 0.199 0.016 0.03  0.034 0.06 ]]
The mean of the samples was -0.586
Iteration 167
Acquisition function convergence reached at iteration 100.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.208 0.302 0.03  0.019 0.015 0.045]]
The mean of the samples was -0.288
Iteration 168
Acquisition function convergence reached at iteration 151.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.816 0.027 0.022 0.042 0.062 0.044]]
The mean of the samples was -0.426
Iteration 169
Acquisition function convergence reached at iteration 5.
The final EI loss was 0.0 with predicted mean of [0.945]
The next parameters to simulate from are [[0.113 0.888 0.016 0.077 0.028 0.015]]
The mean of the samples was 1.19
Iteration 170
Acquisition function convergence reached at iteration 114.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.098 0.435 0.007 0.013 0.043 0.054]]
The mean of the samples was -0.659
Iteration 171
Acquisition function convergence reached at iteration 147.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.082 0.435 0.007 0.012 0.04  0.044]]
The mean of the samples was -0.676
Iteration 172
Acquisition function convergence reached at iteration 156.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.668 0.578 0.007 0.026 0.043 0.018]]
The mean of the samples was -0.553
Iteration 173
Acquisition function convergence reached at iteration 6.
The final EI loss was 0.0 with predicted mean of [0.339]
The next parameters to simulate from are [[0.658 0.888 0.025 0.078 0.062 0.007]]
The mean of the samples was 0.41
Iteration 174
```

```
Acquisition function convergence reached at iteration 173.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.452 0.266 0.03  0.04  0.056 0.018]]
The mean of the samples was -0.836
Iteration 175
Acquisition function convergence reached at iteration 10.
The final EI loss was 0.0 with predicted mean of [0.67]
The next parameters to simulate from are [[0.559 0.611 0.025 0.096 0.068 0.014]]
The mean of the samples was 1.308
Iteration 176
Acquisition function convergence reached at iteration 225.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.124 0.685 0.03  0.018 0.038 0.043]]
The mean of the samples was -0.676
Iteration 177
Acquisition function convergence reached at iteration 174.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.553 0.288 0.027 0.036 0.058 0.012]]
The mean of the samples was -0.431
Iteration 178
Acquisition function convergence reached at iteration 223.
The final EI loss was -0.399 with predicted mean of [-0.53]
The next parameters to simulate from are [[0.598 0.737 0.032 0.055 0.014 0.022]]
The mean of the samples was -0.786
Iteration 179
Acquisition function convergence reached at iteration 9.
The final EI loss was 0.0 with predicted mean of [0.875]
The next parameters to simulate from are [[0.164 0.771 0.006 0.069 0.071 0.023]]
The mean of the samples was 1.035
Iteration 180
Acquisition function convergence reached at iteration 122.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.668 0.628 0.007 0.018 0.04  0.007]]
The mean of the samples was -0.508
Hyperparameter convergence reached at iteration 963.
The minimum predicted mean of the observed indices is -1.057 at the point
[0.361 0.891 0.009 0.041 0.025 0.016]
Iteration 181
Acquisition function convergence reached at iteration 7.
The final EI loss was 0.0 with predicted mean of [0.789]
The next parameters to simulate from are [[0.5   0.67  0.026 0.084 0.055 0.05 ]]
The mean of the samples was 0.91
Iteration 182
```

```
Acquisition function convergence reached at iteration 229.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.368 0.254 0.027 0.041 0.047 0.018]]
The mean of the samples was -0.581
Iteration 183
Acquisition function convergence reached at iteration 7.
The final EI loss was 0.0 with predicted mean of [0.383]
The next parameters to simulate from are [[0.899 0.448 0.012 0.014 0.013 0.031]]
The mean of the samples was 0.377
Iteration 184
Acquisition function convergence reached at iteration 190.
The final EI loss was -0.397 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.576 0.085 0.027 0.059 0.008 0.04 ]]
The mean of the samples was -0.484
Iteration 185
Acquisition function convergence reached at iteration 169.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.408 0.171 0.018 0.029 0.033 0.063]]
The mean of the samples was -0.614
Iteration 186
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.261]
The next parameters to simulate from are [[0.928 0.101 0.008 0.044 0.022 0.052]]
The mean of the samples was 0.284
Iteration 187
Acquisition function convergence reached at iteration 497.
The final EI loss was -0.399 with predicted mean of [-0.531]
The next parameters to simulate from are [[0.056 0.679 0.003 0.038 0.01  0.042]]
The mean of the samples was -0.52
Iteration 188
Acquisition function convergence reached at iteration 140.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.153 0.199 0.029 0.019 0.025 0.057]]
The mean of the samples was -0.544
Iteration 189
Acquisition function convergence reached at iteration 132.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.327 0.864 0.011 0.022 0.019 0.022]]
The mean of the samples was -0.601
Iteration 190
Acquisition function convergence reached at iteration 7.
The final EI loss was 0.0 with predicted mean of [0.744]
The next parameters to simulate from are [[0.206 0.689 0.031 0.086 0.066 0.056]]
```

```
The mean of the samples was 1.242
Iteration 191
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [1.109]
The next parameters to simulate from are [[0.234 0.691 0.017 0.089 0.028 0.046]]
The mean of the samples was 1.135
Iteration 192
Acquisition function convergence reached at iteration 3.
The final EI loss was 0.0 with predicted mean of [0.713]
The next parameters to simulate from are [[0.659 0.444 0.026 0.094 0.025 0.02 ]]
The mean of the samples was 0.803
Iteration 193
Acquisition function convergence reached at iteration 108.
The final EI loss was -0.399 with predicted mean of [-0.526]
The next parameters to simulate from are [[0.658 0.794 0.03  0.056 0.012 0.015]]
The mean of the samples was -0.748
Iteration 194
Acquisition function convergence reached at iteration 6.
The final EI loss was 0.0 with predicted mean of [0.861]
The next parameters to simulate from are [[0.264 0.418 0.003 0.074 0.002 0.014]]
The mean of the samples was 0.692
Iteration 195
Acquisition function convergence reached at iteration 154.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.058 0.601 0.029 0.016 0.034 0.051]]
The mean of the samples was -0.394
Iteration 196
Acquisition function convergence reached at iteration 11.
The final EI loss was 0.0 with predicted mean of [0.187]
The next parameters to simulate from are [[0.198 0.287 0.03  0.    0.034 0.044]]
The mean of the samples was 0.341
Iteration 197
Acquisition function convergence reached at iteration 146.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.005 0.653 0.004 0.041 0.01  0.037]]
The mean of the samples was -0.465
Iteration 198
Acquisition function convergence reached at iteration 326.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.254 0.165 0.027 0.024 0.019 0.053]]
The mean of the samples was -0.474
Iteration 199
Acquisition function convergence reached at iteration 165.
```

```
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.31  0.159 0.02  0.029 0.029 0.012]]
The mean of the samples was -0.479
Iteration 200
Acquisition function convergence reached at iteration 227.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.607 0.849 0.033 0.062 0.003 0.012]]
The mean of the samples was -0.18
Hyperparameter convergence reached at iteration 975.
The minimum predicted mean of the observed indices is -1.057 at the point
[0.361 0.891 0.009 0.041 0.025 0.016]
Trained parameters:
amplitude_champ:0 is 0.419

length_scales_champ:0 is [0.25  0.25  0.008 0.016 0.018 0.018]

observation_noise_variance_champ:0 is 0.0

bias_mean:0 is 0.346

Iteration 201
Acquisition function convergence reached at iteration 10.
The final EI loss was 0.0 with predicted mean of [0.404]
The next parameters to simulate from are [[0.777 0.543 0.026 0.095 0.011 0.029]]
The mean of the samples was 0.444
Iteration 202
Acquisition function convergence reached at iteration 345.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.763 0.334 0.    0.063 0.001 0.014]]
The mean of the samples was 0.013
Iteration 203
Acquisition function convergence reached at iteration 144.
The final EI loss was -0.399 with predicted mean of [-0.53]
The next parameters to simulate from are [[0.611 0.604 0.008 0.019 0.039 0.005]]
The mean of the samples was -0.285
Iteration 204
Acquisition function convergence reached at iteration 136.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.478 0.941 0.002 0.036 0.017 0.018]]
The mean of the samples was -0.67
Iteration 205
Acquisition function convergence reached at iteration 124.
The final EI loss was -0.399 with predicted mean of [-0.528]
```

```
The next parameters to simulate from are [[0.793 0.044 0.02  0.046 0.071 0.056]]
The mean of the samples was -0.404
Iteration 206
Acquisition function convergence reached at iteration 161.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.433 0.023 0.021 0.057 0.013 0.048]]
The mean of the samples was -0.463
Iteration 207
Acquisition function convergence reached at iteration 152.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.147 0.131 0.028 0.021 0.023 0.059]]
The mean of the samples was -0.496
Iteration 208
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.699]
The next parameters to simulate from are [[0.547 0.705 0.019 0.074 0.057 0.016]]
The mean of the samples was 0.529
Iteration 209
Acquisition function convergence reached at iteration 105.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.788 0.056 0.02  0.038 0.064 0.044]]
The mean of the samples was -0.43
Iteration 210
Acquisition function convergence reached at iteration 10.
The final EI loss was 0.0 with predicted mean of [0.378]
The next parameters to simulate from are [[0.892 0.256 0.009 0.028 0.015 0.063]]
The mean of the samples was 0.382
Iteration 211
Acquisition function convergence reached at iteration 99.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.633 0.028 0.027 0.057 0.007 0.038]]
The mean of the samples was -0.538
Iteration 212
Acquisition function convergence reached at iteration 126.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.128 0.159 0.03  0.02  0.027 0.062]]
The mean of the samples was -0.428
Iteration 213
Acquisition function convergence reached at iteration 32.
The final EI loss was -0.364 with predicted mean of [-0.497]
The next parameters to simulate from are [[0.545 0.083 0.021 0.036 0.053 0.038]]
The mean of the samples was -0.755
Iteration 214
```

```
Acquisition function convergence reached at iteration 168.
The final EI loss was -0.399 with predicted mean of [-0.526]
The next parameters to simulate from are [[0.671 0.62  0.004 0.017 0.043 0.009]]
The mean of the samples was -0.395
Iteration 215
Acquisition function convergence reached at iteration 3.
The final EI loss was 0.0 with predicted mean of [0.981]
The next parameters to simulate from are [[0.398 0.769 0.009 0.076 0.042 0.007]]
The mean of the samples was 0.933
Iteration 216
Acquisition function convergence reached at iteration 92.
The final EI loss was -0.398 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.64  0.133 0.021 0.039 0.058 0.041]]
The mean of the samples was -0.669
Iteration 217
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.915]
The next parameters to simulate from are [[0.348 0.237 0.002 0.08  0.066 0.025]]
The mean of the samples was 1.132
Iteration 218
Acquisition function convergence reached at iteration 94.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.129 0.874 0.02  0.032 0.049 0.014]]
The mean of the samples was -0.493
Iteration 219
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [1.624]
The next parameters to simulate from are [[0.121 0.119 0.007 0.088 0.051 0.038]]
The mean of the samples was 1.614
Iteration 220
Acquisition function convergence reached at iteration 162.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.608 0.96  0.004 0.037 0.014 0.017]]
The mean of the samples was -0.566
Hyperparameter convergence reached at iteration 1003.
The minimum predicted mean of the observed indices is -1.056 at the point
[0.361 0.891 0.009 0.041 0.025 0.016]
Iteration 221
Acquisition function convergence reached at iteration 87.
The final EI loss was -0.01 with predicted mean of [0.004]
The next parameters to simulate from are [[0.114 0.999 0.009 0.001 0.071 0.071]]
The mean of the samples was 0.173
Iteration 222
```

```
Acquisition function convergence reached at iteration 125.
The final EI loss was -0.008 with predicted mean of [0.018]
The next parameters to simulate from are [[0.999 0.004 0.    0.1   0.    0.071]]
The mean of the samples was 0.396
Iteration 223
Acquisition function convergence reached at iteration 160.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.065 0.169 0.03  0.022 0.027 0.06 ]]
The mean of the samples was -0.725
Iteration 224
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.212]
The next parameters to simulate from are [[0.704 0.841 0.021 0.013 0.052 0.058]]
The mean of the samples was 0.219
Iteration 225
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [1.226]
The next parameters to simulate from are [[0.015 0.789 0.01  0.083 0.035 0.002]]
The mean of the samples was 1.848
Iteration 226
Acquisition function convergence reached at iteration 216.
The final EI loss was -0.399 with predicted mean of [-0.526]
The next parameters to simulate from are [[0.278 0.251 0.026 0.041 0.045 0.018]]
The mean of the samples was -0.705
Iteration 227
Acquisition function convergence reached at iteration 110.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.323 0.02  0.022 0.055 0.006 0.049]]
The mean of the samples was -0.59
Iteration 228
Acquisition function convergence reached at iteration 6.
The final EI loss was 0.0 with predicted mean of [0.32]
The next parameters to simulate from are [[0.943 0.919 0.01  0.029 0.052 0.058]]
The mean of the samples was 0.344
Iteration 229
Acquisition function convergence reached at iteration 5.
The final EI loss was 0.0 with predicted mean of [0.092]
The next parameters to simulate from are [[0.739 0.709 0.013 0.029 0.015 0.035]]
The mean of the samples was 0.069
Iteration 230
Acquisition function convergence reached at iteration 365.
The final EI loss was -0.399 with predicted mean of [-0.526]
The next parameters to simulate from are [[0.146 0.29  0.03  0.02  0.01  0.048]]
```

```
The mean of the samples was -0.228
Iteration 231
Acquisition function convergence reached at iteration 126.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.656 0.671 0.004 0.021 0.043 0.011]]
The mean of the samples was -0.591
Iteration 232
Acquisition function convergence reached at iteration 4.
The final EI loss was 0.0 with predicted mean of [0.387]
The next parameters to simulate from are [[0.945 0.002 0.018 0.087 0.069 0.007]]
The mean of the samples was 0.16
Iteration 233
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.883]
The next parameters to simulate from are [[0.324 0.973 0.006 0.097 0.043 0.034]]
The mean of the samples was 1.28
Iteration 234
Acquisition function convergence reached at iteration 166.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.186 0.802 0.022 0.032 0.046 0.015]]
The mean of the samples was -0.606
Iteration 235
Acquisition function convergence reached at iteration 244.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.161 0.118 0.03  0.025 0.027 0.062]]
The mean of the samples was -0.753
Iteration 236
Acquisition function convergence reached at iteration 8.
The final EI loss was 0.0 with predicted mean of [0.88]
The next parameters to simulate from are [[0.095 0.375 0.003 0.07  0.053 0.044]]
The mean of the samples was 1.041
Iteration 237
Acquisition function convergence reached at iteration 7.
The final EI loss was 0.0 with predicted mean of [0.747]
The next parameters to simulate from are [[0.519 0.049 0.007 0.079 0.058 0.036]]
The mean of the samples was 0.877
Iteration 238
Acquisition function convergence reached at iteration 8.
The final EI loss was 0.0 with predicted mean of [0.661]
The next parameters to simulate from are [[0.587 0.962 0.029 0.097 0.055 0.023]]
The mean of the samples was 1.012
Iteration 239
Acquisition function convergence reached at iteration 306.
```

```
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.271 0.9   0.013 0.039 0.031 0.013]]
The mean of the samples was -0.615
Iteration 240
Acquisition function convergence reached at iteration 112.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.278 0.115 0.029 0.028 0.03  0.067]]
The mean of the samples was -0.741
Hyperparameter convergence reached at iteration 997.
The minimum predicted mean of the observed indices is -1.056 at the point
[0.361 0.891 0.009 0.041 0.025 0.016]
Iteration 241
Acquisition function convergence reached at iteration 3.
The final EI loss was 0.0 with predicted mean of [0.857]
The next parameters to simulate from are [[0.264 0.495 0.001 0.072 0.026 0.027]]
The mean of the samples was 0.899
Iteration 242
Acquisition function convergence reached at iteration 121.
The final EI loss was -0.008 with predicted mean of [0.015]
The next parameters to simulate from are [[0.999 1.    0.    0.068 0.001 0.071]]
The mean of the samples was 0.537
Iteration 243
Acquisition function convergence reached at iteration 137.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.23  0.861 0.019 0.024 0.052 0.016]]
The mean of the samples was -0.638
Iteration 244
Acquisition function convergence reached at iteration 3.
The final EI loss was 0.0 with predicted mean of [0.739]
The next parameters to simulate from are [[0.646 0.223 0.003 0.088 0.063 0.031]]
The mean of the samples was 0.778
Iteration 245
Acquisition function convergence reached at iteration 12.
The final EI loss was 0.0 with predicted mean of [0.889]
The next parameters to simulate from are [[0.191 0.334 0.03  0.093 0.045 0.032]]
The mean of the samples was 1.526
Iteration 246
Acquisition function convergence reached at iteration 4.
The final EI loss was 0.0 with predicted mean of [0.547]
The next parameters to simulate from are [[0.685 0.393 0.001 0.065 0.003 0.026]]
The mean of the samples was 0.03
Iteration 247
Acquisition function convergence reached at iteration 125.
```

```
The final EI loss was -0.398 with predicted mean of [-0.524]
The next parameters to simulate from are [[0.47  0.272 0.026 0.042 0.047 0.02 ]]
The mean of the samples was -0.687
Iteration 248
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.425]
The next parameters to simulate from are [[0.696 0.986 0.029 0.087 0.064 0.01 ]]
The mean of the samples was 0.511
Iteration 249
Acquisition function convergence reached at iteration 130.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.705 0.097 0.    0.045 0.029 0.023]]
The mean of the samples was -0.616
Iteration 250
Acquisition function convergence reached at iteration 196.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.265 0.85  0.019 0.029 0.046 0.013]]
The mean of the samples was -0.686
Trained parameters:
amplitude_champ:0 is 0.416

length_scales_champ:0 is [0.25  0.25  0.008 0.017 0.018 0.018]

observation_noise_variance_champ:0 is 0.001

bias_mean:0 is 0.35

Iteration 251
Acquisition function convergence reached at iteration 144.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.661 0.623 0.005 0.018 0.047 0.012]]
The mean of the samples was -0.383
Iteration 252
Acquisition function convergence reached at iteration 1062.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.602 0.709 0.032 0.057 0.022 0.024]]
The mean of the samples was -0.466
Iteration 253
Acquisition function convergence reached at iteration 142.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.727 0.125 0.001 0.043 0.024 0.023]]
The mean of the samples was -0.508
Iteration 254
```

```
Acquisition function convergence reached at iteration 195.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.648 0.054 0.021 0.041 0.065 0.045]]
The mean of the samples was -0.496
Iteration 255
Acquisition function convergence reached at iteration 83.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.697 0.03  0.001 0.044 0.033 0.019]]
The mean of the samples was -0.912
Iteration 256
Acquisition function convergence reached at iteration 317.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.744 0.651 0.007 0.024 0.051 0.013]]
The mean of the samples was -0.412
Iteration 257
Acquisition function convergence reached at iteration 3.
The final EI loss was 0.0 with predicted mean of [0.82]
The next parameters to simulate from are [[0.241 0.714 0.001 0.07  0.068 0.038]]
The mean of the samples was 1.068
Iteration 258
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [1.156]
The next parameters to simulate from are [[0.175 0.749 0.02  0.079 0.023 0.013]]
The mean of the samples was 1.155
Iteration 259
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.743]
The next parameters to simulate from are [[0.193 0.556 0.019 0.066 0.021 0.006]]
The mean of the samples was 0.882
Iteration 260
WARNING:tensorflow:5 out of the last 325 calls to <function update_var_EI.<locals>.opt_var at
Acquisition function convergence reached at iteration 8.
The final EI loss was 0.0 with predicted mean of [0.578]
The next parameters to simulate from are [[0.79  0.366 0.002 0.092 0.045 0.025]]
The mean of the samples was 0.698
Hyperparameter convergence reached at iteration 998.
The minimum predicted mean of the observed indices is -1.054 at the point
[0.361 0.891 0.009 0.041 0.025 0.016]
Iteration 261
Acquisition function convergence reached at iteration 195.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.265 0.467 0.03  0.018 0.048 0.037]]
The mean of the samples was -0.697
```

```
Iteration 262
Acquisition function convergence reached at iteration 221.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.767 0.667 0.007 0.026 0.057 0.01 ]]
The mean of the samples was -0.56
Iteration 263
Acquisition function convergence reached at iteration 5.
The final EI loss was 0.0 with predicted mean of [0.276]
The next parameters to simulate from are [[0.965 0.682 0.025 0.083 0.03  0.041]]
The mean of the samples was 0.285
Iteration 264
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.804]
The next parameters to simulate from are [[0.017 0.819 0.02  0.076 0.065 0.032]]
The mean of the samples was 1.315
Iteration 265
Acquisition function convergence reached at iteration 124.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.14  0.46  0.006 0.012 0.044 0.043]]
The mean of the samples was -0.574
Iteration 266
Acquisition function convergence reached at iteration 127.
The final EI loss was -0.399 with predicted mean of [-0.526]
The next parameters to simulate from are [[0.158 0.499 0.006 0.022 0.036 0.049]]
The mean of the samples was -0.388
Iteration 267
Acquisition function convergence reached at iteration 106.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.467 0.021 0.021 0.057 0.006 0.038]]
The mean of the samples was -0.463
Iteration 268
Acquisition function convergence reached at iteration 6.
The final EI loss was 0.0 with predicted mean of [0.808]
The next parameters to simulate from are [[0.332 0.114 0.021 0.088 0.039 0.016]]
The mean of the samples was 1.185
Iteration 269
Acquisition function convergence reached at iteration 114.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.119 0.424 0.007 0.014 0.039 0.059]]
The mean of the samples was -0.61
Iteration 270
Acquisition function convergence reached at iteration 457.
The final EI loss was -0.399 with predicted mean of [-0.527]
```

```
The next parameters to simulate from are [[0.619 0.696 0.032 0.053 0.027 0.024]]
The mean of the samples was -0.599
Iteration 271
Acquisition function convergence reached at iteration 133.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.773 0.005 0.02  0.042 0.059 0.044]]
The mean of the samples was -0.56
Iteration 272
Acquisition function convergence reached at iteration 4.
The final EI loss was 0.0 with predicted mean of [0.365]
The next parameters to simulate from are [[0.676 0.185 0.009 0.004 0.033 0.042]]
The mean of the samples was 0.338
Iteration 273
Acquisition function convergence reached at iteration 59.
The final EI loss was -0.013 with predicted mean of [-0.019]
The next parameters to simulate from are [[0.034 0.982 0.    0.027 0.    0.071]]
The mean of the samples was 0.437
Iteration 274
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.883]
The next parameters to simulate from are [[0.298 0.197 0.009 0.077 0.043 0.066]]
The mean of the samples was 1.018
Iteration 275
Acquisition function convergence reached at iteration 140.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.073 0.835 0.021 0.031 0.052 0.018]]
The mean of the samples was -0.592
Iteration 276
Acquisition function convergence reached at iteration 169.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.032 0.76  0.027 0.016 0.039 0.048]]
The mean of the samples was -0.782
Iteration 277
Acquisition function convergence reached at iteration 206.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.497 0.871 0.01  0.041 0.019 0.032]]
The mean of the samples was -0.534
Iteration 278
Acquisition function convergence reached at iteration 3.
The final EI loss was 0.0 with predicted mean of [0.857]
The next parameters to simulate from are [[0.048 0.557 0.01  0.062 0.034 0.008]]
The mean of the samples was 0.868
Iteration 279
```

```
Acquisition function convergence reached at iteration 377.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.254 0.112 0.03  0.022 0.022 0.056]]
The mean of the samples was -0.526
Iteration 280
Acquisition function convergence reached at iteration 7.
The final EI loss was 0.0 with predicted mean of [0.106]
The next parameters to simulate from are [[0.816 0.837 0.025 0.076 0.066 0.01 ]]
The mean of the samples was 0.431
Hyperparameter convergence reached at iteration 1001.
The minimum predicted mean of the observed indices is -1.055 at the point
[0.361 0.891 0.009 0.041 0.025 0.016]
Iteration 281
Acquisition function convergence reached at iteration 5.
The final EI loss was 0.0 with predicted mean of [0.567]
The next parameters to simulate from are [[0.709 0.869 0.023 0.08  0.049 0.022]]
The mean of the samples was 0.558
Iteration 282
Acquisition function convergence reached at iteration 118.
The final EI loss was -0.006 with predicted mean of [0.024]
The next parameters to simulate from are [[0.997 0.998 0.033 0.1   0.    0.071]]
The mean of the samples was 0.396
Iteration 283
Acquisition function convergence reached at iteration 6.
The final EI loss was 0.0 with predicted mean of [0.691]
The next parameters to simulate from are [[0.584 0.473 0.023 0.09  0.032 0.039]]
The mean of the samples was 0.565
Iteration 284
Acquisition function convergence reached at iteration 108.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.228 0.382 0.007 0.018 0.038 0.053]]
The mean of the samples was -0.553
Iteration 285
Acquisition function convergence reached at iteration 229.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.426 0.965 0.001 0.038 0.019 0.023]]
The mean of the samples was -0.432
Iteration 286
Acquisition function convergence reached at iteration 5.
The final EI loss was 0.0 with predicted mean of [0.525]
The next parameters to simulate from are [[0.284 0.587 0.027 0.067 0.054 0.065]]
The mean of the samples was 0.731
Iteration 287
```

```
Acquisition function convergence reached at iteration 240.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.422 0.047 0.026 0.027 0.028 0.059]]
The mean of the samples was -0.66
Iteration 288
Acquisition function convergence reached at iteration 188.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.577 0.283 0.025 0.036 0.053 0.015]]
The mean of the samples was -0.862
Iteration 289
Acquisition function convergence reached at iteration 139.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.7   0.019 0.017 0.046 0.066 0.049]]
The mean of the samples was -0.48
Iteration 290
Acquisition function convergence reached at iteration 101.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.749 0.037 0.017 0.04  0.059 0.046]]
The mean of the samples was -0.559
Iteration 291
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.964]
The next parameters to simulate from are [[0.025 0.852 0.005 0.068 0.043 0.01 ]]
The mean of the samples was 1.028
Iteration 292
Acquisition function convergence reached at iteration 287.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.043 0.807 0.028 0.012 0.039 0.048]]
The mean of the samples was -0.341
Iteration 293
Acquisition function convergence reached at iteration 313.
The final EI loss was -0.399 with predicted mean of [-0.53]
The next parameters to simulate from are [[0.681 0.854 0.029 0.055 0.015 0.015]]
The mean of the samples was -0.646
Iteration 294
Acquisition function convergence reached at iteration 63.
The final EI loss was -0.124 with predicted mean of [-1.222]
The next parameters to simulate from are [[0.283 0.312 0.028 0.031 0.041 0.033]]
The mean of the samples was -1.002
Iteration 295
Acquisition function convergence reached at iteration 3.
The final EI loss was 0.0 with predicted mean of [0.529]
The next parameters to simulate from are [[0.827 0.459 0.016 0.095 0.044 0.036]]
```

```
The mean of the samples was 0.3
Iteration 296
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [1.395]
The next parameters to simulate from are [[0.25  0.591 0.026 0.093 0.033 0.023]]
The mean of the samples was 1.286
Iteration 297
Acquisition function convergence reached at iteration 125.
The final EI loss was -0.399 with predicted mean of [-0.525]
The next parameters to simulate from are [[0.519 0.765 0.029 0.027 0.019 0.027]]
The mean of the samples was -0.544
Iteration 298
Acquisition function convergence reached at iteration 246.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.034 0.64  0.029 0.017 0.035 0.057]]
The mean of the samples was -0.67
Iteration 299
Acquisition function convergence reached at iteration 3.
The final EI loss was 0.0 with predicted mean of [0.512]
The next parameters to simulate from are [[0.831 0.709 0.024 0.088 0.046 0.032]]
The mean of the samples was 0.323
Iteration 300
Acquisition function convergence reached at iteration 139.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.251 0.035 0.022 0.057 0.005 0.049]]
The mean of the samples was -0.395
Hyperparameter convergence reached at iteration 1005.
The minimum predicted mean of the observed indices is -1.054 at the point
[0.361 0.891 0.009 0.041 0.025 0.016]
Trained parameters:
amplitude_champ:0 is 0.408

length_scales_champ:0 is [0.25  0.25  0.008 0.017 0.018 0.018]

observation_noise_variance_champ:0 is 0.001

bias_mean:0 is 0.351

Iteration 301
Acquisition function convergence reached at iteration 293.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.788 0.329 0.002 0.061 0.01  0.014]]
The mean of the samples was -0.222
```

```
Iteration 302
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.818]
The next parameters to simulate from are [[0.426 0.317 0.001 0.099 0.069 0.032]]
The mean of the samples was 1.474
Iteration 303
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.894]
The next parameters to simulate from are [[0.095 0.758 0.017 0.093 0.055 0.071]]
The mean of the samples was 1.597
Iteration 304
Acquisition function convergence reached at iteration 98.
The final EI loss was -0.396 with predicted mean of [-0.522]
The next parameters to simulate from are [[0.747 0.037 0.001 0.043 0.02  0.028]]
The mean of the samples was -0.235
Iteration 305
Acquisition function convergence reached at iteration 122.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.554 0.004 0.021 0.04  0.059 0.043]]
The mean of the samples was -0.673
Iteration 306
Acquisition function convergence reached at iteration 188.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.424 0.16  0.028 0.033 0.053 0.014]]
The mean of the samples was -0.421
Iteration 307
Acquisition function convergence reached at iteration 10.
The final EI loss was 0.0 with predicted mean of [0.315]
The next parameters to simulate from are [[0.912 0.148 0.009 0.037 0.008 0.062]]
The mean of the samples was 0.409
Iteration 308
Acquisition function convergence reached at iteration 338.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.273 0.923 0.012 0.025 0.02  0.029]]
The mean of the samples was -0.75
Iteration 309
Acquisition function convergence reached at iteration 402.
The final EI loss was -0.398 with predicted mean of [-0.522]
The next parameters to simulate from are [[0.512 0.875 0.006 0.041 0.024 0.025]]
The mean of the samples was -0.516
Iteration 310
Acquisition function convergence reached at iteration 259.
The final EI loss was -0.399 with predicted mean of [-0.528]
```

```
The next parameters to simulate from are [[0.727 0.032 0.    0.045 0.029 0.028]]
The mean of the samples was -0.51
Iteration 311
Acquisition function convergence reached at iteration 113.
The final EI loss was -0.399 with predicted mean of [-0.529]
The next parameters to simulate from are [[0.272 0.833 0.019 0.019 0.055 0.017]]
The mean of the samples was -0.525
Iteration 312
Acquisition function convergence reached at iteration 134.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.053 0.7   0.004 0.037 0.015 0.043]]
The mean of the samples was -0.381
Iteration 313
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.927]
The next parameters to simulate from are [[0.019 0.488 0.015 0.088 0.067 0.026]]
The mean of the samples was 1.737
Iteration 314
Acquisition function convergence reached at iteration 198.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.222 0.379 0.008 0.018 0.038 0.061]]
The mean of the samples was -0.48
Iteration 315
Acquisition function convergence reached at iteration 3.
The final EI loss was 0.0 with predicted mean of [0.576]
The next parameters to simulate from are [[0.458 0.581 0.028 0.088 0.032 0.054]]
The mean of the samples was 0.768
Iteration 316
Acquisition function convergence reached at iteration 130.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.545 0.074 0.024 0.055 0.004 0.036]]
The mean of the samples was -0.694
Iteration 317
Acquisition function convergence reached at iteration 301.
The final EI loss was -0.399 with predicted mean of [-0.527]
The next parameters to simulate from are [[0.505 0.01  0.024 0.06  0.012 0.048]]
The mean of the samples was -0.324
Iteration 318
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.632]
The next parameters to simulate from are [[0.284 0.21  0.012 0.073 0.032 0.069]]
The mean of the samples was 0.646
Iteration 319
```

```
Acquisition function convergence reached at iteration 257.
The final EI loss was -0.399 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.175 0.177 0.03  0.023 0.011 0.053]]
The mean of the samples was -0.4
Iteration 320
Acquisition function convergence reached at iteration 87.
The final EI loss was -0.357 with predicted mean of [-0.516]
The next parameters to simulate from are [[0.857 0.996 0.007 0.043 0.057 0.001]]
The mean of the samples was -0.364
Hyperparameter convergence reached at iteration 1050.
The minimum predicted mean of the observed indices is -1.051 at the point
[0.361 0.891 0.009 0.041 0.025 0.016]
Iteration 321
Acquisition function convergence reached at iteration 209.
The final EI loss was -0.399 with predicted mean of [-0.526]
The next parameters to simulate from are [[0.486 0.274 0.031 0.037 0.056 0.012]]
The mean of the samples was -0.49
Iteration 322
Acquisition function convergence reached at iteration 420.
The final EI loss was -0.399 with predicted mean of [-0.524]
The next parameters to simulate from are [[0.673 0.827 0.028 0.056 0.02  0.02 ]]
The mean of the samples was -0.489
Iteration 323
Acquisition function convergence reached at iteration 5.
The final EI loss was 0.0 with predicted mean of [0.646]
The next parameters to simulate from are [[0.516 0.483 0.015 0.084 0.043 0.056]]
The mean of the samples was 0.808
Iteration 324
Acquisition function convergence reached at iteration 44.
The final EI loss was -0.007 with predicted mean of [-0.043]
The next parameters to simulate from are [[0.643 0.489 0.    0.065 0.001 0.045]]
The mean of the samples was 0.357
Iteration 325
Acquisition function convergence reached at iteration 233.
The final EI loss was -0.399 with predicted mean of [-0.526]
The next parameters to simulate from are [[0.851 0.33  0.002 0.061 0.008 0.01 ]]
The mean of the samples was -0.276
Iteration 326
Acquisition function convergence reached at iteration 8.
The final EI loss was 0.0 with predicted mean of [0.284]
The next parameters to simulate from are [[0.703 0.387 0.014 0.009 0.032 0.065]]
The mean of the samples was 0.333
Iteration 327
```

```
Acquisition function convergence reached at iteration 14.
The final EI loss was 0.0 with predicted mean of [0.776]
The next parameters to simulate from are [[0.207 0.087 0.027 0.093 0.057 0.03 ]]
The mean of the samples was 1.619
Iteration 328
Acquisition function convergence reached at iteration 7.
The final EI loss was 0.0 with predicted mean of [0.629]
The next parameters to simulate from are [[0.74  0.271 0.006 0.086 0.05  0.041]]
The mean of the samples was 0.542
Iteration 329
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.617]
The next parameters to simulate from are [[0.616 0.342 0.005 0.085 0.056 0.056]]
The mean of the samples was 0.795
Iteration 330
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.85]
The next parameters to simulate from are [[0.423 0.768 0.013 0.096 0.041 0.056]]
The mean of the samples was 1.023
Iteration 331
Acquisition function convergence reached at iteration 214.
The final EI loss was -0.399 with predicted mean of [-0.524]
The next parameters to simulate from are [[0.431 0.78  0.03  0.026 0.018 0.032]]
The mean of the samples was -0.71
Iteration 332
Acquisition function convergence reached at iteration 9.
The final EI loss was 0.0 with predicted mean of [0.339]
The next parameters to simulate from are [[0.987 0.207 0.006 0.024 0.017 0.059]]
The mean of the samples was 0.456
Iteration 333
Acquisition function convergence reached at iteration 156.
The final EI loss was -0.399 with predicted mean of [-0.526]
The next parameters to simulate from are [[0.315 0.147 0.017 0.028 0.025 0.013]]
The mean of the samples was -0.548
Iteration 334
Acquisition function convergence reached at iteration 4.
The final EI loss was 0.0 with predicted mean of [0.215]
The next parameters to simulate from are [[0.82  0.743 0.024 0.019 0.054 0.059]]
The mean of the samples was 0.239
Iteration 335
Acquisition function convergence reached at iteration 4.
The final EI loss was 0.0 with predicted mean of [0.783]
The next parameters to simulate from are [[0.067 0.525 0.01  0.067 0.041 0.052]]
```

```
The mean of the samples was 0.865
Iteration 336
Acquisition function convergence reached at iteration 175.
The final EI loss was -0.399 with predicted mean of [-0.526]
The next parameters to simulate from are [[0.429 0.144 0.028 0.049 0.02  0.059]]
The mean of the samples was -0.433
Iteration 337
Acquisition function convergence reached at iteration 4.
The final EI loss was 0.0 with predicted mean of [0.751]
The next parameters to simulate from are [[0.349 0.447 0.018 0.077 0.006 0.013]]
The mean of the samples was 0.734
Iteration 338
Acquisition function convergence reached at iteration 163.
The final EI loss was -0.399 with predicted mean of [-0.526]
The next parameters to simulate from are [[0.215 0.206 0.026 0.021 0.028 0.049]]
The mean of the samples was -0.684
Iteration 339
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.894]
The next parameters to simulate from are [[0.359 0.478 0.029 0.082 0.065 0.014]]
The mean of the samples was 1.087
Iteration 340
Acquisition function convergence reached at iteration 6.
The final EI loss was 0.0 with predicted mean of [0.834]
The next parameters to simulate from are [[0.345 0.556 0.005 0.087 0.018 0.037]]
The mean of the samples was 0.855
Hyperparameter convergence reached at iteration 1075.
The minimum predicted mean of the observed indices is -1.051 at the point
[0.361 0.891 0.009 0.041 0.025 0.016]
Iteration 341
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [1.655]
The next parameters to simulate from are [[0.196 0.795 0.011 0.09  0.03  0.023]]
The mean of the samples was 1.494
Iteration 342
Acquisition function convergence reached at iteration 5.
The final EI loss was 0.0 with predicted mean of [0.169]
The next parameters to simulate from are [[0.604 0.536 0.009 0.018 0.023 0.05 ]]
The mean of the samples was 0.176
Iteration 343
Acquisition function convergence reached at iteration 4.
The final EI loss was 0.0 with predicted mean of [0.493]
The next parameters to simulate from are [[0.811 0.385 0.016 0.083 0.039 0.015]]
```

```
The mean of the samples was 0.464
Iteration 344
Acquisition function convergence reached at iteration 13.
The final EI loss was 0.0 with predicted mean of [0.333]
The next parameters to simulate from are [[0.832 0.467 0.01  0.017 0.027 0.045]]
The mean of the samples was 0.369
Iteration 345
Acquisition function convergence reached at iteration 5.
The final EI loss was 0.0 with predicted mean of [0.592]
The next parameters to simulate from are [[0.65  0.53  0.012 0.089 0.036 0.044]]
The mean of the samples was 0.688
Iteration 346
Acquisition function convergence reached at iteration 5.
The final EI loss was 0.0 with predicted mean of [0.373]
The next parameters to simulate from are [[0.804 0.247 0.007 0.004 0.04  0.046]]
The mean of the samples was 0.425
Iteration 347
Acquisition function convergence reached at iteration 6.
The final EI loss was 0.0 with predicted mean of [0.587]
The next parameters to simulate from are [[0.62  0.081 0.023 0.082 0.027 0.011]]
The mean of the samples was 0.448
Iteration 348
Acquisition function convergence reached at iteration 160.
The final EI loss was -0.399 with predicted mean of [-0.525]
The next parameters to simulate from are [[0.442 0.119 0.028 0.033 0.05  0.017]]
The mean of the samples was -0.664
Iteration 349
Acquisition function convergence reached at iteration 7.
The final EI loss was 0.0 with predicted mean of [0.772]
The next parameters to simulate from are [[0.116 0.957 0.009 0.085 0.029 0.055]]
The mean of the samples was 1.194
Iteration 350
Acquisition function convergence reached at iteration 173.
The final EI loss was -0.399 with predicted mean of [-0.524]
The next parameters to simulate from are [[0.008 0.631 0.03  0.015 0.035 0.062]]
The mean of the samples was -0.576
Trained parameters:
amplitude_champ:0 is 0.398

length_scales_champ:0 is [0.25  0.25  0.008 0.017 0.018 0.018]

observation_noise_variance_champ:0 is 0.001
```

```
bias_mean:0 is 0.377

Iteration 351
Acquisition function convergence reached at iteration 195.
The final EI loss was -0.399 with predicted mean of [-0.525]
The next parameters to simulate from are [[0.103 0.537 0.03  0.02  0.033 0.052]]
The mean of the samples was -1.05
Iteration 352
Acquisition function convergence reached at iteration 109.
The final EI loss was -0.399 with predicted mean of [-0.525]
The next parameters to simulate from are [[0.644 0.053 0.021 0.037 0.066 0.041]]
The mean of the samples was -0.84
Iteration 353
Acquisition function convergence reached at iteration 104.
The final EI loss was -0.399 with predicted mean of [-0.525]
The next parameters to simulate from are [[0.358 0.084 0.02  0.053 0.007 0.05 ]]
The mean of the samples was -0.551
Iteration 354
Acquisition function convergence reached at iteration 5.
The final EI loss was 0.0 with predicted mean of [0.396]
The next parameters to simulate from are [[0.99  0.372 0.009 0.02  0.024 0.058]]
The mean of the samples was 0.458
Iteration 355
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.251]
The next parameters to simulate from are [[0.726 0.934 0.02  0.016 0.046 0.054]]
The mean of the samples was 0.229
Iteration 356
Acquisition function convergence reached at iteration 286.
The final EI loss was -0.399 with predicted mean of [-0.526]
The next parameters to simulate from are [[0.277 0.168 0.028 0.025 0.031 0.069]]
The mean of the samples was -0.623
Iteration 357
Acquisition function convergence reached at iteration 6.
The final EI loss was 0.0 with predicted mean of [0.558]
The next parameters to simulate from are [[0.506 0.63  0.027 0.09  0.024 0.06 ]]
The mean of the samples was 0.667
Iteration 358
Acquisition function convergence reached at iteration 3.
The final EI loss was 0.0 with predicted mean of [0.681]
The next parameters to simulate from are [[0.4   0.616 0.021 0.069 0.069 0.018]]
The mean of the samples was 0.493
Iteration 359
```

```
Acquisition function convergence reached at iteration 6.
The final EI loss was 0.0 with predicted mean of [0.892]
The next parameters to simulate from are [[0.119 0.158 0.019 0.094 0.058 0.055]]
The mean of the samples was 1.595
Iteration 360
Acquisition function convergence reached at iteration 251.
The final EI loss was -0.399 with predicted mean of [-0.524]
The next parameters to simulate from are [[0.239 0.946 0.012 0.019 0.022 0.029]]
The mean of the samples was -0.515
Hyperparameter convergence reached at iteration 1079.
The minimum predicted mean of the observed indices is -1.049 at the point
[0.361 0.891 0.009 0.041 0.025 0.016]
Iteration 361
Acquisition function convergence reached at iteration 4.
The final EI loss was 0.0 with predicted mean of [0.436]
The next parameters to simulate from are [[0.701 0.314 0.021 0.095 0.061 0.064]]
The mean of the samples was 0.73
Iteration 362
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [1.602]
The next parameters to simulate from are [[0.066 0.28  0.006 0.089 0.06  0.046]]
The mean of the samples was 1.702
Iteration 363
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.96]
The next parameters to simulate from are [[0.602 0.416 0.006 0.086 0.042 0.017]]
The mean of the samples was 0.839
Iteration 364
WARNING:tensorflow:5 out of the last 260 calls to <function update_var_EI.<locals>.opt_var at
Acquisition function convergence reached at iteration 149.
The final EI loss was -0.399 with predicted mean of [-0.525]
The next parameters to simulate from are [[0.138 0.135 0.028 0.025 0.016 0.06 ]]
The mean of the samples was -0.457
Iteration 365
Acquisition function convergence reached at iteration 233.
The final EI loss was -0.399 with predicted mean of [-0.524]
The next parameters to simulate from are [[0.657 0.773 0.029 0.053 0.027 0.02 ]]
The mean of the samples was -0.588
Iteration 366
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.733]
The next parameters to simulate from are [[0.628 0.153 0.009 0.095 0.059 0.019]]
The mean of the samples was 1.029
```

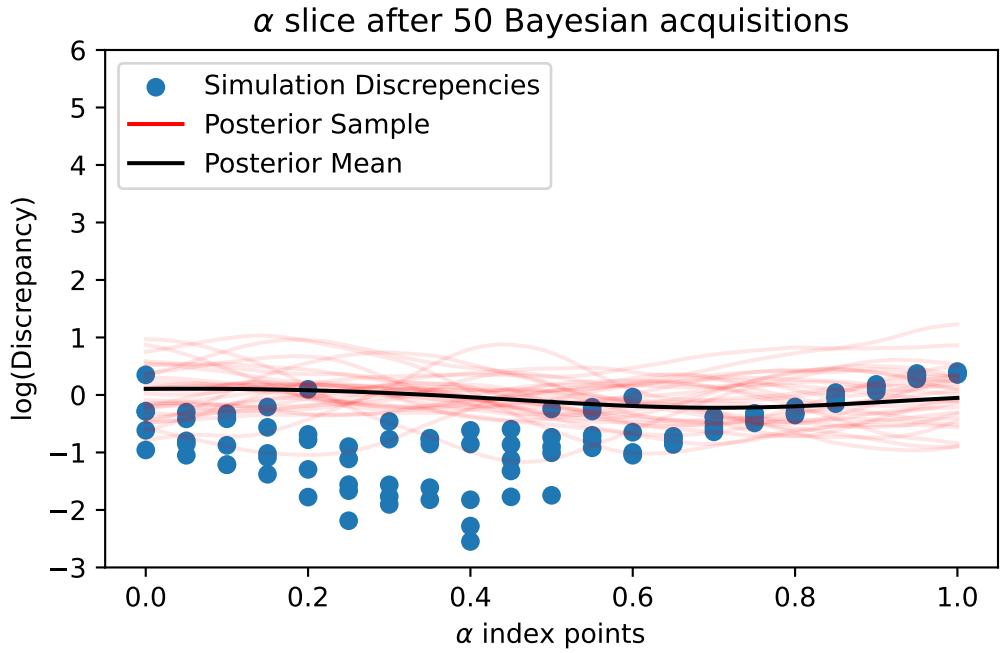
```
Iteration 367
Acquisition function convergence reached at iteration 6.
The final EI loss was 0.0 with predicted mean of [0.652]
The next parameters to simulate from are [[0.498 0.643 0.008 0.08  0.026 0.033]]
The mean of the samples was 0.69
Iteration 368
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [1.131]
The next parameters to simulate from are [[0.556 0.501 0.017 0.099 0.031 0.004]]
The mean of the samples was 1.295
Iteration 369
Acquisition function convergence reached at iteration 116.
The final EI loss was -0.399 with predicted mean of [-0.525]
The next parameters to simulate from are [[0.032 0.353 0.008 0.009 0.036 0.053]]
The mean of the samples was -0.439
Iteration 370
Acquisition function convergence reached at iteration 167.
The final EI loss was -0.399 with predicted mean of [-0.524]
The next parameters to simulate from are [[0.326 0.822 0.01  0.021 0.018 0.016]]
The mean of the samples was -0.684
Iteration 371
Acquisition function convergence reached at iteration 10.
The final EI loss was 0.0 with predicted mean of [0.721]
The next parameters to simulate from are [[0.418 0.93  0.011 0.089 0.042 0.065]]
The mean of the samples was 0.964
Iteration 372
Acquisition function convergence reached at iteration 3.
The final EI loss was 0.0 with predicted mean of [0.721]
The next parameters to simulate from are [[0.133 0.881 0.008 0.064 0.066 0.012]]
The mean of the samples was 0.679
Iteration 373
Acquisition function convergence reached at iteration 142.
The final EI loss was -0.399 with predicted mean of [-0.525]
The next parameters to simulate from are [[0.866 0.945 0.006 0.041 0.057 0.001]]
The mean of the samples was -0.395
Iteration 374
Acquisition function convergence reached at iteration 196.
The final EI loss was -0.399 with predicted mean of [-0.524]
The next parameters to simulate from are [[0.627 0.298 0.027 0.036 0.061 0.015]]
The mean of the samples was -1.151
Iteration 375
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [1.198]
```

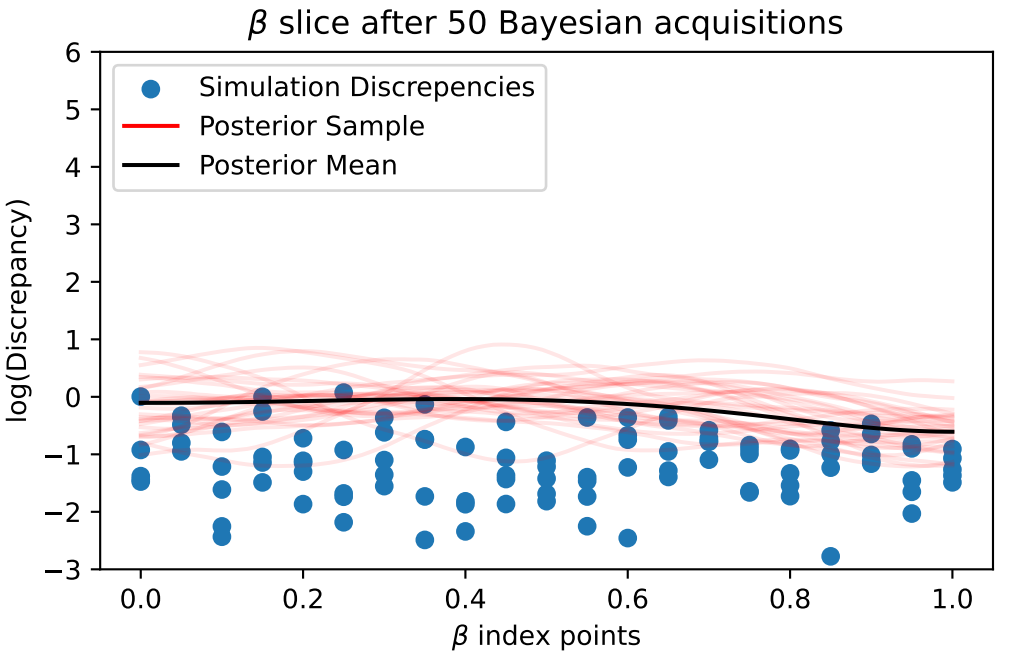
```
The next parameters to simulate from are [[0.371 0.54  0.014 0.094 0.048 0.001]]
The mean of the samples was 1.64
Iteration 376
Acquisition function convergence reached at iteration 161.
The final EI loss was -0.399 with predicted mean of [-0.555]
The next parameters to simulate from are [[0.024 0.648 0.029 0.015 0.03  0.062]]
The mean of the samples was -0.435
Iteration 377
Acquisition function convergence reached at iteration 178.
The final EI loss was -0.399 with predicted mean of [-0.555]
The next parameters to simulate from are [[0.049 0.373 0.011 0.013 0.037 0.051]]
The mean of the samples was -0.645
Iteration 378
Acquisition function convergence reached at iteration 114.
The final EI loss was -0.399 with predicted mean of [-0.554]
The next parameters to simulate from are [[0.096 0.321 0.007 0.01  0.044 0.048]]
The mean of the samples was -0.603
Iteration 379
Acquisition function convergence reached at iteration 135.
The final EI loss was -0.399 with predicted mean of [-0.554]
The next parameters to simulate from are [[0.698 0.064 0.018 0.04  0.055 0.046]]
The mean of the samples was -0.677
Iteration 380
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.644]
The next parameters to simulate from are [[0.524 0.696 0.014 0.075 0.038 0.001]]
The mean of the samples was 0.892
Hyperparameter convergence reached at iteration 1094.
The minimum predicted mean of the observed indices is -1.098 at the point
[0.627 0.298 0.027 0.036 0.061 0.015]
Iteration 381
Acquisition function convergence reached at iteration 9.
The final EI loss was 0.0 with predicted mean of [0.262]
The next parameters to simulate from are [[0.981 0.928 0.007 0.036 0.047 0.07 ]]
The mean of the samples was 0.364
Iteration 382
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.957]
The next parameters to simulate from are [[0.14  0.694 0.027 0.078 0.052 0.029]]
The mean of the samples was 1.208
Iteration 383
Acquisition function convergence reached at iteration 7.
The final EI loss was 0.0 with predicted mean of [0.513]
```

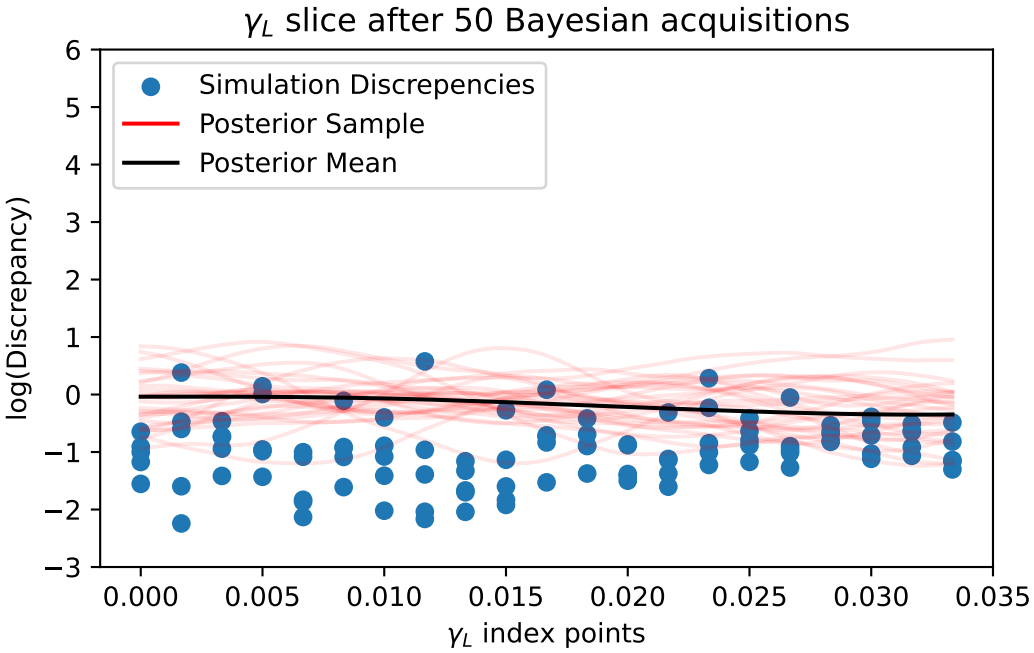
```
The next parameters to simulate from are [[0.726 0.202 0.02  0.096 0.054 0.069]]
The mean of the samples was 0.572
Iteration 384
Acquisition function convergence reached at iteration 130.
The final EI loss was -0.399 with predicted mean of [-0.549]
The next parameters to simulate from are [[0.548 0.066 0.021 0.052 0.004 0.036]]
The mean of the samples was -0.451
Iteration 385
Acquisition function convergence reached at iteration 77.
The final EI loss was -0.399 with predicted mean of [-0.544]
The next parameters to simulate from are [[0.342 0.199 0.026 0.022 0.036 0.063]]
The mean of the samples was -0.472
Iteration 386
Acquisition function convergence reached at iteration 3.
The final EI loss was 0.0 with predicted mean of [0.619]
The next parameters to simulate from are [[0.498 0.363 0.022 0.073 0.043 0.026]]
The mean of the samples was 0.494
Iteration 387
Acquisition function convergence reached at iteration 130.
The final EI loss was -0.399 with predicted mean of [-0.55]
The next parameters to simulate from are [[0.767 0.105 0.021 0.037 0.062 0.036]]
The mean of the samples was -0.589
Iteration 388
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.645]
The next parameters to simulate from are [[0.404 0.172 0.013 0.077 0.032 0.06 ]]
The mean of the samples was 0.446
Iteration 389
Acquisition function convergence reached at iteration 7.
The final EI loss was 0.0 with predicted mean of [0.626]
The next parameters to simulate from are [[0.718 0.047 0.015 0.088 0.057 0.022]]
The mean of the samples was 0.541
Iteration 390
Acquisition function convergence reached at iteration 5.
The final EI loss was 0.0 with predicted mean of [0.741]
The next parameters to simulate from are [[0.027 0.764 0.017 0.066 0.049 0.058]]
The mean of the samples was 0.833
Iteration 391
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.911]
The next parameters to simulate from are [[0.652 0.373 0.017 0.099 0.057 0.021]]
The mean of the samples was 0.971
Iteration 392
```

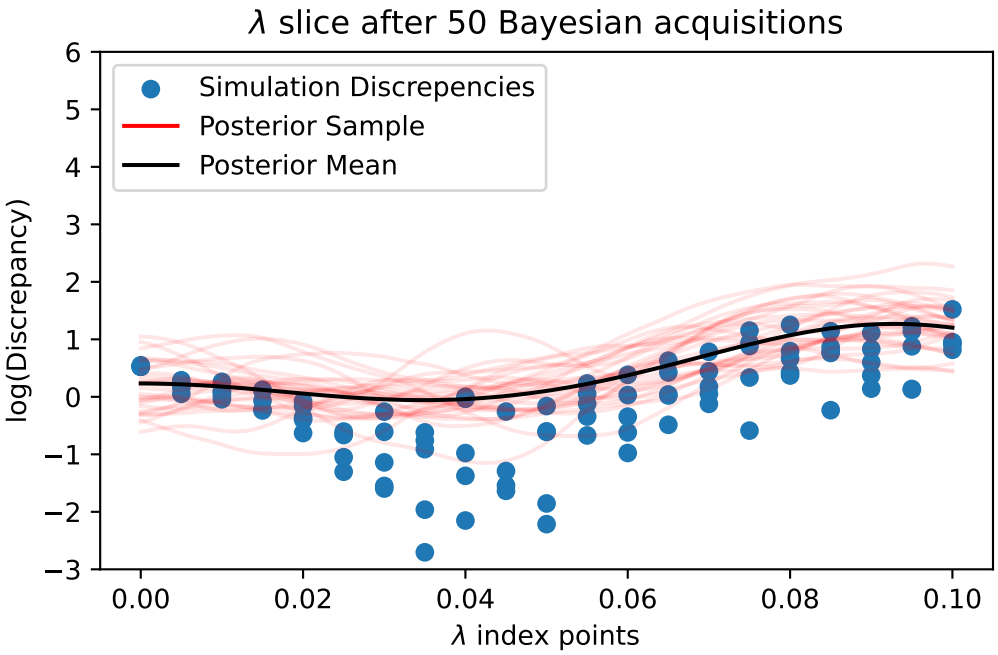
```
Acquisition function convergence reached at iteration 120.
The final EI loss was -0.399 with predicted mean of [-0.549]
The next parameters to simulate from are [[0.078 0.295 0.006 0.015 0.027 0.052]]
The mean of the samples was -0.388
Iteration 393
Acquisition function convergence reached at iteration 215.
The final EI loss was -0.399 with predicted mean of [-0.548]
The next parameters to simulate from are [[0.801 0.138 0.021 0.035 0.058 0.03 ]]
The mean of the samples was -0.553
Iteration 394
Acquisition function convergence reached at iteration 382.
The final EI loss was -0.399 with predicted mean of [-0.55]
The next parameters to simulate from are [[0.746 0.824 0.029 0.057 0.018 0.015]]
The mean of the samples was -0.65
Iteration 395
Acquisition function convergence reached at iteration 111.
The final EI loss was -0.399 with predicted mean of [-0.55]
The next parameters to simulate from are [[0.201 0.949 0.011 0.02  0.024 0.031]]
The mean of the samples was -0.616
Iteration 396
Acquisition function convergence reached at iteration 222.
The final EI loss was -0.399 with predicted mean of [-0.549]
The next parameters to simulate from are [[0.793 0.826 0.03  0.057 0.02  0.019]]
The mean of the samples was -0.414
Iteration 397
Acquisition function convergence reached at iteration 4.
The final EI loss was 0.0 with predicted mean of [0.643]
The next parameters to simulate from are [[0.532 0.921 0.017 0.091 0.04  0.068]]
The mean of the samples was 0.645
Iteration 398
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.923]
The next parameters to simulate from are [[0.561 0.095 0.012 0.09  0.048 0.019]]
The mean of the samples was 0.996
Iteration 399
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.427]
The next parameters to simulate from are [[0.273 0.731 0.022 0.066 0.038 0.057]]
The mean of the samples was 0.53
Iteration 400
Acquisition function convergence reached at iteration 2.
The final EI loss was 0.0 with predicted mean of [0.796]
The next parameters to simulate from are [[0.35  0.753 0.028 0.084 0.037 0.009]]
```

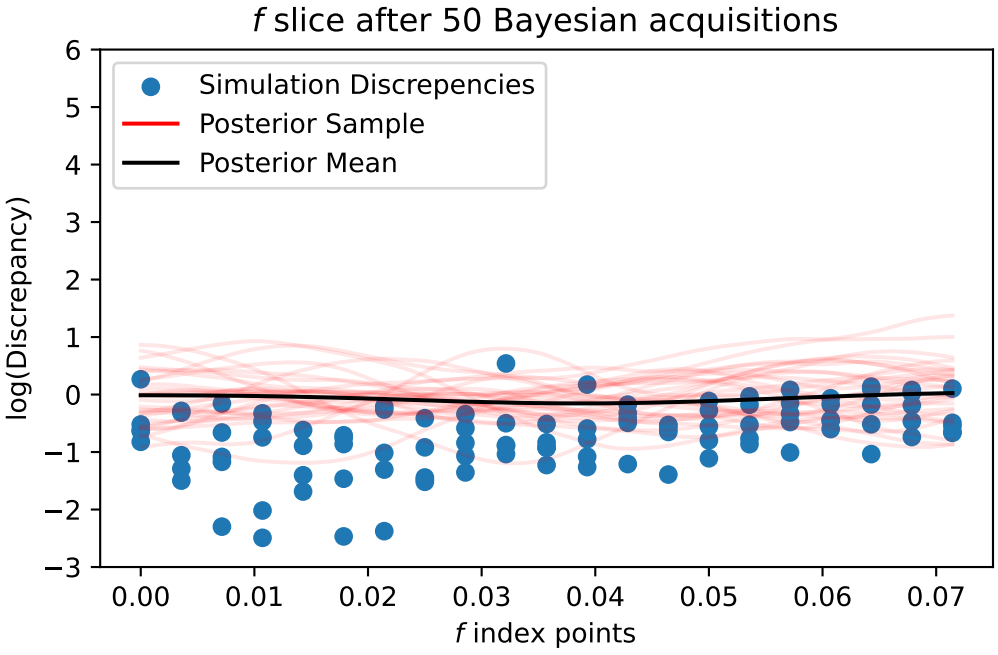
```
The mean of the samples was 1.256
Hyperparameter convergence reached at iteration 1097.
The minimum predicted mean of the observed indices is -1.097 at the point
[0.627 0.298 0.027 0.036 0.061 0.015]
Trained parameters:
amplitude_champ:0 is 0.387

length_scales_champ:0 is [0.25  0.25  0.008 0.017 0.018 0.018]

observation_noise_variance_champ:0 is 0.002

bias_mean:0 is 0.382
```

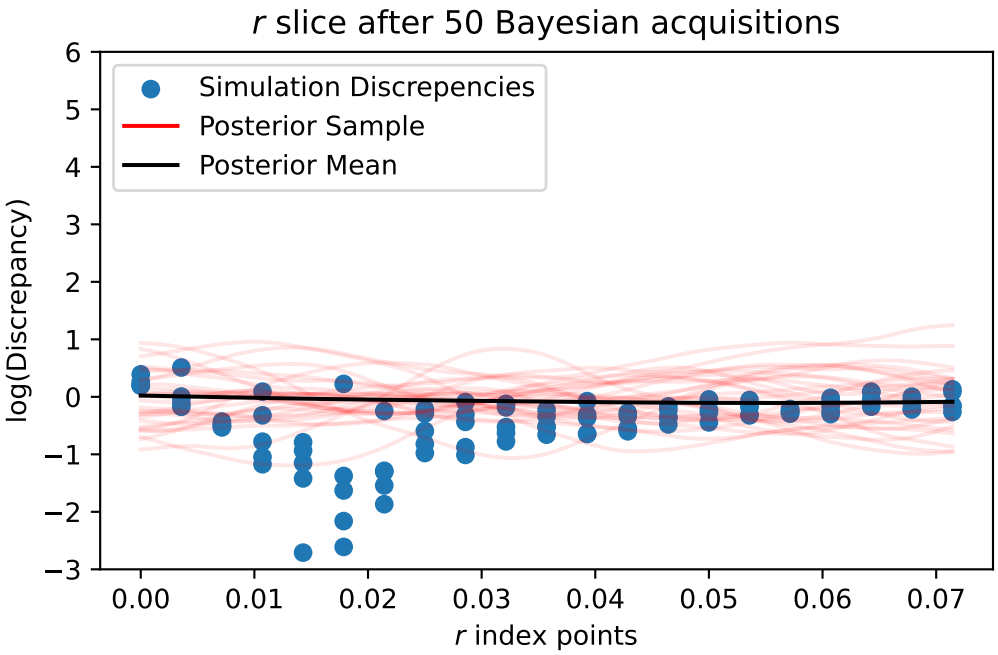


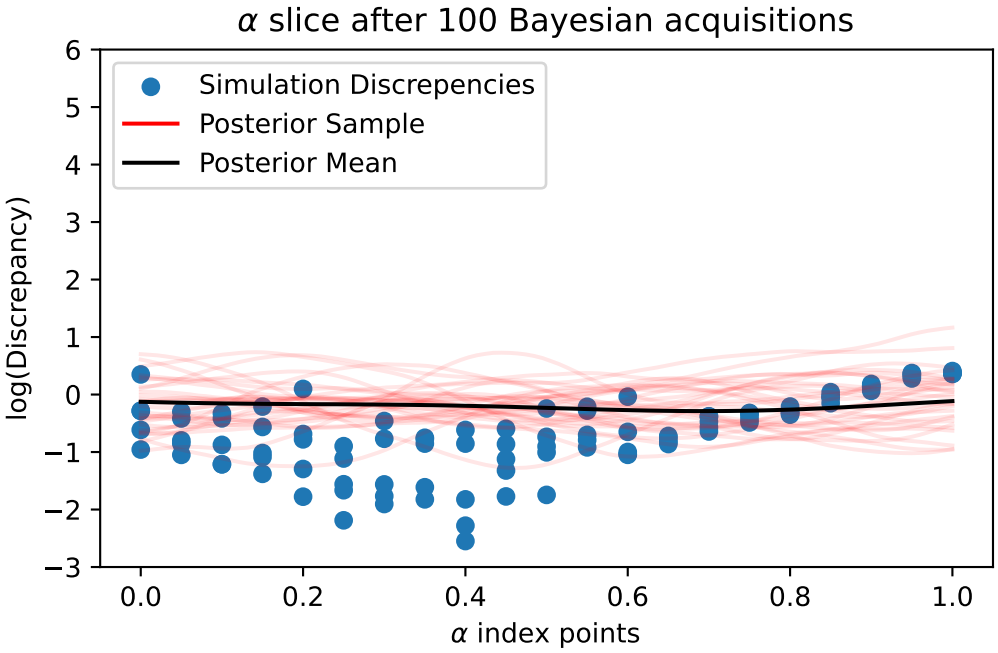

β slice after 50 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
β index points

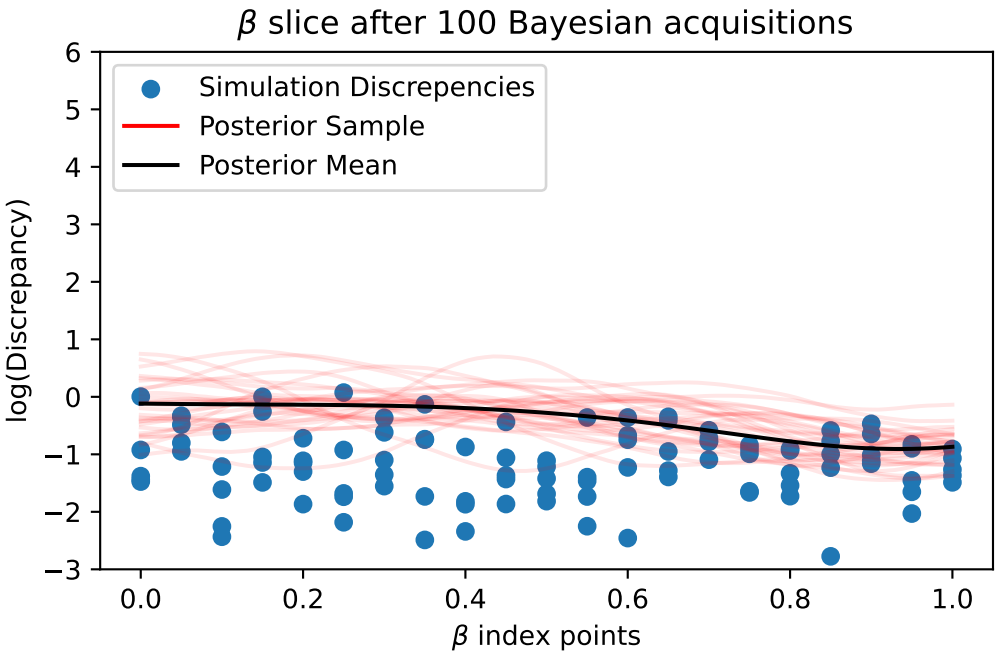

γ_L slice after 50 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
γ_L index points

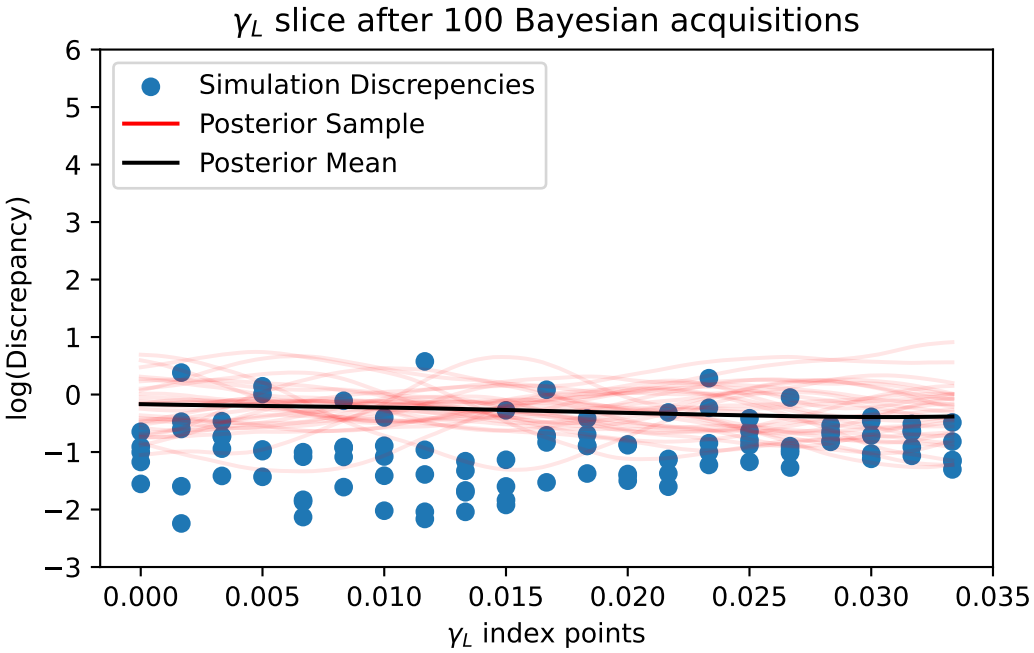

λ slice after 50 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
λ index points

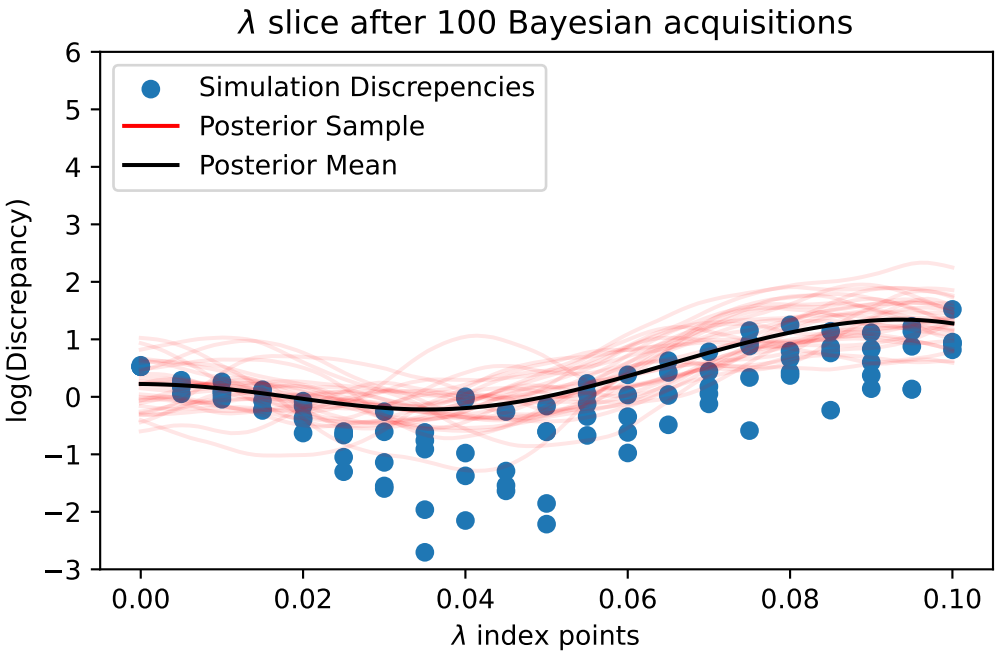

f slice after 50 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
f index points

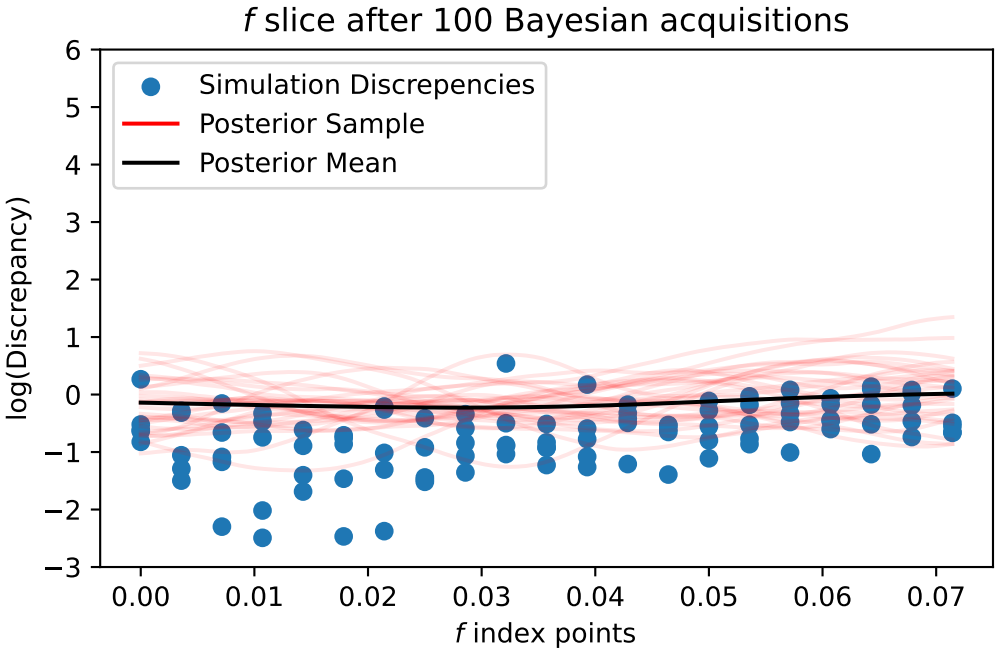

r slice after 50 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
r index points
6
5
4
3
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07

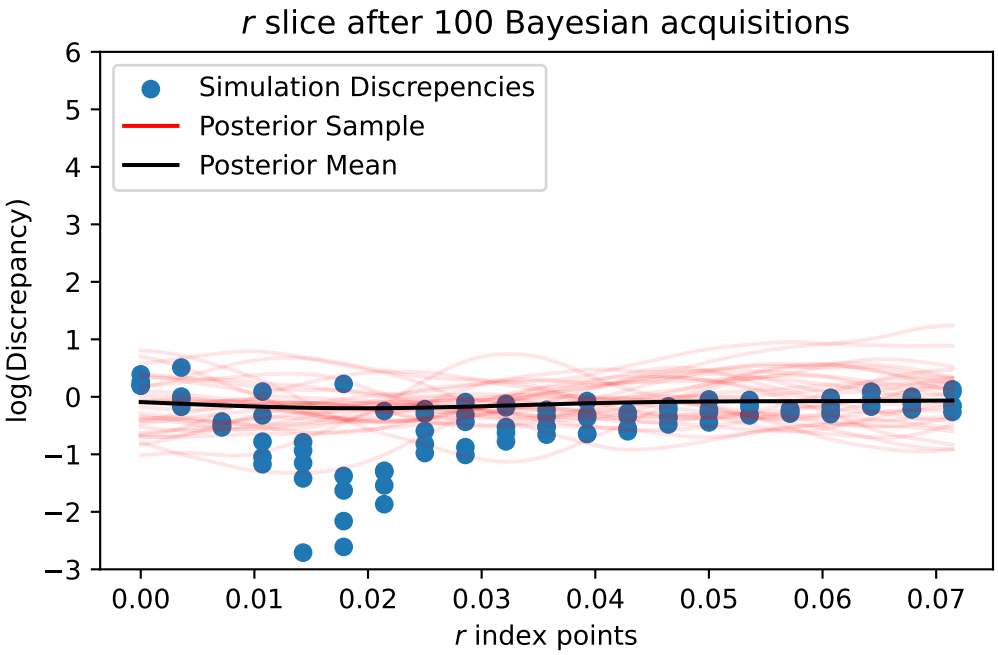

α slice after 100 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
α index points
6
5
4
3
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0

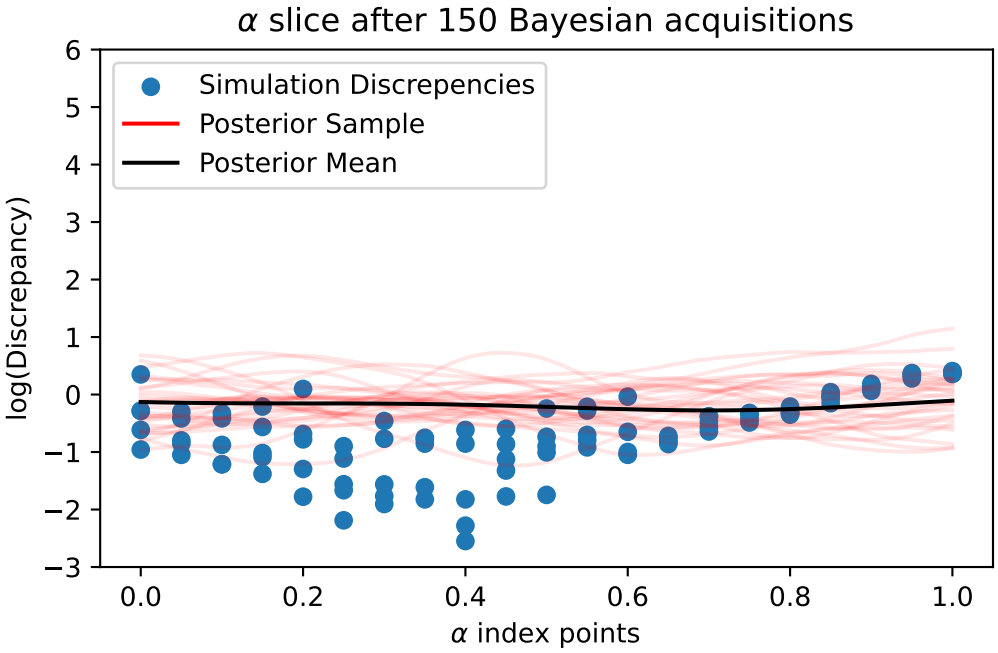

β slice after 100 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
β index points

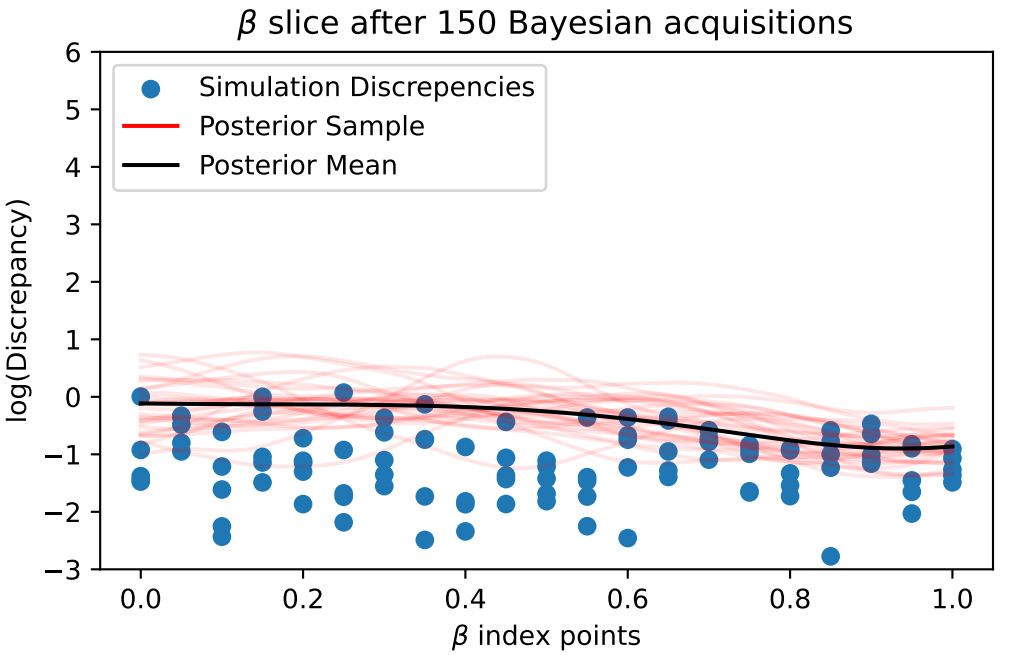

γ_L slice after 100 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
γ_L index points

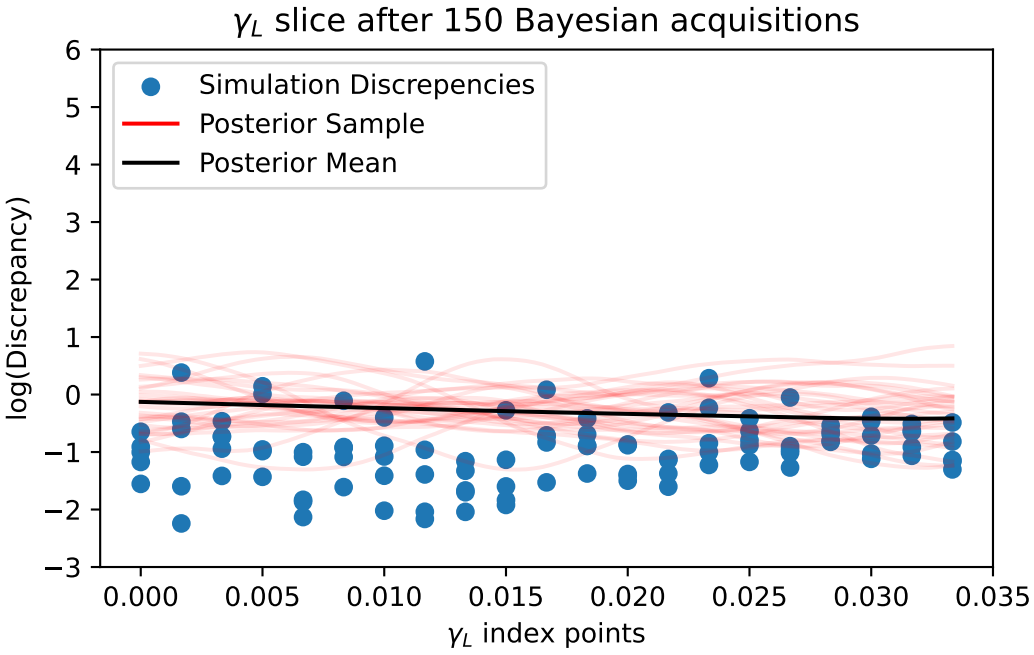

λ slice after 100 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
λ index points
6
5
4
3
2
1
0
−1
−2
−3
0.00
0.02
0.04
0.06
0.08
0.10

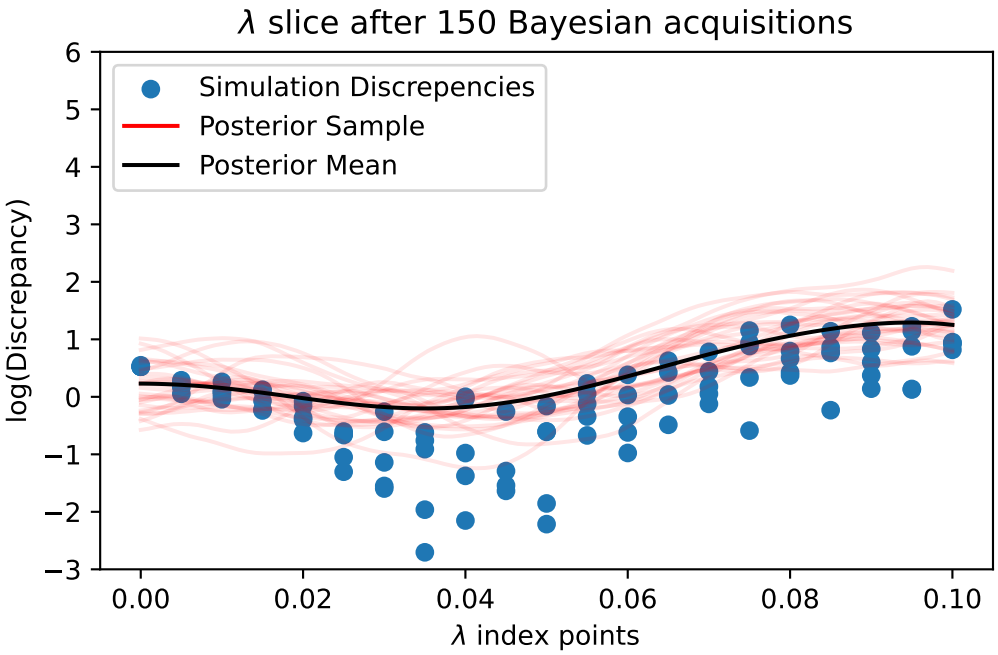

f slice after 100 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
f index points
6
5
4
3
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07

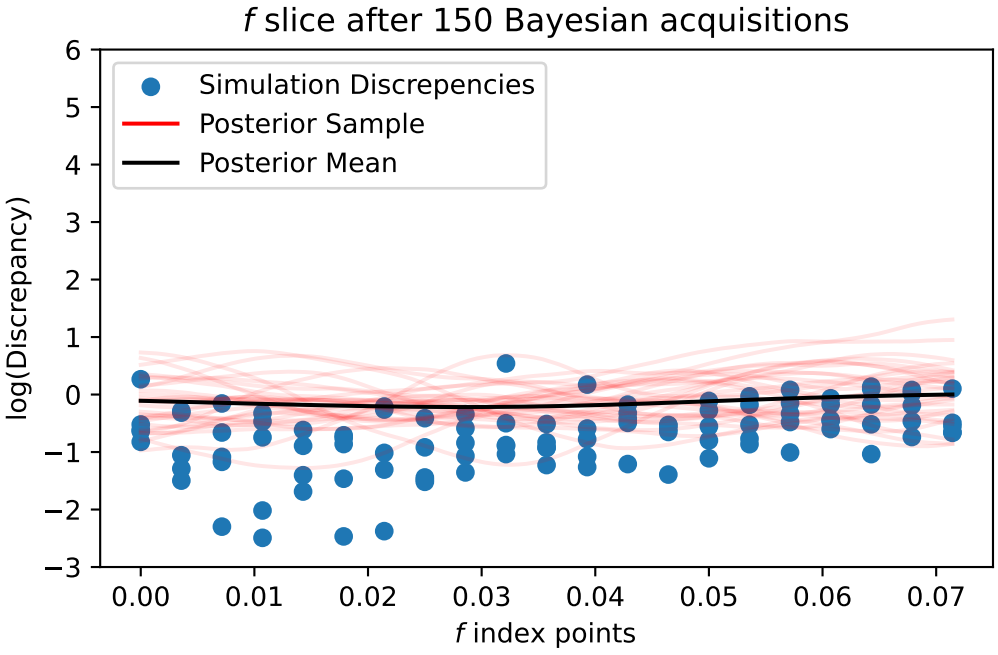

r slice after 100 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
r index points

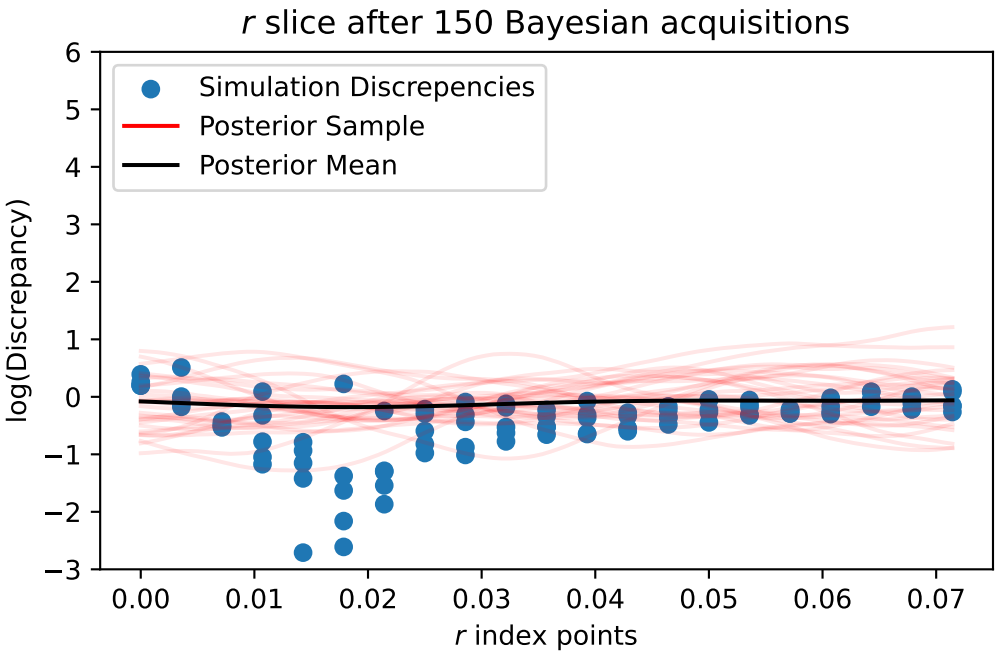

α slice after 150 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
α index points

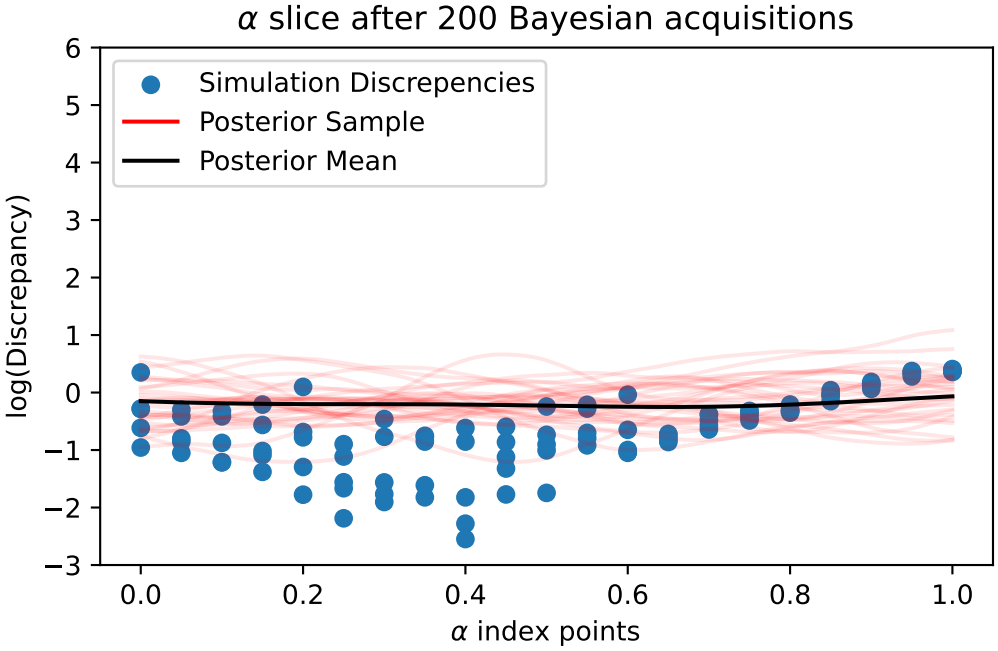

β slice after 150 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
β index points

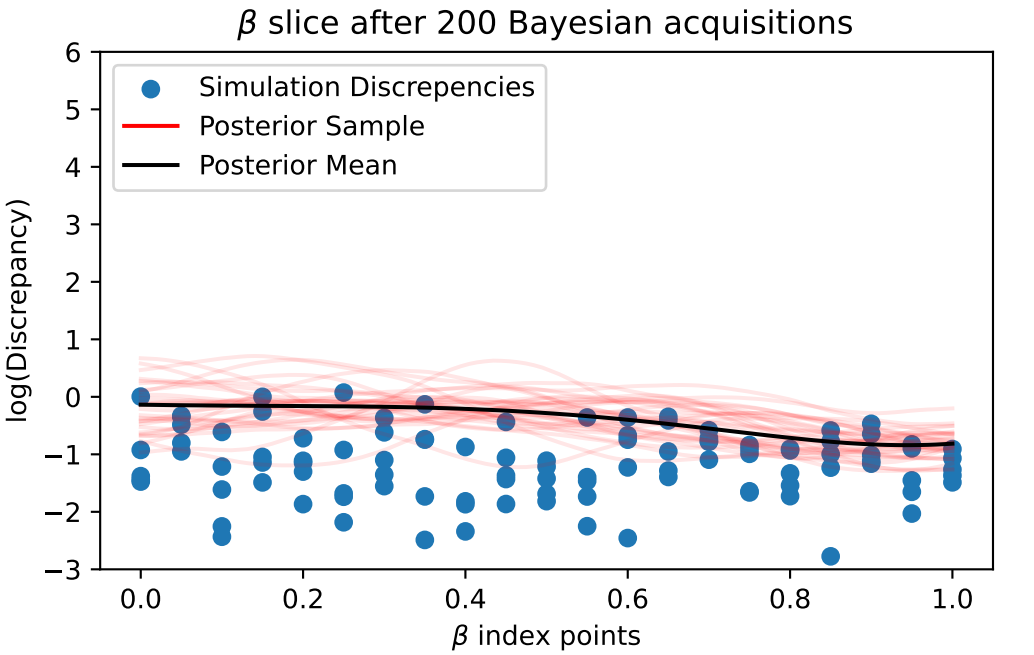

γ_L slice after 150 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
γ_L index points

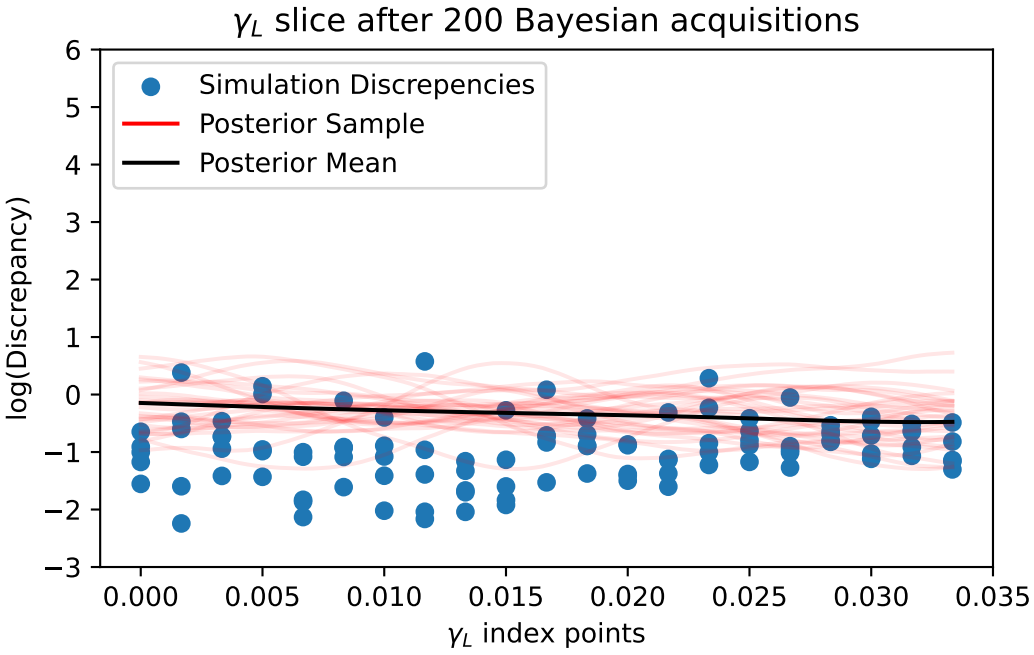

λ slice after 150 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
λ index points

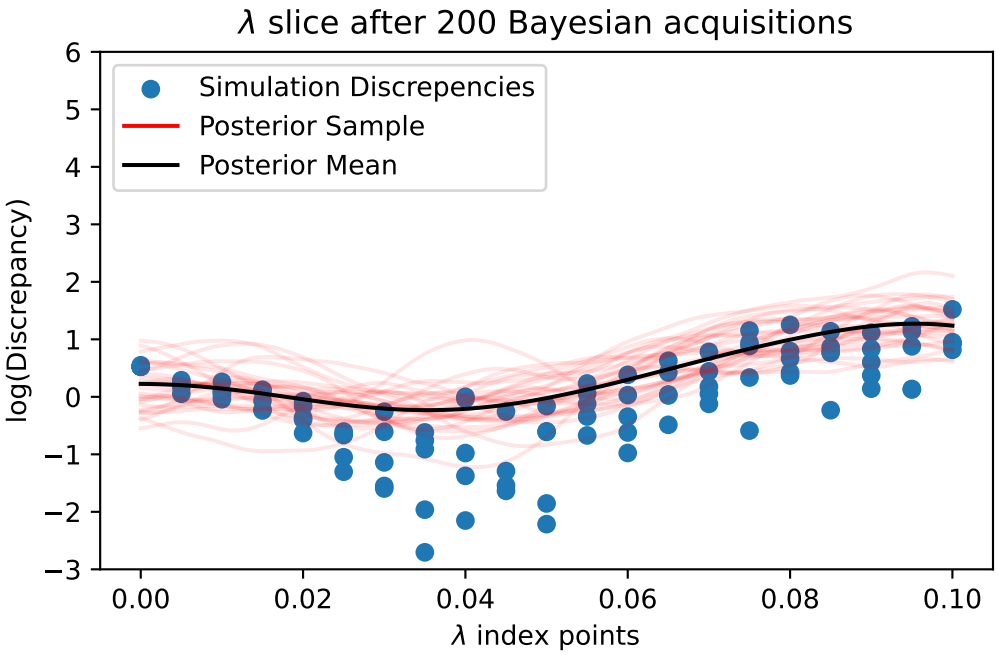

f slice after 150 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
f index points

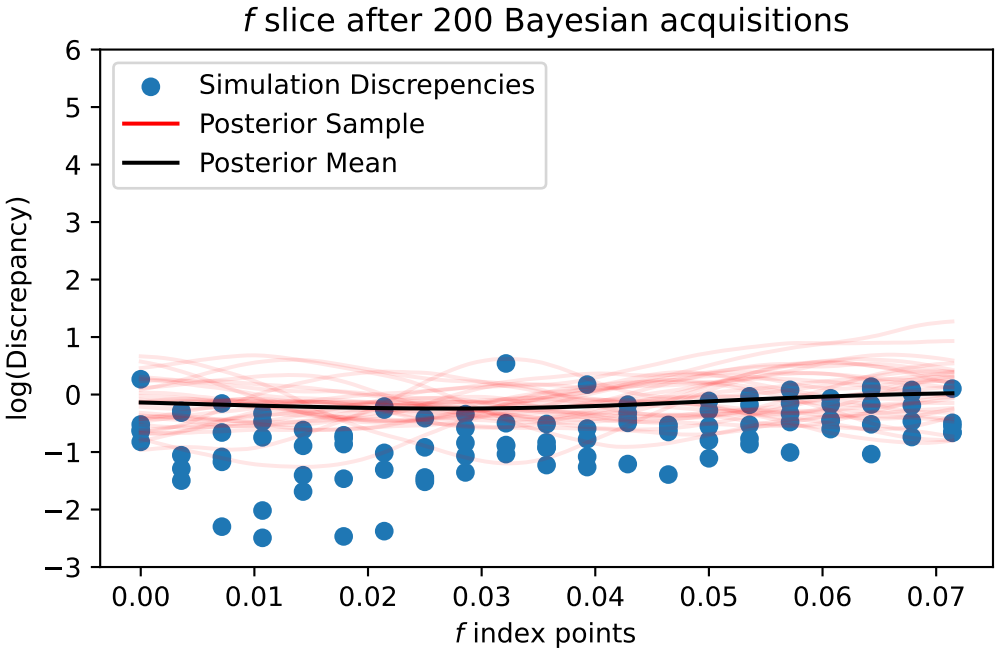

r slice after 150 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
r index points

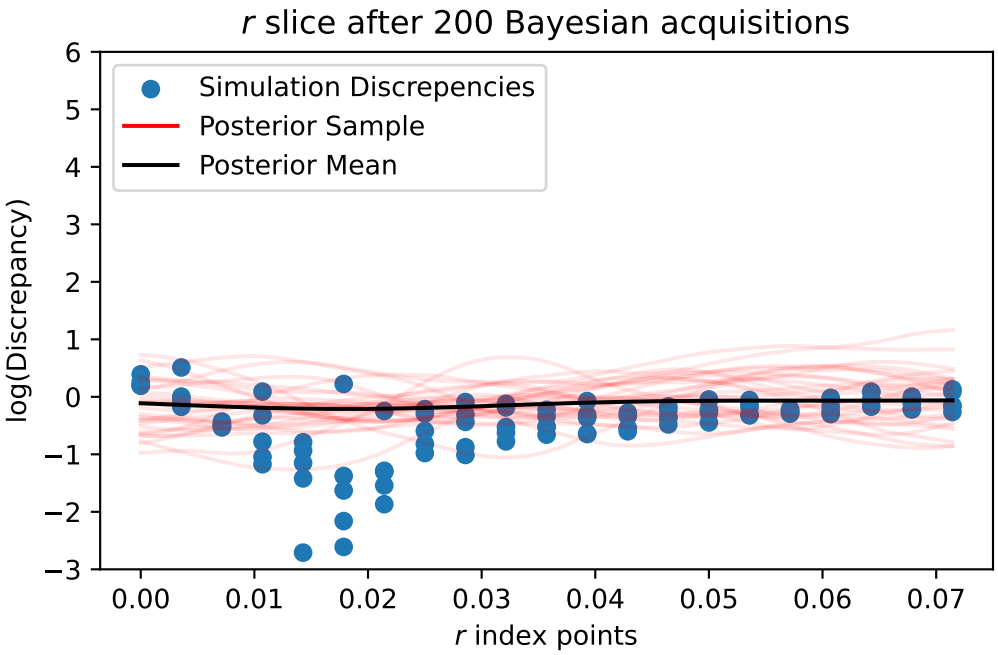

α slice after 200 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
α index points

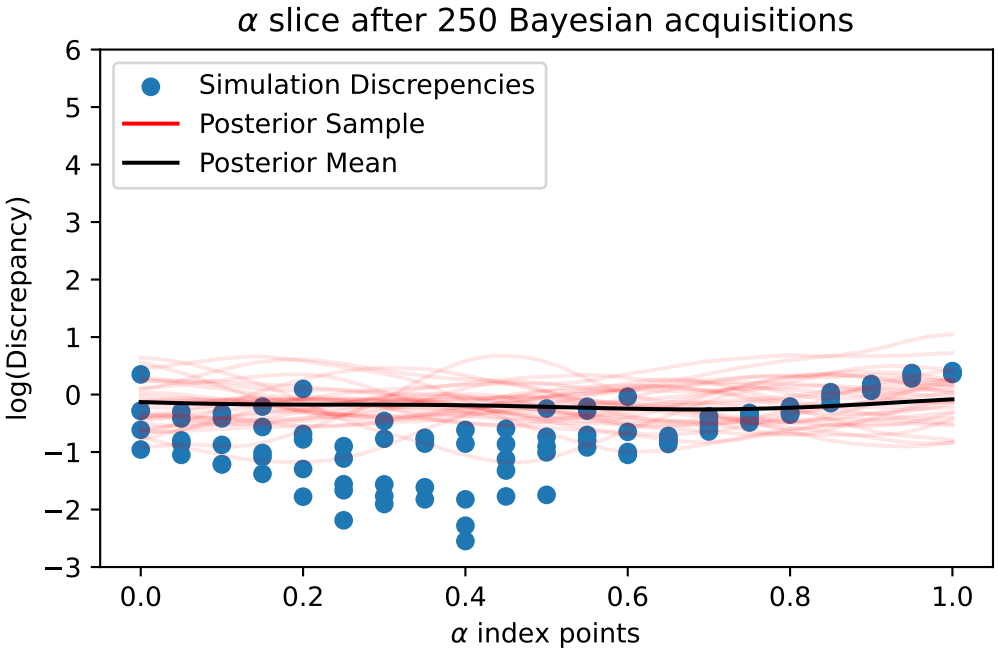

β slice after 200 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
β index points

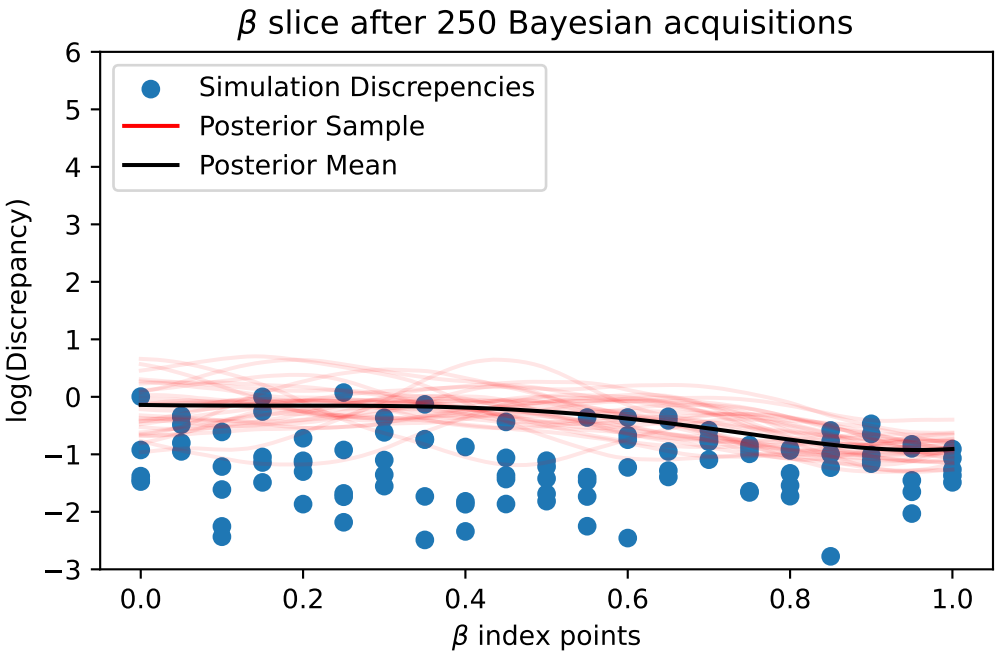

γ_L slice after 200 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
γ_L index points

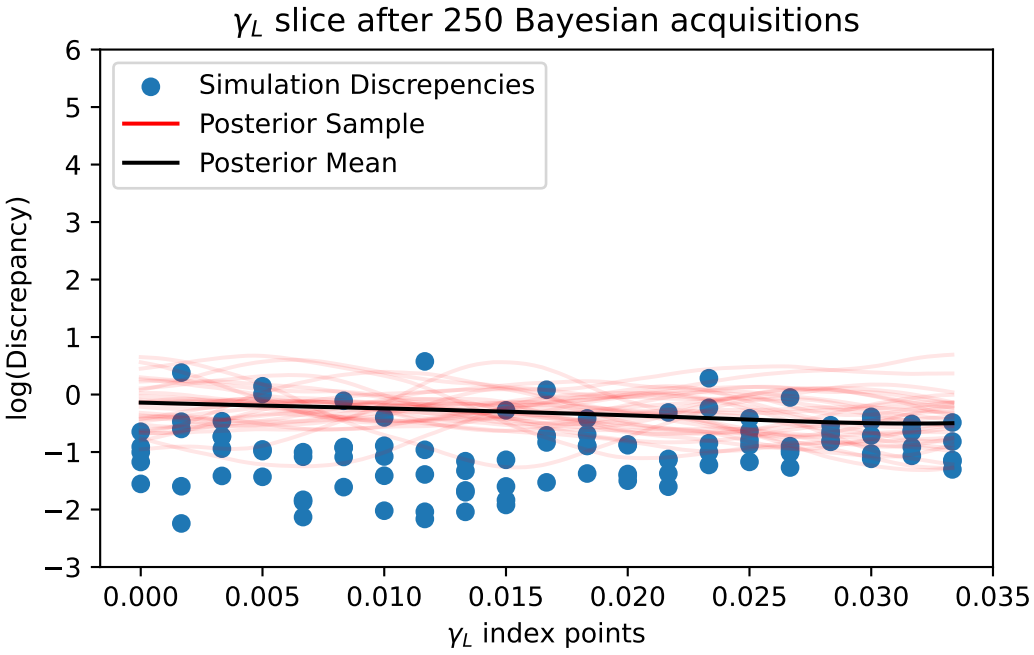

λ slice after 200 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
λ index points

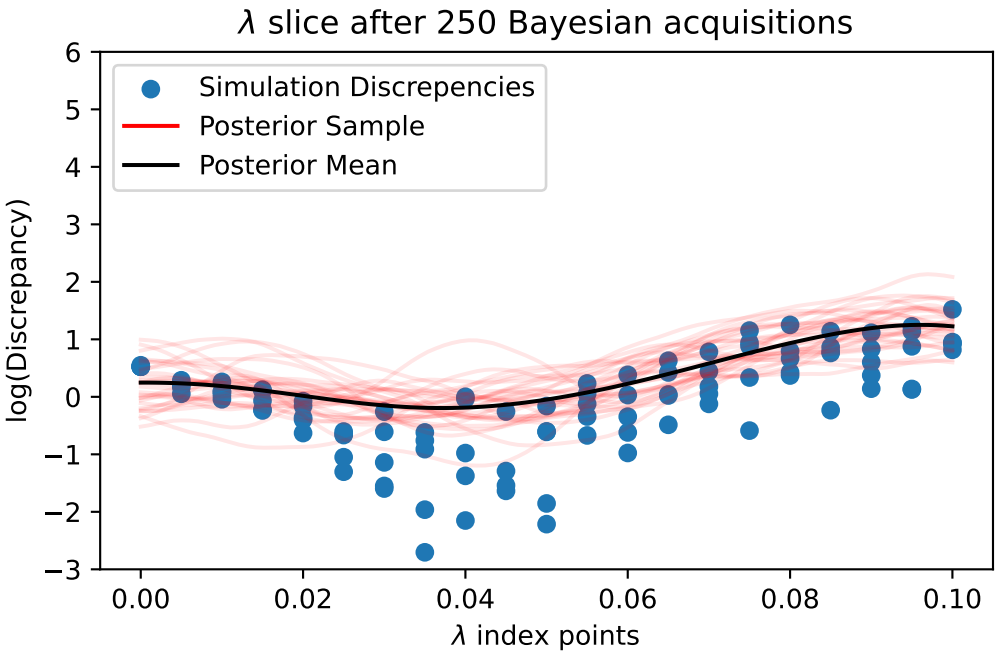

f slice after 200 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
f index points

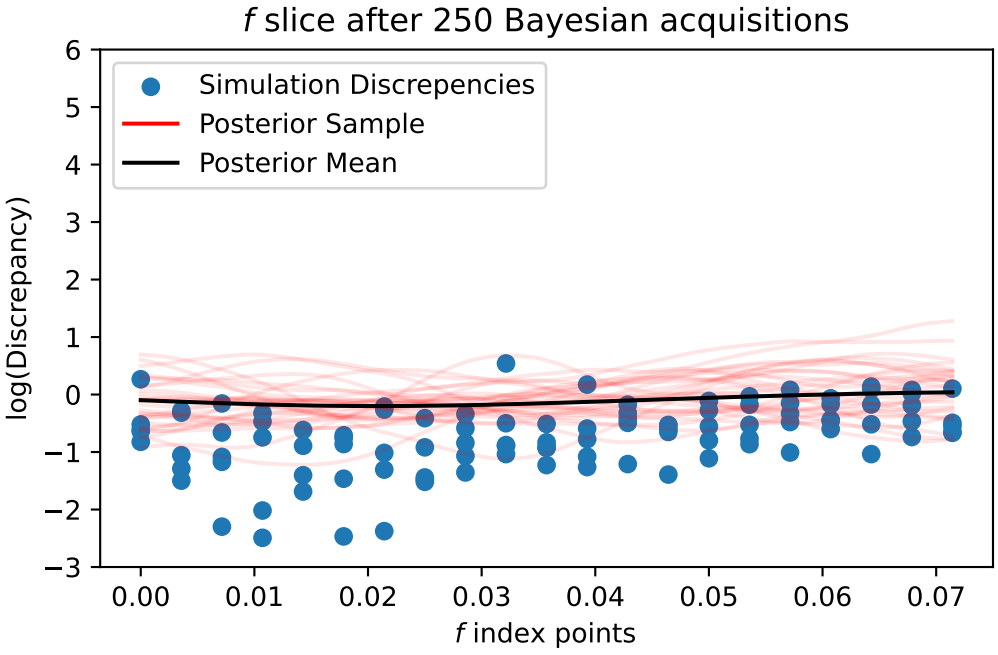

r slice after 200 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
r index points

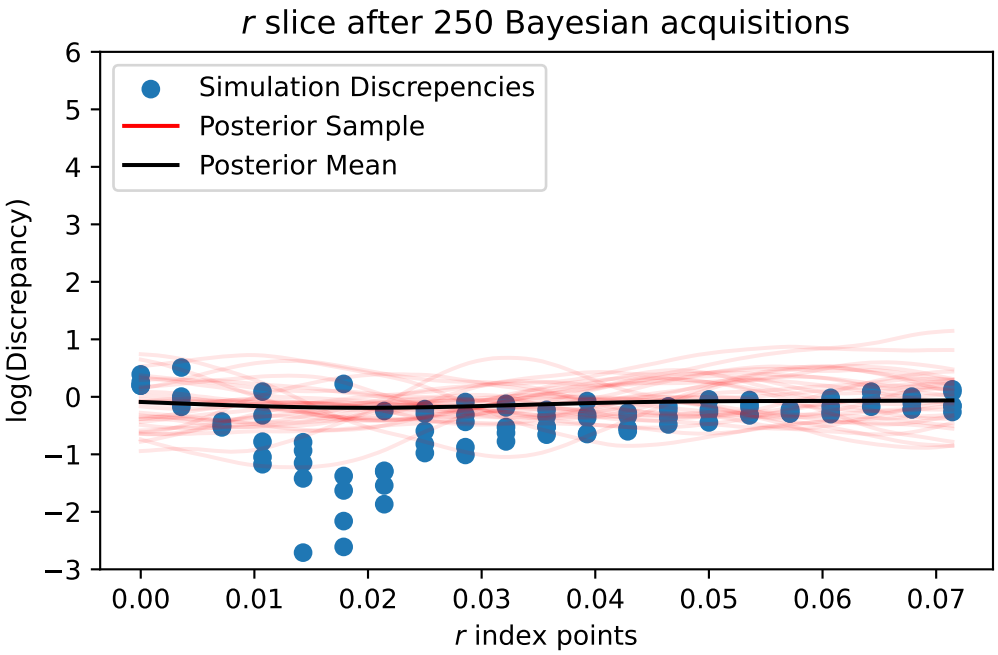

α slice after 250 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
α index points

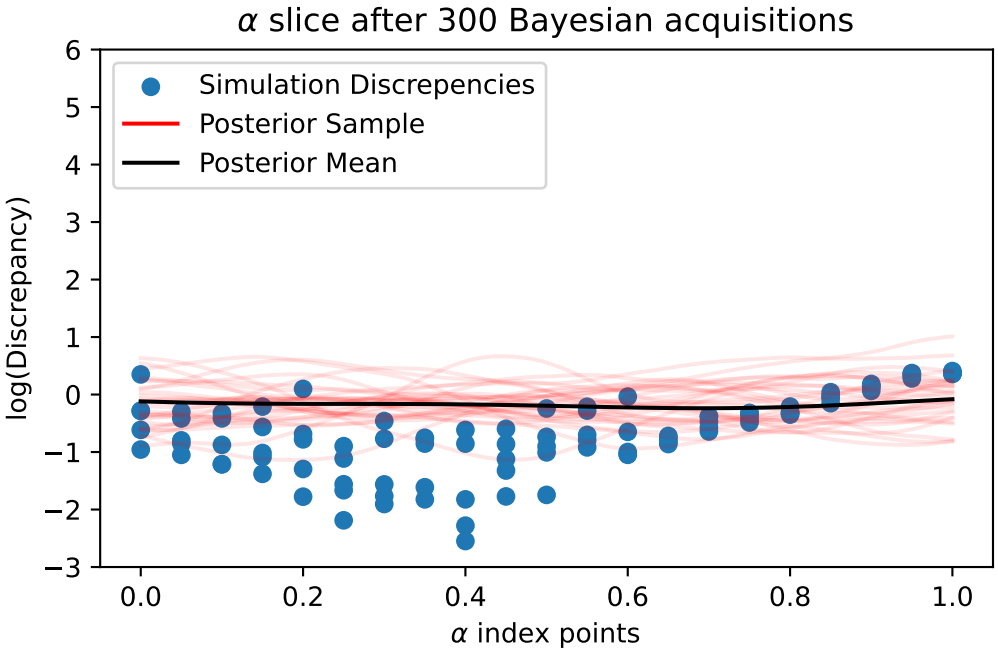

β slice after 250 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
β index points

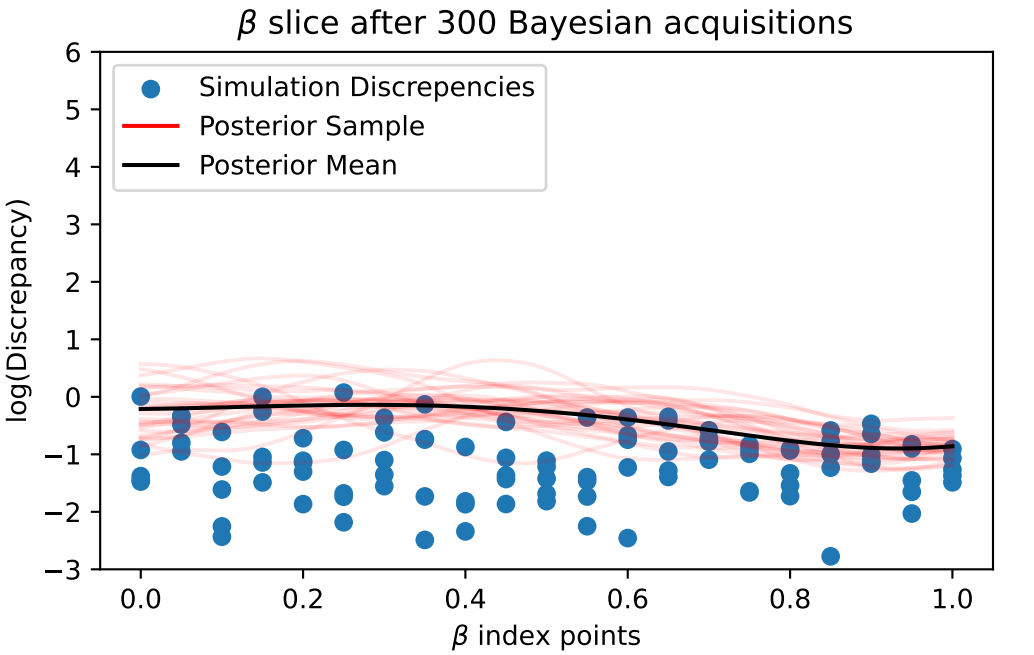

γ_L slice after 250 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
γ_L index points

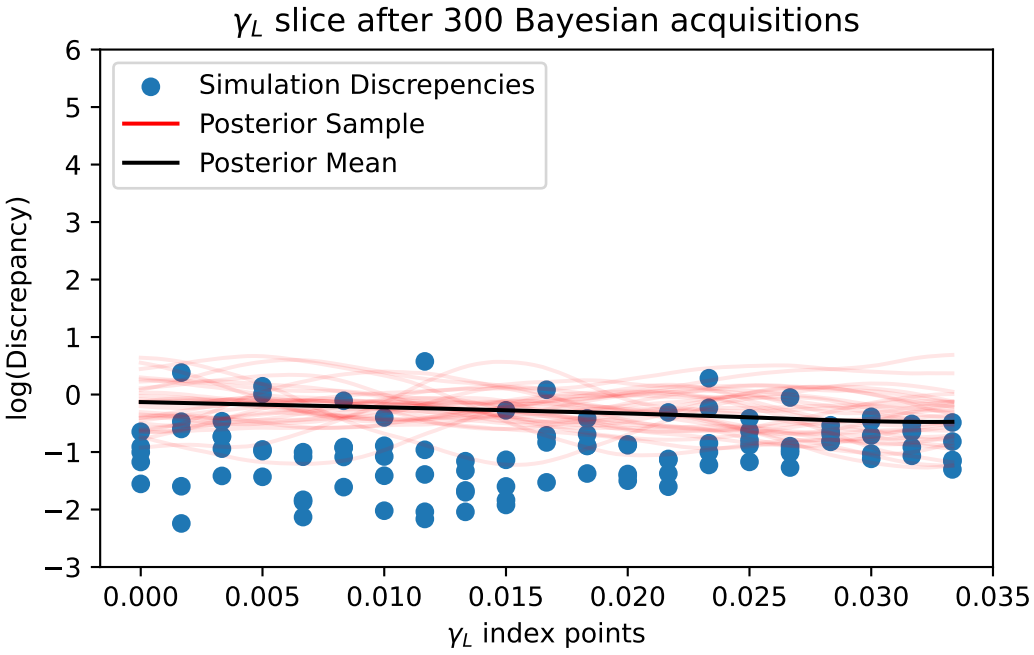

λ slice after 250 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
λ index points

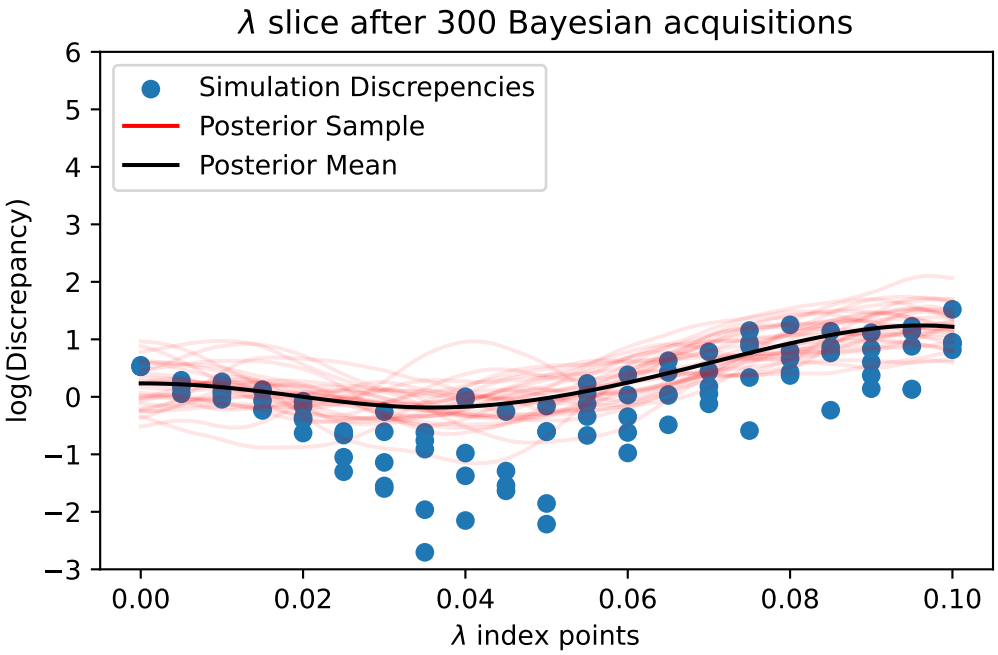

f slice after 250 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
f index points

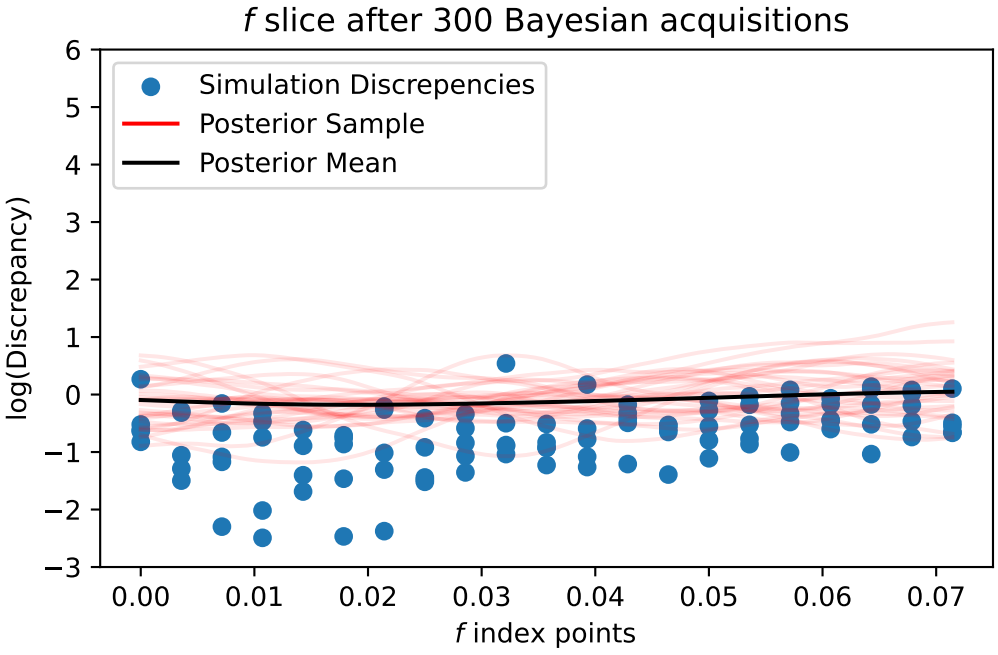

r slice after 250 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
r index points

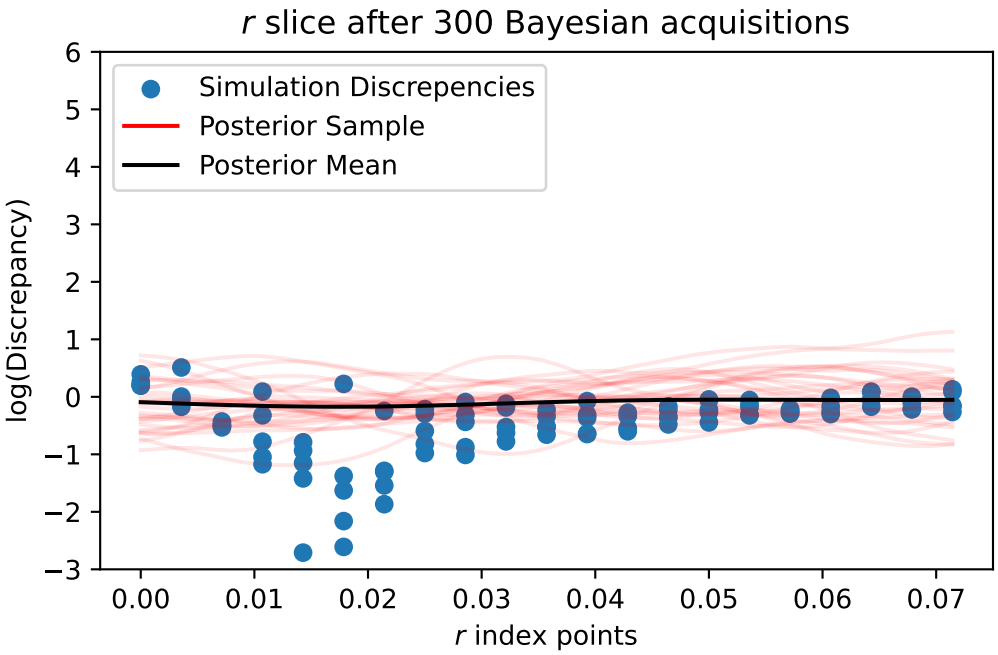

α slice after 300 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
α index points

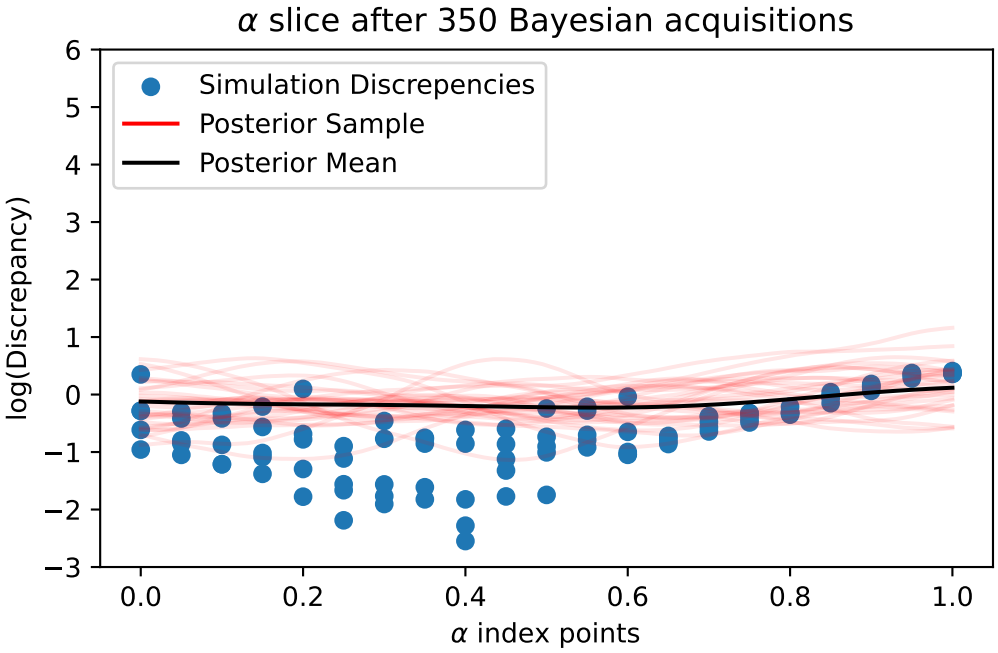

β slice after 300 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
β index points

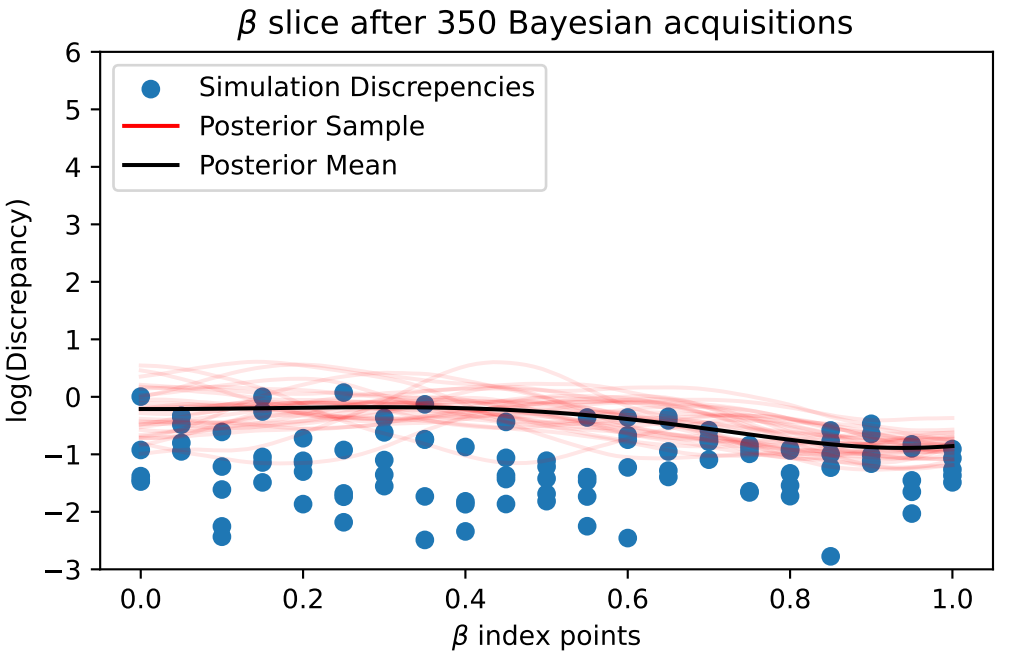

γL slice after 300 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
γL index points

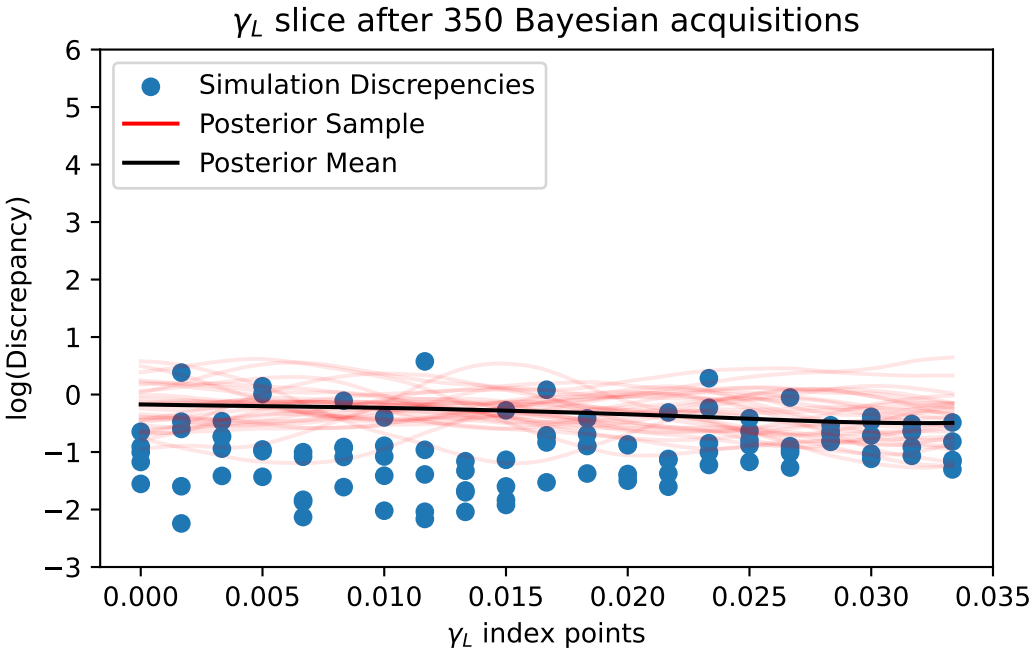

λ slice after 300 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
λ index points

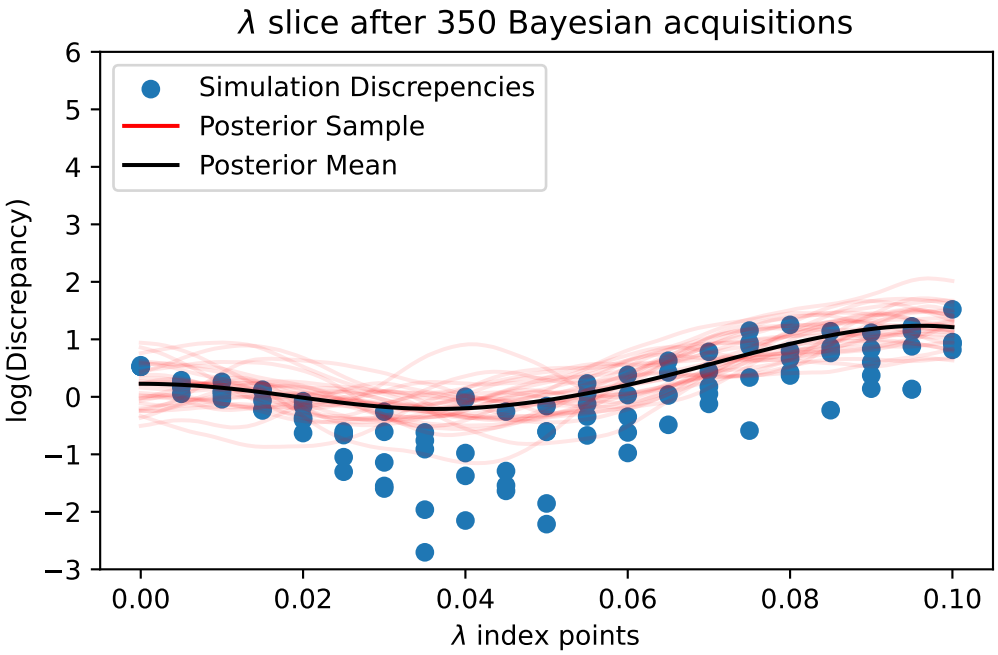

f slice after 300 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
f index points

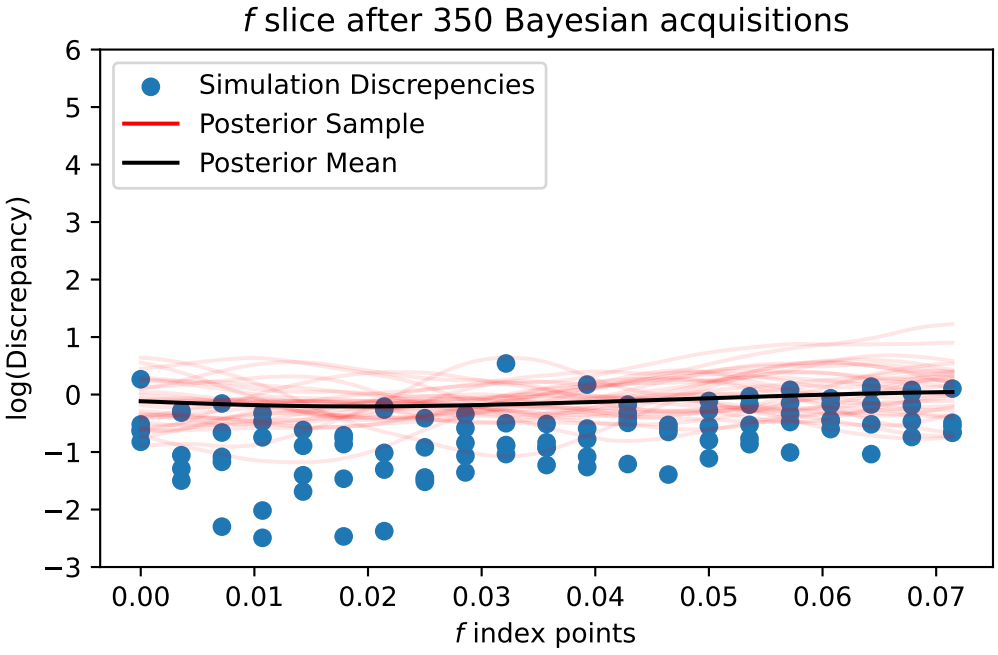

r slice after 300 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
r index points

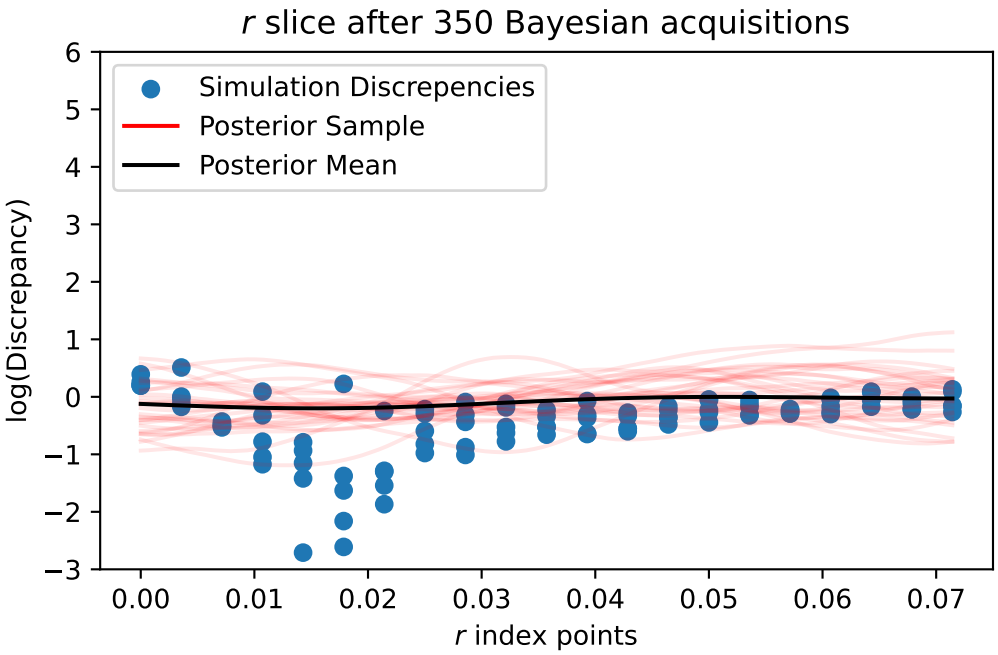

α slice after 350 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
α index points

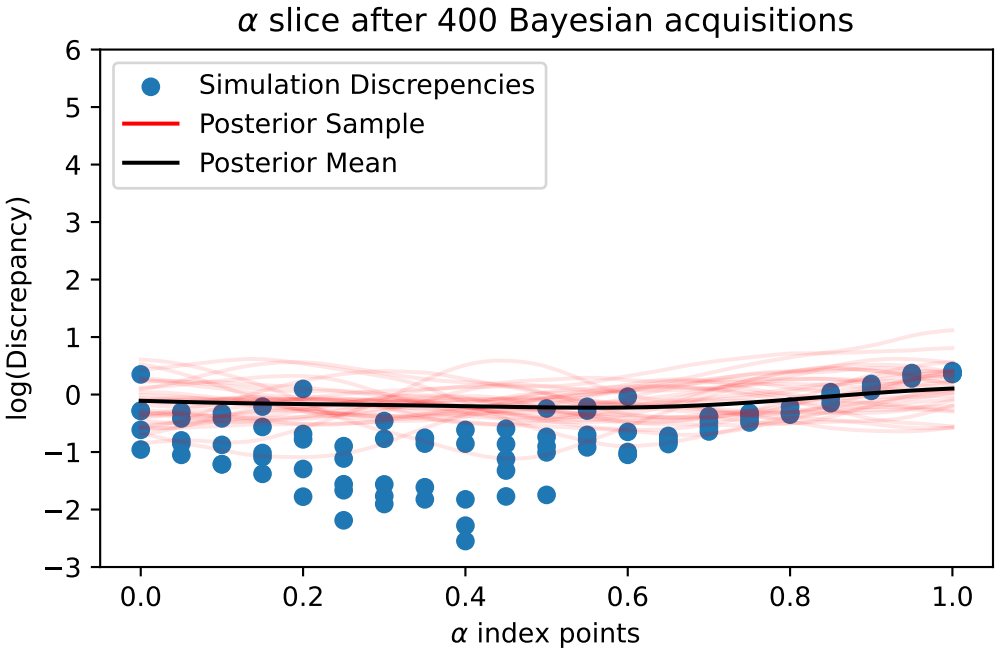

β slice after 350 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
β index points

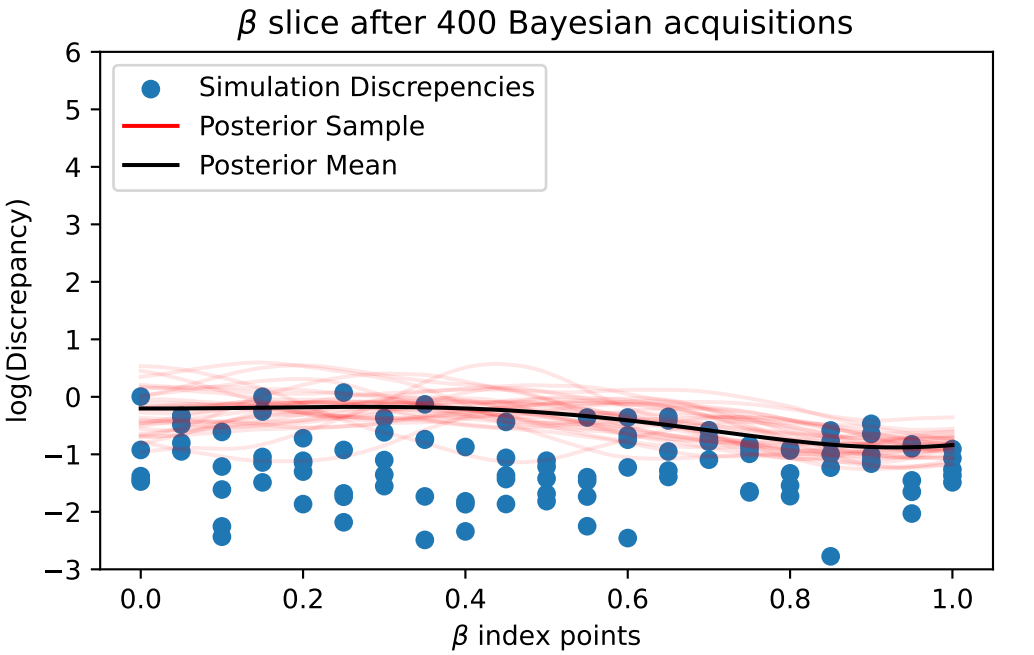

γL slice after 350 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
γL index points

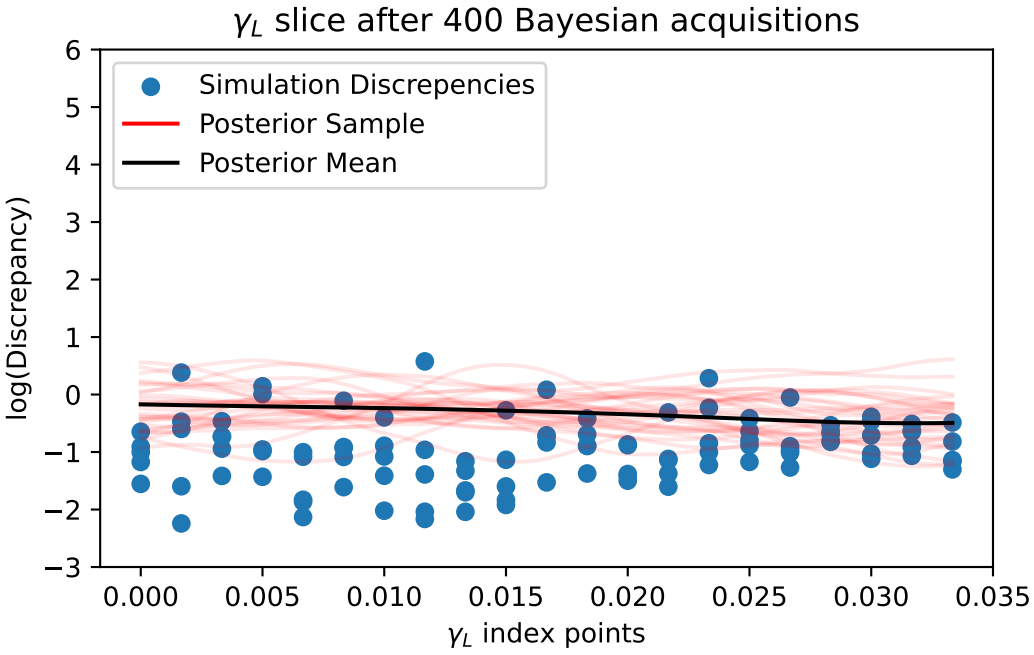

λ slice after 350 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
λ index points

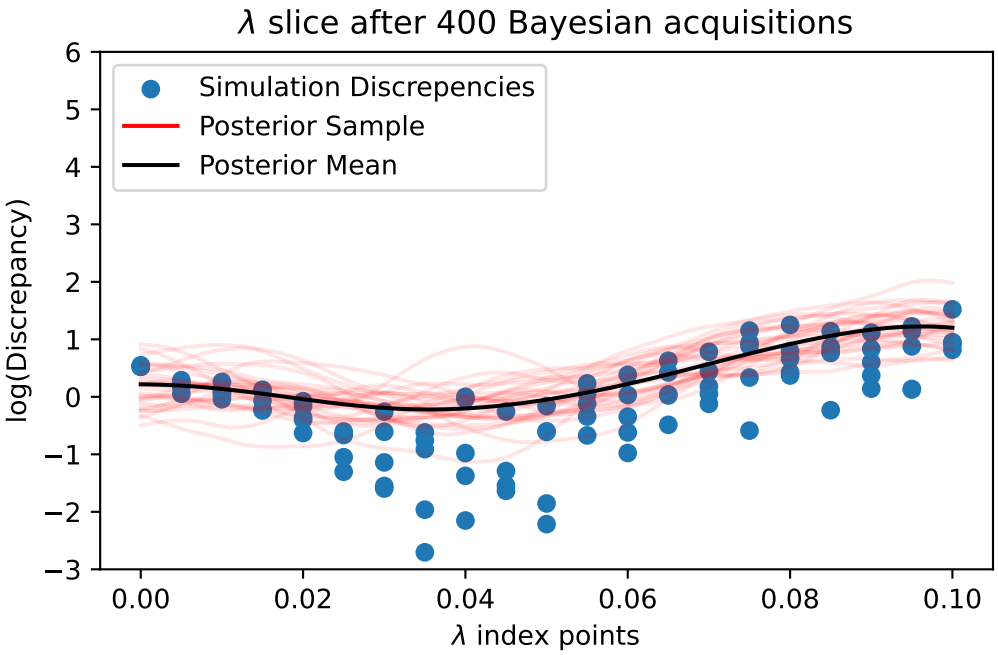

f slice after 350 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
f index points

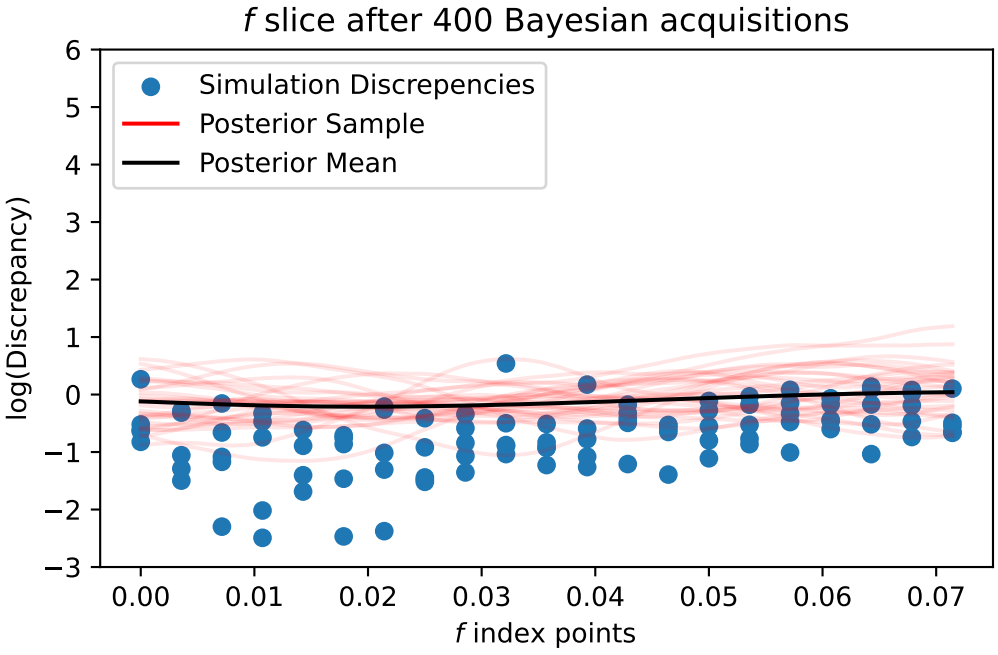

r slice after 350 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
r index points

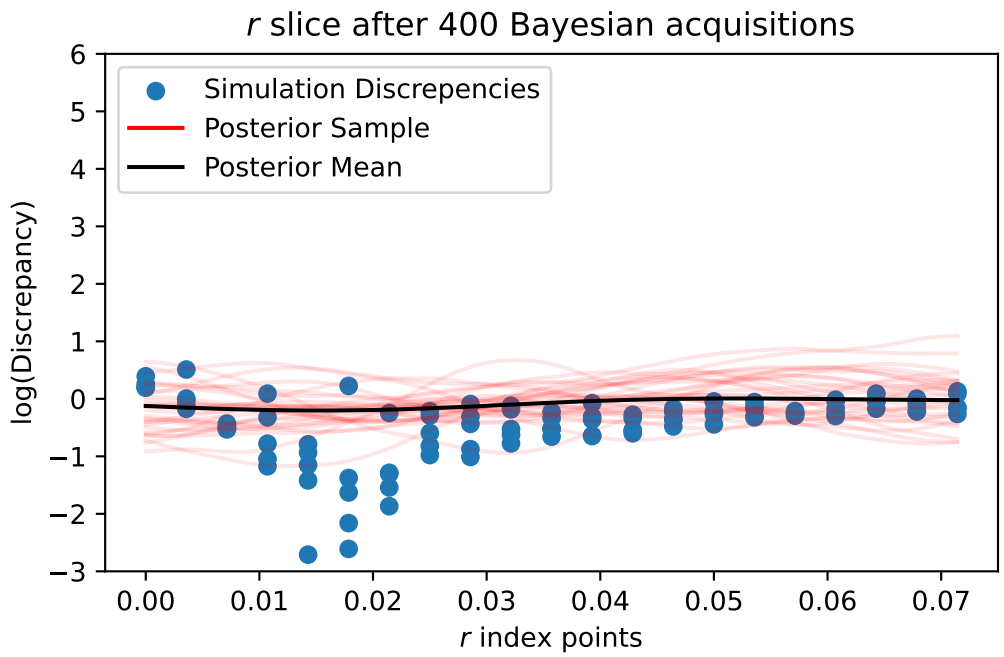

α slice after 400 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
α index points


β slice after 400 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
β index points


γ_L slice after 400 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
γ_L index points

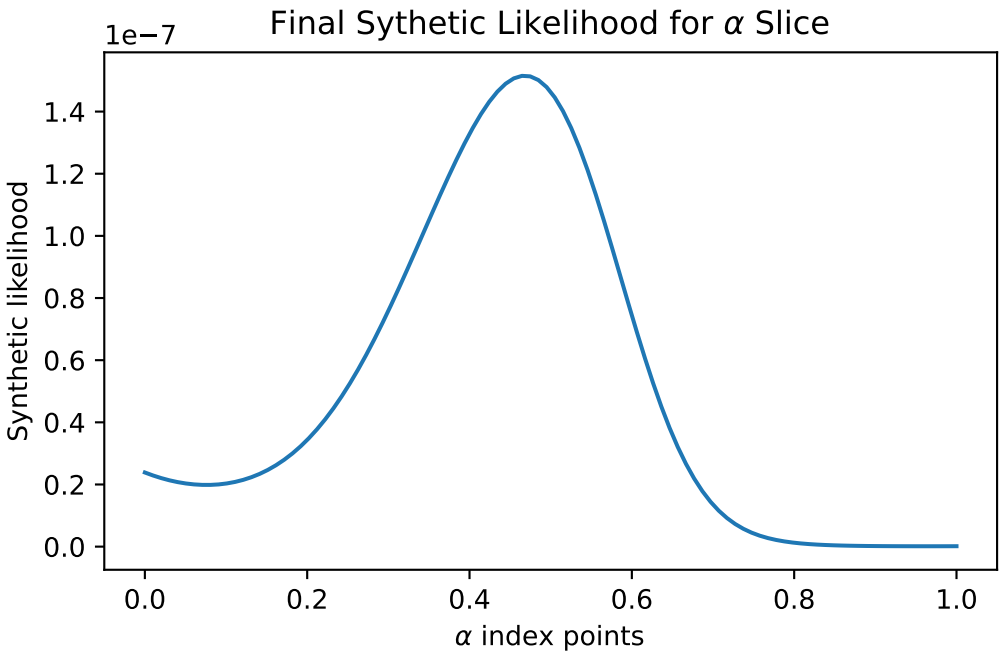

λ slice after 400 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
λ index points

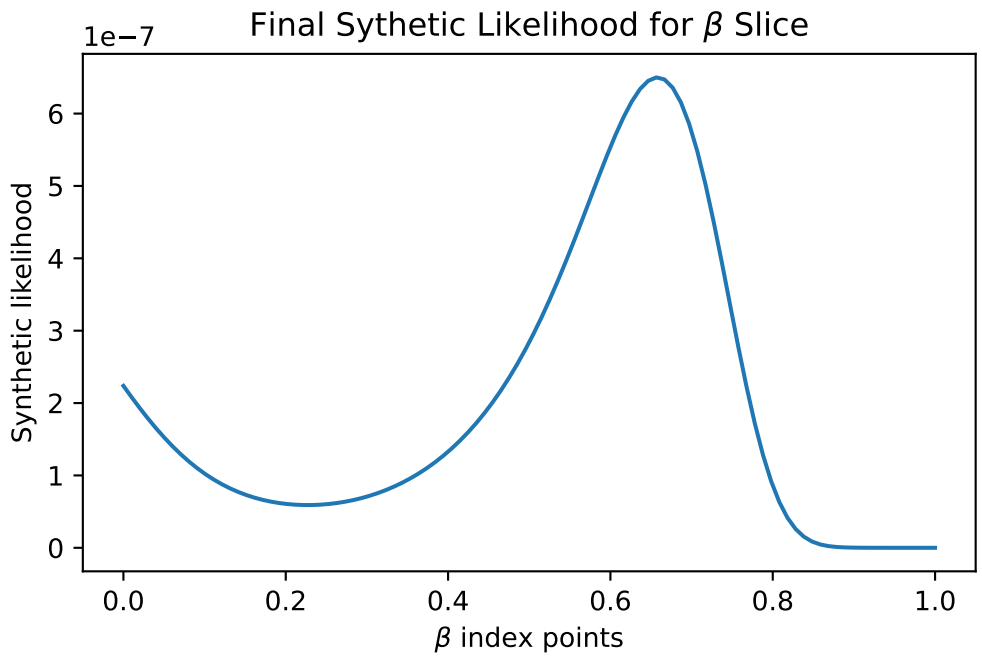

f slice after 400 Bayesian acquisitions
Simulation Discrepencies
Posterior Sample
Posterior Mean
log(Discrepancy)
f index points

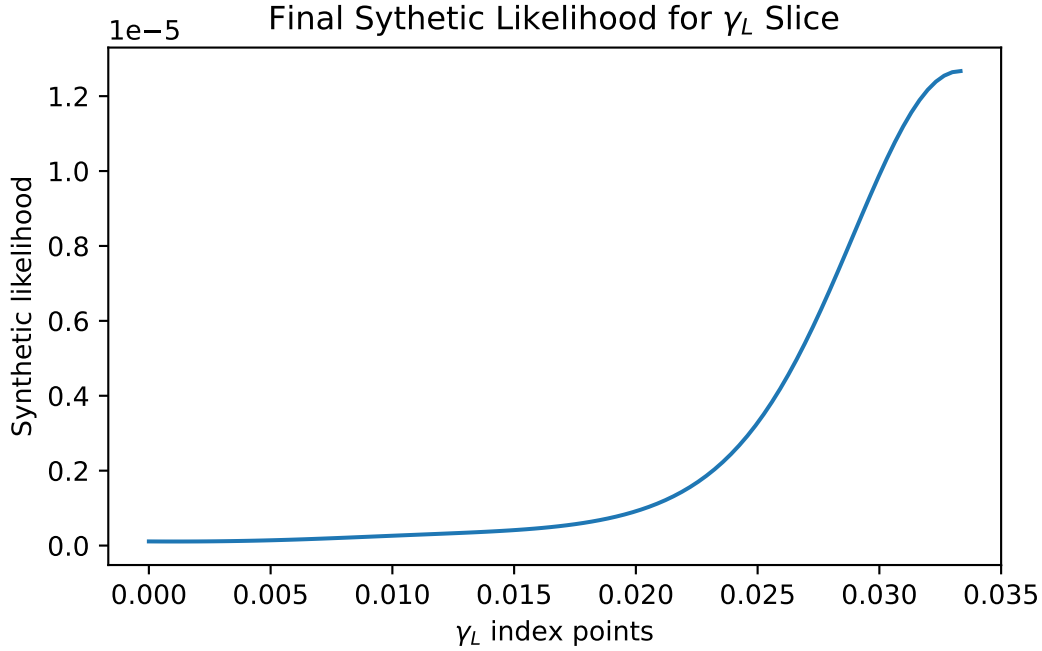


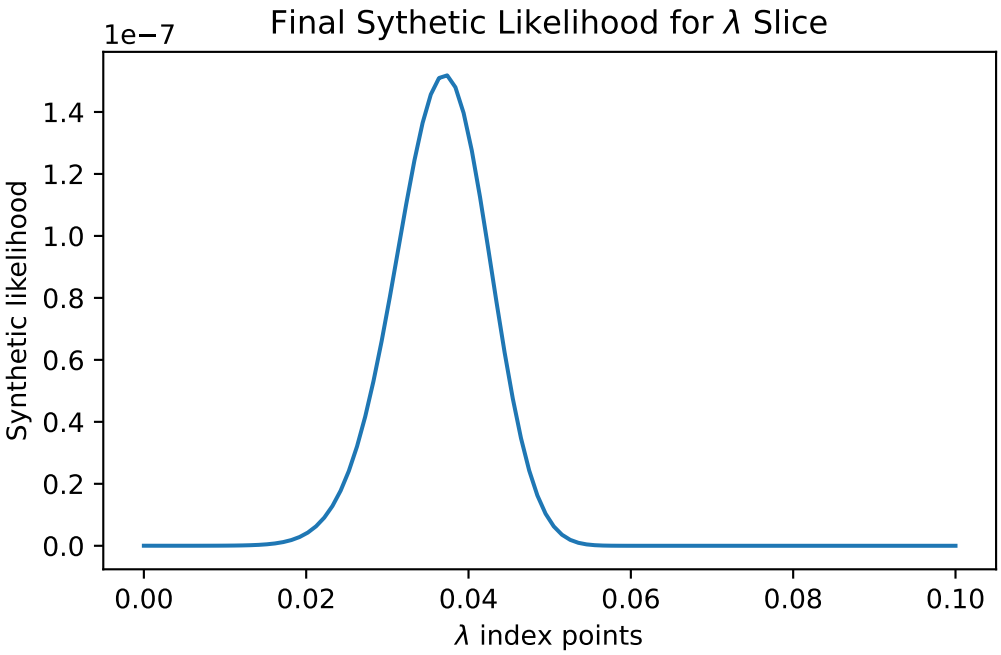

```
epsilon = -2.
for var in vars:
    champ_GP_reg = tfd.GaussianProcessRegressionModel(
        kernel=kernel_champ,
        index_points=slice_indices_dfs_dict[var + "_gp_indices_df"].values,
        observation_index_points=index_vals,
        observations=obs_vals,
        observation_noise_variance=observation_noise_variance_champ,
        predictive_noise_variance=0.0,
        mean_fn=const_mean_fn(),
    )

    indices_for_lik = slice_indices_dfs_dict[var + "_gp_indices_df"].values

    mean = champ_GP_reg.mean_fn(indices_for_lik)
    variance = champ_GP_reg.variance(index_points=indices_for_lik)
    post_std = np.sqrt(variance)
    cdf_vals = tfd.Normal(mean, post_std).log_cdf(epsilon)

    plt.figure(figsize=(6, 3.5))
    plt.plot(
        slice_indices_dfs_dict[var + "_gp_indices_df"][var].values,
        np.exp(cdf_vals),
```

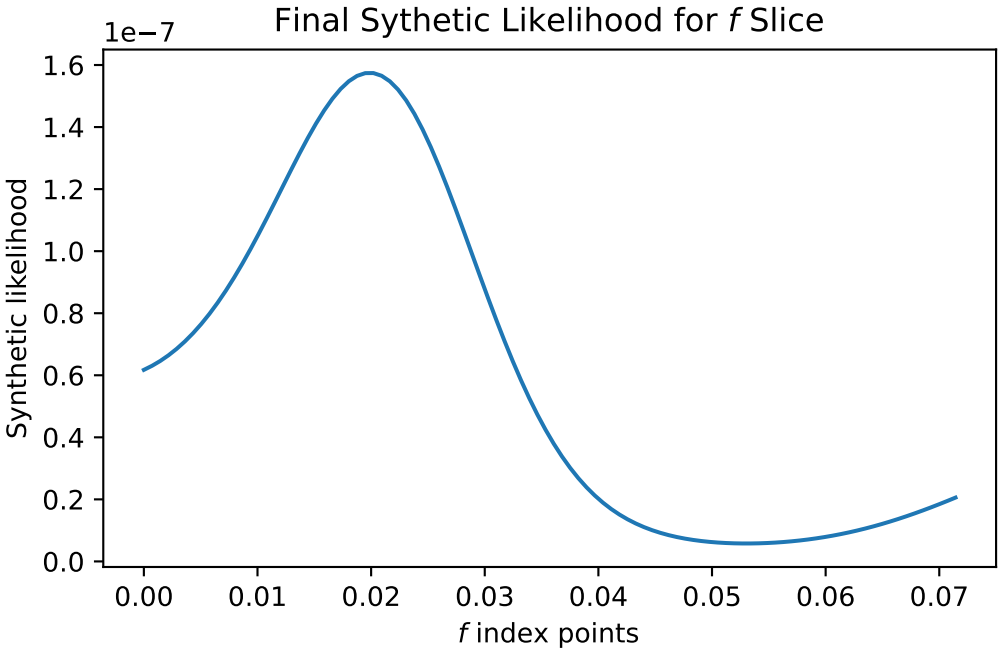
```
    )
    if var in ["f", "r"]:
        plt.xlabel("$" + var + "$ index points")
        plt.title("Final Sythetic Likelihood for $" + var + "$ Slice")
    else:
        plt.xlabel("$\\" + var + "$ index points")
        plt.title("Final Sythetic Likelihood for $\\" + var + "$ Slice")
    plt.ylabel("Synthetic likelihood")
    plt.savefig(
        "champagne_GP_images/"
        + var
        + "_slice_"
        + str(t)
        + "_synth_likelihood.pdf"
    )
    plt.show()
```

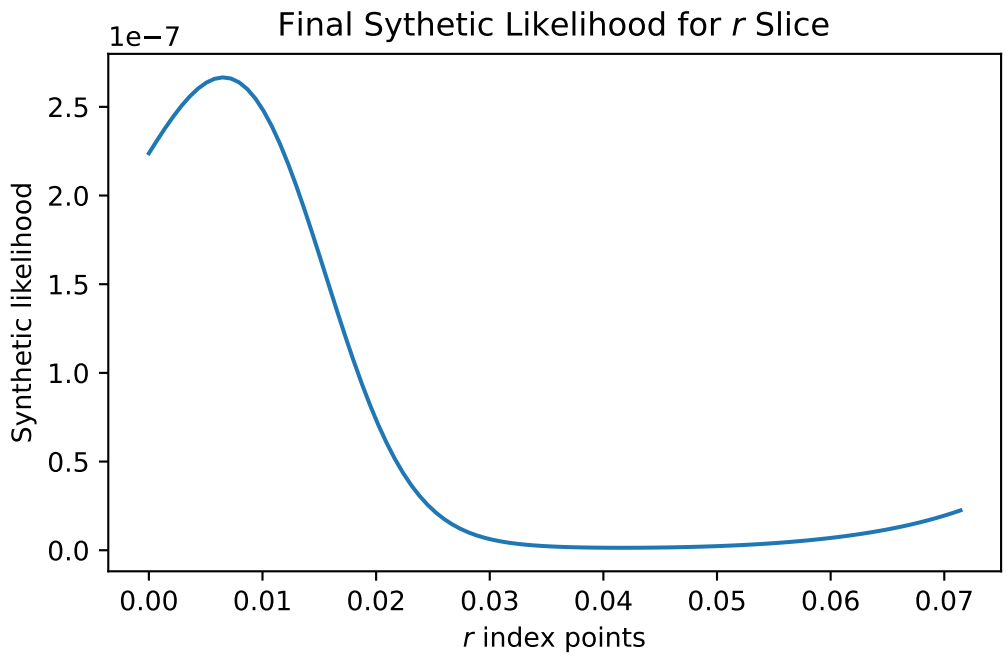



Final Sythetic Likelihood for β Slice
1e−7
Synthetic likelihood
0
1
2
3
4
5
6
0.0
0.2
0.4
0.6
0.8
1.0
β index points


Final Sythetic Likelihood for γ_L Slice
1e−5
Synthetic likelihood
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0.000
0.005
0.010
0.015
0.020
0.025
0.030
0.035
γ_L index points


Final Sythetic Likelihood for λ Slice
1e−7
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Synthetic likelihood
0.00
0.02
0.04
0.06
0.08
0.10
λ index points


Final Sythetic Likelihood for f Slice
1e−7
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Synthetic likelihood
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
f index points




```
# print(index_vals[-600,].round(3))
print(index_vals[-400,].round(3))
print(index_vals[-200,].round(3))
print(index_vals[-80,].round(3))
print(index_vals[-40,].round(3))
print(index_vals[-20,].round(3))
print(index_vals[-8,].round(3))
print(index_vals[-4,].round(3))
print(index_vals[-2,].round(3))
print(index_vals[-1,].round(3))
```

```
[0.601 0.754 0.032 0.061 0.011 0.022]
[0.777 0.543 0.026 0.095 0.011 0.029]
[0.486 0.274 0.031 0.037 0.056 0.012]
[0.701 0.314 0.021 0.095 0.061 0.064]
[0.981 0.928 0.007 0.036 0.047 0.07 ]
[0.801 0.138 0.021 0.035 0.058 0.03 ]
[0.532 0.921 0.017 0.091 0.04  0.068]
[0.273 0.731 0.022 0.066 0.038 0.057]
[0.35  0.753 0.028 0.084 0.037 0.009]
```